Sampling Internet Topologies: How Small Can We Go?

Vaishnavi Krishnamurthy*, Junhong Sun†, Michalis Faloutsos‡, Sudhir Tauro§
Department of Computer Science, U.C.Riverside, Riverside, CA 92521, Tel:909-787-2434, Fax:909-787-4643
This work was supported by the NSF CAREER grant ANIR 9985195, and DARPA award FTN F03602-01-20535

Abstract*— *In this paper, we develop methods to "sample" a large real network into a small realistic graph. Although topology modeling has received a lot attention lately, it has not yet been completely resolved. Several methods create arguably realistic topologies from scratch. Our approach moves in the exact opposite direction. First, we observe that many real topologies are available to the networking community. However, their size makes them expensive to use in simulations as is. This brings up the following question: how can we shrink a graph, so that it still retains its essential properties? We propose an iterative sampling framework and seven different "sampling" methods. We show that some of our methods can be very effective: they reduce a graph by 70%, and maintain several topological properties within 22% of the expected value. An advantage of this method is that it can potentially maintain topological properties that we are not yet aware: all we have to is sample "fairly". In addition, our methods are statistically robust and reliable. We find that apart from its practical applications, the problem of graph sampling is of interest in its own right.

I. Introduction

"How can I reduce a large graph to a smaller graph?" This paper revolves around this question. The reduced graph must have properties similar to the original graph. We develop a framework for sampling real graphs. The motivation for this attempt is twofold. First, the graph reduction can help us see how topological properties "scale" with size. Second, the approach can help generate small realistic graphs which are useful for simulation purposes. Despite recent significant activities, a definite appropriate topology which can be used for all empirical purposes has not been developed. Although there exist several real instances, they are quite large for simulation purposes [12] [13], especially if the details of the simulation are fine as in packet level simulations.

In [5], Faloutsos et al. showed that the Internet followed power-laws with a very high correlation coefficient. Following these power-law observations, several new generators create power-law topologies [3] [4] [15] [16] in contrast to the generators before them [1] [2] [9] [10]. For example, the popular Barabasi-Albert model [3] generates a graph by adding nodes preferentially to the existing nodes: it prefers nodes with high degree. It is easy to see that the specifics of the preferential function can give rise to different classes of graphs [15]. In addition, these methods attempt to match the degree distribution, but they often miss significant other properties [15] [4]. In other words, the small changes in constants, let alone the principles of construction have significant effects in the resulting topology. We call these methods of generating a graph, the constructive approach. The graph partitioning in [24] [25] bears some similarities to the problem of graph sampling, but the metrics of the former are different from ours presented here, and so, the algorithms for graph partitioning cannot be used here.

In contrast to the constructive approach, we sample real topologies and "shrink" them, naming this the reductive approach. Our contributions can be summarized in the following points.

- We suggest an iterative framework to generate small realistic topologies from an initial topology.
- We propose seven methods to do the reduction at each step of the framework. These methods dictate how we select, remove or merge nodes or edges.
- We compare the proposed sampling methods with respect to some chosen metrics. Some reduction methods seem to work very well. They can reduce a graph by as much as 70%, preserving sufficiently its topological properties (within 22%).
- Our methods seem statistically robust. They show relatively little sensitivity to both the random initial seed and the initial topology.

It is worth noting that the reductive approach has two attractive properties:

1) If the reduction is "statistically fair", it may preserve graph properties that we have not yet identified.
2) The approach can be used to reduce significantly different types of graphs [1].

The rest of the paper is organized as follows: Section II presents the topological metrics we have used to gauge the methods, Section III our iterative reduction algorithm and proposes seven graph reduction methods. The performance of each method is studied using various metrics in Section IV. Section V concludes the paper with some future work.

II. Topological Properties and Metrics

In this section, we introduce the topology model and several topological properties of the Internet. In our study, we focus on the inter-domain topology. We model the Internet as an undirected graph whose nodes are domains and whose edges are inter-domain connections. We select 104 real inter-domain topology instances from November 1997 to November 1999, one instance per week. All of these instances are provided by the National Laboratory for Applied Network Research. Each instance is named using its date, e.g. the instance collected on November 8, 1997 is named 971108, this naming convention will be followed in this paper.

To evaluate our method, we use a subset of the observed properties [5]. These properties are a necessary condition but may not form a sufficient condition

[1]One reduction method may not be the best for all types of topologies, but it may be good for several types of topologies.

for the realism of a graph. The main metrics used to evaluate the generated graphs are listed below.

Average Degree: The average degree of a graph is defined as 2m/n, where m is the number of inter-domain links and n is the number of domains, which indicates the density of the graph. The average degree increases over time, growing from 3.42 in November 1997 to 3.93 in November 1999 (15% growth). At the same time, the size of the Internet approximately doubled (100% growth). We fit a straight line for the variation of average degree for a corresponding change in the size of the graph. From this equation, we found that if we reduce the 981103 graph by 70%, we expect a 8% decrease in the average degree.

Power Laws 1 and 2: Faloutsos et al [5] have revealed the existence of power-laws in the Internet topology. The power-laws describe succinctly the skewed distributions of the graph properties, such as the degree distribution. Both the existence and the slope of the power-laws can be used as criteria for the comparison of graphs [6]. Here, we focus on power-law 1 (Rank Exponent) and 2 (Degree Exponent), as they seem to be the most effective in distinguishing different kinds of topologies according to [6]. We can claim that a reduced graph is very similar to the original graph, if it has the same slope value as that of the original graph with a very high correlation coefficient.

Clustering Coefficient: Clustering coefficient [15] characterizes the connectivity of the neighborhood of a node. This metric captures the local density of a graph. For a particular node, it is defined as the ratio of the number of edges in the neighborhood of that node to the total number of possible edges in the neighborhood. The overall clustering coefficient of a graph is the average of the clustering coefficient of all the nodes with an outdegree greater than one. Intutively, it answers the question: are my neighbors connected among themselves? We calculate the clustering coefficient for the original Internet over a period of 3 years from 1997 - 2000. We note that there is a steady increase in the clustering coefficient as the number of nodes increases over that period of time, with very few exceptions. Figure 2 shows the best fit curve for the above with a correlation coefficient of about 99.2%. We then extrapolate this line to calculate the clustering coefficient of a graph with about 1250 nodes (i.e. a 70% reduced 981103 graph). This is our expected value of clustering coefficient which is about 0.303.

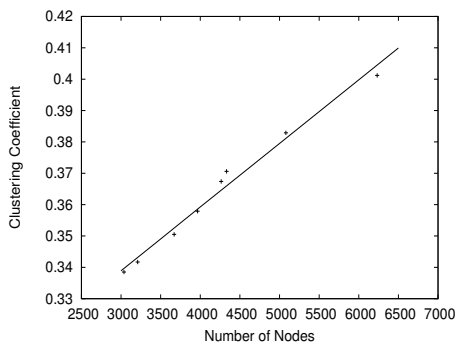

Fig. 1. Clustering Coefficient Vs Number of nodes in the Internet

III. Graph Reduction

In this section, we present our approach for reducing a real graph to a small realistic topology. We develop seven methods of shrinking that proceed by (a) deleting nodes or edges, (b) merging nodes, (c) retaining the induced subgraph. The resulting reduced topology is compared with the expected topology.

A. Iterative Graph Reduction Algorithm

Our framework reduces the graph iteratively by removing a small percentage, s, of the graph in each iteration. After each reduction, the set of topological metrics described in Section II is calculated. The algorithm stops when the graph reaches the desired size. In our experiments, we reduce the graph until the properties diverge too much.

Given a graph G with n nodes and the total percentage of nodes to be removed as P, the algorithm uses a reduction method to reduce G to a smaller graph with $n * (P/100)$ nodes. The metric values are compared with those of the original graph to evaluate the realism of the reduced graph.

In some of our reduction methods, the graph can become disconnected. In such cases, we choose the largest connected component and discard the rest. By reducing a small percentage of the graph in each iteration, we are able to meet the target size more accurately, thus providing more control. In practice, a reduction of 3% to 5% of the nodes at each step seemed to work out very well. Note that after every reduction step we find the largest connected component and use it as the beginning topology for the next reduction step. This process continues until we reach the total percentage reduction.

B. Graph Reduction Methods

We propose five graph reduction methods for our iterative graph reduction algorithm.

Random Vertex Deletion (RVD): We randomly pick one node, and delete all edges between it and its neighbors, and then delete the node itself from the graph. The graph might become disconnected after the deletion. As we already mentioned, we keep the largest connected component.

True Random Edge Detection (TRED): In this method, we arrange all the edges and randomly pick an edge. All the edges have equal probability of being chosen for deletion.

Random Edge Detection (RED): We randomly select one vertex, then randomly pick one of its neighbor vertices and delete the edge between them. If we pick a isolated vertex, we ignore it.

n-Neighbor Clustering (CLST): We define clustering as follows: we merge neighbor vertices of a node and the node itself into one single node. The neighbors of all the merged nodes become neighbors of the new node. The n-Neighbor Clustering method aggregates a node and n of its neighbors into one node. In our experiments we use n = 2. We tried larger values but the results were worse and they are not shown here.

Edge Shrinking (SRK): The Edge Shrinking method is a special case of the n-Neighbor Clustering method with n = 1. This method is very similar to the random matching method described in [24] or the edge coarsening method described in [25].

C. Induced Subgraph Methods

We propose two ways to keep a part of the initial topology. We pick an initial node randomly and then explore to choose a part of its neighbourhood which

TABLE I

COMPARISON OF THE SEVEN REDUCTION METHODS FOR A REDUCTION OF ABOUT 68% - 70%

Metric	RVD	TRED	RED	CLST	SRK	BFS	DFS	Target
Average Degree	-15%	-39%	-3%	-10%	+11%	+14%	+13%	-8%
Rank exponent	+5%	+8%	-4%	+13%	-8%	+3%	-8%	0%
Degree exponent	-7%	-11%	-5%	-20%	-1%	-12%	+4%	0%

we retain in the final graph.

Subgraph by BFS (BFS): We randomly select one node, and then do breadth-first search (BFS) starting from that node, until the desired size has been achieved. All nodes that have been visited are retained in the final graph. All other nodes and the edges between them and their neighbors are deleted.

Subgraph by DFS (DFS): We do a depth-first search (DFS) on the graph, starting from a random node. After the desired number of nodes has been visited, we keep all nodes that have been visited and delete all other components.

IV. PERFORMANCE ANALYSIS

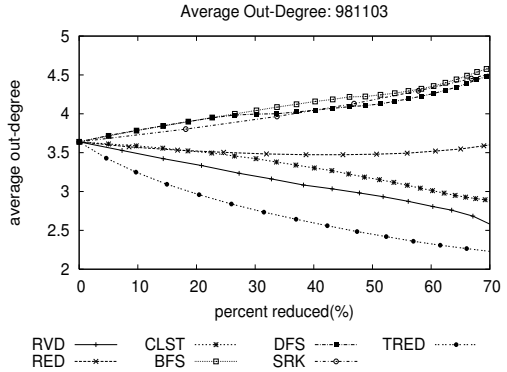

Fig. 2. Average degree comparison of the seven methods

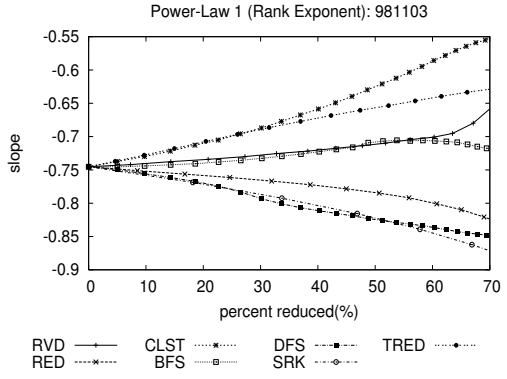

Fig. 3. Rank exponent comparison of the seven methods

In this section, we examine the performance of our sampling methods. Our experimental results show that among the four methods RED, RVD, TRED and CLST, RED and RVD gives the best overall results, followed by TRED, while CLST is the worst. We will explain below why we have considered only these four methods for a final judgement.

If we consider only the average degree, CLST has the best performance followed by RED. Figure 2, 3 and 4 show the mean variations of the average degree, the rank exponent and the degree exponent respectively, when the seven methods are applied to

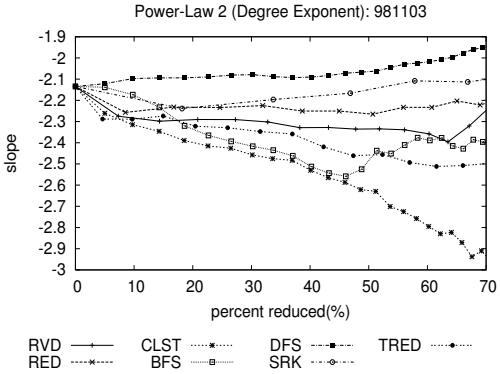

Fig. 4. Degree exponent comparison of the seven methods

instance 981103. As mentioned in section II, we expect a reduction of roughly -8% in the average degree for a 70% topology reduction. Based on these observations and the results of Table I, we see that CLST deviates the least from the expected -8%, followed by RED and RVD. RED produces a very small decrease in the average degree. TRED also decreases the average degree but the percentage of reduction is very large. The last three methods, SRK, BFS, DFS donot fare well in this metric comparison, since they actually increase the average degree and are therefore not considered further.

Considering the power-law metrics, RED and RVD are the best methods. The slope of the rank exponent and the degree exponent for these two methods varies the least from the original value of slope. TRED deviates more than these two methods, while CLST fails completely on the power-law metrics. Therefore, we only consider RVD, TRED and RED; the latter performs better with regard to power-laws.

Considering clustering coefficient, RVD seems to outperform TRED but is followed very closely by RED. We test our methods RED, RVD and TRED using the clustering coefficient [15]. We compare the graphs from RED, RVD and TRED with the expected real topology. Table II shows the summarized results of the clustering coefficients for the original Internet expected, RED, RVD and TRED. None of the methods have a value very close to the expected value. Among these three methods, the percentage deviation of RVD from the expected value is about 19% followed closely by RED with a 22% deviation. Since TRED has an extremely low value of clustering coefficient, we think RED and RVD are better than TRED.

It is fair to claim that both RVD and RED performs well in shrinking an Internet graph. Even though RED seems to perform the best, the difference in deviation between its performance and that of RVD is small, even negligible. RED deviates 3% from the 981103's average degree, while the expected deviation is about 8%. RED is followed closely by RVD with a 15%

deviation. RED performs very well in power-laws 1 and 2 and is once again followed very closely by RVD with nearly the same percentage of deviations. RVD is better with respect to clustering coefficient, but RED follows RVD closely in this metric. Since, RVD and RED seems to perform equivalently with respect to the four metrics, both seem to be good methods of shrinking a graph.

TABLE II

COMPARISON OF CLUSTERING COEFFICIENT OF RED, RVD, TRED WITH THE INTERNET FOR A GRAPH HAVING 1250 NODES

Metric	Expected Value	RED	RVD	TRED
Clustering Coeff	0.303	0.37	0.36	0.05

V. CONCLUSION

In this paper, we propose a framework for sampling real network topologies. With our framework, we show that we can reduce a graph by as much as 70% in size and maintain several topological properties. This makes our framework a promising tool for generating graphs for network simulations. An additional advantage is that the resulting graphs may preserve some unknown topological properties of the Internet. Our contributions can be summarized in the following points.

- We propose a small-decrement iterative framework that offers more control over the reduction process, so that we can create graphs of desired size.
- We propose seven reduction methods to reduce the graphs that we can group in three main categories: removal of components (nodes or edges), merging clusters of nodes and retaining an induced subgraph.
- The Random Edge Deletion (RED) and Random Vertex Deletion (RVD) seems to perform better in comparison to the other five methods.
- All methods seem robust to the randomization seed and the initial topology.

Our experiments lead to the following tips for practitioners.

- RED and RVD methods can be used for graph reduction in practice for Internet like topologies.
- It is advisable to pick a small incremental step (e.g 3% or 5%), in order to reach desired size more accurately.
- Reduction of the graph by more than 70% is not advisable as at that point the graph starts to diverge from the initial properties significantly.

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

Future Work We are developing analytical proofs to examine why some of our methods adhere to power-laws with such a high correlation coefficient. Our initial analysis is in agreeement with the experimental results we get here.

REFERENCES

[1] K. Calvert, E. Zegura and M. Doar *Modeling Internet Topology*, IEEE Trans. on Communications, 160-163, December 1997

[2] M. Doar *A Better Model for Generating Test Networks*, Proceedings of Global Internet, November 1996.

[3] A. Barabasi and R. Albert *Emergence of Scaling in Random Networks*, Science, vol. 286, 509-512, October 1999.

[4] C. Jin, Q. Chen and S. Jamin *INET: Internet Topology Generator*, U. Michigan, Tech. Report CSE-TR-433-00, 2000.

[5] M. Faloutsos, P. Faloutsos and C. Faloutsos *On Power-Law Relationships of the Internet Topology*, ACM SIGCOMM 1999.

[6] A. Medina, I. Matta and J. Byers *On the Origin of Power Laws in Internet Topologies*, ACM Computer Communication Review, 30(2):18-28, April 2000.

[7] R. Govindan and A. Reddy *An analysis of Internet inter-domain topology and route stability*, IEEE INFOCOM, April 1997.

[8] V. Paxson and S. Floyd *Why We Don't Know How to Simulate the Internet*, Proceedings of the 1997 Winter Simulation Conference, December 1997.

[9] E. Zegura, K. Calvert and S. Bhattacharjee *How to Model an Internetwork*, Proceedings of IEEE INFOCOM, San Francisco, CA, April 1996.

[10] E. Zegura, K. Calvert and M.J. Donaloo *A quantitative comparison of graph based models for Internetworks*, IEEE/ACM Transactions on Networking, 5(6): 770-783, December 1997.

[11] B.M. Waxman *Routing of multipoint connections*, IEEE journal of Selected Areas in Communications, 1617-1622, 1998.

[12] R. Govindan and H. Tangmunarunkit *Heuristics for Internet Map Discovery*, INFOCOM, March 2000.

[13] B. Cheswick and H. Burch *Internet Mapping Project*, Wired Magazine, December 1998.

[14] R. Albert and A. Barabasi *Topology of an Evolving Network: Local Events and Universality*, Physical Review Letters, 85:5234-5237, 2000.

[15] T. Bu and D. Towsley *On Distinguishing between Internet Power Law Topology Generators*, INFOCOM 2002.

[16] W. Aiello, F. Chung and L. Lu *A Random graph model for massive graphs*, Proc. 32nd ACM Symposium on Theory of Computing, 2000.

[17] D. Magoni and J. Pansiot *Analysis of the Autonomous System Network Topology*, Computer Communication Review, July 2001.

[18] S. Tauro, C. Palmer, G. Siganos and M. Faloutsos *A Simple Conceptual Model for The Internet Topology*, 2001 Global Internet.

[19] Z. Ge, D. Figueiredo, S. Jaiswal and L. Gao *On the Hierarchical Structure of the Logical Internet Graph*, ITCOM'2001.

[20] G. Siganos, M. Faloutsos, C. Faloutsos and P. Faloutsos *Power-laws and the internet topology*, Submitted to IEEE Transactions on Networking.

[21] S. Yook, H. Jeong and A. Barabasi *Modeling the Internet's large-scale topology*, Proceedings of the nat'l Academy of Sciences 99, 13382-13386.

[22] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker and W. Willinger *The Origin of Power Laws in Internet Topologies Revisited*, Proc. IEEE INFOCOM, June 2002.

[23] Z. Ge, D. Figueiredo, S. Jaiswal and L. Gao *On the Hierarchical Structure of the Logical Internet Graph*, ITCOM'2001.

[24] G. Karypis, V. Kumar *A Fast and High Quality Scheme for Partitioning Irregular Graphs*, Technical Report, Department of Computer Science, University of Minnesota: 95-035.

[25] G. Karypis *Multilevel Hypergraph Partitioning*, Technical Report, Department of Computer Science, University of Minnesota: 02-025.

[26] J.Sun *Generating Realistic Network Topologies for Simulation Purposes*, Thesis, Department of Computer Science, University of California, Riverside, Dec 2000.