Orbis: Rescaling Degree Correlations to Generate Annotated Internet Topologies

Priya Mahadevan
UC San Diego
pmahadevan@cs.ucsd.edu

Calvin Hubble
UC San Diego
chubble@cs.ucsd.edu

Dmitri Krioukov
CAIDA
dima@caida.org

Bradley Huffaker
CAIDA
brad@caida.org

Amin Vahdat
UC San Diego
vahdat@cs.ucsd.edu

ABSTRACT

Researchers involved in designing network services and protocols rely on results from simulation and emulation environments to evaluate correctness, performance and scalability. To better understand the behavior of these applications and to predict their performance when deployed across the Internet, the generated topologies that serve as input to simulated and emulated environments must closely match real network characteristics, not just in terms of graph structure (node interconnectivity) but also with respect to various node and link annotations. Relevant annotations include link latencies, AS membership and whether a router is a peering or internal router. Finally, it should be possible to rescale a given topology to a variety of sizes while still maintaining its essential characteristics.

In this paper, we propose techniques to generate annotated, Internet router graphs of different sizes based on existing observations of Internet characteristics. We find that our generated graphs match a variety of graph properties of observed topologies for a range of target graph sizes. While the best available data of Internet topology currently remains imperfect, the quality of our generated topologies will improve with the fidelity of available measurement techniques or next generation architectures that make Internet structure more transparent.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Network topology; G.2.2 [**Graph Theory**]: Network problems

General Terms

Measurement, Design, Theory

Keywords

Network topology, degree correlations

SIGCOMM'07, August 27–31, 2007, Kyoto, Japan.

1. INTRODUCTION

There has been growing interest in understanding the structure of Internet topologies. Some highlights of this work include the observation that: i) Internet AS peering relationships follow a power law distribution [12], ii) simply considering the degree distribution is insufficient to reproduce the complexity found in router-level topologies [16], and iii) considering the interconnectivity of neighborhoods of increasing size can reproduce arbitrary graph metrics (with neighborhoods of size 3 sufficient to reproduce all known important properties of Internet graphs) [18].

We are interested in applying recent strides in the understanding of Internet structure to generate random Internet router graphs appropriate for simulation [24], emulation [10, 30, 32], and testbed deployment [2, 14] studies. Topology characteristics can impact a range of experiments, including performance of routing protocols, the spread of worms, end-to-end application behavior and congestion, and the resilience of distributed protocols and services to network failures. These studies typically require as input a range of Internet topologies. We identify a number of requirements for any generator of such topologies.

First, such generators should produce both router- and AS-level graphs that accurately reflect the interconnectivity characteristics of the Internet based on the best available measurements. Further, the tool should export knobs to make it easy to explore alternatives to measured network characteristics, e.g., to understand application sensitivity to current Internet structure, to project to some future network topology based on an understanding of evolving trends in network connectivity, or to propose corrections based on suspicions of bias in measurement or sampling error in available datasets. Third, the tool should produce router and AS topologies of a range of sizes, from small-scale topologies appropriate for deployment to larger-scale topologies to support simulation and emulation. In all cases, the topologies should maintain important characteristics of measured Internet topologies. That is, users should be able to feed the tool measured topologies of a range of sizes, and the tool should produce random graphs of a target (rescaled) size that still reproduce characteristics of the input graph to the extent possible.

Finally, the tool should support a range of annotations important to higher-level studies. For instance, nodes in the generated topology should be annotated with AS membership information to enable studies that account for routing behavior. Similarly, the business relationships among peering ASes (peering, customer, etc.) should be included. Without such annotations, Internet routing would have to default to shortest path between end hosts. Other important annotations include link latencies, loss rates, and capacities.

Unfortunately, the current state of the art in topology generation fails to meet the above requirements along a number of dimensions. Existing techniques either produce AS-level graphs, representing entire Autonomous Systems as a single node in the graph with links between ASes representing peering relationships, or router-level graphs with no associated AS information. In the latter case, earlier work [16] shows that existing tools do not reproduce the complex structure of router topologies. These tools include either no annotations or use simple heuristics known not to reflect Internet characteristics. Finally, existing techniques typically either cannot perform graph rescaling or do not do so in a manner that reflects known patterns of network evolution.

Thus, the goal of this paper is to produce a topology generator capable of outputing a range of annotated Internet topologies of varying sizes based on available measurements of network connectivity and characteristics. We employ earlier work on the dK-series [18] to characterize and reproduce network topologies. The dK-series uses degree distributions of node sets of increasing size to characterize an input topology. This model has the advantage of capturing and reproducing all of the important graph properties proposed in the literature to date [18]. As one example, we have used it to produce random graphs that capture the complex interconnectivity of Internet router topologies.

In this context, this paper makes three primary contributions. First, we present an algorithm to rescale an input degree distribution to a graph of different size from the original (Section 4). There are a range of possible techniques for performing rescaling; we propose considering historical Internet connectivity data to inform such rescaling. We experimentally show that we are able to produce graphs of a variety of sizes while still maintaining other important graph characteristics. Second, we present a top-down technique for generating router-level topologies annotated with AS membership (Section 5). Once again, starting with observations regarding AS interconnectivity, we generate rescaled AS graphs and then backfill per-AS router topologies that follow appropriate size and degree distributions based on available measurement data.

Finally, we compare our randomly generated, annotated router topologies to observed Internet router topologies (Section 6). We find close matches for a range of graph metrics proposed in the literature, demonstrating that our techniques maintain important graph properties while rescaling and annotating our generated topologies. We are making the source code for our topology generator publicly available and hope that it will benefit a range of studies. In Section 7, we discuss a number of scenarios that could benefit from our topology generation techniques.

2. BACKGROUND

We use the dK-series [18] as the basis for characterizing a given graph and also for generating random graphs that match a given dK-distribution. The goal of the dK-series is to unify the wide range of graph metrics proposed in the literature. These metrics directly influence the performance and behavior of various network applications and services. We briefly discuss some of the more widely known ones in this section, while a more thorough discussion of these metrics and their impact on networking applications and protocols can be found in [19].

- The *spectrum* of a graph is the set of eigenvalues of its Laplacian matrix, with all the eigenvalues lying between 0 and 2. The smallest non-zero and largest eigenvalues, λ_1 and λ_{n-1}, where n is the graph size, are especially significant.

Table 1: Scalar metric notations.

Metric	Notation
Average degree	k
Maximum degree	k_{max}
Assortativity coefficient	r
Average clustering	$\bar{C}$
Average distance	$\bar{d}$
Standard deviation of distance distribution	σ_d
Smallest eigenvalue of the Laplacian	λ_1
Largest eigenvalue of the Laplacian	λ_{n-1}

- The *distance distribution* $d(x)$ or path-length distribution is the number of pairs of nodes at a distance x, divided by the total number of node pairs n^2 (self-pairs included).

- *Betweenness*, a commonly used measure of centrality, is the weighted sum of the number of shortest paths passing through a given node or link. It estimates the potential traffic load on a node or link, assuming uniformly distributed traffic following shortest paths.

- The *assortativity coefficient* r [22] suggested as a summary statistic of node interconnectivity. Its low (high) values indicate that links connecting nodes with dissimilar (similar) degrees prevail.

- *Clustering* $C(k)$ is a measure of how close neighbors of the average k-degree node are to forming a clique: $C(k)$ is the ratio of the average number of links between the neighbors of k-degree nodes to the maximum number of such links $\binom{k}{2}$.

Throughout this paper, we use the notations from Table 1 to refer to the scalar statistics associated with the metrics mentioned above.

The dK-series is based on the observation that the most basic property of a network topology characterizes its connectivity. Each property in the dK-series provides information on connectivity within groups of d nodes with degrees $k_1, ..., k_d$, where d takes values from 0 to the total number of nodes, n, in the graph. In other words, the dK-series describes correlations amongst degrees of nodes in subgraphs of size d, for $d = 0, 1, ..., n$. Each element in the dK-series allows us to generate dK-graphs that reproduce the specific correlation among the degrees in all d-sized subgraphs of the given graph.

For $d = 0$, $0K$-graphs reproduce the average degree (also referred to as $0K$*-distribution*) of the given graph $G(V, E)$. The average degree is $\bar{k} = 2m/n$, where $n = |V|$ and $m = |E|$ are the numbers of nodes and links in G. When $d = 1$, $1K$-graphs reproduce the degree distribution ($1K$*-distribution*), $P(k) = n(k)/n$, of the given graph with $n(k)$ nodes of degree k (k-degree nodes). The $1K$-distribution contains more information on node connectivity than $0K$-distribution, and it is, therefore, a more restrictive metric: the set of $1K$-graphs is more constrained than the set of $0K$-graphs and thus is a subset of the set of $0K$-graphs. When $d = 2$, $2K$-graphs reproduce the joint degree distribution (JDD) of the given graph. The JDD, or the $2K$*-distribution*, is given by $P(k_1, k_2) = m(k_1, k_2)\mu(k_1, k_2)/(2m)$, where $m(k_1, k_2)$ is the number of edges between k_1- and k_2-degree nodes, and $\mu(k_1, k_2)$ is 2 if $k_1 = k_2$ or 1 otherwise.

Extending this series in a similar fashion, when $d = n$, the generated nK-graphs are isomorphic to the given graph, and thus are guaranteed to reproduce any graph metric currently proposed in the literature or any future metric of interest. In summary, each property in the dK-series embeds increasingly more information about the given graph structure, and thus the corresponding dK-graphs are increasingly constrained, until they finally converge to the given graph.

Earlier work [18] presented several different techniques to construct *dK-random graphs*, which are random graphs that have the same dK-distribution as the given graph and that are unbiased with respect to any other more constraining property. One significant limitation of [18] was that the generated random graphs had the same number of nodes as the original graph. One of the principal goals of this work is to devise techniques for generating graphs of different sizes, i.e., graphs reproducing appropriately rescaled dK-distributions of the original graph.

2.1 Other Related Work

One of the earliest network topology models was proposed by Waxman [31] and is based on the classical Erdős-Rényi random graphs [11]. This model later was abandoned in favor of other models such as GT-ITM [34] that incorporated hierarchical structures observed in the Internet. Seminal work in 1999 [12] presented evidence that the degree distribution of Internet ASes followed a power law. Structural models such as GT-ITM failed to reproduce this specific form of degree distribution. Later, generators such as PLRG [1], Inet [33], and BRITE [21] focused on reproducing the observed power-law degree distribution; however, the graphs so generated do not match the observed topologies with respect to a wide range of metrics considered important in the literature.

Li *et al.* [16] consider router capacity constraints, as well as likelihood to model router-level topologies and advocate understanding the evolution of networks in order to accurately model router-level graphs. For AS-level topologies, recent work [7] considers the technological, economic and political considerations behind whether pairs of ASes peer with another. Thus, this and related efforts consider the driving evolutionary forces behind the growth of particular topologies. We consider the study of evolutionary forces to be complementary to the techniques described in this paper. While we present an approach to scaling graphs based on observed historical changes in important graph metrics, evolutionary studies focus on the external stimuli behind growth.

3. METHODOLOGY OVERVIEW

To justify our rescaling approach, we first study historic data for AS topologies collected over the past few years. The historic data provide insight into how the global and local structures of the network topology change as the network grows. Specifically, our random graph generation technique is based on the dK-series [18], which shows that reproducing the $1K$-distribution for AS graphs matches most important topology metrics proposed in the literature. We thus want to check if the $1K$-distribution of the AS topology is an invariant of Internet growth. Reproducing $2K$-distributions almost suffices for router topologies [18],[1] and in this paper we present techniques to generate $1K$- and $2K$-random graphs of any specified size.

A graph that merely captures the structure of the Internet's AS-level or router-level topology, though still useful, might not help researchers who would like to evaluate the performance of their network applications and services. One reason is that influence of node and link *annotations* such as latency, loss-rate, AS membership, *etc.*, can significantly impact application performance and protocol behavior. For example, several research studies show that actual packet paths between Internet hosts are substantially longer than the corresponding shortest paths [13, 26, 29]. In the absence of AS membership information in the generated graphs, studies that require a router-level topology end up computing shortest paths between pairs of end nodes. These studies will likely yield incorrect results for packet round-trip times and thus potentially incorrect results for application performance. Annotating each router with AS membership will enable us to implement correct routing paths in the topology, as we can now employ modified shortest-path algorithms such as [20] that respect AS membership and routing policies between the ASes. In this paper, we focus on techniques for annotating topologies with AS membership. We have also developed techniques to annotate topologies with per-hop latencies to match end-to-end distributions of latency distributions observed in the Internet. We omit details of our techniques for latency annotation for brevity.

[1]We strongly emphasize, however, that one can easily construct regular synthetic graphs (e.g., chains, rings, grids, etc.) such that their $2K$-random counterparts would be extremely dissimilar to the original graphs. In other words, it is an empirical observation that Internet topologies are *specific* in the sense that they are almost $1K$- or $2K$-random.

4. RESCALING TECHNIQUES

In this section, we present techniques to generate different-sized graphs using dK-distributions. Specifically, we describe algorithms to generate $1K$-random and $2K$-random graphs with variable numbers of nodes given a target distribution for some fixed-sized graph.

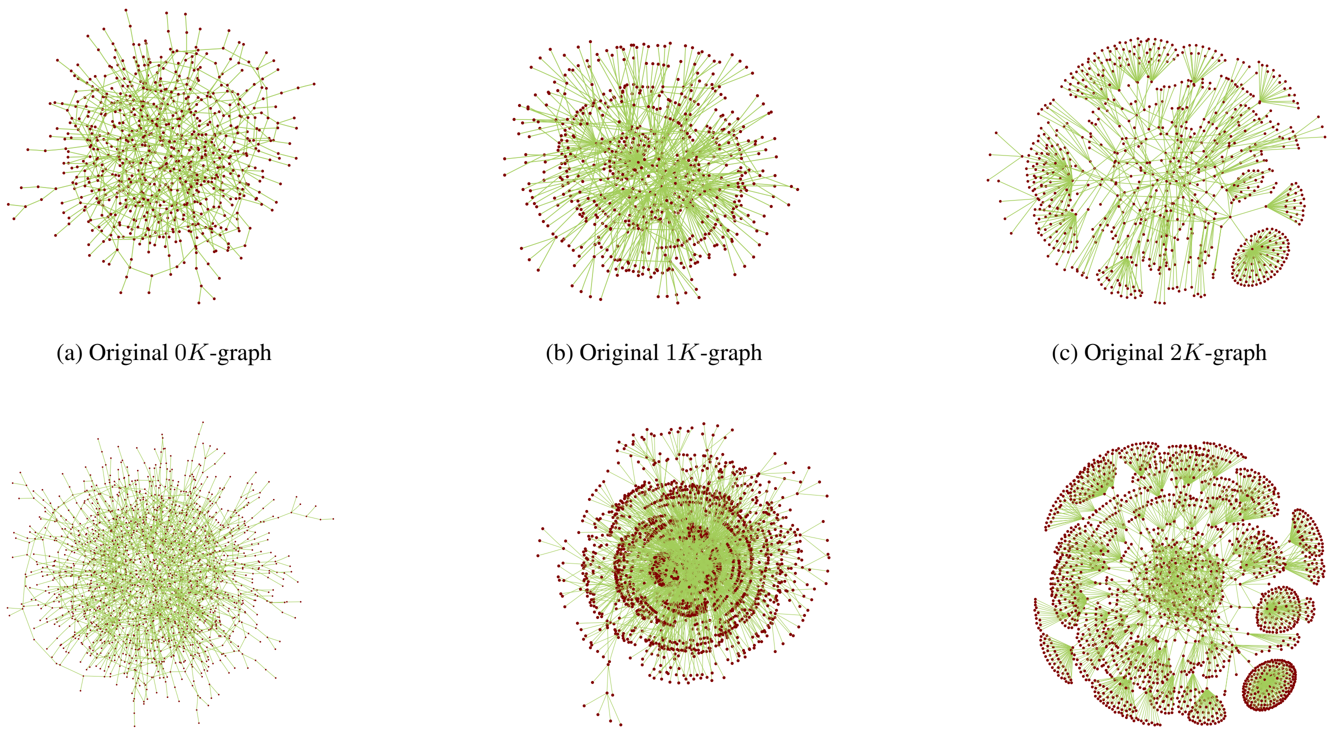
We do not need to rescale a $0K$-distribution because it is a single scalar equal to the average degree. Procedures to construct arbitrary-sized graphs with a given average degree are straightforward [18].

Consider the $1K$-distribution $P(k)$ however. It is a function of one integer variable, i.e., node degree k. The support[2] of $P(k)$ lies within $[0, n-1]$, $0 \leqslant \text{supp}(P) \leqslant n-1$, while the values of $P(k)$ are between 0 and 1. If we wish to generate a random graph of a different size n', then the main question is how the support and values of $P(k)$ should change to result in the appropriately rescaled degree distribution $P'(k')$ of the new graph. For example, when scaling a graph from 1000 to 2000 nodes, how should the degrees itself as well as their corresponding distribution be rescaled in the new graph? The question becomes more complex when considering, for instance, $2K$-distributions $P(k_1, k_2)$ that are functions of two arguments.

We believe that appropriate rescaling techniques depend on the characteristics of the class of graphs being considered. That is, individual types of graphs will scale up and down in an application-specific manner. The way that Internet graphs grow may very well be different from how social or biological networks grow. Similarly, growth characteristics for a given organization's router topology may well differ from the growth characteristics of the Internet's global AS peering graph.

To motivate how our rescaling works in practice, we present, in Figure 1, visualizations of the $0K$-, $1K$-, and $2K$-randomized versions of the original-sized HOT router graph from [16] and their rescaled counterparts. While we will quantify our ability to reproduce important graph metrics in subsequent sections, visually we see that the rescaled graphs maintain much of the same connectivity structure of their original-sized versions.

[2]The support $\text{supp}(f)$ of function $f(x)$ is the set of values of its argument x such that $f(x) \neq 0$.



(a) Original $0K$-graph (b) Original $1K$-graph (c) Original $2K$-graph

(d) Rescaled $0K$-graph (e) Rescaled $1K$-graph (f) Rescaled $2K$-graph

Figure 1: The original-sized (939 nodes) dK-random HOT graphs and their rescaled (2000 nodes) versions.

Table 2: Scalar metric values for historic skitter AS-level topologies.

Year	Nodes	k	k_{max}	r	C	d	σ_d
2000	3308	5.77	836	-0.25	0.38	3.14	0.41
2001	6021	5.48	1461	-0.22	0.40	3.21	0.42
2002	8359	6.49	2355	-0.24	0.44	2.99	0.35
2003	8512	5.17	1443	-0.23	0.38	3.30	0.43
2004	9204	6.23	2070	-0.24	0.46	3.12	0.37
2005	8500	5.97	1783	-0.23	0.45	3.17	0.39

4.1 Internet topology input data

4.1.1 AS topology historic data

Since we are interested in building an Internet router topology generator that includes AS information for each router, we decided to study the historical growth characteristics of Internet AS graphs. In particular, we considered the dK-distributions for historical AS-level topologies extracted from skitter [5] and RouteViews [25] data in March of each year between 2000 and 2005. The AS-level topology obtained from skitter grew by almost a factor of 3 during this time period. Our hypothesis is that the graph will demonstrate some steady growth characteristics and that we could then apply our understanding of the Internet's AS growth to generating graphs of a range of sizes given some initial dK distribution. For brevity, we only present results for the skitter data, though the conclusion from the RouteViews data is statistically similar. Figure 2 plots degree distribution and Table 2 presents some of the commonly used graph metrics for skitter data during this time period.

From the historic skitter AS-level data we make the following observations:

- As is well-known, the degree distribution follows a power-law, $P(k) \sim k^{-\gamma}$, with $\gamma \approx 2.1$, and the maximum degree k_{max} scales almost linearly with the graph size.[3]

- Up to a certain threshold degree, the values of the degree distribution stay the same and the power-law exponent γ remains unchanged with the evolution of the topology. The average degree stays the same due to linear scaling of values of high degrees in the power-law tail.

- The assortativity coefficient, a scalar summary of the $2K$-distribution, and mean clustering, a partial summary of the $3K$-distribution, remain almost constant over the five year period.

- Some other global metrics such as the distance distribution, do not drastically change either.

4.1.2 Router topology data

We next consider the characteristics of router topologies within individual ASes. Unlike AS-level topologies, we do not have access to detailed historic router-level topologies to understand their growth patterns. Instead we gathered router-level topologies for all the ASes observed in the skitter traceroute data from September 1-15, 2006, by executing the following steps:

[3]In fact, the expected maximum among n samples of a random variable distributed according to a power law with exponent $-\gamma$ is $k_{\max} \sim n^{1/(\gamma-1)}$ [3]. For the observed values of γ, this scaling is almost linear since $1/(2.1-1) \approx 0.9$.

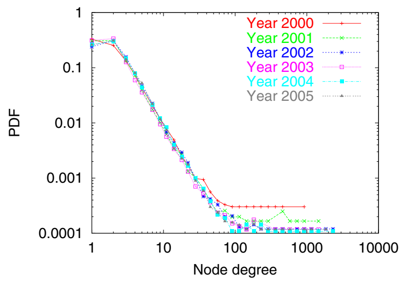


Figure 2: Degree distribution for historical AS-level skitter topologies

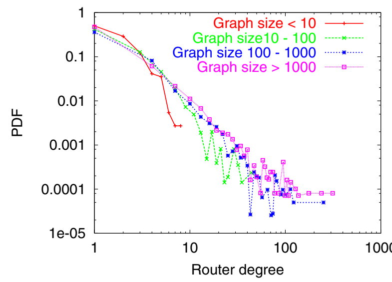


Figure 3: Degree distribution for skitter router-level topologies of different sizes

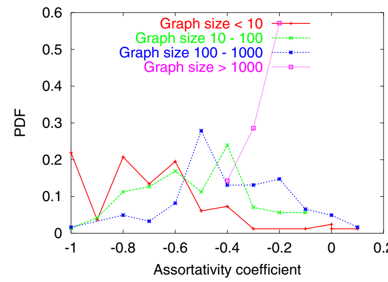


Figure 4: Assortativity for skitter router-level topologies of different sizes

1. From the traceroute traces, we first extract IP links. Using the RouteViews data, we assign the source AS of the longest matching prefix to each IP link.

2. Interface aliases are merged using `iffinder` [4] into routers. `iffinder` sends UDP probe packets to all IP addresses seen in the traces with destination UDP ports set to presumably unused values. If router R receives such a packet from prober P destined to R's IP interface X, while R's route to P goes via some other R's IP interface Y, then R is supposed to reply to P with ICMP `port unreachable` message with the source address set to Y. Prober P can thus conclude that X and Y belong to the same router [15].

3. We translate IP links from step 1 to router links using the results from step 2. Next, we assign AS numbers corresponding to the IP links to the router links. Since all traceroute traces are directed from source to destination, the graph obtained at this step is also directed, *i.e.*, each node is characterized by its in- and out-degree.

4. We discard all nodes with either in-degree 0 or out-degree 0 as majority of nodes removed this way are end hosts.

After applying the above steps, we obtain a router-level topology of the Internet with more than 200,000 routers. This router-level topology has information on AS membership, *i.e.*, the AS number that each link in the graph belongs to. By putting together links that belong to the same AS, we are able to create a router-level topology for each of the ASes observed in the data.

The resulting topology suffers from several limitations. In particular, since skitter sends traceroute probes from the source monitors to the hosts in its destination lists, typically not all routers inside an AS will be discovered. As a result, the router-level topology for a particular AS need not be connected (disjoint traceroutes may discover different portions of the same AS). We extract the giant connected component (GCC) from the router-level topology for each AS and discard ASes where the GCC is less than 75% of the total number of routers in that AS. We consider the connectivity information for these ASes to be insufficient to inform subsequent topology generation. We are left with router-level topologies of varying sizes for approximately 5700 ASes. There are a number of other concerns with the quality of obtained topologies. We list just a few:

- Traceroute explorations are widely known to introduce sampling biases since they find only those links that, roughly, lie in a collection of shortest-path trees rooted at the monitors; other links are missing [9].
- Traceroute probes are also susceptible to degree inflation, since they do not account for layer-2 connectivity. For example, when several routers are connected through a switch, traceroute techniques can mistakenly assume that these routers are directly connected to each other. As a result, the degrees of the routers need not be accurate.
- Several concerns stem from using `iffinder` to map IP interfaces to routers. In particular, while there are no false positives in the mapping, it is difficult to quantify the false negatives.

Despite these limitations, the skitter data remains one of the few sources of router-level data that can be used for topological modeling and inference.

We next smooth the obtained router topology statistics. Specifically, we perform logarithmic binning to group all the router graphs obtained for each AS into the following four categories based on the AS size equal to the number of routers within an AS:

AS category	AS size S
T_1	$S < 10$
T_2	$10 \leqslant S < 100$
T_3	$100 \leqslant S < 1000$
T_4	$S \geqslant 1000$

We plot the degree distributions and assortativity coefficients averaged for each category in Figures 3 and 4, and make the following observations:

- Router graphs with more than 100 nodes have degree distributions that can be loosely approximated by power laws, although the power law is not as good a fit as was observed for AS-level topologies. Graphs smaller than 100 nodes have such scarce degree statistics that any discussion whether their degree distributions follow any specific laws or not is impossible.
- The observed maximum degree does not scale in a similar fashion as observed in historic AS-level graphs. In fact, the maximum degree does not increase significantly with increase in graph size and the observed maximum node degree is about 330 in the largest router graphs. This observation agrees with the intuition behind router topologies: the number of interfaces per router cannot be arbitrarily large; it is bounded by simple technical network design constraints [16].
- We observe no specific and statistically significant values of assortativity coefficients for the obtained router graphs of different sizes.

These observations suggest that our rescaling techniques should be appropriately adjusted to account for the specifics of router topologies. The maximum degree, for example, cannot scale linearly as in the AS topology case. Instead, it should be bounded above by a specific value representing current technological and practical limitations.

4.2 $1K$-rescaling

We base our rescaling techniques on the observations made in Section 4.1. In our $1K$-rescaling, we attempt to preserve the shape of the PDF of the graph's degree distribution. We do so by: i) keeping the proportion of low-degree nodes in the rescaled graph the same as in the original graph, ii) scaling linearly, with the size of a rescaled graph, the values of high degrees, and iii) keeping the number of nodes of rescaled high degrees the same as the corresponding number of nodes in the original graph.

Specifically, our algorithm to rescale $1K$-distributions of AS graphs works as follows.

Input:

- The original graph size n.
- The original degree distribution $P(k)$.
- The new graph size n'.

Output:

- The degree distribution $P'(k')$ in the new graph.

Procedure:

1. Find the low-degree threshold $k_l > 1$ defined as the lowest degree value such that the smoothed degree distribution behaves as $P(k_l + 1) > P(k_l)$, *i.e.*, value k_l is such that statistical noise in $P(k)$ becomes significant for $k > k_l$. We require that k_l be sufficiently large such that the nodes of the remaining degrees account for less than 10% of the total nodes (the degree distribution PDF does not typically follow a power law for the first few degrees and most of the nodes in the graph fall in this range).

2. Find the high-degree threshold k_h defined as the lowest degree value such that the smoothed degree distribution $P(k)$ is a constant function, *i.e.*, value k_h is such that the number of nodes $n(k) = nP(k)$ is approximately the same for all $k > k_h$.

3. Let $k \in \text{supp}(P)$ be the set of degree values k such that $P(k) \neq 0$, and $k'(k) \in \text{supp}(P')$ be the set of corresponding degree values in the new graph.

4. For $k \leqslant k_l$, $k'(k) = k$ and $P'(k) = P(k)$.

5. For $k \geqslant k_h$, $k'(k) = kn'/n$ and $n'(k'(k)) = n(k) \Leftrightarrow P'(kn'/n) = n/n'P(k)$.[4]

6. Let $M = (k_l, k_h)$ be an open interval between k_l and k_h, and $M' = (k_l, k_h n'/n)$ be an open interval between $k'(k_l)$ and $k'(k_h)$. For $k \in M$, we use linear rescaling to glue the two regimes of $P'(k')$ defined at steps 4 and 5 as follows:

   (a) Let u be the number of nodes in the original graph with degrees $k \in M$, $u = n\sum_{k\in M} P(k)$, let $i = 1, \ldots, u$ be the rank of node i in the list of nodes with degrees in M sorted in the order of non-increasing degrees, and let $k(i)$ be the degree of node i. This way $k(1) = k_h$ and $k(u) = k_l$.

   (b) Let u' be the number of nodes in the new graph that should have degrees $k' \in M'$, $u' = n'(1 - \sum_{k'\notin M'} P'(k'))$.

   (c) For nodes $j = 1, \ldots, u'$ in the new graph, compute their linearly rescaled degree values by $k'(j) = \left(\frac{1-n'/n}{u'-1}(j-1) + \frac{n'}{n}\right) k\left[\frac{u-1}{u'-1}(j-1)+1\right]$.

   (d) Let $l(x)$ be a linear function such that $l(k'_l) = P'(k'_l)$ and $l(k'_h) = P'(k'_h)$, *i.e.*, $l(x) = ax + b$, where $a = (n/n'P(k_h n'/n) - P(k_l))/(k_h n'/n - k_l)$ and $b = P(k_l) - ak_l$. Let ρ be a small random variable uniformly and symmetrically distributed around 0. For values k' in the new graph produced by step 6c, compute the corresponding values of the degree distribution by $P'(k') = c(l(k') + \rho)$, where the constant c is determined from the normalization condition for the whole $P'(k')$, *i.e.*, $\sum_{k'\in\text{supp}(P')} P'(k') = 1$.

[4]Rounding to closest integers is assumed whenever needed.

We then supply the output degree distribution $P'(k')$ as input to the $1K$-random topology generation algorithms described in [18] to obtain the final graph. Similar to the methodology followed in [18], we extract the GCC from the generated graph.

To rescale router topologies of sizes smaller than 100 nodes, we also maintain the degree distribution of the given graph. Rescaling router topologies larger than 100 nodes, we impose an additional constraint on the maximum degree to not exceed 330. We stress that this value can be configured by the user based on better data for Internet connectivity or updated router technology information.

4.3 $2K$-rescaling

The main idea behind our $2K$-rescaling is the same as in the $1K$ case—we try to preserve the shape of the $2K$-distribution $P(k_1, k_2)$. In other words, we want the degree correlation "profile" of rescaled graphs be similar to the original. We preserve it by means of the following algorithm:

Input:

- The original graph size n.
- The original joint degree distribution (JDD) $P(k_1, k_2)$.
- The new graph size n'.

Output:

- The JDD in the new graph $P'(k'_1, k'_2)$.

Procedure:

1. Compute the $1K$-distribution from the given $2K$-distribution by $P(k) = \bar{k}/k \sum_{k_1} P(k, k_1)$.

2. Rescale $P(k)$ to $P'(k')$ of the new graph as in Section 4.2.

3. Let $\hat{k}'(k)$ be the mapping between the old and new degree values induced by $1K$-rescaling. Specifically, let $M = (k_l, k_h)$ be as in Section 4.2. If $k \notin M$, then $\hat{k}'(k) = k'(k)$ from steps 4 and 5 in Section 4.2. Otherwise, if $k \in M$, $\hat{k}'(k)$ is given by the degree mapping from step 6c in Section 4.2.

4. Let $X = |\text{supp}(P)|_M$ and $X' = |\text{supp}(P')|_{M'}$ be the sizes of supports of $P(k)$ and $P'(k')$ within the M and M' intervals respectively. When scaling up, $n' > n$, $X' \geqslant X$, and

the $2K$-distribution is computed as follows:

$$P'(\hat{k}'_1(k_1), \hat{k}'_2(k_2)) = \begin{cases} P(k_1, k_2), & \text{if } k_1 \notin M \text{ and } k_2 \notin M, \\ \left(\frac{X}{X'}\right)^2 P(k_1, k_2), & \text{if } k_1 \in M \text{ and } k_2 \in M, \\ \left(\frac{X}{X'}\right) P(k_1, k_2), & \text{otherwise.} \end{cases}$$

When scaling down, $n' < n$, $X' \leqslant X$, and

$$P'(k'_1, k'_2) = \sum_{k_{1,2} | \hat{k}'_{1,2}(k_{1,2}) = k'_{1,2}} P(k_1, k_2)$$

That is, the JDD shapes before and after rescaling are the same, while the above expressions guarantee that the JDDs of both original and new graphs are properly normalized.

As in the $1K$ case, we then supply the produced $2K$-distribution $P'(k'_1, k'_2)$ to $2K$-random graph construction algorithms from [18] to obtain the final graph.

4.4 $1K + r$-rescaling

We have also experimented with a rescaling technique lying somewhat in-between $1K$- and $2K$-rescaling. The motivation for it is that for the AS-level graphs, the assortativity coefficient r, a summary statistic of the $2K$-distribution, has remained roughly constant over time (see Table 2). We can thus perform $1K$-rescaling and then move the resulting graph to a target value of r by a sequence of $1K$-preserving r-targeting rewirings. At each rewiring step, we select a random pair of edges (v_1, v_2) and (v_3, v_4) and rewire them to the pair (v_1, v_4) and (v_2, v_3) only if the $1K$-distribution does not change and the value of r after rewiring is closer to its target value than before (see [18] for further details).

Compared to $2K$-rescaling, this technique is simpler and can easily be extended to higher-order statistics since we can always compute scalar summaries of a given dK-distribution [18]—the mean clustering and correlation of degrees of nodes located at distance 2 from each other are such summaries for the $3K$-distribution, for example. However, this simplicity comes with a price: no scalar metric can capture all of the information contained in, for instance, a $2K$-distribution (that must be encoded as a matrix). Hence, $1K + r$-rescaling may give up some accuracy. We present detailed comparison of our rescaling techniques in Section 6.

5. AS ANNOTATIONS

Given our ability to generate graphs of a range of sizes, in this section we describe techniques for annotating generated router-level topologies with AS membership information. The traceroute data described in Section 4.1 includes information on both ASes and router connectivity. Given AS annotation information in an original router topology, there are two possible techniques for maintaining AS annotations in the randomly generated rescaled graph. The first, bottom up technique, would simply rescale the input router topology and then devise techniques to "grow" contiguous ASes to match some target number of ASes, making some assumptions about how the number of ASes scales with the total number of routers and using observations of the number of routers per AS in the original topology.

Unfortunately, we could not devise any straightforward techniques for filling in the details of this bottom up technique. Thus, we propose a top down technique for generating a rescaled, annotated topology. This technique consists of the following high-level phases illustrated in Figure 5:

1. Generate AS-level topology of desired size.
2. Populate each AS with a router-level topology using information on correlations between AS degrees and sizes measured by the number of routers within an AS.
3. Select peering (*i.e.*, inter-AS or border) routers for each AS based on the peering router statistics extracted from the traceroute data.
4. Glue per-AS router topologies into a global router topology by connecting peering routers.

While we describe details of our methodology in the context of skitter below, we note that our approach is general to a variety of data sources. For example, we could use Rocketfuel [27] or iPlane [17] data to generate router-level topologies for each AS.

To provide more details about our annotation techniques, we first describe some additional post-processing of the skitter data from Section 4.1. In addition to size-based AS categories $T_1, \ldots, T_4$ from Section 4.1.2, we also use logarithmic binning to coarsely smooth the AS degree distribution and split all ASes into the following degree-based classes C_1, C_2, C_3:

AS class	AS degree K
C_1	$K < 10$
C_2	$10 \leqslant K < 100$
C_3	$K \geqslant 100$

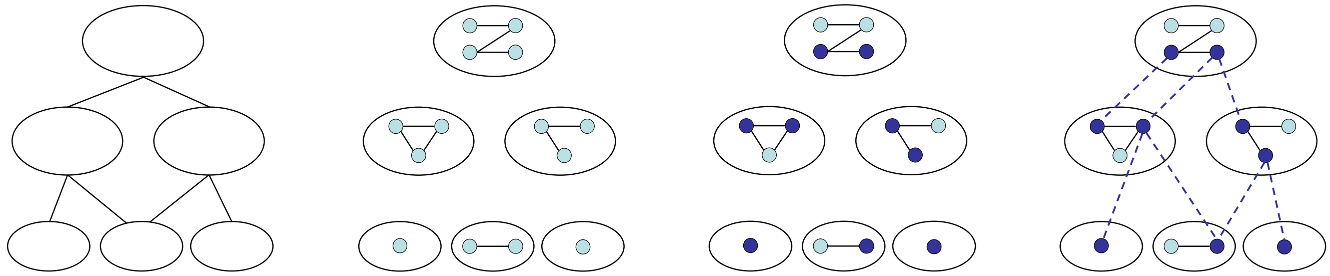
Note that after the data processing in Section 4.1.2, we do not have ASes with $K > 1000$.

We next statistically relate AS classes $c = C_1, C_2, C_3$ and categories $t = T_1, \ldots, T_4$. Let $A(c, t)$ be the set of ASes of class c and category t. For each combination (c, t), we keep the following statistics:

- The number of ASes $N(c, t)$ in $A(c, t)$.
- The collection $J_a(c, t)$, $a = 1, \ldots, N(c, t)$, of the intra-AS router topology JDDs of all ASes in $A(c, t)$. Each $J_a(c, t)$ denotes the whole $2K$-distribution $P(k_1, k_2)$ of the router topology of AS $a \in A(c, t)$.
- The corresponding collection $D_a(c, t)$ of the peering router degree distributions. For each AS a, $D_a(c, t) = P_B(k)$, where $P_B(k)$ is proportional to the number of routers within AS a that have degree k and that peer with routers in other ASes.

Having prepared the statistics above, we generate rescaled AS-annotated router graphs as follows:

1. Rescale the AS-level graph as in Section 4 (Figure 5(a)).
2. For each AS A in the rescaled AS graph:
   (a) Determine A's class c.
   (b) Randomly select A's category t with a conditional probability proportional to $N(c, t)$.
   (c) Select an AS a from $A(c, t)$ uniformly at random.
   (d) Populate A with a $2K$-random router topology based on $P(k_1, k_2) = J_a(c, t)$ (Figure 5(b)).
   (e) Choose peering routers within AS a having degree k based on a probability proportional to $P_B(k) = D_a(c, t)$ (Figure 5(c)).
3. Walk over the pairs of adjacent ASes in the rescaled AS graph and connect random pairs of designated peering routers in each AS.



(a) Generating AS-graph

(b) Generating router graphs for each AS

(c) Assigning peering routers for each AS

(d) Connecting peering routers

Figure 5: Generating a router-level topology annotated with AS-membership

6. EVALUATION

In this section, we conduct a number of experiments to demonstrate the ability of our approach to reproduce important graph metric values discussed in Section 2. Using these metrics, we compare our generated topologies with the original observed topologies extracted from the data described in Section 4.1. We use notations in Table 1. The results below represent averages over 10 generated graphs in each case. The standard deviations for all the metrics from their mean values are negligible.

Table 3: Scalar metrics for $1K$-rescaled skitter AS topologies (Section 4.2)

	Number of nodes					
Metric	Original (9200)	9200	6000	12000	15000	30000
k	6.29	6.34	6.17	6.38	6.35	6.44
r	-0.24	-0.24	-0.23	-0.23	-0.22	-0.21
$\bar{C}$	0.46	0.25	0.23	0.25	0.27	0.27
$\bar{d}$	3.12	3.11	3.18	3.15	3.13	3.1
σ_d	0.37	0.4	0.44	0.42	0.41	0.39
λ_1	0.1	0.03	0.1	0.09	0.09	0.08
λ_{n-1}	1.9	1.97	1.89	1.9	1.9	1.93

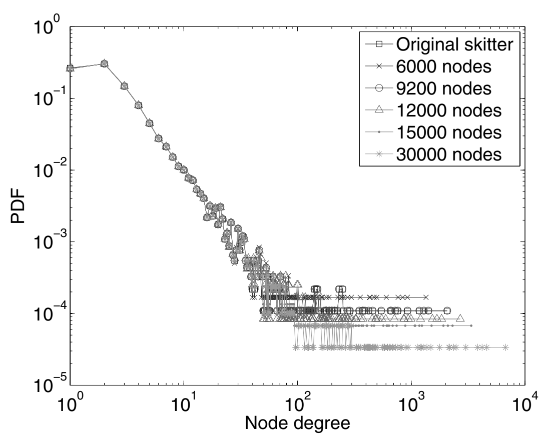


Figure 6: Degree distribution for $1K$-rescaled skitter AS topologies

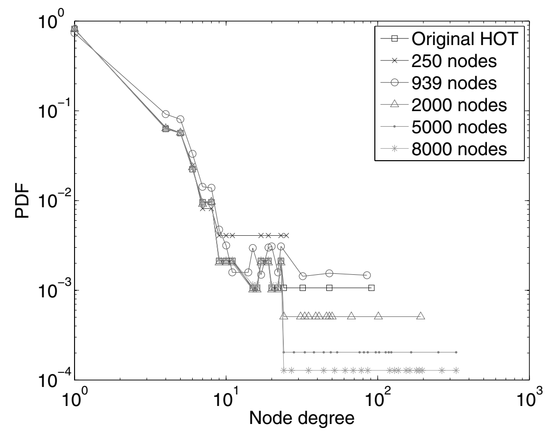


Figure 7: Degree distribution for $1K$-rescaled HOT topologies

6.1 $1K$-rescaling for AS-graphs

The skitter AS-level topology for March 2004 has 9200 ASes, with an average degree of 6.29 and an assortativity coefficient of -0.24. As shown in [18], a $1K$-random graph with 9200 nodes reproduces most metric values of the original skitter topology except for clustering. We generate random graphs of varying sizes using the $1K$-rescaling algorithm and summarize our results in Table 3. In Figure 6, we plot the degree distribution of these different-sized graphs. We find that the metric values are invariant for most graph sizes, and they closely match the corresponding values of the input skitter topology. The average degree, average distance and the assortativity coefficient remain constant even as the graph grows or shrinks in size. The distance distribution across different-sized skitter graphs is the same as that of the original skitter graph, hence we do not show the plot for brevity. We generated larger size graphs containing up to 80,000 nodes and note that the average degree and assortativity coefficient values are maintained for these graphs as well.

6.2 $1K$-rescaling for router-graphs

Next, we experimented with a synthetic router-level topology, the HOT graph from [16]. As shown in [18], $2K$-random graphs reproduce metric values of the original graph much better than their $1K$-random counterparts. The original HOT graph has 939 nodes with an assortativity coefficient of -0.22. We generate graphs of a variety of sizes for the HOT topology using our $1K$-rescaling tech-

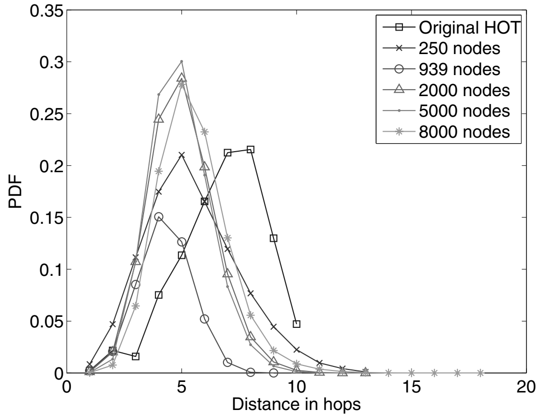


Figure 8: Distance distribution for $1K$-rescaled HOT topologies

Table 4: Scalar metrics for $1K$-rescaled HOT graphs (Section 4.2)

	Number of nodes					
Metric	Original (939)	939	250	2000	5000	8000
k	2.1	2.58	2.09	2.24	2.41	2.52
r	-0.22	-0.14	-0.32	-0.11	-0.1	-0.11
$\bar{C}$	0	0.009	0.002	0.002	0.005	0.006
$\bar{d}$	6.81	4.41	5.4	5.02	4.94	5.4
σ_d	0.57	0.72	1.09	0.91	0.86	1.01
λ_1	0.004	0.034	0.006	0.013	0.015	0.007
λ_{n-1}	1.997	1.967	1.994	1.987	1.985	1.994

nique for router-graphs described in Section 4.2. We report metric values for different-sized HOT graphs in Table 4 and Figures 7 and 8. Unlike AS-level topologies, we notice that the $1K$-random graphs do not accurately reproduce most metric values such as assortativity coefficient and average distance. These observations, once again, verify that reproducing HOT's $1K$-distribution is insufficient for accurate capturing important global properties of the HOT topology [18].

6.3 $2K$-rescaling for AS-graphs

Next, we generate different-sized skitter graphs using our $2K$-rescaling algorithm (Section 4.3) and present metric values for these graphs in Table 5. As in the $1K$ case, we notice that for variable-sized $2K$-random graphs, all metric values accurately mimic that of the original skitter graph, except for clustering. Earlier work [18] shows that clustering can be reproduced by a $3K$-generator. We have not yet implemented $3K$-rescaling, though we note that employing rewiring toward a target clustering value should result in appropriate clustering values for the rescaled graphs. For brevity, we do not plot the degree distribution of these graphs as they match their $1K$-counterparts. Finally, Figure 9 plots the distance distribution of these graphs. We see that for all the graphs, the distance distribution remains almost unchanged, matching the trends in the input graph data for a variety of graph sizes.

6.4 $1K + r$-rescaling for AS-graphs

Next, we generate the graphs using our $1K$-rescaling algorithm and then subject them to the $1K$-preserving r-targeting rewiring

Table 5: Scalar metrics for $2K$-rescaled skitter AS topologies (Section 4.3)

	Number of nodes					
Metric	Original (9200)	9200	6000	12000	15000	30000
k	6.29	6.29	6.09	6.31	6.35	6.44
r	-0.24	-0.24	-0.22	-0.23	-0.23	-0.22
$\bar{C}$	0.46	0.29	0.26	0.28	0.28	0.27
$\bar{d}$	3.12	3.08	3.1	3.1	3.1	3.12
σ_d	0.37	0.35	0.38	0.36	0.35	0.35
λ_1	0.1	0.15	0.13	0.13	0.15	0.12
λ_{n-1}	1.9	1.85	1.87	1.88	1.88	1.91

Table 6: Scalar metrics for $1K + r$-rescaled skitter AS topologies (Section 4.4)

	Number of nodes				
Metric	Original (9200)	6000	12000	15000	30000
k	6.29	6.17	6.38	6.35	6.44
r	-0.24	-0.24	-0.24	-0.24	-0.24
$\bar{C}$	0.46	0.24	0.26	0.25	0.25
$\bar{d}$	3.12	3.1	3.09	3.09	3.2
σ_d	0.37	0.24	0.36	0.33	0.34
λ_1	0.1	0.13	0.13	0.15	0.12
λ_{n-1}	1.9	1.88	1.88	1.89	1.91

process. The process terminates upon reaching required value of r. We present the metric values for the skitter graphs in Table 6. Since all the generated $1K$-random skitter graphs have values for r close to the required r value of the original graph, a few rewirings are sufficient for all the graphs to reach their target state. In the case of skitter, the loss of accuracy from using the r value instead of the entire JDD matrix appears minimal. In fact, for all metrics we considered, the r-targeting rewiring performs as well as the $2K$-rescaling technique. The reason behind this effect is that the skitter AS topology is almost $1K$-random [18], *i.e.*, it can be accurately captured using only its $1K$-distribution.

6.5 $2K$-rescaling for router-graphs

Next, we generate different-sized $2K$-random HOT graphs using our $2K$-rescaling technique. We present the metric values for these graphs in Table 7 and plot the distance distribution in Figure 10 and the normalized betweenness distribution in Figure 11. Betweenness is a hard metric to reproduce for the HOT graph, but all the 2K random-graphs reproduce the shape of the betweenness curve exactly. The difference in the betweenness values for these graphs is due to the difference in their sizes. The $2K$-random hot graphs better reproduce the metric values of the original HOT graph than the $1K$-random HOT graphs, even when scaling up by a factor of 10 or scaling down by a factor of 4.

6.6 $1K + r$-rescaling for router-graphs

Next, we present results for our $1K + r$ rescaling of router graphs. Unlike $1K$-random AS graphs, for the $1K$-random HOT graphs, reaching the target r values takes longer as the r values of $1K$-random HOT graphs are not close to the target value. We report some of the scalar metric values for the HOT graph in Table 8. While most metric values are similar to the HOT graphs from the $2K$-rescaling technique, the average distance value is higher for all

Table 7: Scalar metrics for $2K$-rescaled HOT graphs(Section 4.3)

	Number of nodes					
Metric	Original (939)	939	250	2000	5000	8000
k	2.1	2.18	2.2	2.19	2.3	2.41
r	-0.22	-0.23	-0.36	-0.19	-0.18	-0.18
$\bar{C}$	0	0.0001	0.0005	0.0004	0.0005	0.0001
$\bar{d}$	6.81	6.32	5.4	6.4	6.6	6.92
σ_d	0.57	0.71	0.84	0.83	0.97	1.02
λ_1	0.004	0.005	0.01	0.004	0.004	0.005
λ_{n-1}	1.997	1.996	1.986	1.997	1.996	1.996

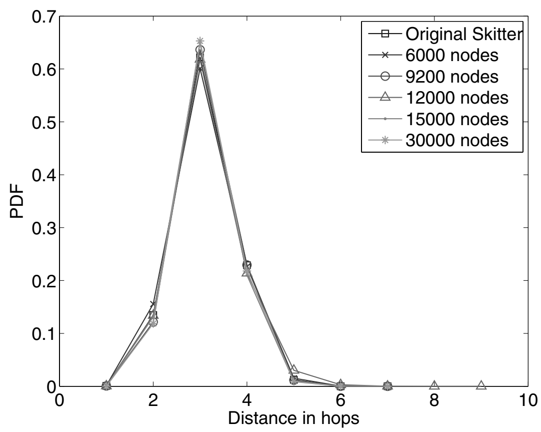


Figure 9: Distance distribution for $2K$-rescaled skitter AS topologies

the graphs obtained using the r-targeting technique. This increase in value for the average distance and, correspondingly, in the distance distribution results from employing the scalar summary of the $2K$-distribution instead of the entire function. Since the HOT graph is not $1K$-random but almost $2K$-random [18], the full information contained in its $2K$-distribution is required to accurately capture all its global properties. Of course, its $1K + r$-random version must closer to the original than the $1K$-random version, but farther than the $2K$-random one.

6.7 AS annotations

Given our ability to accurately rescale graphs, we now evaluate our techniques for performing AS-membership annotations in our generated router topologies. Since this topology combines both router and AS information, in addition to evaluating metrics for the overall graph, we present results on the fraction of peering-routers within an AS, and degree distribution of routers within an AS.

We classify the ASes in the AS-level topology based on their degrees (see Section 5). Our AS-level topology consists of a total of 5662 ASes. Of all the ASes, 97% belong to class C_1, 2.6% belong to class C_2, and the remaining ASes belong to class C_3. Figure 12 plots the PDF for the number of routers in an AS for every AS class. As expected, we observe that the majority of the ASes belonging to class C_1 have fewer than 10 routers in their router-level topology. The distribution of the number of routers belonging to C_1-ASes appears more widespread than for C_2- and C_3-ASes. The maximum number of routers observed for a C_1-AS is 1774. The number of

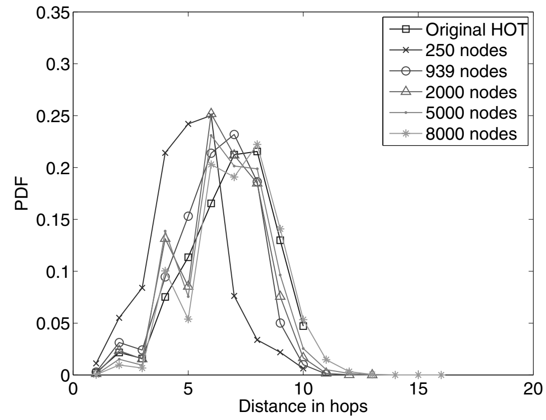


Figure 10: Distance distribution for $2K$-rescaled HOT topologies

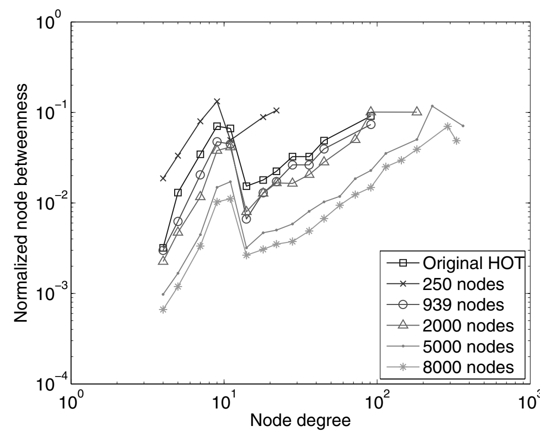


Figure 11: Betweenness distribution $2K$-rescaled HOT topologies

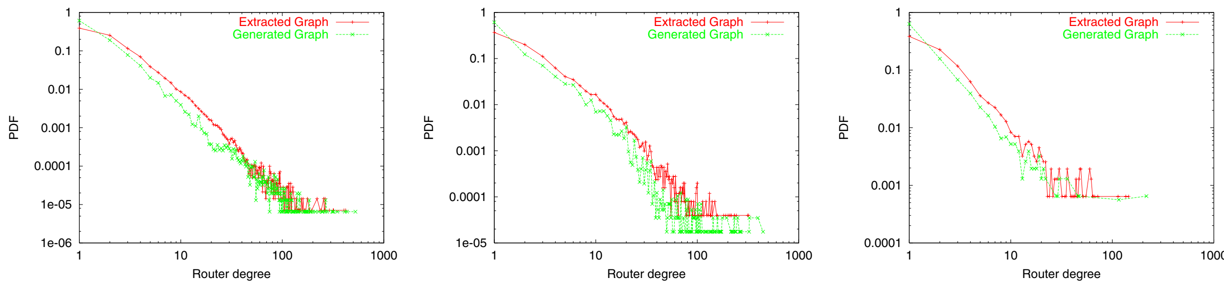
routers for C_2-ASes ranges from 10 to 1996. For C_3-ASes, the minimum number of routers is 150. This agrees well with the previous results relating AS degrees and sizes [28].

Next, we present the degree distributions of all routers within an AS for each AS class for the original and generated graphs in Figures 13(a), 13(b), and 13(c). These plots allow us to compare how close the extracted and generated topologies match with respect to the degree distribution of all the routers belonging to each AS class. We observe reasonably good matches between the two for all AS classes.

Finally, Table 9 presents some scalar metric values to compare the generated combined AS+router-level graph with the corresponding topology extracted from the skitter traces. These metrics are for the overall router-level topology across all of the ASes for both the original as well as the generated graphs. We randomly choose 50,000 unique paths from our original and generated graphs and compute the average distance for these sampled paths using shortest-path algorithm. We note that average distance for the generated graph compares well with that of the original graph.

7. DISCUSSION AND CONCLUSIONS

While we are satisfied with our ability to both rescale Internet topologies and to annotate them, there are a number of interesting



(a) Degree distribution for class C_1 (b) Degree distribution for class C_2 (c) Degree distribution for class C_3

Figure 13: Degree distribution for all routers belonging to an AS class

Table 8: Scalar metrics for $1K + r$-rescaled HOT graphs (Section 4.4)

	Number of nodes				
Metric	Original (939)	250	2000	5000	8000
k	2.1	2.09	2.24	2.41	2.52
r	-0.22	-0.22	-0.2	-0.19	-0.18
$\bar{C}$	0	0.005	0.0006	0.0003	0.0005
$\bar{d}$	6.81	5.4	7.3	7.41	7.8
σ_d	0.57	0.89	0.82	.73	0.74
λ_1	0.004	0.01	0.001	0.002	0.003
λ_{n-1}	1.997	1.989	1.998	1.998	1.998

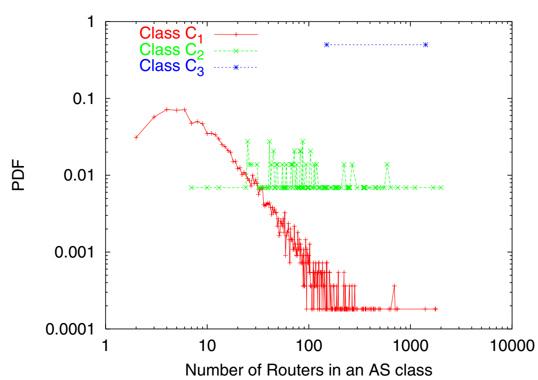


Figure 12: PDF of number of routers in an AS class

questions that we intend to explore as future work. For example, given our heuristics for rescaling, one question is the maximum rescaling factor that we can safely apply to a given graph. We believe that given the data sources available to us and our analysis of this data, scaling by a factor of approximately 10 and perhaps 100 is meaningful using our methodology. However, it would not, for instance, be meaningful to scale a 10-node topology to one with a million nodes. Such scaling would likely require a fundamental understanding of the laws governing the evolution of a given graph [7] rather than the observation-based techniques we employ in our rescaling techniques. Given that we do not yet have a firm grasp of such evolutionary laws, our methodology and our generator that combines both AS and router-level information presents

Table 9: Scalar metric values for router-level topologies annotated with AS membership

	k	r	k_{max}	d
Original	4.25	0.006	1141	9.14
Generated	3.9	-0.0009	1333	8.53

an intermediate solution for researchers interested in evaluating the performance of services and protocols for a range of graph sizes.

Our ability to reproduce observed input graph properties is limited by the quality of available Internet measurements. In fact, we found limitations with existing measurements that impacted our ability to reconstruct Internet graph connectivity characteristics. However, the methodology and algorithms for generating annotated, rescaled topologies constitute the primary contributions of this work, and not necessarily the particular generated graphs. While we can verify that our topologies match observed router topology characteristics, we cannot claim that either the input or generated topologies accurately reflect reality. The quality of our generated topologies will improve with the quality of available measurements.

At the same time, improvements in Internet measurement techniques or some future network architecture that exports topology will not obviate the need to generate random network topologies for at least two reasons. First, even with complete knowledge of network topology, we still require techniques to either scale the graphs up (e.g., to understand how routing behavior scales with graph size) or down (e.g., to serve as manageable input for simulations, emulations, or testbed deployments). Second, our methodologies for random graph generation enable rewiring techniques to explore variations in a particular graph property/parameter while holding all other graph characteristics steady. Even with much improved understanding of Internet topologies, research ranging from routing, to congestion control, to overlay protocols would benefit from quantifying behavioral sensitivity to alternate network topologies. These alternate network topologies may be a consequence, for instance, of a shift in the evolutionary behavior of the Internet or some suspected bias in the measurement methodology.

This paper restricts its attention to AS annotations for generated router topologies. We are currently investigating a number of techniques to also support annotations for link latency and capacity based on available measurement sources. It remains to be seen whether such annotations can match the distributions found in real networks but we are encouraged by available data sets both for end-to-end and per-hop latency distributions. Similarly, data sources are available for distributions of access link bandwidths and network capacity in the core of the Internet.

Overall, we believe that our topology generator will serve as valuable input to a range of research studies. We outline a number of cases here. Studying routing protocol scalability and convergence requires knowledge of both topology and AS relationships and hence our work can serve as valuable input to such work. Many studies of congestion control protocols employ simple "dumbbell"-style topologies. While such simple topologies are appropriate starting points, it will be valuable to consider more complex topologies, for instance with more variable round trip times and multiple, changing bottlenecks. Many overlay and peer-to-peer systems attempt to create application-level logical topologies that match the characteristics of the underlying network. Similarly, developing network coordinates [8, 23] and geo-localization [6] has recently become an important research area. Our topology generator can supply a range of inputs and potential deployment scenarios in support of such studies.

Emerging network testbeds such as VINI [2] and GENI [14] will enable network topology configuration for deployed systems running across the wide area. Once again, running with a range of topologies, scaled to fit available resources, will allow more accurate conclusions to be drawn for emerging network architectures. Finally, multiple aspects of network security efforts, including defenses against denial of service attacks and large-scale worm outbreaks depend on network topology. The ability to both experiment with a range of random graphs that match Internet characteristics to understand the sensitivity of particular techniques to network topology (and to variations in network topology) will be of significant value.

Acknowledgments

We wish to thank the anonymous referees and our shepherd, Walter Willinger, for their valuable feedback on earlier drafts of this paper. This research was supported in part by a National Science Foundation Grant (CNS-0627145) and UCSD's Center for Networked Systems.

8. REFERENCES

[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *STOC*, 2000.

[2] A. Bavier, N. Feamster, M. Huang, L. Peterson, and J. Rexford. In VINI Veritas: Realistic and controlled network experimentation. In *SIGCOMM*, 2006.

[3] M. Boguñá, R. Pastor-Satorras, and A. Vespignani. Cut-offs and finite size effects in scale-free networks. *European Physical Journal B*, 38:205–209, 2004.

[4] CAIDA. Iffinder. `http://www.caida.org/tools/measurement/iffinder/`.

[5] CAIDA. Macroscopic topology AS adjacencies. `http://www.caida.org/tools/measurement/skitter/as_adjacencies.xml`.

[6] M. Casado and M. J. Freedman. Peering through the shroud: The effect of edge opacity on IP-based client identification. In *USENIX*, 2007.

[7] H. Chang, S. Jamin, and W. Willinger. To peer or not to peer: Modeling the evolution of the Internet's AS-level topology. In *INFOCOM*, 2006.

[8] F. Dabek, R. Cox, F. Kaashoek, and R. Morris. Vivaldi: A decentralized network coordinate system. In *SIGCOMM*, 2004.

[9] L. Dall'Asta, I. Alvarez-Hamelin, A. Barrat, A. Vázquez, and A. Vespignani. Exploring networks with traceroute-like probes: Theory and simulations. *Theoretical Computer Science, Special Issue on Complex Networks*, 2005.

[10] DETER. `http://www.isi.edu/deter/`.

[11] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, 1959.

[12] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *SIGCOMM*, 1999.

[13] L. Gao and F. Wang. The extent of AS path inflation by routing policies. In *IEEE Global Internet Symposium*, 2002.

[14] GENI: Global environment for network innovations. `http://www.geni.net/`.

[15] R. Govindam and H. Tangmunarunkit. Heuristics for Internet map discovery. In *INFOCOM*, 2000.

[16] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the Internets router-level topology. In *SIGCOMM*, 2004.

[17] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani. iPlane: An information plane for distributed services. In *OSDI*, 2006.

[18] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. In *SIGCOMM*, 2006.

[19] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, kc claffy, and A. Vahdat. The Internet AS-level topology: Three data sources and one definitive metric. *Computer Communication Review*, 36(1), 2006.

[20] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang. On AS-level path inference. In *SIGMETRICS*, 2005.

[21] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *MASCOTS*, 2001.

[22] M. E. J. Newman. Assortative mixing in networks. *Physical Review Letters*, 89:208701, 2002.

[23] T. S. E. Ng and H. Zhang. A network positioning system for the Internet. In *USENIX*, 2004.

[24] The network simulator `ns-2`. `http://www.isi.edu/nsnam/ns/`.

[25] University of Oregon RouteViews Project. `http://www.routeviews.org/`.

[26] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *SIGCOMM*, 2003.

[27] N. Spring, R. Mahajan, and D. Wetherall. Measuring ISP topologies with Rocketfuel. In *SIGCOMM*, 2002.

[28] H. Tangmunarunkit, J. Doyle, R. Govindan, S. Jamin, W. Willinger, and S. Shenker. Does AS size determine AS degree? *Computer Communication Review*, 31(5), 2001.

[29] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on Internet paths. In *INFOCOM*, 2001.

[30] A. Vahdat, K. Yocum, K. Walsh, P. Mahadevan, D. Kostic, J. Chase, and D. Becker. Scalability and accuracy in a large-scale network emulator. In *OSDI*, 2002.

[31] B. M. Waxman. Routing of multipoint connections. *IEEE JSAC*, 1988.

[32] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. An integrated experimental environment for distributed systems and networks. In *OSDI*, 2002.

[33] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report UM-CSE-TR-456-02, University of Michigan, 2002.

[34] E. Zegura, K. Calvert, and S. Bhattacharjee. How to model an internetwork. In *INFOCOM*, 1996.