Inet: Internet Topology Generator

Cheng Jin Qian Chen Sugih Jamin[*]
Department of EECS
University of Michigan
Ann Arbor, MI 48109-2122
{*chengjin,qianc,jamin*}*@eecs.umich.edu*

Abstract

Network research often involves the evaluation of new application designs, system architectures, and protocol implementations. Due to the immense scale of the Internet, deploying an Internet-wide system for the purpose of experimental study is nearly impossible. Instead, researchers evaluate their designs using generated random network topologies. In this report, we present a topology generator that is based on Autonomous System (AS) connectivity in the Internet. We compare the networks generated by our generator with other types of random networks and show that it generates topologies that best approximate the actual Internet AS topology.

1 Using Inet

NAME

inet - an AS-level Internet topology generator.

SYNOPSIS

inet -n N [-d k] [-p n] [-s sd] [-f of]

DESCRIPTION

The "Inet" generator generates an AS-level representation of the Internet with qualitatively similar connectivity. It is important to note that Inet only provides the connectivity information; the generated topologies do not have any information pertaining to latency, bandwidth etc. It generates random networks with characteristics similar to those of the Internet from November 1997 to June 2000, and beyond. The generator should be used to generate network of no less than 3037 nodes, which is the number of ASs on the Internet in November 1997. The software package with source code for Unix, can be found at:
`http://topology.eecs.umich.edu/inet/`

[*] This project is funded in part by NSF grant number ANI-0082287. Sugih Jamin is further supported by the NSF CAREER Award ANI-9734145 and the Presidential Early Career Award for Scientists and Engineers (PECASE) 1998. Additional funding is provided by AT&T Labs-Research, and by equipment grants from Sun Microsystems Inc. and Compaq Corp.

OPTIONS

-n N**:** the total number of nodes in the topology.
-d k**:** the fraction of degree-one nodes. Default is 0.3.
-p n**:** the size of the plane used for node placement. Default is 10,000.
-s sd: the seed to initialize the random number generator. Default is 0.
-f of: the debugging output file name. Default is `stderr`.

EXAMPLES

To generate a 6,000-node network with default values:

```
    example% inet -n 6000 > Inet.6000
```

To generate a 6,000-node network on a 10,000 by 10,000 plane with 30% of total nodes as degree-one nodes, random seed of 16, and debug file `debug`:

```
    example% inet -n 6000 -f debug -d .3 -s 16 -p 10000 > Inet.6000
```

OUTPUT FORMAT

The output of `inet` follows the following format:

```
nodes links
...
id x y
...
id1 id2 weight
```

The first line of the output specifies the number of nodes with `nodes`, and the number of undirected links with `links`. The next section contains the location of each node. Each line contains the node id, `id`, and the xy-coordinate, `x` and `y`. The last section contains the list of links in the topology. Each line contains the node ids of the two end points of the link, `id1` and `id2`, and the link weight, `weight`.

2 Overview

The need for realistic random topologies in simulations has long been recognized by researchers working on routing and multicast protocols, e.g. [BE90, ZGLA91, WE94]; more recently, the need for realistic random topologies has also been voiced by researchers studying traffic dynamics and protocol behavior [MS94, FGHW99, F$^+$00]. In recognition of this need, several topology generators have been proposed in the literature, e.g. [Wax88, Doa96, CDZ97, J$^+$00, MM00] (see Section 5 for a more detailed description of these generators). To illustrate the current state-of-the-art in the area of generating Internet-like topologies, we generate topologies using each of these available generators and evaluate how well they resemble the Internet AS topology based on the properties described in [FFF99]. Our analysis showed that the characteristics of these synthetically generated topologies significantly deviate from those of the actual Internet in one or more ways. The results of this comparative study are presented in Section 5.

In this report, we describe an updated Internet AS-level topology generator, *Inet-2.0*, (version 1.0 was presented in [J$^+$00]). We show that Inet-2.0 generates topologies with characteristics more faithful to Internet topologies of comparable sizes.

Table 1: A sample of ASconnlist data.

127	$->$	4	:226:127:127:297
131	$->$	3	:1852:701:3561
132	$->$	3	:668:7170:1914
137	$->$	7	:5441:3561:6682:5443:5455:5448:5445

3 The Data Sets

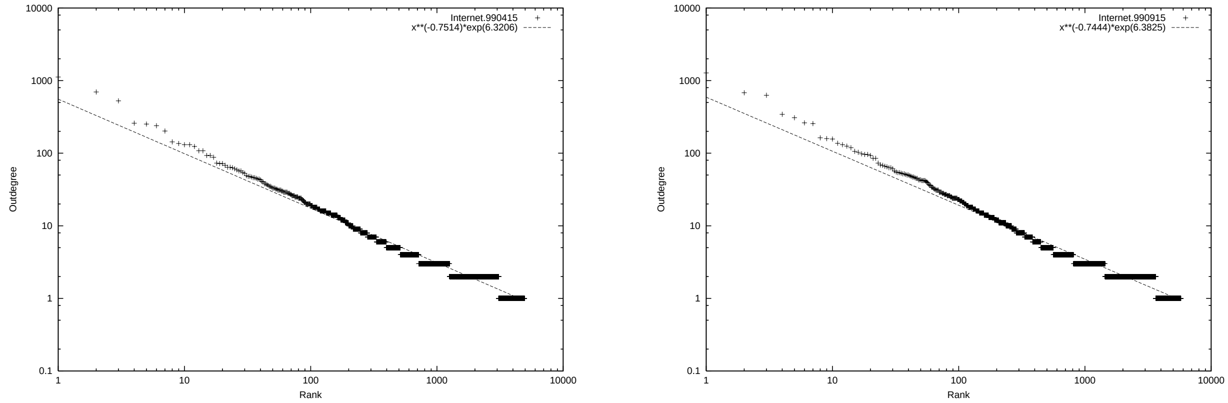
An Autonomous System (AS) is a network under a single administrative authority. ASs connect to each other through border routers, so the Internet can be considered as consisting of interconnected ASs. Within each AS, the network could be further divided into subnetworks connected by internal routers. Hence, the Internet can either be modeled as a graph where each node represents a router, or as a graph where each node represents an AS. In building Inet-2.0, we model Internet as a network of interconnected ASs.

Border routers exchange BGP (Border Gateway Protocol) route updates to propagate reachability information, which is stored in a routing table in each BGP router. Starting from November 1997, The National Laboratory for Applied Network Research (NLANR) [NLA99] collects BGP routing tables once a day from the route server `route-views.oregon-ix.net`. This route server connects to several operational routers for the sole purpose of obtaining routing tables. NLANR processes the routing tables to generate several sets of statistics, one of which (*ASconnlist*) lists the neighboring ASs each AS is connected to. For our analysis, we obtain 32 sets of ASconnlist, each on the 15th of a month starting from November 1997. Table 1 shows a sample ASconnlist segment. Each line has an AS number, its degree of connectivity (or *outdegree*), and the ASs it is connected to. For example, the second line of Table 1 says that AS131 has an outdegree of 3 and is connected to AS1852, AS701, and AS3561. Notice that AS127 is listed as having an outdegree of 4 and is connected to itself twice. Our analysis script removes such self-referential and duplicate entries. A more detailed explanation of the data reduction process can be obtained from `http://moat.nlanr.net/AS/background.html`.

¿From studying NLANR's ASconnlist statistics of November 1997, April 1998, and December 1998, the authors of [FFF99] concluded that the outdegree d_v, of a node, v, is proportional to the rank of the node on a sorted list in decreasing order of node outdegree, r_v, raised to the power of a constant, R (Power-Law 1): $d_v \propto r_v^R$. In [FFF99] the frequency of a node outdegree is said to be proportional to the outdegree raised to the power of a constant, O (Power-Law 2): $f_d \propto d^O$.

We verify our handling and processing of the ASconnlist data by reproducing the results presented in [FFF99] from the same data sets. Figs. 1 and 2 show the relationships between rank and outdegree and between outdegree and frequency respectively for two snapshots on April 15, 1999 and September 15, 1999. As the figures show, both power-laws presented in [FFF99] still hold in April 1999 and September 1999. The outdegree exponent ($O = -2.2$) and rank exponent ($R = -0.75$) are also in agreement with the constants computed in [FFF99]. Due to the small number of nodes with outdegrees larger than 20, we exclude them from our outdegree frequency analysis. This translates into excluding about 1.5 to 2% of samples from each of the 32 ASconnlist data set, which is in keeping with the percentages of samples excluded in [FFF99]. The higher outdegree samples are captured by the rank exponent power-law.

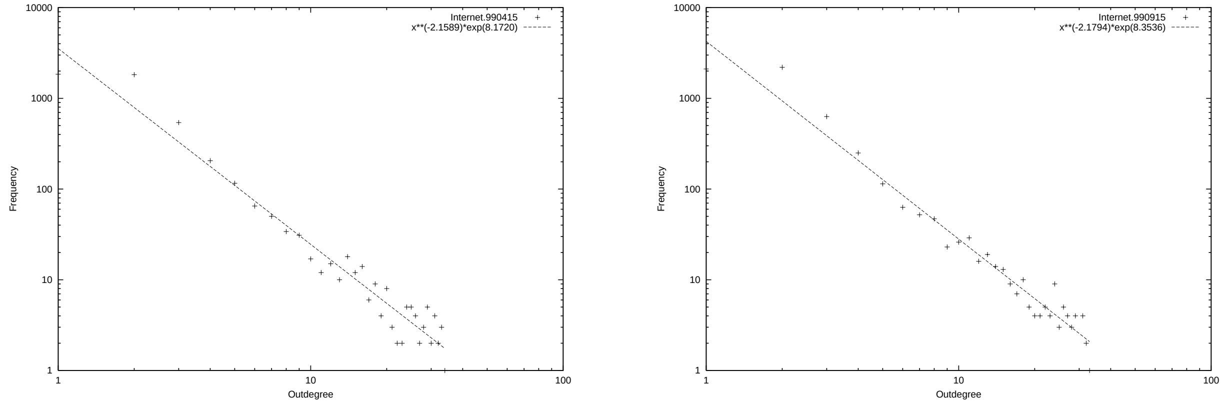
Finally, AS connectivity is characterized in [FFF99] by the eigenvalues of the Internet's connectivity matrix. The connectivity matrix of a graph is a square matrix $A = \{a_{i,j}\}$, where $a_{i,j} = 1$ if nodes i is connected to node j, 0 otherwise. A node is not considered connected to itself.



a. April 15, 1999.

b. September 15, 1999.

Figure 1: Power-Law 1: Outdegree versus rank in 1999.



a. April 15, 1999.

b. September 15, 1999.

Figure 2: Power-Law 2: Frequency distribution of AS outdegree in 1999.

4 Exponential Growth Laws

Two observations have often been made of Internet AS topology: (1) the number of ASs has a high growth rate [Bat98], and (2) there is an increasing preference for ASs to make direct peering arrangements with other ASs. Given these two observations, we would like to quantify how connectivities between nodes are formed as the Internet grows and evolves. The power-laws presented in [FFF99] are a summary of AS connectivity of static snapshots (in time) of the Internet topology. What we want to explore next is how to capture and characterize the high growth rate of the number of ASs, as well as the increasing preference for direct peering between ASs. ASs prefer to peer directly to other ASs with whom they exchange large amount of traffic to avoid going through NAPs (Network Access Points) that are often congested.

4.1 Frequency of Outdegree Growth

The second power-law relating frequency with AS outdegree can be written as:

$$f_d = Cd^O, \tag{1}$$

where O is a constant [FFF99]. Since the sum of the frequencies of all outdegrees is the number of ASs on the Internet, which changes with time, C must also change with time. We will show this using proof by contradiction. Assuming C is a constant, applying the first power-law to outdegrees 1 to 20, we get the following set of equations:

$$\begin{aligned} f_1 &= C1^O, \\ f_2 &= C2^O, \\ f_3 &= C3^O, \\ &\vdots \\ f_{20} &= C20^O, \end{aligned}$$

which sums up to:

$$\sum_{i=1}^{20} f_i = \sum_{i=1}^{20} Ci^O. \tag{2}$$

If C were a constant, the left hand side of the equation would also be a constant over time; however, the sum of ASs with degrees from 1 to 20 went from 2992 in November 1997 to 7665 in June 2000. Therefore, the assumption that C is a constant is false, and C is probably an increasing function of time.

To find a close-form expression for C, we take the log of both sides of Eqn. 1:

$$\log f_d = \log C + O \log d, \tag{3}$$

If Eqn. 1 were indeed correct, we would expect to see a linear relationship when we plot the log of the frequency, f_d, of AS outdegree versus the log of AS degree, d. Fig. 3 shows the plot of power-law approximations of 32 ASconnlist snapshots taken on the 15th of each month from November 1997 to June 2000. For each snapshot, we fit a power-law to the frequency versus degree graph of the real data, we then compute the constants C and O and plot the expression in Eqn. 1. We observe that the lines have similar slopes, strengthening the claim that O remains constant over time. Furthermore, when we plot the 32 intercepts of Fig. 3 against time in Fig. 4, we find that C is approximately linear with respect to time.

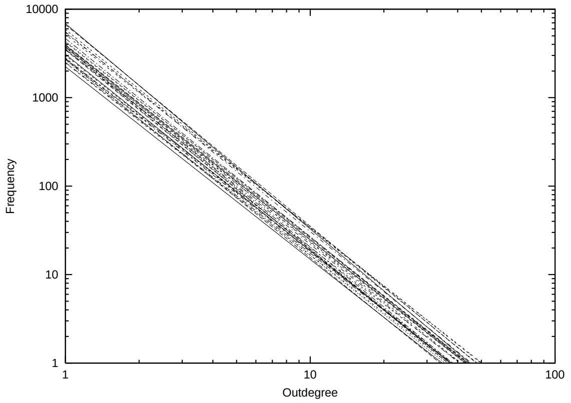


Figure 3: Lines fitting Power-Law 2 data for 32 monthly snapshots of the Internet since November 1997.

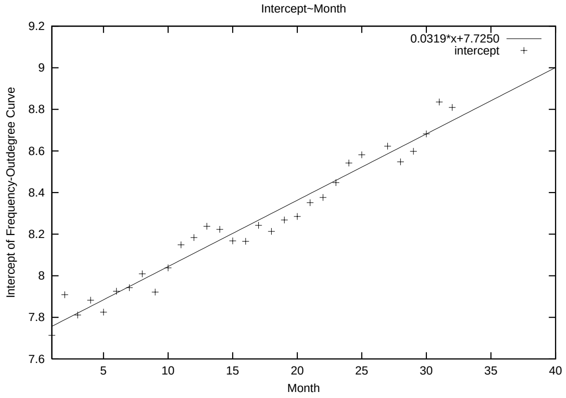


Figure 4: Relationship between 32 Power-Law 2 intercepts and time.

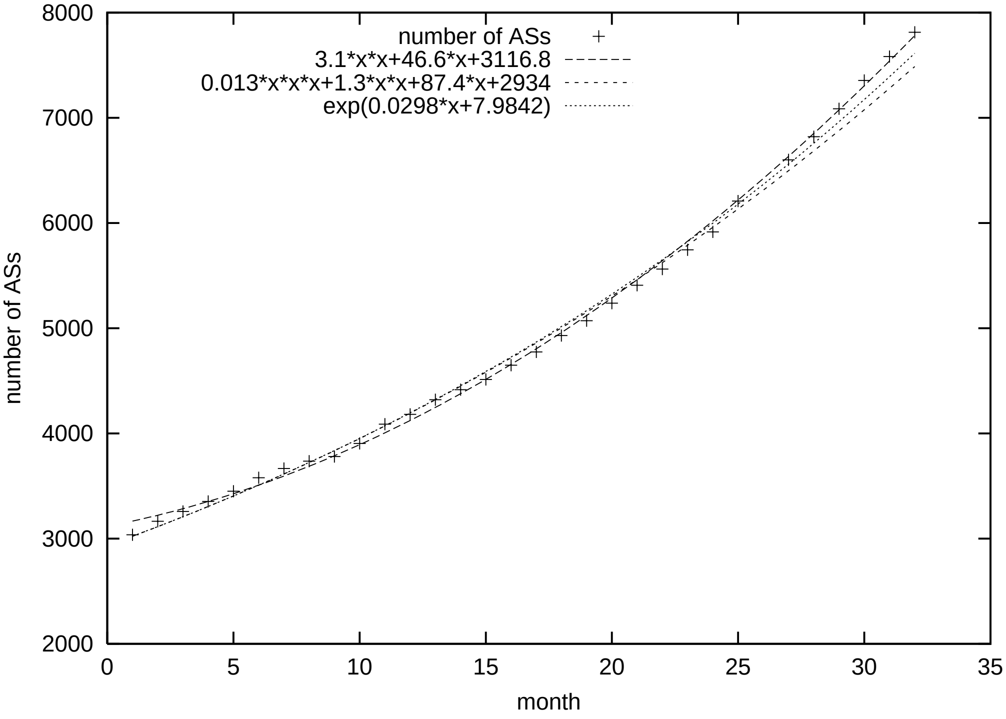


Figure 5: The number of ASs vs. month since November 1997

We approximate C as a linear relation, $at + b$ and obtain our first exponential law:

Exponential-Law 1 (frequency growth)
The frequency, f_d, of an outdegree, d, grows exponentially over time according to:

$$f_d = e^{at+b} d^O, \tag{4}$$

where a, b, and O are known constants and t is the number of months since November 1997.

Exponential-Law 1 says that, given the frequency of an outdegree in November 1997, we can predict the outdegree's frequency for a number of months into the future. Also note Eqn. 4 tells us that the number of ASs, not just the number of hosts, on the Internet has been growing exponentially over time. To verify whether the number of ASs indeed grows exponentially with time, we perform the following computations[1]:

1. We obtain an exponential expression, $e^{0.0298x+7.9842}$, by fitting the number ASs vs. time plot.
2. We apply Taylor series expansion on the expression to the cubic term around $x_0 = 0$. The resulting expression is $0.013x^3 + 1.3x^2 + 87.4x + 2934$.
3. To test whether the cubic term is necessary, we also fit the data to a quadratic form, $3.1x^2 + 46.6x + 3116.8$, using the least mean-square method.

Looking at Fig. 5, it is not clear which expression best approximates the grow of the number of ASs (although the actual NLANR data does rise more steeply than the others towards the end); we choose the exponential form for its relative parsimony. We recognize that the AS address space is currently limited to 16-bit in the BGP standard [KSR90]. If the number of ASs continues to grow exponentially, the AS address space would need a larger allocation.

4.2 Outdegree at Rank Growth

Traditionally, non-transit ASs connect to one or more transit ASs and reach other non-transit ASs indirectly through these transit ASs. Recently, there is an increasing preference among many non-transit ASs to peer directly with other "nearby" non-transit ASs instead of going through one or more transit ASs. This suggests that each AS is expected to have growing connectivity over time. Thus we expect that the outdegree at a given rank (ranked by outdegrees in descending order), expressed as,

$$d = e^D r^R \tag{5}$$

also increases over time. Applying the same methodology as in the previous section, we find that this is indeed the case, shown in Fig. 6. Furthermore, the exponent, D, grows linearly with time as shown in Fig. 7. We can again approximate D with a linear function of time, $pt + q$, and obtain our second exponential law:

Exponential-Law 2 (outdegree growth)
The outdegree, d, at a given rank, r, grows exponentially over time according to:

$$d = e^{pt+q} r^R, \tag{6}$$

where p, q, and R are known constants, and t is the number of months since November 1997.

[1] We thank Scott Shenker for suggesting this test.

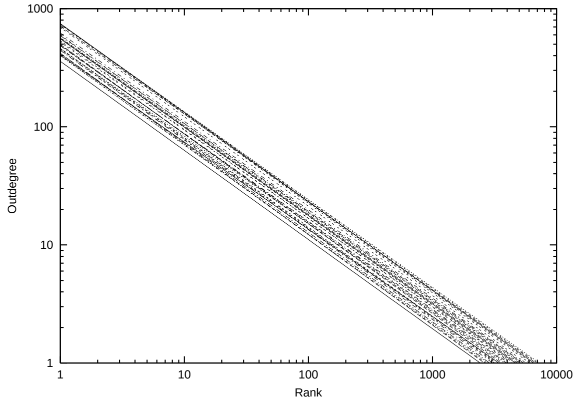


Figure 6: Lines fitting Power-Law 1 data for 32 monthly snapshots of the Internet since November 1997.

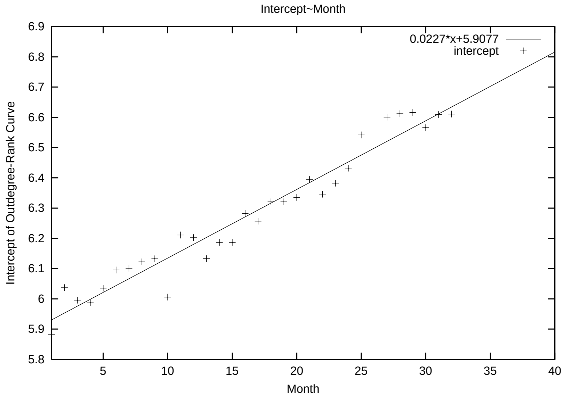


Figure 7: Relationship between 32 Power-Law 1 intercepts and time.

Note that this does not mean every AS's outdegree grows exponentially with time since the rank of a particular AS changes as the number of ASs increases; instead, this law tells us that the value of the i-th largest outdegree of the Internet grows exponentially.

4.3 Pair Size and Neighborhood Size Growths

In addition to Power-Laws 1 and 2, the authors of [FFF99] also studied the average neighborhood size of a node and the reachable pair size of the network. The neighborhood size, $A_v(h)$, of an AS v within h hops is the number of ASs reachable within h hops from v. The reachable pair size of a network reflects the node connectivity; it is defined as the number of reachable node pairs within h hops over the entire network, including self-pairs, and counting all other pairs exactly twice. The reachable pair size within h hops, $P(h)$, is thus the sum of neighborhood sizes of all ASs within h hops. For $h = 0$, $P(0) = N$, the number of nodes in the network. The *average* number of nodes reachable within h hops is $A(h) = \frac{P(h)}{P(0)}$. For all h's greater than or equal to the diameter of the network, $P(h) = N^2$.

The authors of [FFF99] then went on to present an approximation which states that $P(h) \propto h^H$, with H being a constant and $h \ll D$, the diameter of a network. Independently, the authors of [PST99] observed from the March 1999 AS connectivity data that $A(h)$ is an exponential function ($A(h) \approx e^{\lambda h}$, for some λ), which contradicts the pair-size power-law (which implies $A(h) \approx h^H$, for some H). As the authors of [FFF99, PST99, McM99] have also observed, we find that almost 95% of the ASs on the Internet are reachable among themselves within $h = 5$ hops. We show this in Fig. 8 for 6 snapshots of the Internet topology between November 1997 and July 1999. Given the small hop count before Internet reachability reaches saturation (N^2 number of nodes), we decided not to resolve this difference, but to concentrate instead on observing how the reachable pair sizes and neighborhood sizes at various hop count grow overtime. Fig. 9 plots $P(h)$ of the Internet, for $h = 0, 1, 2, 3, 4$ on the 15th of each month between November 1997 and September 1999, for a total of 23 months. We find that pair size grows exponentially with time, and arrive at our third exponential law:

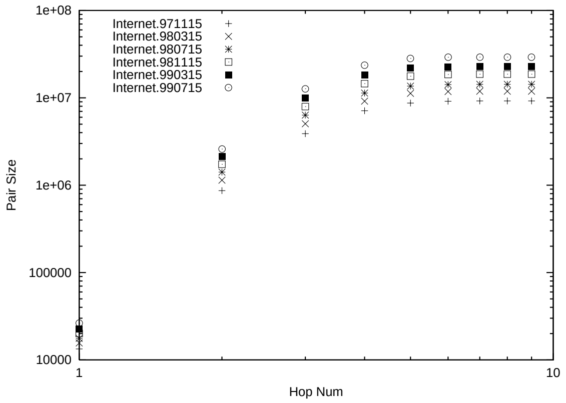


Figure 8: Pair size versus hop.

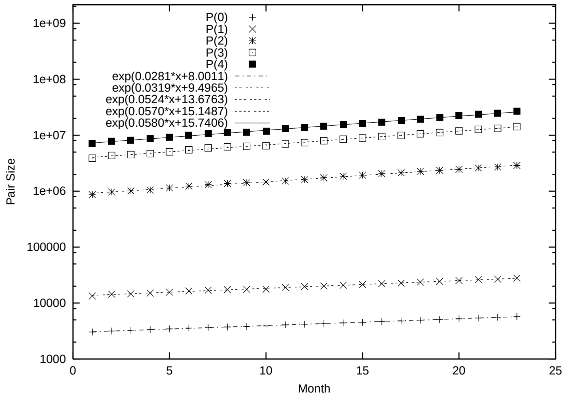


Figure 9: Growth of pair size within h hops over time.

Exponential-Law 3 (pair size growth)
Pair size within h hops, $P(h)$, grows exponentially over time according to:

$$P_t(h) = e^{s_h t} P_0(h) \tag{7}$$

where $P_0(h)$ is the pair size within h hops at time 0 (November 1997), s_h the pair size growth rate, and t the number of months since November 1997.

And a corollary:

Corollary to Exponential-Law 3 (neighborhood size growth)
Number of nodes reachable, i.e. the neighborhood size, within h hops, grows exponentially over time according to:

$$A_t(h) = \frac{P_t(h)}{P_t(0)} = e^{(c_h - c_0)+(s_h - s_0)t} = A_0(h) e^{(s_h - s_0)t}, \tag{8}$$

where $c_h = \log P_0(h)$, $c_0 = \log P_0(0)$, $A_0(h)$ is the neighborhood size at time 0 (November 1997) and and t is the number of months since November 1997.

5 Topology Generators

In this section, we compare the output of six random topology generators (also referred to as "topology models," or just "models") against snapshots of Internet topology at the AS-level. The six random topology generators we study here are: Waxman [Wax88], Tiers [Doa96], GT-ITM [CDZ97], Inet-1.0 [J$^+$00], BRITE [MM00], and Inet-2.0. All the comparisons reported in this section apply *only* when the output of these generators are used as AS-level topology. Tiers and GT-ITM in particular are designed to provide topologies for different kinds of network: LAN (Local Area Network), MAN (Metropolitan Area Network), WAN (Wide Area Network), and transit-stub networks with router-level details, all of which may not share any characteristics with AS-level Internet connectivity.

The process of generating a random topology can generally be summarized as follows: given an input of N nodes and a 2-dimensional plane of size n by m, we first decide where to place the nodes. The nodes can be distributed uniformly across the plain, or clustered around some regions. For each node, we decide its outdegree, i.e., how many other nodes it should be connected to. Then we decide which node should connect to which other node(s). The probability of creating an edge between two nodes can be uniformly distributed, or weighted by the Euclidean distance between them. We continually add edges to the network until all nodes have their outdegrees satisfied. A minimum spanning tree may be built prior to the generation of other edges to ensure that the resulting graph is a connected graph. Otherwise, a walk through of the generated topology is necessary to ensure a connected graph. If the generated graph is disconnected, extra edges may be added with nodes connected at random to form a connected graph. Or the graph can be discarded and the process repeated to search for a connected graph. If different types of nodes are to be generated, e.g., to represent transit vs. stub networks, the process may be repeated, recursively replacing some nodes with similarly generated networks. Reference [ZCD97] provides a good overview of various graph generation methods.

For this study, we generate several random topologies, ranging from 3,000 to 8,000 nodes, using each of the above generators and compare them against snapshots of the Internet topology. In this report, we show only results from studying 6,000-node topologies, and compare them against the Internet topology of October 1999, which has about the same number of nodes. The basis for comparison is whether the topologies generated exhibit the power-law relationships observed on the Internet topology.

5.1 Topology Generators Used

Before showing the comparison results, we briefly describe each topology generator and the parameters used to generate the 6,000-node topologies.

Waxman. The Waxman model has been widely used to generate random topologies for network simulations. It starts by placing N nodes uniformly on an n by n plane. Once all nodes have been placed on the plane, the model computes the probability of creating an edge between two nodes u and v with the following probability function:

$$P(u,v) = \alpha e^{-d(u,v)/\beta L} \tag{9}$$

where $d(u,v)$ is the Euclidean distance between u and v, α the average outdegree, L the maximum Euclidean distance between any two nodes, and β determines the average edge length. Then a random number is generated between 0 and 1. An edge is created between u and v only if the random number is smaller than $P(u,v)$. We use $\alpha = 0.0015$ and $\beta = 0.6$. Finally a spanning tree is created, adding edges where necessary so that the resulting topology is connected.

Tiers. The Tiers generator is based on a three level hierarchy that represents WAN, MAN, and LAN. To generate a random topology using Tiers, one specifies a target number of LANs and MANs. Currently Tiers cannot generate more than one WAN per random topology. For each level of hierarchy, one also specifies a fixed number of nodes per network. A minimum spanning tree is computed to connect all edges, then other edges are created based on user-specified average inter-level and intra-level redundancy. Edge formation favors close-by nodes, resulting in topologies with large diameters. For our study, we generate a 6,026-node network with 47 WAN, 20 LAN, 10 nodes/WAN, 8 nodes/MAN, and 6 nodes/LAN. The redundancy numbers are: WAN 1, MAN 5, LAN 1, MAN to WAN 2, and LAN to MAN 3.

GT-ITM. GT-ITM generates topologies based on several different models. We are particularly interested in the transit-stub model because it most closely resembles the Internet topology, albeit at the router-level [CDZ97]. Similar to Tiers, the transit-stub model has a well-defined hierarchical structure. It gen-

Table 2: Transit-Stub Model Parameters.

Parameter	Meaning	Value Used
T	number of transit ASs	30
N_t	avg. # nodes / transit ASs	8
K	avg. # stub domains / transit node	8
N_s	avg. # nodes / stub AS	3
N	total number of nodes	6,000
E_t	extra transit-stub links	30
E_s	extra stub-stub links	100
$N = T \times N_t \times (1 + K \times N_s)$		

erates topologies with two levels of hierarchy: one consisting of transit ASs, and the other consisting of stub ASs.

To generate a topology, GT-ITM first generates a connected random graph of T nodes; each node represents a transit AS. Each transit AS is then instantiated as, and replaced by, a connected random graph with an average of N_t number of nodes. Next, each node in the transit AS are connected to, on the average, K number of stub ASs. Each stub AS consists of a connected graph with an average of N_s number of nodes. The connectivity used to generate each connected graph can be selected from one of six methods: PureRandom, Waxman1, Waxman2, Doar-Leslie, Exponential, or Locality. We decided to use the PureRandom method. We refer the interested readers to the GT-ITM manual [GI97] for a more detailed explanations of these connectivity models. Similar to Tiers, GT-ITM also allows for extra edges to be added between stub ASs and between stub and transit ASs. Table 2 lists the values we use for the parameters in GT-ITM transit-stub to generate a 6,000-node topology.

Inet-1.0. The Inet-1.0 model generates a topology by placing N nodes on an n by n plane. Each node is assigned an outdegree based on Power-Law 2. Then a full mesh is used to connect the top τ most connected nodes. For these nodes, 25% of their edges are connected to randomly selected nodes with outdegree 2. To create a fully connected topology, the remaining nodes are either connected to one of these τ nodes or connected to a node that can reach one of these τ nodes. The Inet-1.0 model has a second phase where the top κ most connected nodes are expanded into networks with η nodes each. This phase is used to expand the top most connected ASs into networks with router-level connectivity. In this report, we use $\tau = 5$ and omitted the second phase of the generation process since we are only interested in AS-level topology.

BRITE. BRITE [MM00] is another generator based on the AS power-laws. Furthermore, BRITE also incorporates recent findings on the origin of power-laws presented in [BA99] and observations of skewed network placement and locality in network connections on the Internet. By studying a number of existing topology generators, the authors of BRITE claim that the preferential connectivity and incremental growth presented in [BA99] are the primary reasons for power-laws on the Internet. For completeness, we generated topologies that incorporates both skewed node placement and locality in network connections as well as topologies with just incremental growth and preferential connectivity.

To generate a topology on a plane, the plane is first divided into $HS \times HS$ squares, then the number of nodes in each square is assigned according to the placement, NP, which is either a uniform random distribution or a bounded Pareto distribution. The bounded Pareto distribution gives a skewed node placement where a non-negligible number of squares have a large number of nodes in them. Each square is

Table 3: BRITE Model Parameters.

Parameter	Meaning	Scenario I	Scenario II
HS	size of one side of the plane	1,000	
LS	size of one side of a high-level square	10	
NP	clustered node placement	uniform random	pareto
m	number of links added per new node	1, 2, 3, 4, 5	
PC	preferential connectivity	degree-based only	degree and locality based
IG	incremental growth	enabled	

further divided into $LS \times LS$ smaller squares, and the assigned nodes are then uniformly distributed among the smaller squares. A backbone node is selected from each of the top-level squares populated with nodes, and a spanning tree is formed among the backbone nodes. Nodes are then connected one at a time to nodes that are already connected to the backbone. This is referred to as "incremental growth" (IC in the table) in [BA99]. A new node can have preferential connectivity (PC) in its choice of neighboring nodes: locality-based, outdegree-based, or both. The locality-based preferential connectivity uses a Waxman probability function to connect nodes in the topology. In outdegree-based preferential connectivity, the probability of a new node connecting to an existing node is the ratio of the existing node's outdegree over the sum of all outdegrees of nodes in the connected network. Finally, when mixing both locality-based and outdegree-based preferential connectivities, the probability of connecting to an existing node under outdegree-based preferential connectivity is weighted by the Waxman probability between the new node and the existing node. Each new node introduces m new links.

We generate topologies under two scenarios as shown in Table 3: Scenario I includes incremental growth and preferential connectivity based on outdegree, and Scenario II includes incremental growth, skewed placement, and preferential connectivity based on both locality and outdegree. For Scenario I, many of the top-level squares are occupied, which results in a backbone consisting of large number of nodes, and a relative small portion of nodes are then added incrementally. The average degree of such topologies is influenced by the total number of nodes, and in most of our experiments, the average degree comes close to m. For Scenario II, the skewed placement places nodes in a much smaller number of top-level squares, which results in a much smaller backbone size than in Scenario I. Since each incrementally grown node introduces $2m$ new outdegrees, and there are significantly more such nodes than the backbone nodes, the average node degree is approximately $2m$. For both scenarios, we experimented with m ranging in value from 1 through 5. The data presented in Figs. 10 and 12 is for $m = 5$ in Scenario I and $m = 2$ in Scenario II, both of which resulted in an average outdegree of around 3.7.

Inet-2.0. Inet-2.0 follows the basic design of Inet-.10, but uses more systematic approaches to generate node outdegree distribution and to connect nodes in the topology, as follows:

1. The generator takes two numbers from the user, N, the number of nodes, and, k, the fraction of N that has outdegree of 1. Assuming exponential growth rate of number of ASs, we first compute the number of months (t) it would take the Internet to grow from it size in November 1997 to N. With the computed t, we then compute the outdegree-frequency and rank-outdegree distributions using Eqns. 4 and 6 respectively. Recall that Power-Law 2 captures the outdegree distribution of only 98% of the nodes. Accordingly, we use the rank-outdegree distribution (Eqn. 6) to assign the outdegrees of the top 2% of the N nodes. As specified by the user, k percent of the N nodes are assigned outdegree 1. The remaining nodes are assigned outdegrees according to the frequency-outdegree

distribution (Eqn. 6).

2. In this step, we perform the feasibility test to ensure our construction produces a connected network. We construct the network in three steps: (1) forming a spanning tree using nodes with outdegree of at least two, (2) attaching nodes with degree one to the spanning tree, and (3) matching the remaining unfilled degrees of all nodes with each other. Since we generate node degrees ahead of time, the unfilled degrees are simply the difference between the designated degree and the number of current neighboring nodes. The feasibility test checks that there are enough unfilled outdegrees after the spanning tree is constructed to attach degree-one nodes. The feasibility test is written as $F^* >= kN$, where $F^* = F - 2((1-k)N - 1)$, and F is the sum of all the outdegrees of the $(1-k)N$ nodes whose outdegrees are at least two. $(1-k)N - 1$ is the number of links on the spanning tree, and $2((1-k)N - 1)$ is the number of outdegrees consumed by this tree; therefore, F^* gives the number of available outdegrees that degree-one nodes (kN of them) can attach to. The network would be connected if there are enough unfilled outdegrees for degree-one nodes. Under this construction, while we might not always fulfill the outdegree requirement of every node, we can make sure the resulting network is connected.

3. This step builds a spanning tree among nodes with degrees larger than 1. Let G be the graph to be generated, initially empty. Randomly, with uniform probability, a node with outdegree larger than 1 that is not in G is connected to a node in G with proportional probability, namely, the probability of connecting to a node in G is k/K, where k is the outdegree of the node in G, and K is the sum of outdegrees of all nodes already in G that still have at least one unfilled outdegree.

4. Next, we connect the kN nodes with outdegree 1 to nodes in G with proportional probability.

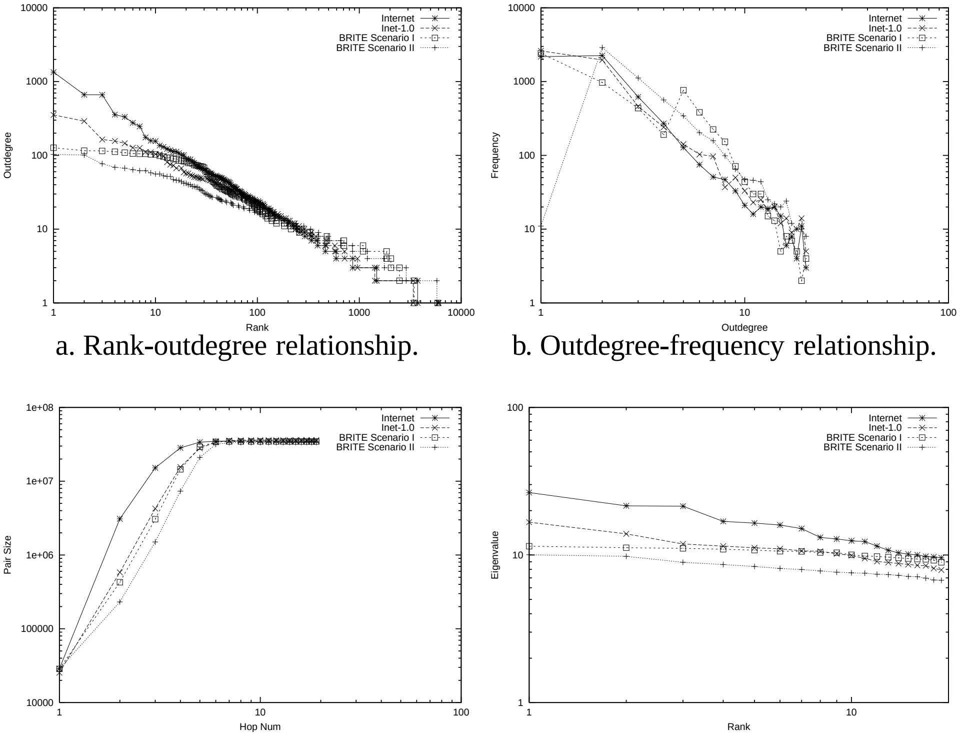
5. Finally, we connect the remaining free outdegrees in G starting from the node with the largest outdegree first. In making these connections, we randomly pick nodes with free outdegrees using proportional probability.

5.2 Comparison Results

We have studied a large number of topologies, ranging in size from 3,000 to 6,000 nodes, generated using all of the six generators. In this section we present only the results of comparing the Internet topology of October 1999 against five randomly generated 6,000-node topologies, one each from the six generators. We first consider the topologies generated by the Inet-1.0 and the BRITE models, then we examine each of the other three topologies in turn.

Fig. 10 shows that the topologies generated by the Inet-1.0 and BRITE models match the Internet topology of similar size quite closely. However, the pair size numbers for the Internet are slightly higher and the eigenvalues of the two generated topologies do not match as closely with those of the Internet. We will come back to these issues at the end of this Section. Fig. 11, on the other hand, shows that none of the topologies generated by Waxman, Tiers, and the GT-ITM models follows any of the power-law relationships. Recall from Eqn. 9 that in the Waxman model, an edge between two nodes is created as a function of the ratio of the distance between them and the network diameter. In the Waxman model, the outdegree of a node is not explicitly specified, instead it is a result of the random formation of edges between nodes. This means that for a network of uniformly placed nodes, the outdegree of most nodes will also be uniform. This explains why the Waxman generated topology has a small maximum outdegree. Note that the y-axis on Fig. 11a only goes up to 100, as opposed to 10,000 as in Fig. 10a.

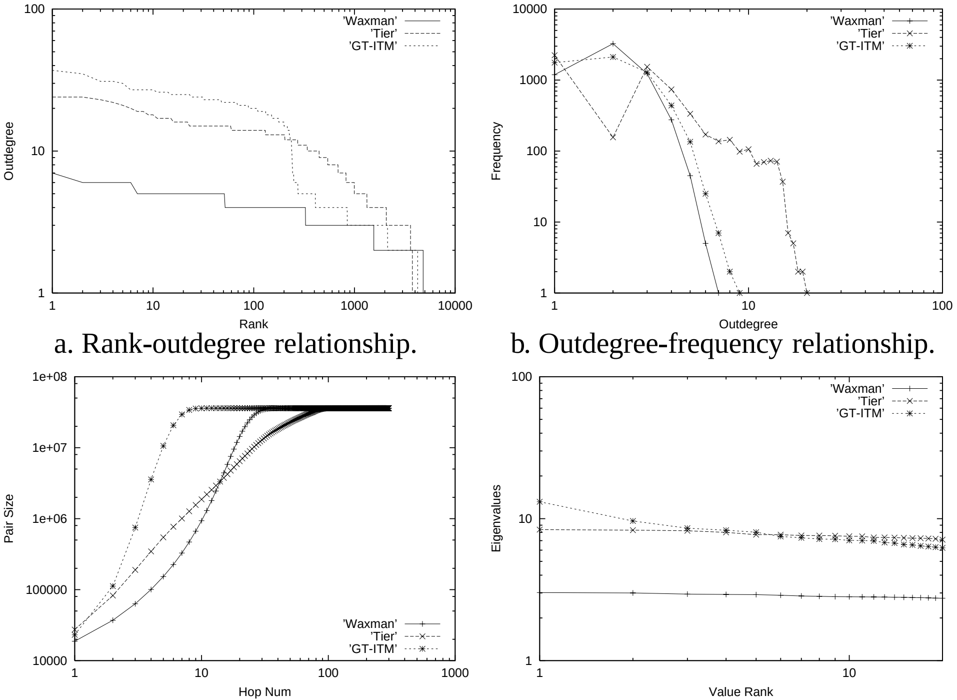
Fig. 11c shows that in the case of Tiers, the pair size is a power-law of the hop count, but for the other two generators, the pair size functions grow faster than a power-law. This can be explained by the



a. Rank-outdegree relationship. b. Outdegree-frequency relationship.

c. Pair size within h hops. d. Eigenvalues of connectivity matrix.

Figure 10: Relatively good match of power-law relationships between Inet-1.0, BRITE generated 6,000-node topologies and Internet topology of October 26, 1999 (around 6,000 nodes).



a. Rank-outdegree relationship. b. Outdegree-frequency relationship.

c. Pair size within h hops. d. Eigenvalues of connectivity matrix.

Figure 11: Characteristics of various relationships for Waxman, Tiers, and GT-ITM generated 6,000-node topologies.

structure of Tiers topologies, where connections to nearby nodes are favored. Tiers has a higher average node degree than both Waxman and GT-ITM, so within the first few hops it is likely to see higher growth in pair size. However, due to Tiers' favoring of nearby nodes, a small number of new nodes will appear with the addition of each hop as distant nodes are gradually reached. This results in pair sizes having a slow but fairly constant growth rate until some saturation point.

The GT-ITM model is strongly hierarchical in nature, with a distinct definition of transit and stub networks. This is reflected by the sudden transition in GT-ITM curve in Fig. 11a. Nodes in transit ASs tend to have higher degrees of connectivity than those in stub ASs. This two-level hierarchy also helps shorten the network diameter because to go from one stub node to another, one simply walks up the tree to a transit AS, transit to zero or more other transit ASs, and walk down the tree to the other stub ASs.

We now return to the topology generated by the Inet-1.0 and BRITE models. In Fig. 10c there are noticeable gaps between the pair size numbers of the generated topologies and those of the Internet. Fig. 10d similarly shows noticeable gaps between the eigenvalues of the generated topologies and those of the Internet. To get a better picture of the difference in pair sizes, we generated 32 topologies of sizes matching those of the Internet between November 1997 and June 2000. Fig. 12a-d show the $P(h)$ of each of the 32 topologies for hop counts 1, 2, 3, and 4. Notice how the pair size growths of the Inet-1.0 and BRITE models, for $h > 1$, do not follow that of the Internet. The jump in the last two pair size data of the Inet-1.0 generated topologies are due to the generator using only Power-Law 2, and not Power-Law 1, in determining node outdegree distribution. We corrected this in Inet-2.0. Finally, we notice that the largest outdegree in topologies generated using the BRITE generator is order of magnitude smaller than that of the Internet (see Fig. 10a).

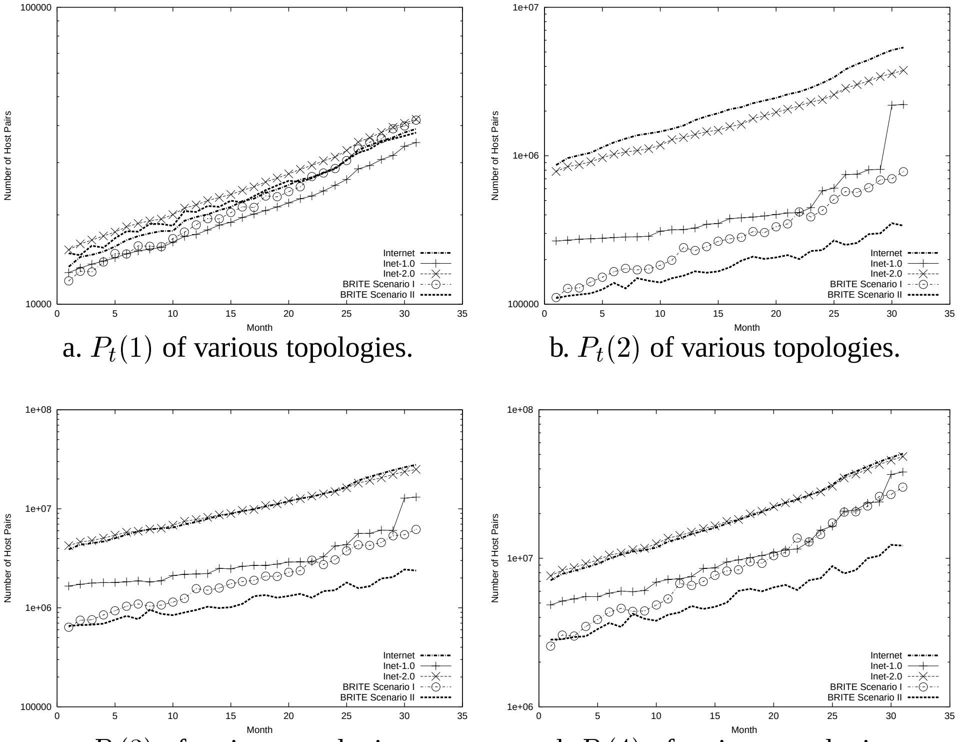
Fig. 13 shows that the 6,000-node topology generated by Inet-2.0 have characteristics that match those of the Internet more closely than topologies generated by the other generators. The pair sizes of Inet-2.0 generated topologies also match very well with those of the actual Internet, even for the last nine data points; recall that we use only the first 23 data sets in computing the power-law distributions encoded in Inet-2.0. In terms of model parsimony, the *only* parameter Inet-2.0 requires from the user is target topology size.

6 Conclusion

In recent years, researchers in many different areas in networking have recognized the need for a random topology generator that produce realistic Internet topology. In order to produce realistic topologies, we need a better understanding of the Internet topology itself. In this report, we present a new Internet topology generator, Inet-2.0, that not only generates topologies with Internet-like characteristics, but is also parsimonious in the number of parameters required. We hasten to add that extant generators, in particular Tiers and GT-ITM which were expressly designed to model router-level connectivity, may model router-level connectivity very well.

Acknowledgement

We thank Michalis Faloutsos, Ramesh Govindan, Danny Raz, Yuval Shavitt, Scott Shenker, and Walter Willinger for discussions on this topic.



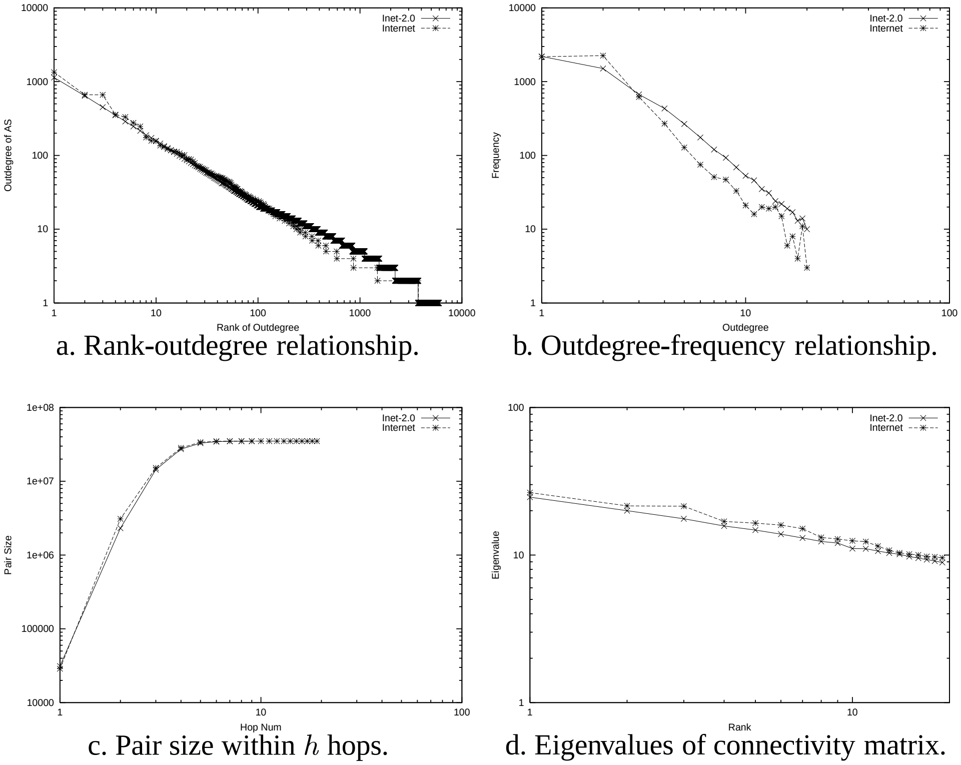
a. $P_t(1)$ of various topologies.

b. $P_t(2)$ of various topologies.

c. $P_t(3)$ of various topologies.

d. $P_t(4)$ of various topologies.

Figure 12: Pair size within h hops of 6,000-node topologies of the Internet and those generated by Inet-1.0 and Inet-2.0.



a. Rank-outdegree relationship.

b. Outdegree-frequency relationship.

c. Pair size within h hops.

d. Eigenvalues of connectivity matrix.

Figure 13: Relatively better match of power-law relationships between Inet-2.0 generated 6,000-node topology and Internet topology of October 26, 1999 (around 6,000 nodes).

References

[BA99] Albert-Laszlo Barabasi and Reka Albert. Emergence of scaling in random networks. *Science*, pages 509–512, Octobor 1999.

[Bat98] T. Bates. *The CIDR Report*. url: http://www.employees.org/~tbates/cidr-report.html, June 1998.

[BE90] L. Breslau and D. Estrin. “Design of Inter-Administrative Domain Routing Protocols”. *Proc. of ACM SIGCOMM ’90*, pages 231–241, Sep. 1990.

[CDZ97] K. Calvert, M.B. Doar, and E.W. Zegura. “Modeling Internet Topology”. *IEEE Communications Magazine*, June 1997.

[Doa96] M. Doar. A Better Model for Generating Test Networks. *Proc. of IEEE GLOBECOM*, Nov. 1996.

[F$^+$00] A. Feldman et al. “NetScope: Traffic Engineering for IP Networks”. *IEEE Network Magazine*, 2000.

[FFF99] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On Power-Law Relationships of the Internet Topology. *Proc. of ACM SIGCOMM ’99*, pages 251–262, Aug. 1999.

[FGHW99] A. Feldman, A.C. Gilber, P. Huang, and W. Willinger. “Dynamics of IP Traffic: A Study of the Role of Variability and the Impact of Control”. *Proc. of ACM SIGCOMM ’99*, pages 301–313, Aug. 1999.

[GI97] GT-ITM. Distribution. http://www.cc.gatech.edu/fac/Ellen.Zegura/graphs.html, 1997.

[J$^+$00] S. Jamin et al. On the Placement of Internet Instrumentation. *Proc. of IEEE INFOCOM 2000*, Mar. 2000.

[KSR90] S. Kirkpatrick, M. Stahl, and M. Recker. Internet numbers, July 1990. RFC-1166.

[McM99] P.R. McManus. A passive system for server selection within mirrored resource environments using as path length heuristics. http://proximate.appliedtheory.com/, Jun. 1999.

[MM00] Alberto Medina and Ibrahim Matta. Brite: A flexible generator of internet topologies. Technical Report BU-CS-TR-2000-005, Boston University, Boston, MA, 2000.

[MS94] D.J. Mitzel and S. Shenker. “Asymptotic Resource Consumption in Multicast Reservation Styles”. *Proc. of ACM SIGCOMM ’94*, 1994. Available from `http://netweb.usc.edu/mitzel/Sigcomm94/sigcomm94.ps.Z`.

[NLA99] NLANR. National laboratory for applied network research routing data. http://moat.nlanr.net/Routing/rawdata/, 1999.

[PST99] G. Phillips, S. Shenker, and H. Tangmunarunkit. “Scaling of Multicast Trees: Comments on the Chuang-Sirbu Scaling L aw”. *Proc. of ACM SIGCOMM ’99*, pages 41–52, Aug. 1999.

[Wax88] B.M. Waxman. Routing of Multipoint Connections. *IEEE Journal of Selected Areas in Communication*, 6(9):1617–1622, Dec. 1988.

[WE94] L. Wei and D. Estrin. “The Trade-offs of Multicast Trees and Algorithms”. *Int’l Conf. on Computer Communications and Networks*, 1994.

[ZCD97] E.W. Zegura, K.L. Calvert, and M.J. Donahoo. “A Quantitative Comparison of Graph-Based Models for Internet Topology”. *ACM/IEEE Transactions on Networking*, 5(6):770–783, Dec. 1997.

[ZGLA91] W.T. Zaumen and J.J. Garcia-Luna Aceves. “Dynamics of Distributed Shortest-Path Routing Algorithms”. *Proc. of ACM SIGCOMM ’91*, pages 31–42, Sep. 1991.