

# Generation and Analysis of Random Graphs to Model Internetworks

Megan Thomas      Ellen W. Zegura

**GIT-CC-94/46**

*August 1994*

## **Abstract**

Graph models are commonly used in studying solutions to internetworking problems. This paper considers several random graph models that have been used to model internetworks, and considers ways to characterize the properties of these graphs. By matching the characteristics of the random graphs to the characteristics of real internetworks, more accurate modeling can be achieved.

College of Computing  
Georgia Institute of Technology  
Atlanta, Georgia 30332-0280

# 1 Introduction

Accompanying the rapidly increasing use of large networks and internetworks has been a corresponding growth in research on internetworking. Researchers working in internetwork communication generally adopt one of three methods to evaluate their ideas:

- They implement their ideas on a real internetwork. This approach has obvious difficulties, including the problem of getting access to an internetwork of reasonable size that can tolerate experimentation.
- They implement their ideas on a smaller network and either assume or address the issue of scalability.
- They use a graph to model the internetwork and simulate their ideas on the graph.

The advantage of graph models over the other two methods is that, unlike real internetworks, there is no issue of access and service disruption. Unlike small networks, they can be made arbitrarily large. In addition, the creator of the graph can exercise control over a wider variety of parameters than in the other two methods.

We surveyed the 1993 Infocom Conference Proceedings, the 1990 through 1993 SIGCOMM Conference Proceedings and some other miscellaneous papers, looking at only those papers wherein the authors used a graph or a real network to evaluate their results. Out of the papers that fit our criteria (i.e. they looked at large networks, not just at the events in one router; they simulated or implemented their algorithms; and they included information about the simulation or implementation environment), one used the Internet [10], two used regular graphs [12, 19], two used graphs that were copies of real internetworks [12, 22], one simulated a five node WAN [4], one used two real computers to be the entire internetwork [7], one used a LAN to simulate a WAN [13], and ten, about half of the papers, used random graphs without specifying how they were generated [1, 2, 3, 6, 8, 10, 14, 15, 16, 17]. Only five used clearly specified random graphs [5, 9, 18, 20, 21].

The results of any simulation depend upon the model on which the simulation is carried out. Researchers have been assuming that realistic simulation results can be attained when the graphs used to get the results are far smaller than the real networks they represent (graphs of fewer than fifteen nodes are not rare), and may or may not have the topological characteristics of real networks. Further, there is no way to judge the accuracy of the graph model if the method used to generate the graph and the characteristics of the graph are not specified.

The purpose of this paper is to present some of the random graph models already in use, and to present some metrics for discerning the topological characteristics of graphs. Eventually, in future work, we hope to extend this to matching the characteristics of the random models to those of real internetworks to achieve more accurate internetwork simulation.

The following sections present, first, some graph-related definitions. Then we present random graph generation methods that have already been used for simulations. Next we give results on topological characteristics of graphs generated within each model, and also illustrate an example of random graph use in research. The last section covers areas of possible future work.

## 2 Definitions

In this paper, unless otherwise stated, the term *graph* will refer to a connected graph with undirected edges, no parallel or self-loops, and  $N$  nodes. Nodes in a network simulation are generally used to represent routers and edges to represent the links between the routers. Weights on the edges can be used to indicate bandwidth or some other property of the link. The *degree* of a node is the number of edges incident to the node. The *shortest path* between any two nodes,  $x$  and  $y$ , is the sequence of edges from node  $x$  to node  $y$  whose collective weight sums up to a minimum. The *depth* of a node is the length of the longest of the shortest paths from it to all the other nodes. The *diameter* of a graph is the length of the longest shortest path in the entire graph. Equivalently, the diameter can be thought of as the largest depth in the graph. A *center* of a graph is a node whose depth is minimum over all the nodes in the graph.

## 3 Graph Models

In this section we present three models for generating random graphs that reflect some of the structure of internetwork topologies. Each model is a variation on the basic random graph generation process that distributes vertices in the plane then adds edges between pairs of vertices based on some probability function.

### 3.1 Two-Level Graphs

Calvert et al. propose a two phase method to generate random graphs with a hierarchical structure [5]. In the first phase, nodes are randomly placed in the x-y coordinate plane. Let  $d$  be the Euclidean distance between vertices  $p$  and  $q$ . Let  $R$  be the radius of constant probability of adding an edge. Let  $\alpha$  be a number between zero and one, used to control the edge probability. The probability  $P$  of the edge being added between  $p$  and  $q$  is given by [5]:

$$P = \begin{cases} \alpha & \text{if } d \leq R \\ \alpha(\sqrt{2} - d)/(\sqrt{2} - R) & \text{if } d > R \end{cases}$$

Because the graph generated may not be connected, the generation program iterates until the resulting graph is connected.

After the initial graph has been completed the program enters the second phase. Here program goes through every node in the top level graph and replaces it with a subgraph generated in the same fashion as the original. The edges incident to the top level nodes are connected sequentially to the nodes in the subgraph with the lowest degree greater than one. In this way, the replacement process preserves leaves. The model allows an arbitrary number of levels, but Calvert et al. [5] restricted consideration to two levels. Figure 1 contains a sample Two-Level graph with 100 nodes. The hierarchical nature of its two level structure is quite evident.

### 3.2 Waxman Graphs

Waxman [20] proposed a random graph model that has become something of a *de facto* standard. After distributing  $N$  nodes in the x-y coordinate plane, each pair of nodes  $p$  and

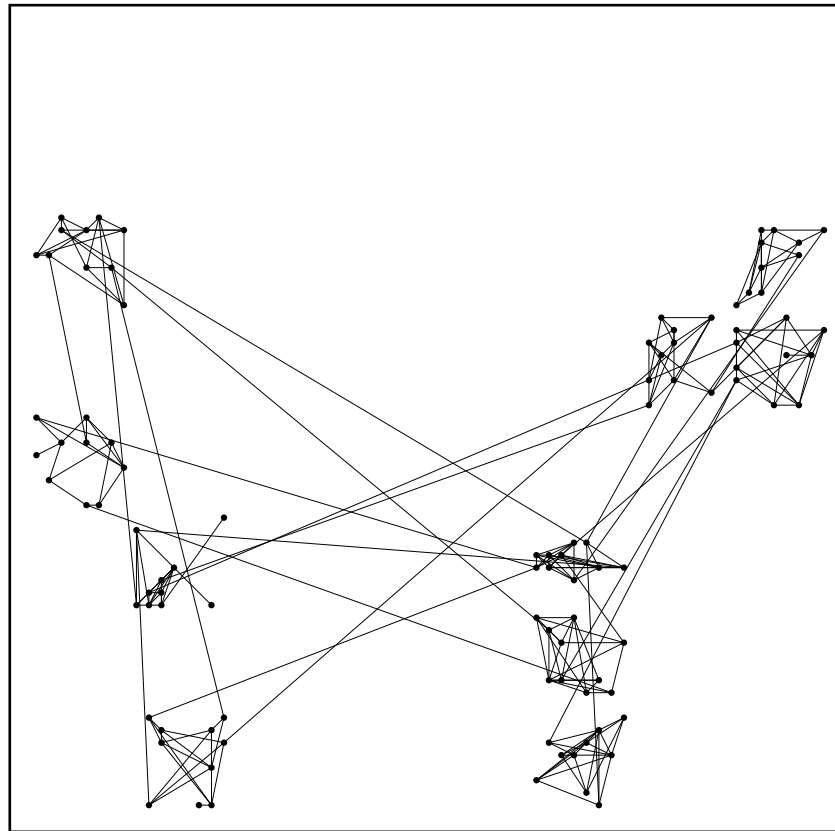


Figure 1: Example Two-Level Graph

$q$  is considered. Let  $d$  be the Euclidean distance between  $p$  and  $q$ . The parameters  $\alpha$  and  $\beta$  are numbers chosen by the user between zero and one that govern the probability of an edge being added. The parameter  $L$  is the maximum Euclidean distance between any two nodes. An edge is added between  $p$  and  $q$  with probability:

$$P = \beta \exp(-d/(L\alpha))$$

The procedure iterates until a connected graph is generated. An alternative to the above method is to create a graph in exactly the same way, but with  $d$  chosen at random between zero and the maximum internodal distance  $L$ . As the two Waxman graph types are extremely similar, for the most part we will not differentiate between them. When we do differentiate, the Waxman graph that uses Euclidean distance will be Waxman 1 and the graphs that use a randomly chosen distance will be Waxman 2. Figure 2 presents a sample Waxman 2 graph with 100 nodes, and Figure 3 presents a sample Waxman 1 graph with 100 nodes. Appearances to the contrary notwithstanding, the Waxman 2 and the Two-Level graphs have nearly the same number of edges (approximately 230), but the Two-Level edges are clearly concentrated within the subgraphs while the Waxman edges span the entire graph freely.

### 3.3 DoarLeslie Graphs

Doar and Leslie proposed a modified version of the Waxman 1 random graph generation scheme [9]. Contending that as the number of nodes in the graph increased, so does the average node degree, Doar and Leslie added a scaling factor to Waxman’s edge probability equation. (Note that Wei and Estrin further refined this model, suggested that coupling  $\beta$  values above one (which Waxman’s original idea did not allow) with very small  $\alpha$  values produced graphs that “appeared to be of practical significance” [21].)

After distributing  $N$  nodes randomly over an x-y coordinate grid, this method goes through every pair of nodes and calculates the Euclidean distance between the nodes. In addition to  $\alpha$ ,  $\beta$  and  $N$ , the user of the program must give the desired average node degree,  $e$ , and a constant,  $k$ , that must be experimentally figured out for every  $\alpha$ ,  $\beta$  and  $e$ . To figure out  $k$ , we ran the program with our  $\alpha$ ,  $\beta$  and  $e$  values and varied  $k$  until we got an average node degree near  $e$ . The average node degree of the final graph will only be close to  $e$  if the correct  $k$  is used. The final probability that the edge will be added is:

$$P = \beta(ke/|N|) \exp(-d/(L\alpha))$$

Figure 4 presents a sample DoarLeslie graph of 100 nodes. Qualitatively, it appears to be very much like the Waxman graphs.

## 4 Characterization of Graphs

Given several methods to generate graphs, the question arises as to what type of graphs are generated by each model. In this section, we propose metrics to help characterize random graphs. The focus here is upon three general graph characteristics.

- the node degree distribution

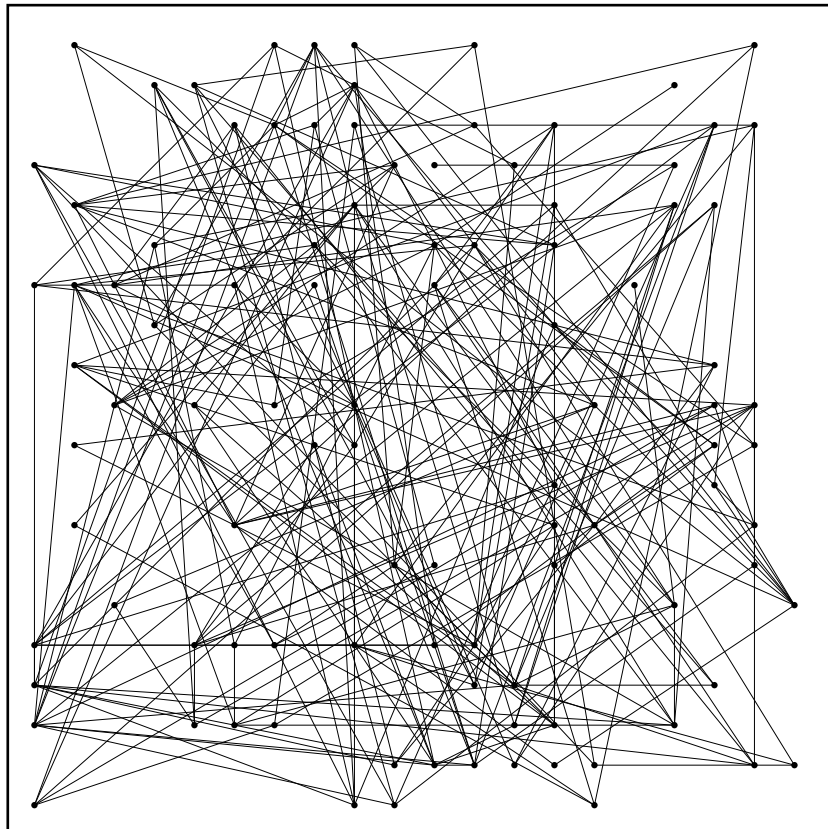


Figure 2: Example Waxman 2 Graph

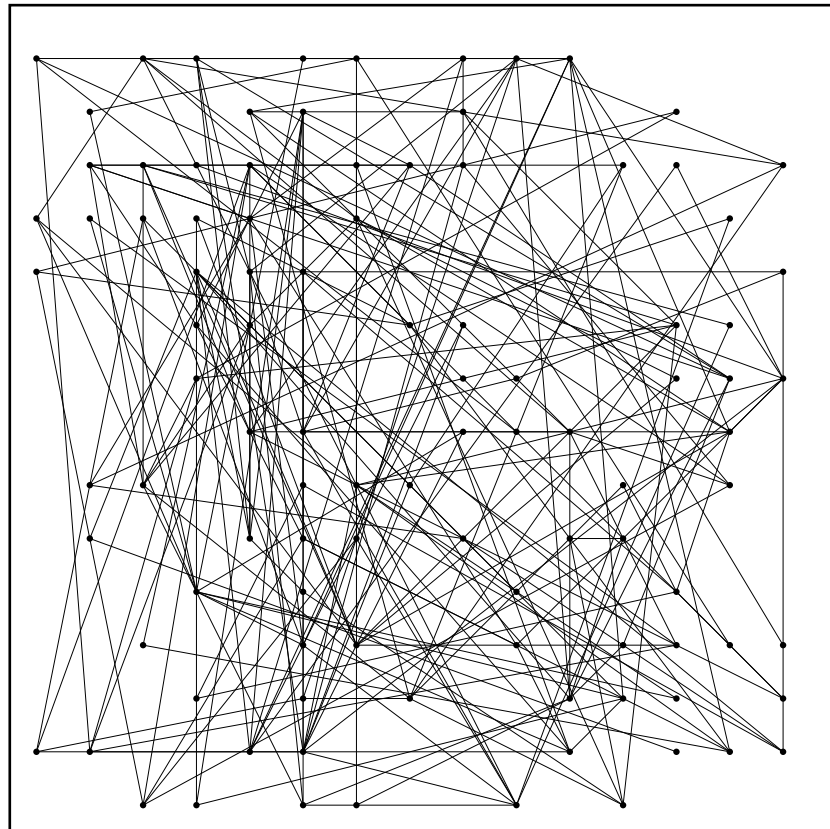


Figure 3: Example Waxman 1 Graph

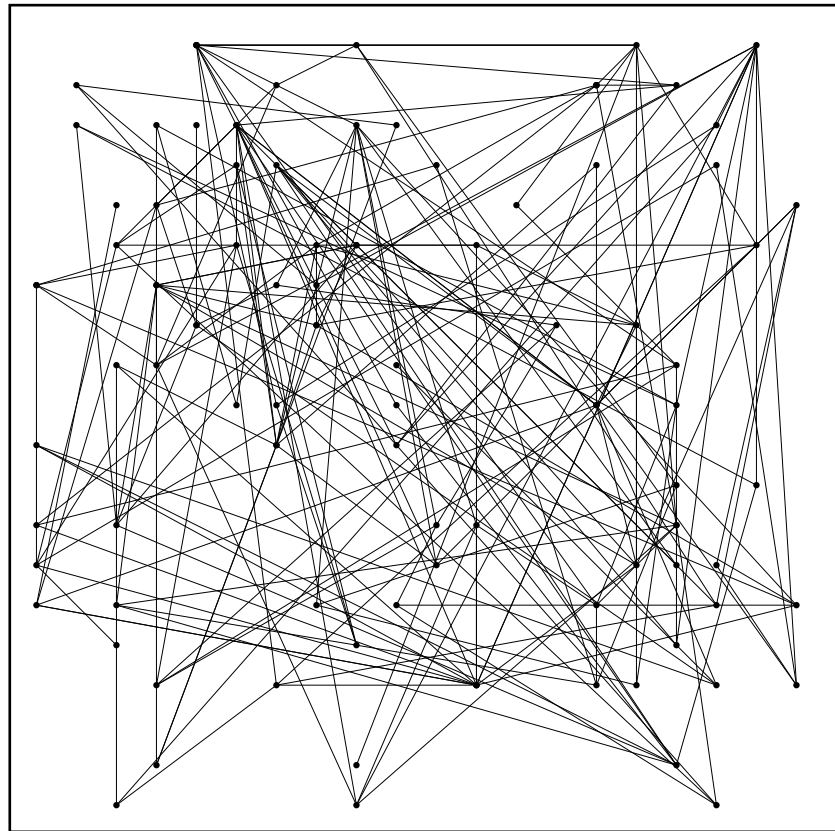


Figure 4: Example DoarLeslie Graph



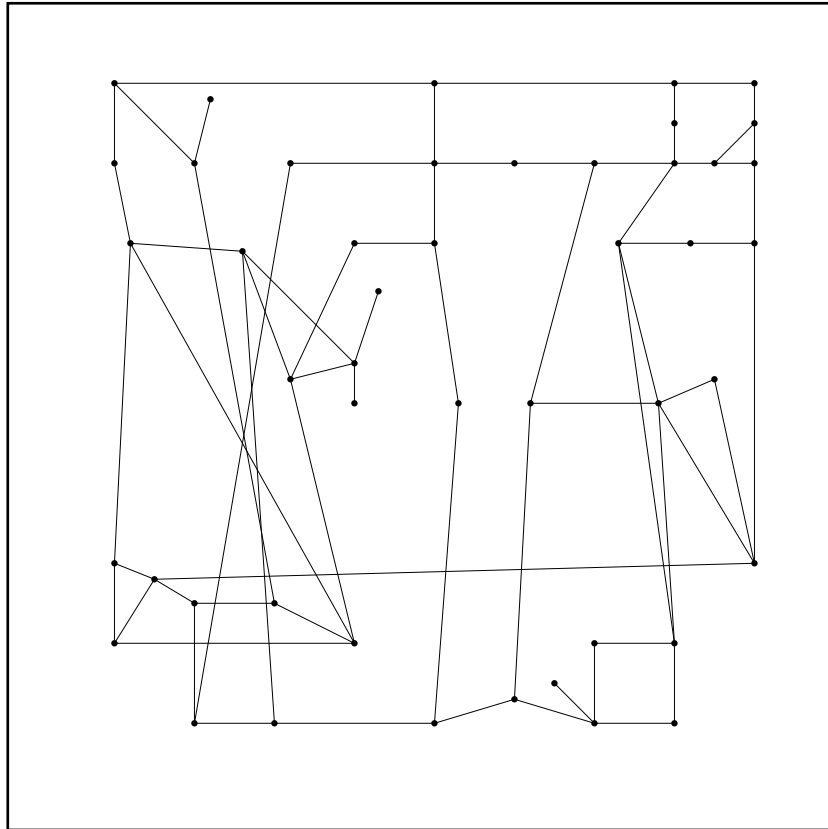


Figure 5: ARPANet

- the diameter of the graph
- the number of centers in the graph

For comparative purposes, the characteristics of the ARPANet are included. The structure of the ARPANet is shown in Figure 5 and its characteristics in Figure 6. As we will see, the ARPANet has a lower average node degree and far fewer nodes than any of the other graphs in this paper, however, its diameter is close to those of the Waxman and DoarLeslie graphs and its number of centers is within the range they encompass.

#### 4.1 Node Degree Distribution

Table 1 gives average node degrees for ten graphs with 400 nodes each of the three graph types. The only difference between two graphs of the same type is the value of the seed for the random number generator. We chose the  $\alpha$ ,  $\beta$ ,  $e$  and  $k$  values such that the average

	No. of Centers	Diameter	Avg. Node Deg.	No. Nodes
ARPA	9	9	2.89	49

Figure 6: Characteristics of ARPAnet

	0	1	2	3	4	5	6	7	8	9
Two-Level	3.28	3.23	3.42	3.42	3.25	3.33	3.39	3.41	3.30	3.47
Waxman 2	4.09	4.09	4.24	4.04	4.56	4.18	4.22	4.12	4.22	3.99
Waxman 1	4.12	4.38	3.94	4.20	4.08	4.12	4.07	4.05	4.22	4.11
DoarLeslie	4.345	4.005	4.220	3.900	4.050	4.105	4.275	4.215	4.205	4.090

Table 1: Average Node Degrees For Various Graphs

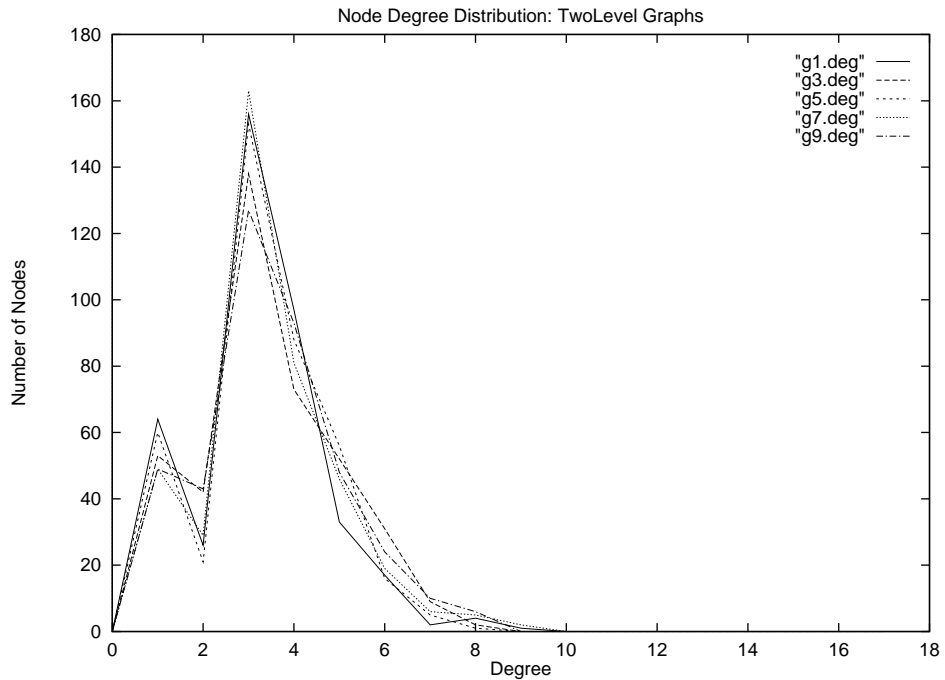


Figure 7: Two-Level Graph Node Degree Distribution

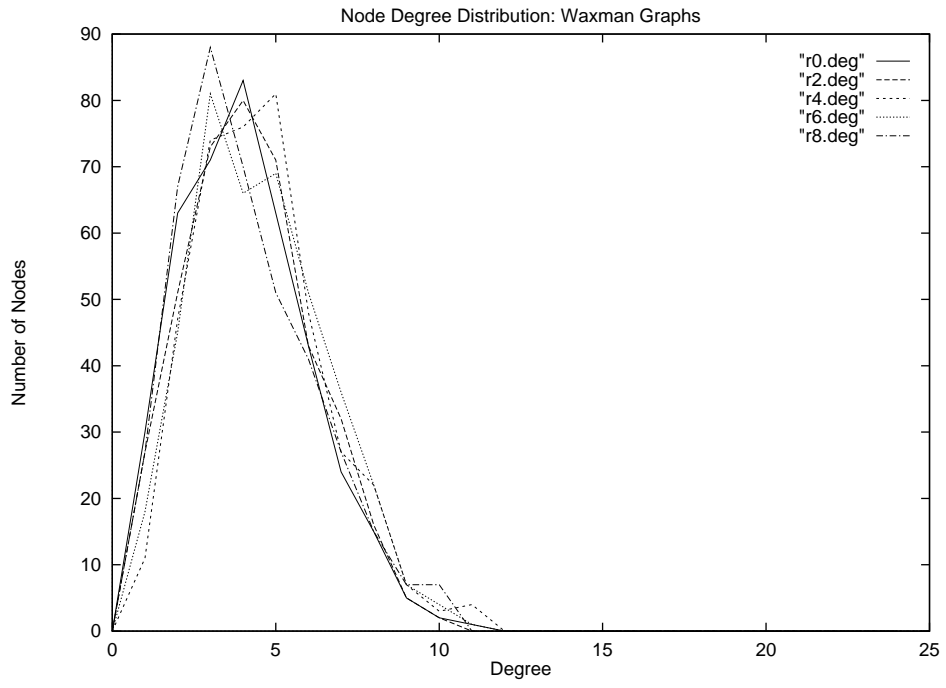


Figure 8: Waxman 2 Graph Node Degree Distribution

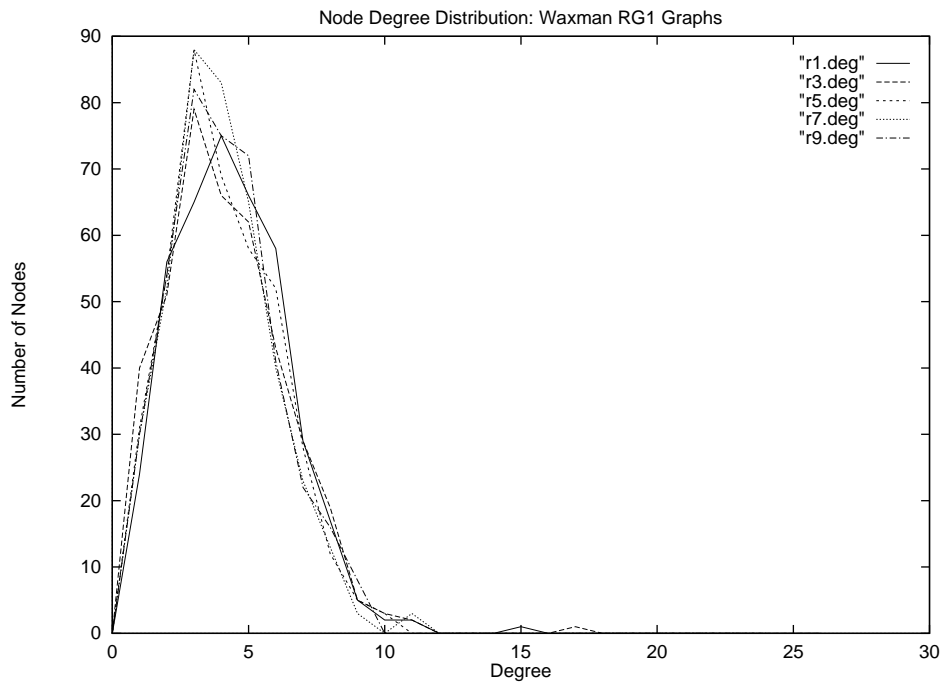


Figure 9: Waxman 1 Graph Node Degree Distribution

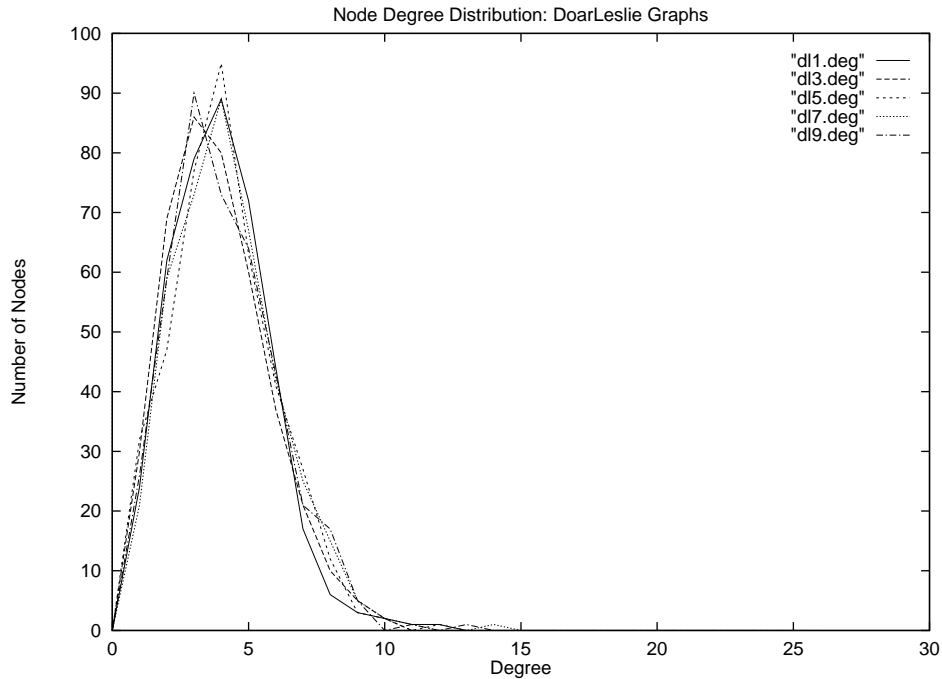


Figure 10: DoarLeslie Graph Node Degree Distribution

	0	1	2	3	4	5	6	7	8	9
Two-Level	29	19	29	26	25	20	22	29	32	24
Waxman 2	9	9	8	10	9	10	9	8	11	9
Waxman 1	9	9	12	10	10	10	10	10	9	9
DoarLeslie	10	9	11	10	10	10	9	8	10	9

Table 2: Diameters of Various Graphs

node degree would be similar in order to provide a point of comparison between the three graph types in regards to the other characteristics. Figures 7, 8, 9 and 10 show node degree distributions for five graphs of each type. As you can see, DoarLeslie and Waxman graphs look quite similar, but Two-Level graphs have a spike at nodes of degree one, which are leaves. That is due to the fact that the process of replacing top level nodes with subgraphs preserves leaves.

## 4.2 Diameter

Note in Table 2 that the diameters of the Two-Level graphs are much higher than the diameters of the Waxman and DoarLeslie graphs. That is caused by the scarcity of edges between subgraphs in a Two-Level graph, which is a result of the fact that edges only exist between two nodes if the nodes are in the same subgraph, or if they are the two nodes

	0	1	2	3	4	5	6	7	8	9
Two-Level	2	1	7	1	1	3	1	3	1	1
Waxman 2	53	45	119	33	5	22	1	1	5	74
Waxman 1	55	1	24	52	35	28	19	16	37	50
DoarLeslie	11	38	18	4	1	11	55	138	34	67

Table 3: Number of Centers in Graphs

connecting one subgraph to another. Given that the Two-Level graphs probably model the hierarchical nature of large networks better than the Waxman or DoarLeslie graphs, the discrepancy in diameters could indicate that Waxman-like graphs are not very realistic when it comes to the number of nodes a packet must, on average, traverse in a real network to get from source to destination.

### 4.3 Centers

Table 3 shows that there are far fewer centers in the Two-Level graphs, on average, than in the other graph types. This is related to Two-Level graphs’ greater diameter, and the scarcity of edges between subgraphs. Due to their nature, small subgraphs with few edges between subgraphs (unless  $\alpha$  was large), nodes in Two-Level graphs tend to be further from the “edge” of the graph. In Waxman and DoarLeslie graphs, an edge can be from any node to any other node — not confined to a particular subgraph, as the Two-Level graph nodes are — so they are more likely to span the graph and put many nodes in the position of a center. Unfortunately, they do not do so in a predictable fashion, which leads to the extreme variance in numbers of centers in the Waxman and DoarLeslie graphs. The unpredictable number of centers to these two graph types is a potential drawback to their use in modeling networks, since the exact same parameters could generate vastly different numbers of centers with a change in the seed for the random number generator.

## 5 An Illustration of Graph Use

The main reason this work on analyzing the characteristics of graphs and the differences in characteristics between different types of graphs was done was as part of a larger project studying the properties of different multicast routing algorithms [5]. Figures 11 and 12 depict the ratios of bandwidth and delay between two different routing schemes for two different types of graph, spanning multicast groups of varying sizes. One routing scheme finds its routing trees through the use of the Steiner tree approximation algorithm proposed by Kou, Markowsky, and Berman [11]. The other scheme picks one of the nodes to be the root of a shortest path tree. The bandwidths and delays generated by the two routing schemes are calculated for randomly chosen multicast groups of size ranging from five to 50. Group size is on the x-axis. The y-axis shows the ratios of the KMB approximation routing trees’ values to the shortest path routing trees’ values. The two lines in each graph represent the bandwidths and delays from the exact same method of multicast group choices and the same size multicast groups run over ten network-modelling graphs with 400 nodes each. The only difference is the

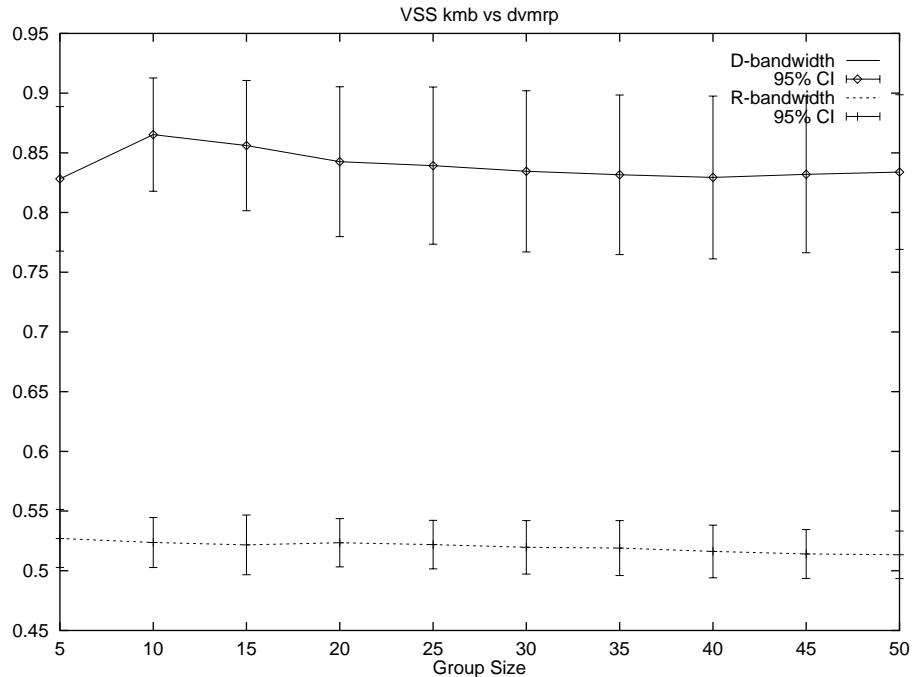


Figure 11: Differences in Bandwidth Between Different Graph Types

type of graph - the lines labeled D represent the results from Two-Level graphs and the lines labeled R represent the results from Waxman 2 graphs. The simulation results are significantly different for the different types of random graphs.

## 6 Additional Comments

One of the problems with using realistically large random graphs to model internetworks is that it is quite difficult to display pictures of the graphs on one page legibly. Nodes and edges may overlap and figuring out whether an edge is connecting two nodes or merely overlapping one of them on its way to a third node may require actually looking into the file the graph is stored in - not a pleasant prospect, as the files may contain hundreds or even thousands of nodes and edges. However, nice pictures of the graphs can be extremely useful in assisting the researcher to get a better handle on what, exactly, the model looks like or the simulation is doing. This is especially true if the nodes in the picture are numbered. AT&T has an experimental service where, if one sends graphs in a certain file format to a certain address, they will send back a PostScript picture of the graph with nice layout <sup>1</sup>. Unfortunately, AT&T limits the graphs they will process to only twenty-five nodes.

One way to get around the problem of not being able to tell if an edge is going to or through a particular node is to have two versions of the program that generates graph pictures. One version can lay out the nodes in the positions they were randomly placed in at the time of

<sup>1</sup>For more information, ftp to research.att.com and look in dist/drawdag. "mail\_server" contains further instructions and "dotdoc.ps.Z" is a user's guide.

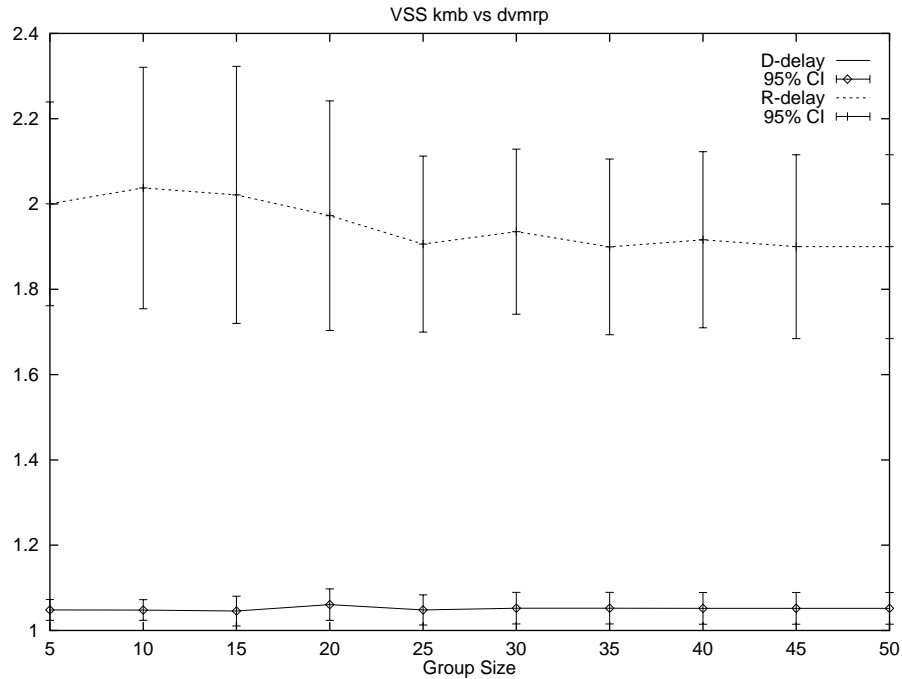


Figure 12: Differences in Delay Between Different Graph Types

graph generation and the other can perturb each node slightly, up or down or side to side. In this way, even if both pictures have places where discerning the crossing and meeting points of nodes and edges is difficult, by comparing the two versions of the graph the structure may be easier to perceive. For graphs of greater than 100 nodes, or especially high average node degree, it still may be difficult to tell what goes where on one piece of paper, but one can at least get an idea.

## 7 Future Work

Another possible random graph model for internetwork simulation is based upon the common use of spines, or backbones. One could develop a library of network backbones - graphs of the major nodes in the ARPAnet, NSFnet, and other real internetworks - and input to a program one of these backbones. The program would generate random subgraphs and attach them to the nodes in the backbone. One could specify the number of subgraphs to attach to each node in the backbone, or a range of numbers of subgraphs to attach. Also, one could specify the number of nodes in each subgraph, or a range of numbers of nodes. The subgraphs could, in fact, have subgraphs of their own.

It is all very well to use random graphs to model internetworks, but are these models realistic? We ought to figure out the rough characteristics of additional real networks, their average node degree, diameters, number of centers (or number of centers per  $x$  number of nodes), and use these characteristics to make the random graphs more closely model the real

networks. The random graphs could be tailored to match a real network - for example, a program could generate a random graph of the Two-Level type where the top level has an average node degree of five, the second level has an average node degree of three, the diameter of the top level graph is  $x$  and the diameters of the subgraphs fall in the range from  $y$  to  $z$ , and so on. Methods could be developed to add and delete nodes from the graphs in a random manner in order to more closely model the dynamic nature of real internetworks.

## References

- [1] Mostafa H. Ammar, Shun Yan Cheung, and Catarina M. Scoglio. Routing multipoint connections using virtual paths in an atm network. In *Proceedings of IEEE INFOCOM '93*, 1993.
- [2] Baruch Awerbuch. Shortest paths and loop-free routing in dynamic networks (extended abstract). In *Proceedings of SIGCOMM '90*, 1990.
- [3] Baruch Awerbuch and David Peleg. Concurrent online tracking of mobile users. In *Proceedings of SIGCOMM '91*, 1991.
- [4] Saewoong Bahk and Magda El Zarki. Dynamic multi-path routing and how it compares with other dynamic routing algorithms for high speed wide area networks. In *Proceedings of SIGCOMM '92*, 1992.
- [5] Ken Calvert, Ramesh Madhavan, and Ellen Zegura. A comparison of two practical multicast routing schemes. Technical Report GIT-CC-94-25, Georgia Institute of Technology, 1994.
- [6] Samuel T. Chanson and Jinsong Zhu. A unified approach to protocol test sequence generation. In *Proceedings of IEEE INFOCOM '93*, 1993.
- [7] Ron Cocchi, Deborah Estrin, Scott Shenker, and Lixia Zhang. A study of priority pricing in multiple service class networks. In *Proceedings of SIGCOMM '91*, 1991.
- [8] Reuven Cohen and Yoram Ofek. Label swapping routing with self-termination. In *Proceedings of IEEE INFOCOM '93*, 1993.
- [9] Matthew Doar and Ian Leslie. How bad is naive multicast routing? In *Proceedings of IEEE INFOCOM '93*, pages 82–89, 1993.
- [10] Sally Floyd and Van Jacobsen. The synchronization of periodic routing messages. In *Proceedings of SIGCOMM '93*, 1993.
- [11] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145, 1981.
- [12] Ming-Jeng Lee and James R. Yee. A design algorithm for reconfigurable atm networks. In *Proceedings of IEEE INFOCOM '93*, 1993.



- [13] Barbara Liskov, Liuba, Shrira, and John Wronclawski. Efficient at-most-once messages based on synchronized clocks. In *Proceedings of SIGCOMM '90*, 1990.
- [14] Richard G. Ogier and Vlad Rutenberg. Robust routing for minimum worst-case expected delay in unreliable networks. In *Proceedings of IEEE INFOCOM '93*, 1993.
- [15] Kihong Park. Warp control: A dynamically stable congestion protocol and its analysis. In *Proceedings of SIGCOMM '93*, 1993.
- [16] Deepinder Sidhu, Tayang Fu, Shukri Abdallah, Raj Nair, and Rob Coltun. Open shortest path first routing protocol simulation. In *Proceedings of SIGCOMM '93*, 1993.
- [17] Deepinder Sidhu, Raj Nair, and Shukri Abdallah. Finding disjoint paths in networks. In *Proceedings of SIGCOMM '91*, 1991.
- [18] Dinesh C. Verma and P.M. Gopal. Routing reserved bandwidth multi-point connections. In *Proceedings of SIGCOMM '93*, 1993.
- [19] Zheng Wang and Jon Crowcroft. Shortest path first with emergency exits. In *Proceedings of SIGCOMM '90*, 1990.
- [20] Bernard M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.
- [21] L. Wei and D. Estrin. The trade-offs of multicast trees and algorithms. In *To appear in Proceedings of ICCCN '94*, 1994.
- [22] William T. Zaumen and J.J. Garcia-Luna Aceves. Dynamics of distributed shortest-path routing algorithms. In *Proceedings of SIGCOMM '91*, 1991.