How to Model an Internetwork

Ellen W. Zegura
ewz@cc.gatech.edu

Kenneth L. Calvert
calvert@cc.gatech.edu

Samrat Bhattacharjee
bobby@cc.gatech.edu

College of Computing
Georgia Institute of Technology
Atlanta, GA 30332–0280

Abstract

Graphs are commonly used to model the structure of internetworks, for the study of problems ranging from routing to resource reservation. A variety of graph models are found in the literature, including regular topologies such as rings or stars, "well-known" topologies such as the original ARPAnet, and randomly generated topologies. Less common is any discussion of how closely these models correlate with real network topologies. We consider the problem of efficiently generating graph models that accurately reflect the topological properties of real internetworks. We compare properties of graphs generated using various methods with those of real internets. We also propose efficient methods for generating topologies with particular properties, including a Transit-Stub model that correlates well with Internet structure. Improved models for internetwork structure have the potential to impact the significance of simulation studies of internetworking solutions, providing basis for the validity of the conclusions.

1 Introduction

The explosive growth of internetworking, and particularly of the Internet, has been accompanied by a wide range of internetworking problems related to routing, resource reservation, and administration. The study of algorithms and policies to address such problems often involves simulation or analysis using an *abstraction* or model of the actual network structure and applications. The reason is clear: networks that are large enough to be interesting are also expensive and difficult to control, therefore they are rarely available for experimental purposes. Moreover, it is generally more efficient to assess solutions using analysis or simulation — *provided* the model is a "good" abstraction of the real network and application. It is therefore remarkable that studies based on randomly-generated or trivial network topologies are so common, while rigorous analyses of how the results scale or how they can be applied to actual networks are so rare.

A selected review of the literature indicates the use of network models to study a wide range of problems. The state of the art in modeling includes:

- Regular topologies, such as rings, trees and stars (e.g., [3, 8, 14])
- "Well-known" topologies, such as the ARPAnet or NSFnet backbone (e.g., [1, 9, 15])
- Randomly generated topologies (e.g., [10, 11, 12])

The limitations of each of these are obvious: well-known and regular topologies reflect only parts of current or past real networks; random topologies may not reflect any (past, present or future) real network. What is additionally clear from the cited references is the diverse set of problems that rely on network models to evaluate performance.

To illustrate the important role that the network model can play in assessing algorithms, consider the following results:

- Doar and Leslie found that the inefficiency of their dynamic multicasting algorithms was up to twice as large when using random graphs versus using hierarchically structured graphs designed to reflect some of the properties of real internetworks. (See Figure 5 in [6].)
- Wei and Estrin found that the traffic concentration in core-based trees is comparable to traffic concentration in shortest-path trees for a network model with average node degree of about 3.0, but the traffic concentration is almost 30% higher in core-based trees than shortest-path trees when average node degree increased to 8.0. (See Figure 9(a) in [13].)
- Mitzel and Shenker found that multicast resource reservation styles compared quite differently on many metrics for linear, tree and star topologies. (See Tables 4 and 5 in [8].)

It should be evident from these examples that the network model *does* matter: the conclusions reached about the suitability and performance of algorithms may vary depending on the methods used to model the network.

In this paper we address the problem of *efficient* and *accurate* generation of graph models for internetworks. We begin in Section 2 by proposing a set of properties that can be used to characterize network topologies. These properties enable us to quantify the comparison of one model to another and to real networks. A rigorous method of quantifying the characteristics of models is essential; the somewhat common

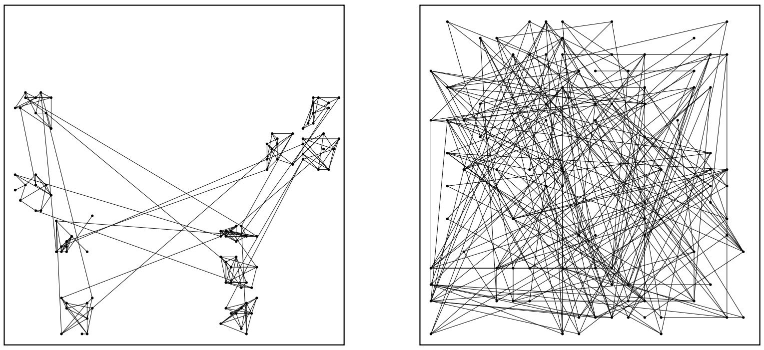

Figure 1: The Dangers of Visual Representation

practice of relying on visual representation (e.g., noting that a layout bears a visual resemblance to geographic maps of the Internet) can be terribly misleading. For example, the two topologies in Figure 1 each have 100 vertices and approximately the same number of edges (230 versus 231), a fact that is counter to the visual impression.

We conclude Section 2 with the properties of a collection of real networks. In Section 3, we study random graph models, all variations on the standard random graph generation method that constructs a graph by randomly adding edges to a set of vertices. We also give the properties of some regular graphs. In Section 4 we propose a new model that represents a hybrid approach, imposing some high-level structure, and using random graph models to fill in details. We compare the properties of topologies generated by the various methods, and reach conclusions about the suitability of the models for reflecting Internet structure.

The contributions of this paper include:

- We characterize the similarities and differences among various models for internetwork structure. For example, we find that the usual random graph model results in far longer paths than any of the graph models that are specifically designed to reflect networking structure.

- We compare our models, and those from the literature, with a collection of real topologies. We conclude that non-hierarchical models, while the predominant choice by many researchers, miss essential details of real topologies (e.g., lower average node degree).

- We propose a new model — the Transit-Stub model — that accurately reflects a wide range of the properties of real internetworks, including hierarchy and locality.

2 Internet Topology Characteristics

Throughout this paper it is assumed that the goal is to model the *paths* (i.e., sequences of nodes) along which information flows betwen two nodes in an internetwork. Nodes represent *switches* or *routers*, and edges represent forwarding paths between switches. We do not model individual hosts; it is therefore sensible to have a switching node that is connected to only one other node. Thus it is the *logical* structure with which we are primarily concerned, as opposed to the physical structure. For example, a FDDI ring to which four IP routers are connected would be represented as a clique of four nodes.

In what follows, references to "the path between two nodes" always refer to the primary path that would be followed by packets traveling from one node to the other; this is usually the "shortest" path according to some metric or other. We assume such paths are symmetric.

A variety of criteria may be applied to assessing a network model, depending in large part on the intended use. For example, if the purpose is to stress-test the algorithm, the model should generate instances which are, in some sense, "difficult." If the purpose is to model a particular (static) network (e.g., a campus or corporate network), then the graph should reflect the actual topology. In many cases, particularly in wide-area networking, the purpose is to study the performance of algorithms operating in an environment whose structure is only partly known. For these cases, the model should reflect the known properties, and instantiate the remainder of the topology in some random (but reasonable) fashion.

Because the "best" form for a network topology model depends on its intended use, we focus here on various *topological* attributes of graph models. We consider both "hop-based" metrics, in which each edge has unit weight, and "length-based" metrics, in which each edge has weight equal to its Euclidean length.

For a graph with n nodes and m edges, we consider the following:

- **Node degree distribution.** We are particularly interested in the average node degree, $2m/n$, and the number of leaves (degree = 1).

Network	Nodes	Avg Deg	Diam.	Bicomp.
ARPAnet	47	2.9	9	5
CERFnet	63	2.1	5	59
Sesquinet	94	2.1	9	77
PGE	137	2.2	10	103
Ga. Tech	30	3.9	3	16

Table 1: Properties of real internetwork topologies

- **Hop-depth distribution.** The hop-depth at node u is the depth of the shortest-path tree rooted at u to all other nodes. We are particularly interested in the **hop-based diameter** of the graph, that is the largest hop-depth.
- **Length-depth distribution.** This is equivalent to hop-depth using the length metric. We are particularly interested in the **length-based diameter** of the graph.
- **Number of biconnected components.** A biconnected component ("bicomponent") is a maximal set of edges such that any two edges in the set are on a common simple cycle. This quantity is a measure of the degree of "connectedness" or "redundancy" in a graph.

These characteristics are selected for two reasons: they are quantitative abstractions of some aspects of the structure of the graph, and they are relatively easy to measure. They are not independent. In particular, increasing average node degree correlates with decreasing diameter and fewer bicomponents.

In the next subsection, we consider some measurements of real networks with respect to these parameters.

2.1 Characteristics of Actual Networks

Unfortunately, it is difficult to obtain complete topological descriptions of even a modest portion of the Internet, due to the lack of any centralized administration. In fact, some administrations go to great lengths to hide the topology of their network from the outside world, for reasons of security [4]. We do have data for several backbone networks and one campus network (namely the Georgia Tech campus).

The results are shown in Table 1. The ARPAnet entries refer to the "old ARPAnet" backbone, which has been used as a topology in a number of simulation studies. CERFnet and Sesquinet are backbone regional networks, i.e., they connect to (and carry traffic from) many smaller "stub" networks. (The hierarchical structure of the Internet is discussed further in Section 4.) "PGE" is a large private network; the final entry reflects the characteristics of our local campus internet.

The statistics for ARPAnet reflect the backbone topology only, while those for CERFnet and Sesquinet include edges to other routing domains. This gives the latter networks a more tree-like structure, and thus a greater number of bicomponents and lower average node degree. PGE also has a substantial number of "leaf" nodes, while the Georgia Tech network features a relatively high degree of connectivity.

We now proceed to discuss methods for generating graphs intended to model the structure of actual internetworks.

3 Random Graph Models

3.1 Five Flavors of Random Graphs

The networking literature contains a variety of random graphs used to model internetworks. All are variations on the standard random graph model that distributes vertices at random locations in a plane and then considers each pair of vertices; an edge is added between a pair of vertices with probability p. We call this the Pure Random model or simply the Random model. While it does not explicitly attempt to reflect the structure of real internetworks, it is commonly used to study networking problems.

The other models also distribute the vertices randomly in the plane, however they alter the function used for the probability of an edge, in an attempt to better reflect real network structure. After the Pure Random Model, perhaps the most common random graph model is one proposed by Waxman [12], with the probability of an edge from u to v given by:

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

where $0 < \alpha, \beta \leq 1$ are parameters of the model, d is the Euclidean distance from u to v, and L is the maximum distance between any two nodes. An increase in α will increase the number of edges in the graph, while an increase in β will increase the ratio of long edges relative to shorter edges.

Several variations on the Waxman model have been proposed, including (1) replacing d by a random number between 0 and L [12], (2) scaling $P(u,v)$ by a factor $k\epsilon/n$, where ϵ is the desired average node degree, n is the number of nodes and k is a constant that depends on α and β [6], and (3) allowing $\alpha > 1.0$ [13]. The second variation, proposed by Doar and Leslie, is included in our study. Note that the addition of the factor $k\epsilon/n$ gives more direct control over the number of edges in the graphs that are generated, however the model is not fundamentally different than the Waxman model since the α parameter of the Waxman model can be chosen to be equivalent to any particular setting of parameters in the Doar-Leslie model.

We also propose two new models, intended to relate edge probability to distance between vertices (as in the Waxman model), but with more straightforward probability functions. Our Exponential model uses:

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

The probability of an edge in this model decreases exponentially with the distance between the two vertices. Our Locality model partitions the edges into discrete categories based on length, and assigns a different edge probability for each category. With two categories we

Model	Edge Probability
Pure Random	p
Waxman	$\alpha e^{-d/(\beta L)}$
DoarLeslie	$\alpha k\epsilon/n e^{-d/(\beta L)}$
Exponential	$\alpha e^{-d/(L-d)}$
Locality	α if $d < r$ β if $d \geq r$

Table 2: Random graph models

use one parameter r to define the boundary:

$$P(u,v) = \begin{cases} \alpha & \text{if } d < r \\ \beta & \text{if } d \geq r \end{cases}$$

One nice feature of the Locality model is that we have been able to extend many of the analytic results for (pure) random graphs to this model. Any finite number of categories are allowed, giving a reasonable amount of control over the topologies generated.

To summarize, Table 2 indicates the probability of an edge between two vertices at Euclidean distance d for each model in our study.

3.2 Evaluation Methodology

Our aim is to investigate these random graph models, to determine how the parameters affect the topologies and to examine similarities and differences between models. To establish a basis for comparison, we fix a value for n, the number of vertices, m, the number of edges, and L, the maximum distance between two nodes. Specifically, we select $n = 100$ and $m = 175$; this corresponds to an average node degree of 3.5. We fix the area in which the nodes are generated to be 100 by 100, thus $L = 100\sqrt{2} \approx 141$.

We have three primary reasons for focusing on this set of parameters. First, the data that we have from real topologies is in this ballpark, facilitating comparisons between models and reality. Second, graphs of 100 nodes are on par with the graphs typically used by other researchers for experimentation. Third, the random methods have a limitation on the relationship between node degree and size: as the graph size increases, graphs with low average node degree are generally not connected. (See Section 4 for a specific example.)

In each model, we explore the combinations of parameters that yield a fixed number of edges. After characterizing the combinations of parameters that can produce some connected graphs with the target number of edges, we pick one particular set of parameters for each model and determine the additional properties of interest for graphs generated using these parameters.

3.3 Results

3.3.1 Parameter Selection

For all of the properties we will examine, the most useful result would be an analytic expression giving the value of the property as a function of the model parameters. Unfortunately, most of the models do not admit an analytic expression even for fairly simple properties. There are two exceptions to this: first, a wide body of work on the properties of (pure) random graphs has been developed, particularly for $n \to \infty$ (e.g., [2]); second, we have extended a number of these results to the Locality model. For the remaining models, we rely on empirical results.

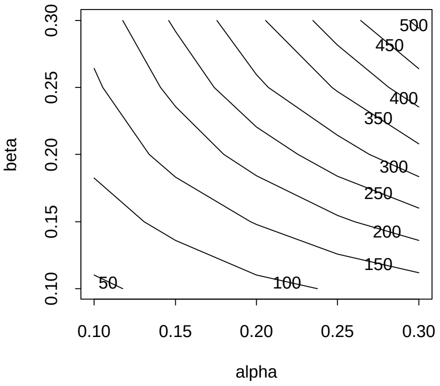

Figure 2: Effect of α and β on number of edges in Waxman model

The expected node degree in the Pure Random model is given by $p(n-1)$, thus $p = .035$ will produce graphs with $m = 175$ edges. The expected degree of a node in the Exponential model is $E[n\alpha e^{-d/(L-d)}] = n\alpha E[e^{-d/(L-d)}]$, scaling linearly with α. Empirical results indicate that the average number of edges is approximately 175 when $\alpha = .06$; about 3 out of 100 of these graphs are connected. The fraction of connected graphs sharply increases as α is increased; this is typical of all of the types of random graphs we study, and has a basis in theoretical analysis of random graphs [2].

The relationship between the number of edges and the parameters of the Waxman model is somewhat more complex. Through empirical studies, we have determined that the parameter L has essentially *no effect* on the number of edges, for fixed α and β, because a change in L is accompanied by a change in the distance d between an arbitrary pair of vertices, keeping the quantity d/L essentially constant. For a target number of edges, many combinations of α and β can achieve the target. Figure 2 shows the contour lines of equi-number of edges as a function of α and β, for $n = 100$. We select $\alpha = .2$ and $\beta = .15$, yielding $m = 175$.

For the Doar-Leslie model we select $\alpha = .1$ and $\beta = .3$. Through experimentation we determined that $k \approx 27$ will yield $m = 175$ when $\epsilon = 3.5$.

As mentioned earlier, the Locality model has a structure which makes it possible to analytically determine some properties. The average node degree in

Model	Parameters
Random	$p = .035$
Waxman	$\alpha = .2, \beta = .15$
Doar-Leslie	$\alpha = .1, \beta = .3, k = 27$
Exponential	$p = .06$
Locality	$\alpha = .1, \beta = .005,$ $r = .25, L = 35.35$

Table 3: Selected parameters for each random model

this model can be approximated by:

$$\frac{\pi r^2}{L^2}\alpha(n-1) + (1 - \frac{\pi r^2}{L^2})\beta(n-1),$$

where the first term accounts for edges to nodes within distance r and the second accounts for edges to nodes further than distance r. This expression is only an approximation due to the "boundary effects" within the generation region; vertices near the boundary have fewer nearby neighbors than vertices near the middle. The equation predicts average node degree of 4.1 for the parameters used; empirical studies revealed an average node degree of 3.5 for these parameters.

We summarize the choice of parameters for each model in Table 3.

3.3.2 Graph Properties

To evaluate the properties, we generate 100 connected graphs of each model type. For the scalar quantities, we use a boxplot that indicates the median over the 100 values (with a white line), the 25% and 75% range boundaries (with a box), the 5% and 95% range boundaries (with "whiskers") and any outliers with single lines. The upper left plot of Figure 3 compares the average node degree of the five models described thus far, and the Transit-Stub model which we discuss Section 4. This plot confirms that the choice of parameters for each model does produce graphs with average node degree of approximately 3.5.

Figure 3 also shows the other scalar properties, namely the number of bicomponents and the diameter measured in both hops and length. The salient features of these plots are the following:

- The most significant difference in the random models is in the length-based diameter. The Pure Random model is clearly different from the other models, with much longer diameter. The Random model is insensitive to edge length when adding edges to a graph, therefore it is more likely to have long edges and long paths.

- The Doar-Leslie and Exponential models are next most likely to have long paths. The Exponential model does include length in the probability of an edge, however the probability falls off more slowly with increasing edge length than in the other models. The Doar-Leslie model has a higher β value than the Waxman model, leading to longer edges.

- The other two models (Waxman and Locality) are quite similar. Given the differences in edge probability functions, one might have expected more differences in properties of the graphs generated, however this particular choice of parameters results in similar edge length distributions.

- The Transit-Stub model differs significantly from the random models on the bicomponents metric. We discuss this more fully in Section 4.

We have also looked at the distribution (within a graph) on some of these metrics. We measured the distribution on the node degree; all of the models are similar for this metric, with most nodes having degree between 2 and 5, with a few nodes that have degree as high as 11 or 12. All of the models have between 5 and 15 nodes per graph that are leaves (i.e., have degree 1). With random generation methods, it is difficult to get a large number of leaf nodes and still have a connected graph.

We have also measured the distribution on shortest path lengths. The Random distributions are far wider than any of the other models, ranging to paths of length greater than 500, while the Exponential and Doar-Leslie models taper off at about 400 and the other models at about 300. The mean path length in the Random model is also significantly higher than the other models, followed by the Exponential and Doar-Leslie models. These observations are consistent with those noted for the scalar properties.

3.4 Regular Topologies

For completeness, we next consider the properties of some regular topologies, commonly used in studies of algorithms because their structure makes them analytically tractable. Table 4 gives the scalar properties of Linear Chains, Rings, Stars and Meshes, where we have assumed that all edges have unit weight. While a Ring and a Chain differ by only one edge, they have substantially different diameter and number of bicomponents. (Removing one edge in a Ring breaks the symmetry.) A Star is a graph with one designated "center" node, and an edge from this node to each other node. The Mesh is a 2-dimensional array with $\sqrt{n}$ vertices on each side and connectivity only to neighbors in the array. A Star has the smallest diameter of these four regular graphs. Note that the properties of these graphs are quite different than the properties measured from actual networks.

4 A Hybrid Approach

In this section we describe a method of constructing graphs with characteristics that resemble those of (portions of) the Internet. The basic idea of the method is to construct portions of the graph randomly while constraining the gross structure. This can be done very efficiently, because the random parts of the graph are constructed a small portion at a time. Once the topology is generated, edge weights can be assigned in such a way that the shortest paths between

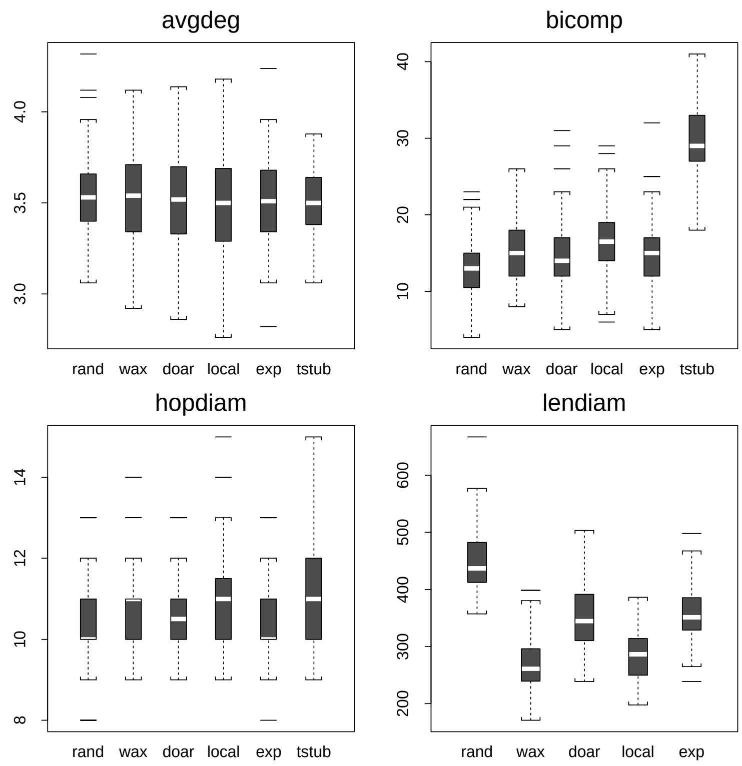

Figure 3: Scalar properties

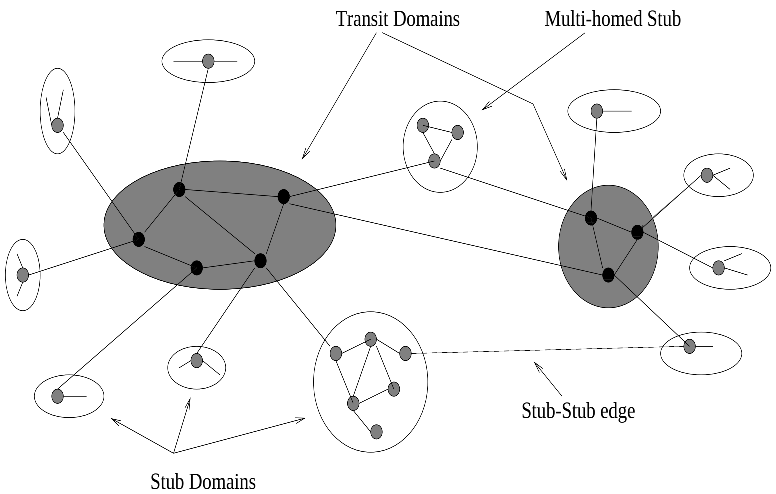

Figure 5: Example of Internet domain structure

	Edges	Diam.	Bicomp.
Linear Chain	$n-1$	n	n
Ring	n	$\lfloor n/2 \rfloor$	1
Star	$n-1$	2	n
Mesh	$2(n-\sqrt{n})$	$2(\sqrt{n}-1)$	1

Figure 4: Properties of regular graphs

nodes exhibit locality characteristics corresponding to those observed in the Internet.

4.1 Domain Structure

Today's Internet can be viewed as a collection of interconnected *routing domains* [5], which are groups of nodes that are under a common administration and share routing information. A primary characteristic of these domains is *routing locality*: the path between any two nodes in a domain stays entirely within the domain. Each routing domain in the Internet can be classified as either a *stub* domain or a *transit* domain. A domain is a stub domain if the path connecting any nodes u and v goes through that domain only if either u or v is in that domain; transit domains do not have this restriction. The purpose of transit domains is to interconnect stub domains efficiently; without them, every pair of stub domains would need to be directly connected. (See Figure 5.)

Stub domains can be further classified as single- or multi-homed. Multi-homed stub domains have connections to more than one other domain. Single-homed stubs connect to only one transit domain. A transit domain comprises a set of *backbone* nodes, which are typically fairly highly connected to each other (at least biconnected). In a transit domain each backbone node also connects to nodes in a number of stub domains. Some backbone nodes also connect to other transit domains. A stub domain usually has one or more *gateway* nodes, which have links to the transit domains. Some stubs also have links to other stubs.

4.2 Constructing Transit-Stub Models

Our model produces graphs composed of interconnected transit and stub domains. We first construct a connected random graph (using any one of the methods discussed earlier); each node in that graph represents an entire transit domain. Each node in that graph is then replaced by another connected random graph, representing the backbone topology of one transit domain. Next, for each node in each transit domain, we generate a number of connected random graphs representing the stub domains attached to that node. Finally, we add some number of additional edges between pairs of nodes, one from a transit domain and one from a stub, or one from each of two different stub domains. Clearly, if the random graphs generated are all connected, this construction results in a connected graph. We note that Doar and Leslie proposed a graph model that has a similar hierarchical structure to our model, but is more limited in scope [6]. In our implementation, which uses the Stanford Graph Base [7] for portable representation and storage of graphs, different random graph methods and parameters may be plugged in to construct transit and stub domain subgraphs with different characteristics.

The size of the graph (number of nodes) and distribution of nodes between transit and stub domains is controlled by the following parameters:

	Meaning	Ex.
T	# transit domains	6
N_t	(avg) nodes/transit domain	15
K	(avg) stub domains/transit node	12
N_s	(avg) nodes/stub domain	8
	total nodes	8730

As the table shows, it is not difficult to generate rather large graphs with this method.

Once the complete graph is constructed, integer edge weights are assigned in such a way that the shortest paths computed using the weights obey certain constraints (described below). The edge weights to be calculated and assigned are:

W_{tt}: transit-transit edge weight
W_{ts}: transit-stub edge weight
W_{ss}: stub-stub edge weight

Note that all edges of the same type are given the same weight. Also, all *intra*domain edges are given unit edge weight. By making the weights of the *inter*domain edges sufficiently large, we guarantee that the path between any two nodes in the same domain will remain within that domain.

In calculating the weights of the interdomain edges, the following quantities, taken from the constructed graph, are used:

D_{top} transit domain connectivity diameter
D_t max transit domain diameter
D_s max stub domain diameter

The following constraints guarantee the desired locality characteristics when paths are computed using the assigned weights. The constraint $2W_{ss} > D_s$ ensures that an intra-stub path is preferred over any inter-stub path; similarly $2W_{tt} > D_t$ implies that any intra-transit path is preferred over every inter-transit path. The constraint $2W_{ts} > D_{top}W_{tt} + (D_{top}+1)D_t$ ensures that there are no "shortcuts" between transit nodes via a stub node connected to both of them. The constraint $2W_{ss} > 2D_s + 2W_{ts} + D_{top}W_{tt} + D_{top}D_t$ guarantees that every all-transit path is shorter than any path passing through three stubs. Finally, $W_{ss} \approx 2W_{ts} + \epsilon$ allows direct connection between two stub domains to be preferred over a transit path in some cases.

When weights are assigned as follows, it is not difficult to show that the above constraints are all satisfied:

$$
\begin{aligned}
W_{tt} &:= \lceil D_t/2 \rceil \\
W_{ts} &:= \lceil D_{top}W_{tt}/2 \rceil + \lceil (D_{top}+1)D_t \rceil \\
W_{ss} &:= D_s + 2W_{ts}
\end{aligned}
$$

4.3 Characteristics of Transit-Stub Graphs

To compare the structural characteristics of transit-stub graphs with those of random graphs, we constructed and measured 100 transit-stub graphs of 100 nodes each, with average node degree about 3.5. Each graph had 1 transit domain of 4 nodes, 2 stub domains per transit node, and 12 nodes per stub domain. The measurements are shown in the Figures discussed earlier. None of these graphs had any "extra" transit-stub or stub-stub edges, so each stub domain connects to exactly one transit domain; this accounts for the significantly higher number of bicomponents in these graphs compared to the random methods. The other measurements appear to be similar to those of the other methods for graphs of this size.

One of the advantages of the transit-stub graph model, however, is its ability to generate *large* graphs efficiently, while maintaining low average node degree. The other methods require the average node degree to grow as n grows in order to generate connected graphs with reasonable efficiency. This increase in average node degree in turn affects other parameters such as diameter and number of bicomponents. For example, with the Pure Random method, using edge probability $p = 0.0123 \approx .99(\ln n/n)$, generating 20 connected graphs of 500 nodes each required 145 attempts. The average node degree of each of those 20 connected graphs was about six! In contrast, the formula for the average node degree of a transit stub graph (ignoring "extra" edges) is

$$\frac{2(E_t + (1 + E_s N_s)K)}{1 + KN_s}$$

where E_t and E_s are the edge densities (number of edges per node) of the transit and stub domains, respectively. If the average node degree is 3 for transit backbones (i.e. excluding transit-stub edges) and 2.5 for stub domains, using the example parameters given above for the 8730-node graph, the average node degree of the entire graph will be about 2.75.

It is possible to control the diameter and number of bicomponents of transit-stub graphs rather precisely. Figure 6 shows box plots of the number of bicomponents and the diameter (measured in hops, but using shortest paths calculated using edge weights) for three different sets of 10 graphs of 1056 nodes each. In each set, every graph had four transit domains that averaged eight nodes each. In the leftmost set of values, each graph had one stub domain per transit node, with 32 nodes per stub. In the middle set, each graph averaged four stubs per transit node, with eight nodes per stub, while the rightmost set averaged eight stubs per transit node, with an average of four nodes per stub. None of the graphs had any "extra" transit-stub or stub-stub edges.

As Figure 6 shows, the diameter of the graphs decreased with decreasing stub domain size, i.e., increasing number of stubs per node. For a fixed-size top-level topology, the diameter is an approximately linear function of the stub diameter, which decreases with stub size. The other graph shows how the number of bicomponents increases with increasing number of stubs per transit node. Because each stub domain attaches to a transit node with exactly one edge, each stub adds at least one bicomponent to the graph.

5 Conclusions and Future Work

We set out to develop efficient and accurate methods for generating graph models of internetworks. We have accomplished the following:

- Surveyed and analyzed the behavior of graph models commonly used in studies of networks.

  The literature contains numerous random and regular topologies used to study networking algorithms, with little guidance for the user of network models. We have outlined the role of the parameters in each graph model and included plots that can be used for parameter lookup by users of these models.

  We conclude that the Pure Random model is significantly different from the five other random models, generating edges that are longer, leading to longer paths (in terms of Euclidean distance). Given the tendency towards locality in internetworking connectivity, models that include edge length in determining edge probability are more realistic. The Exponential and Doar-Leslie models are next most likely to have long edges; the other two models are similar in all properties we considered.

- Identified a random model (the Locality model) that uses edge length in determining edge probability, but is amenable to analytic expression for the properties of the graphs that are generated. These expressions are more accurate as the number of nodes increases, but can give reasonable approximations for smaller numbers of nodes. The general form of this model is quite flexible, allowing an arbitrary number of categories.

- Identified a fundamental limitation in all of the random graph models.

  We found that it is not practical to generate even moderate-sized graphs (e.g., $n = 1500$) that are connected and have realistic average node degrees (e.g., less than about 6). Based on our measurements of real topologies, this limitation is significant, especially if a model is to be used to obtain *quantitative* results that are valid for the Internet (e.g. in measuring the difference in efficiency of multicast routing algorithms [13]).

- Developed a hybrid generation method, the Transit-Stub model, capable of creating large graphs by composing smaller random graphs.

  By imposing a domain structure resembling that of the Internet, the Transit-Stub model allows creation of large random graphs having realistic average node degree. Moreover, edge weights can be assigned to these graphs in such a way that intra- and interdomain paths in the graph behave

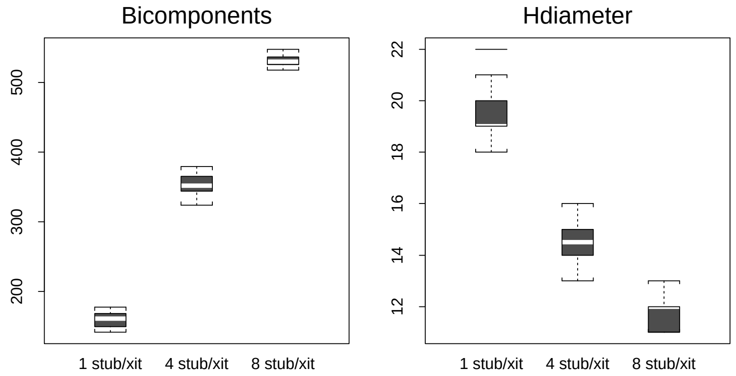

Figure 6: Controlling bicomponents and hop diameter

in a realistic manner. Finally, the Transit-Stub model allows pretty direct control over attributes such as hop diameter and average node degree.

As the Internet continues to grow in size and importance, realistic network topology models will be critical for quantitative assessment of all kinds of algorithms and policies. One goal of our future work is to understand how measurements taken on modest-sized graphs can be *scaled* to apply to much larger networks. Work is in progress to establish a public repository of information about real network topologies, including both graph models and measurements, as well as the graph-generating tools described in this paper.

References

[1] J. Behrens and J.J. Garcia-Luna-Aceves. Distributed, scalable routing based on link-state vectors. *Proceedings of ACM SIGCOMM '94*, pages 136–147, 1994.

[2] B. Bollobás. *Random Graphs*. Harcourt Brace Jovanovich, 1985.

[3] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. *Proceedings of ACM SIGCOMM '94*, pages 24–35, 1994.

[4] W. Cheswick and S. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley, 1994.

[5] David Clark. Policy routing in internet protocols. Internet Request for Comments 1102, May 1989.

[6] Matthew Doar and Ian Leslie. How bad is naive multicast routing? In *Proceedings of IEEE INFOCOM '93*, pages 82–89, 1993.

[7] D. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Addison-Wesley, 1994.

[8] D. Mitzel and S. Shenker. Asymptotic resource consumption in multicast reservation styles. *Proceedings of ACM SIGCOMM '94*, pages 226–233, 1994.

[9] S. Sibal and A. DeSimone. Controlling alternate routing in general-mesh packet flow networks. *Proceedings of ACM SIGCOMM '94*, pages 136–147, 1994.

[10] D. Sidhu, T. Fu, S. Abdallah, R. Nair, and R. Coltun. Open shortest path first (OSPF) routing protocol simulation. In *Proceedings of ACM SIGCOMM '93*, pages 53–62, 1993.

[11] Dinesh C. Verma and P.M. Gopal. Routing reserved bandwidth multi-point connections. In *Proceedings of ACM SIGCOMM '93*, 1993.

[12] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[13] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *Proceedings of ICCCN '94*, 1994.

[14] C. Williamson. Optimizing file transfer response time using the loss-load curve congestion control mechanism. In *Proceedings of ACM SIGCOMM '93*, pages 117–126, 1993.

[15] William T. Zaumen and J.J. Garcia-Luna Aceves. Dynamics of distributed shortest-path routing algorithms. In *Proceedings of ACM SIGCOMM '91*, 1991.