On the semantics of Internet topologies

Milena Mihail, Christos Gkantsidis, Amin Saberi, Ellen Zegura
{mihail, gantsich, saberi, ewz}@cc.gatech.edu
College of Computing
Georgia Institute of Technology
Atlanta, GA
Technical Report: GIT-CC-02-07

Abstract— **Models for network topology are necessary for simulation-based studies of a variety of networking problems. Increasingly the research community is interested in problems that arise due to the large scale of the Internet (e.g., BGP routing, performance of peer-to-peer systems). For these sorts of problems, the potential to model the full-scale AS-level topology is appealing. To date, the most successful approach to modeling the AS-level topology is the degree-driven approach of Inet. Inet predicts the degrees of the topology by extrapolation from available data, then constructs a topology meeting the degree sequence using a preferential connectivity heuristic. We focus on two important areas left open by prior work. First, we explore the theoretical foundations of degree-based graph generation. We identify the relevant results from graph theory, and exploit these to improve fundamental understanding and produce richer models. Second, essentially all prior AS-level models have the characteristic that they contain extremely limited *semantics*. The graphs produced are undirected and unlabeled, hence they simply reflect connectivity without any notion of additional semantic information. We address the issue of adding semantics to network topologies, in the areas of peering relationships and clustering into higher-level groupings based on geographic and/or business relationships. Our techniques for adding semantics suggest new methods for evaluating the quality of generated topologies.**

I. Introduction

Models for network topology are necessary for simulation-based studies of a variety of network problems. Early modeling efforts focused primarily on router-level topologies [5], [34] and attempted to capture *geographic locality* – that is, network design favors shorter links, hence routers that are geographically nearby are more likely to be connected and cluster – and *hierarchy* – that is, the Internet is organized into administrative domains with greater likelihood of intra-domain connectivity than inter-domain connectivity. The development of good router-level models has been hampered by the difficulty of obtaining real data, as well as the overwhelming scale of the router-level Internet topology.

More recently, effort has focused on modeling the AS-level topology, where the nodes of the graphs represent autonomous systems (also called domains), and the links represent the exchange of traffic and routing information between the corresponding domains according to the Border Gateway Routing Protocol (BGP). AS-level modeling has the appeal that data is available on (a portion of) the AS-level Internet topology derived by peering with border routers (e.g., NLANR [9], Looking Glass [32]). One can therefore determine some properties of the real data, providing guidance in the development and evaluation of models [8]. In addition, the full Internet AS topology is of somewhat manageable size (on the order of 10,000 nodes in 2001).

The research community currently has considerable interest in problems that specifically arise due to the large scale of the Internet, and therefore require simulations that are "big enough" to reveal the issues and test candidate solutions. Problems of this variety include stability and convergence of BGP routing, efficiency and stability of peer-to-peer lookup systems, and placement of server replicas. For these sorts of problems, the potential to model the full-scale AS-level topology is appealing.

There have been two main approaches to modeling the AS-level topology. The first, typified by the BRITE generator [22], is evolution-based in the sense that it produces a topology incrementally, by adding one node at a time to an existing topology. BRITE uses the principle of preferential connectivity to produce graphs with the skewed statistics present in the real data. The second approach, used in the Inet generator [16], is degree-driven. In particular, Inet first predicts the degrees of the topology by extrapolation from NLANR data, and then constructs a topology meeting the degree sequence using a preferential connectivity heuristic. The topologies generated by Inet come remarkably close to fitting the real data on some measures. This has resulted in prevalent use of Inet, as well as a hypothesis that degree distribution is fundamental in producing topologies that match the real data [36].

We focus on two important areas unexplored by prior work. First, we focus on the theoretical foundations for degree-based graph generation, and exploit them to improve fundamental understanding and produce richer models. Second, essentially all prior models of the AS-level topology have the characteristic that they contain extremely limited *semantics*. The graphs produced are undirected and unlabeled, hence they simply reflect connectivity without any notion of additional semantic information[1]. We address the issue of adding semantics to AS-level network topologies, in the areas of BGP peering relationships and clustering. Our clustering methods make novel use of graph theoretic metrics and spectral filtering (eigenvalue-

[1] In router-level modeling, there have been some efforts to add semantics to topologies. For example, the transit-stub model in the GT-ITM suite contains link weights that reflect routing semantics.

eigenvector) techniques [2], capturing strong global semantics that reflect geographic proximity and/or business relationships. The methods and the results are thus of substantial independent interest.

Our contributions include the following:

- Demonstration that degree-sequence is not sufficient for producing topologies that are a good match to real data, in the sense that one can produce topologies meeting the degree sequence that differ significantly from one another and from the real data. Such topologies can be useful as stress tests for evaluating solutions, since they represent extremes that are more plausible than purely artificial extreme test cases (e.g., pure random graphs, regular graphs).
- A Markov chain Monte Carlo method to produce a random graph meeting a target degree sequence, and evidence that random graphs meeting a degree sequence are a good match to real data in several metrics (comparable to the quality of Inet).
- Enumeration of several important extensions of the fundamental graph theory, for example to produce a graph that meets simultaneously a given in-degree sequence and a given out-degree sequence.
- Development of a method to add peering relationship semantics (provider-customer and peers) to a topology. This method uses an iterated pruning technique to identify the particular semantics to assign to each link.
- Application of the technique of singular value decomposition to identify "clusters" of ASes. The ability to identify high-level relationships between ASes (i.e., beyond local neighbor relationships) is critical to ensuring that synthetic topologies capture important coarse-grain characteristics. High-level groupings may also be valuable for assigning semantics to links (e.g., intra-group links may be of higher bandwidth than inter-group links). Previous work has focused on hierarchy in the AS-level graph [38], [17], [36]. Stated with extreme informality, hierarchy captures the "up-down" characteristics of AS relationships. ASes at the same level of the hierarchy have no explicit relationship to one another beyond similar "status". Clusters, on the other hand, capture relationships both within and across levels of hierarchy, between ASes that form various communities (typically business or geography based).
- Demonstration that while current methods for generating topologies may do a good job capturing hierarchy (indeed, hierarchy has been observed to follow when one meets degree sequence [36]), these methods do not contain the strong clusters present in the real data. This raises the open question of generation methods that meet a degree sequence while also incorporating clustering.

The balance of the paper is organized as follows. In the next section we explore the theoretical foundations of degree-based graph generation. We outline methods to generate qualitatively different graphs meeting the same degree sequence, as well as a Markov chain-based method to produce a random graph meeting a degree sequence. In Section III we evaluate the graphs produced using these methods and compare them to real AS-level data. We then turn to methods for adding peering relationships (Section IV) and for identifying clusters (Section V). We discuss related work in Section VI and summarize in Section VII.

[2]Also known as Singular Value Decomposition, Principal Vector Analysis and Latent Semantic Indexing.

II. Foundations in Degree-Based Graph Generation

In this section we provide the theoretical foundations for degree-based graph generation. In particular, we review the necessary and sufficient conditions for generating a connected and simple graph meeting a sequence of target degrees. This review of the underlying theory points the way to a construction algorithm that contains considerable flexibility in generating a topological model while meeting the degree sequence. The flexibility in the algorithm highlights the fact that, in general, many graphs meet a given degree sequence. This gives rise to the question of generating a "random" instance from the space. We give an efficient method for such random generation based on Markov chain theory. In the next section we evaluate the properties of models generated by using the algorithms developed in this section.

A. Realizing a target degree sequence

Let n denote the number of nodes of the topology we wish to generate. Let $v_i, 1 \leq i \leq n$ denote the nodes and $d_1 \geq d_2 \geq \ldots \geq d_n$ denote the degrees of these nodes. We would like to generate a simple graph, i.e., a graph without self-loops or multiple links between a pair of nodes. In addition, we want the topology to be a connected graph. We deal with these two conditions separately.

In classical graph theory, we call a sequence of degrees $d_1 \geq d_2 \geq \ldots \geq d_n$ "realizable" if and only if there exists a simple graph whose nodes have precisely this sequence of degrees. A necessary condition for a degree sequence to be realizable is that for each subset of the k highest degree nodes, the degrees of these nodes can be "absorbed" within the nodes and the outside degrees, meaning that there are enough edges within the subset and to the outside to account for the necessary degrees of the k nodes. Stated formally, for $1 \leq k \leq n-1$:

$$\sum_{i=1}^{k} d_i \quad \leq \quad k(k-1) + \sum_{i=k+1}^{n} \min\{k, d_i\}$$

A classical theorem of Erdos and Gallai states that this necessary condition is also sufficient [7], [4]. The proof is inductive and provides the following natural construction algorithm [12], [13]. The algorithm is iterative and maintains the residual degrees of vertices. In each iteration, it picks an arbitrary vertex v and adds edges from v to d_v vertices of *highest* residual degree, where d_v is the residual degree of v. The residual degrees of the latter d_v vertices are updated appropriately. The significance of connecting with d_v highest degree vertices is that it ensures the Erdos-Gallai condition holds for the residual problem instance.

For example, the algorithm can start by connecting the highest degree vertex with d_1 other high degree vertices and obtain a residual degree sequence by reducing the degrees of these vertices by one, and repeat the same process until all degrees are satisfied, otherwise output "not realizable". Alternatively, the algorithm can connect the lowest degree vertex d_n, (or a randomly chosen vertex d_i) with the d_n (resp. d_i) highest degree vertices, reduce their degrees and proceed as above.

Clearly the algorithm described above runs in linear time. In addition, since the sequence in which it picks vertices can be chosen, it provides the flexibility alluded to above. For example, when we start with higher degree vertices we get topologies that have very "dense cores", while when we start with low degree vertices we get topologies that have very "sparse cores". We elaborate the notion of "core density" in Section III, however, it should be intuitively clear that these represent extreme instantiations of topologies with a particular degree sequence.

The Erdos-Gallai condition allows for additional flexibility, which results in topologies more closely resembling real data and Inet output. The idea is to use the principle of *preferential attachment* for choosing the d_v vertices to which v will be connected, rather than the maximum degree d_v vertices. Thus, the d_v vertices will be chosen with probabilities proportional to their residual degrees. After each iteration, we need to ensure that the Erdos-Gallai condition is satisfied by the residual graph (this part was automatic in case maximum degree vertices a chosen). If not, the probabilistic choice needs to be repeated. If it fails several times, we can go back to choosing maximum degree vertices.

Next, let us deal with the second requirement of obtaining a connected topology. A necessary and sufficient condition is that the graph must contain a spanning tree. To contain a spanning tree, the sum of the degrees must be at least $2(n-1)$. Suppose that the given degree sequence has sum at least $2(n-1)$. First construct a graph as stated above. If this graph turns out to be unconnected, then one of the connected components must contain a cycle. Let (u,v) be any edge in a cycle and let (s,t) be an edge in a different connected component. Clearly, the graph does not have edges between the pairs u,s and v,t. By removing the edges (u,v) and (s,t), and inserting the edges (u,s) and (v,t), we merge these two components. Note that the resulting graph still satisfies the given degree sequence. Proceeding in this manner, we can get a connected topology.

B. Generating a random graph

We have demonstrated that there are many ways to construct a topological model while meeting the target degree sequence. That is, we have the ability to generate instances with particular structure (e.g., dense core, sparse core) using the above techniques. The ability to generate graphs with particular structure is clearly useful in providing stress test (or worst case) examples for evaluating solutions.

We now turn to the question of generating a *random* instance from the space of all possible graphs that realize the target degree sequence. The ability to generate a random graph from this space is also clearly useful in understanding the expected or average behavior of a solution.

For this purpose we draw on ideas from the theory of Markov chain-based algorithms. In our case the Markov chain is the following process. Start from any realization of the given degree sequence, G. Pick two edges at random in G, (u,v) and (s,t) with distinct endpoints, such that (u,s) and (v,t) are not edges in G. Remove the edges (u,v) and (s,t) and insert the edges (u,s) and (v,t) (repeatedly performing such "local" perturbations is a standard technique in Markov chain-based algorithms [23], [30]; in the context of Internet topologies they can be thought of as "small rewirings" [37]). Observe that the resulting graph does satisfy the given degree sequence. We further have to check whether it is a connected graph. If it is connected then we perform this switching operation, otherwise we do not.

It follows from a theorem of Berge and Taylor [4], [31] and from Markov chain theory [23], [30] that, independent of the starting point, in the limit, this procedure will reach every possible connected realization with equal probability. Hence, in the limit, the Markov chain-based algorithm will generate a random (or "typical") topology with the given degree sequence. To be practical, we must address the question of the time needed to simulate the process before we are close to its limiting condition, as well as the method for detecting when sufficient steps have been taken (i.e., a stopping rule).

Our stopping rule is derived from a non-isomorphism measure between the starting topology and the topology at some point of the simulation. Recall that these topologies have the same degree sequence. For all nodes that have unique degrees (i.e., no other node has the same degree), we sort their neighbors by degree and compute the total number of entries in which the two sorted lists differ. Intuitively, we expect that the larger this difference count becomes, the more "different" the graphs are.

As we simulate the Markov chain, we find that this distance measure initially grows almost linearly and then levels off. Let T denote the time (number of steps) in which the measure levels off. As a heuristic, for instances with up to 12000 nodes (such as today's AS-level topology), we recommend running the simulation for $3T$ total steps. As points of reference, we find T to be less than 10,000 for a 3000 node graph similar to the NLANR data set in December 1997. For a 7500 node graph similar to NLANR data in December 2000, T was less than 75000. For a 11000 node graph similar to the U. of Michigan data set [14], T was less than 180000.

C. Extensions

There are several extensions to the fundamental graph theory presented in this section. While we do not exploit all these extensions in the present paper, we believe they offer potentially powerful tools for topology modeling. Specifically, we have the following capabilities:

- If the underlying graph has weights, representing for ex-

ample more likely or less likely edges (e.g., cheaper, shorter, more compatible), then we can generate a topology of minimum cost. For this we need some additional notions from matching theory (along the lines of [4], [19], [20]). The final algorithms can be implemented efficiently using heuristics along the lines of [35].

- If we consider directed graph models, representing for example important peering relationships ([10], [17] and Section IV), then there is analogous graph theory (based on flows) to generate a directed graph that satisfies simultaneously a given in-degree sequence and out-degree sequence [4].
- If the generated degree sequence is not realizable, it can be shown that a simple adaptation produces a graph whose degree sequence minimizes the l_1 norm from the given degree sequence. We can thus get an optimal efficient approximation.
- The Markov chain simulation method is a special case of Metropolis algorithms that aim to optimize some criterion of interest. Thus Markov chains can be used as "control" or optimization mechanisms to reach a certain target.

III. Evaluation of Degree-Based Generation Methods

In this section we evaluate the properties of the topologies produced by the algorithms developed in the previous section. We must note that topology evaluation remains an open area, with two primary challenges. The first concerns the collection of real data to use as the basis for comparison. The process of collecting data on the real AS-level Internet topology is imperfect. The Route Views data archived by NLANR [9], which many have used as a comparison point, has recently been shown to be missing a significant number of nodes (ASes) and edges (peering relationships). More recent results from U. of Michigan [37][14] are more complete but certainly still miss data. The second challenge is the question of what measures to use in evaluating topologies. A number of measures have been proposed in other work, and we make use of some of these in this section. The second part of this paper proposes additional measures that attempt to capture coarse structural characteristics that we believe are important.

We consider topologies generated with particular structure as well as random topologies. Our evaluation leads to the following conclusions:

- Degree-sequence is not sufficient for producing topologies that are a good match to real data, in the sense that one can produce topologies that differ significantly from one another, despite meeting the same degree sequence.
- On the question of convergence of the simulation to a typical graph based on standard evaluation measures, we provide evidence that about 50,000 steps are sufficient for convergence for a smaller graph of about 7500 nodes and 100,000 steps are sufficient for convergence for a larger graph of about 11300 nodes.
- Randomly generated topologies meeting the degree sequence are a fairly good match to real data for several metrics (e.g. compare to Inet on all the metrics that Inet performs well).

	Michigan	Inet2.2	Dense Core	Sparse Core
diameter	9	13	5	29
avg path len.	3.56	3.86	3.56	5.60
avg eccentricity	6.45	8.72	4.13	17.87

TABLE I
Large topologies (11375 nodes, 32287 edges)

	NLANR	Inet2.2	Dense Core	Sparse Core
diameter	10	13	6	22
avg path len.	3.69	3.79	3.78	4.96
avg eccentricity	6.76	8.62	5.03	13.86

TABLE II
Medium topologies (7657 nodes, 15718edges)

A. Topologies with particular structure

For the same degree sequence, we generate models of three types: sparse core, dense core and preferential, using the basic ideas outlined in the previous section. We have done this using a number of degree sequences, with similar high-level results. Table I shows a set of measures for a topology of 11375 nodes and 32287 edges. We have taken the degree sequence from the more complete Michigan data; we have also run (the most recent version of) Inet with this degree sequence. We show the results for both the sparse core and dense core instances; the preferential instance falls between these two. The measures shown are diameter, average shortest path length and average eccentricity. The eccentricity of node i is the longest distance from i to any other node. We average the eccentricity across all nodes.

We first focus on the sparse and dense core instances. Most striking in these results is the extreme difference in diameter and average eccentricity in the two instances. Consistent with intuition, the sparse core has relatively very large diameter and eccentricity. While not as extreme as the differences in diameter and eccentricity, we also observe significant differences in the average shortest path length, with a factor of about 1.5 from the dense to the sparse core measure.

Table II shows similar results for a smaller topology of 7657 nodes and 15718 edges. In this case the real topology data comes from NLANR. These results are qualitatively similar to the larger topology. From this, it is clear that agreement on degree sequence, alone, does not ensure that topologies are similar.

B. Random topologies

We next examine the convergence properties of the Markov chain algorithm. For this purpose, we use each of the two topologies from the previous subsection as a starting point, then capture snapshots while running the simulation. We use the same two sizes — a large topology and a medium topology. Table III shows a set of measures for the snapshots of a simulation that begins with a sparse core of the

	0	25K	50K	100K	300K	600K
diameter	29	13	11	11	10	10
avgspl	5.60	3.57	3.41	3.36	3.35	3.35
avg-eccentricity	17.87	9.00	7.42	7.35	7.00	6.56

TABLE III
LARGE TOPOLOGY SIMULATIONS, SPARSE INITIAL CORE

	0	25K	50K	100K	300K	600K
diameter	5	10	10	10	10	10
avgspl	3.56	3.23	3.32	3.35	3.35	3.35
avg-eccentricity	4.13	7.14	6.84	6.74	7.08	7.05

TABLE IV
LARGE TOPOLOGY SIMULATIONS, DENSE INITIAL CORE

large size. Each column corresponds to a different snapshot, and the column heading indicates the number of simulation steps when the snapshot was taken. The column labeled "0" corresponds to the initial starting point (and hence repeats part of the data in the two prior tables). Table IV shows the same measures when running the simulation beginning with a dense core. Tables V and VI show the snapshots when starting with sparse and dense cores of the medium size.

There are several observations of interest. First, the greatest difference in the measures occurs from the initial state to the first snapshot taken after 25,000 steps of the simulation. This is most dramatic for the diameter and eccentricity measures, which started out most clearly affected by the dense or sparse core. For example, the diameter of the dense core topology doubles after 25,000 steps, while the diameter of the sparse core topology approximately halves.

Second, we comment on the number of steps needed to reach convergence on these measures. For the larger topology, after 100,000 steps of the simulation, the measures are within 10% of one another, regardless of whether the initial state had a dense core or a sparse core. For the smaller topology, the measures are generally within 10% of one another after 50,000 steps. One exception is the eccentricity measure in the later snapshots for the sparse core.

Third, we note that the average shortest path length exhibits strong stability, with variability of only about 1% after convergence. Diameter and eccentricity exhibit variability of about 10%.

	0	25K	50K	100K	150K	300K	600K
diameter	22	13	12	12	11	11	12
avgspl	4.96	3.51	3.46	3.47	3.45	3.47	3.46
avg-eccentr	13.86	8.29	7.74	7.67	7.50	10.0	10.42

TABLE V
MEDIUM TOPOLOGY SIMULATIONS, SPARSE INITIAL CORE

	0	25K	50K	100K	150K	300K	600K
diameter	6	12	12	11	12	12	12
avgspl	3.78	3.42	3.45	3.46	3.45	3.45	3.47
avg-eccentr	5.03	7.65	8.50	7.38	8.24	7.97	8.31

TABLE VI
MEDIUM TOPOLOGY SIMULATIONS, DENSE INITIAL CORE

IV. AS-GRAPH SEMANTICS: PEERING RELATIONSHIPS

We now turn to the question of *semantics* associated with AS-level topologies. We begin by developing a method to add peering relationship semantics to a topology. This method uses a pruning technique to decompose the topology into layers. In the next section, we consider the issue of higher-level clustering of ASes.

A. Peering relationships

To date, topology modeling has largely concentrated on capturing the connectivity between entities (routers, ASes). This is reasonable in the sense that connectivity is perhaps the most basic characteristic of a topology. On the other hand, Internet routing is significantly determined by peering relationships between autonomous systems (provider, customer, peer, sibling), hence information about these relationships is important in any model used for simulation purposes. In the absence of peering information, many studies simply use shortest paths in the base topology, despite the well-known fact that Internet paths are generally not shortest paths.

Our method for adding peering relationships uses an iterated pruning technique that decomposes the topology. The starting idea of iterated pruning lies in the work of Faloutsos et al. who observed that by pruning all trees, they were left with approximately half of the nodes [8]. Faloutsos et al. called this the "core", and suggested that this decomposition could simplify the generation procedure, since the trees and the core could be generated independently.

A useful way of viewing the Faloutsos decomposition is that they are removing vertices of degree one from the graph, updating the residual degrees each time, until there are no more. This will precisely remove all trees. Let us generalize this as follows. Start with $i = 1$. Repeatedly remove all vertices of degree at most i and update residual degrees. When there are no more vertices of degree less than or equal to i, increase i by one. Define the "level" of each vertex to be the value of i when it was removed. Intuitively, this is a decomposition of the graph into regions of increasing density. For example, if the network contains a clique (completed connected subgraph) of size ten, the pruning procedure will take at least ten iterations.

We propose the following method for adding peering relationship semantics to an undirected topology. The method uses two parameters, k and m. Informally, k represents a lower bound on the number of peering relationships a node has before it is likely to be a provider, while m represents the number of Tier 1 providers (i.e., nodes with no upstream providers). These parameters reflect the intuitive (and experimentally verifiable) notions that most of the peering relationships for small ASes are due to their

providers, while most of the peering relationships for large ASes are due to customers and peers.

The method operates as follows:

Step 1: Perform iterated pruning on the undirected topology. During the first k iterations, each time a node u is pruned and links (u, v) are removed, assign to these links a direction from u to v.

Step 2: Sort the nodes by decreasing total degree. Consider the m highest degree nodes. For each node w in the set of highest degree nodes, assign to all links (w, x) a direction from w to x.

Step 3: If a link between u and v has been assigned a direction from u to v, then define u to be a customer of v and v to be a provider of u. If a link between u and v has been assigned both directions, then define u and v to be peers or siblings (we do not distinguish these).

Step 4: If any links have not been assigned a direction, define a peering relationship using some probabilistic principle. (In our implementation we assigned peer/sibling with high probability (80%), and customer-provider from higher to lower degree with lower probability (20%); of course these probabilities can vary.) The frequency of unassigned links will depend on the parameters k and m used in Steps 1 and 2 above. A high instance of unassigned links is an indication that k and m should be adjusted, by increasing k and/or m.

B. Evaluation and new power laws

We used Gao's script [10] over NLANR data for eight snapshots of the Internet six months apart, between November 1997 and May 2001. As above, if v is a provider of u we introduce a link directed from u to v. If u and v are peers or siblings we introduce two directed links, one from u to v and one from v to u. We thus now have for every node an in-degree and an out-degree. Define further total degree to be the total number of providers, customers, peers and siblings. We report the following new power laws[3].

Let us rank vertices according to their indegree, that is the number of customers and peers/siblings. Then for small ranks, the indegree of the vertex with rank i is proportional to i^{α}, for α between -.98 and -.75, with Pearson correlation coefficient at least 90%; see also Figure 1. In addition, this α is practically identical to the rank versus total degree power law of Faloutsos et al [8]. This expresses the intuitive fact that when we look at nodes of very high degree, representing big providers, almost all of the degree is due to customers and peers/siblings.

The previous power law characterizes big providers. The next power law characterizes small customers. For small i, the number of vertices that have outdegree i, meaning i providers and peers/siblings, is proportional to i^{α}, for α between -2.17 and -2.25, with Pearson correlation coefficient at least 75% (and in all cases, at least as good as [8]); see also Figures 2 and 3. This α is consistently slightly smaller than the corresponding exponent of the frequency

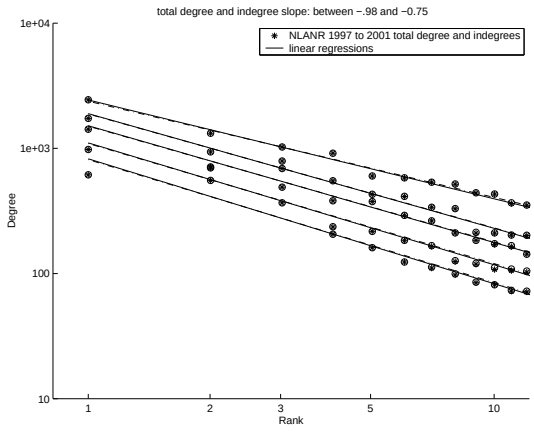

Fig. 1. **The indegree power law**

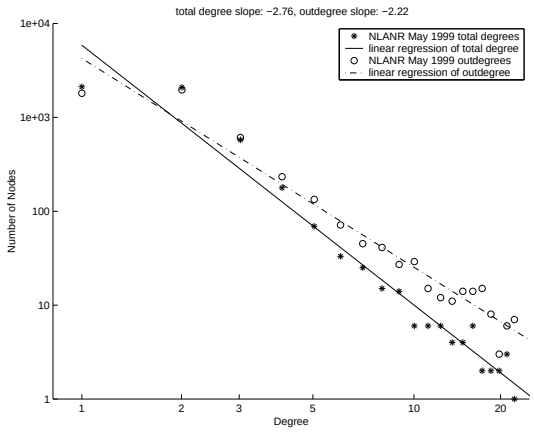

Fig. 2. **The outdegree power law**

versus total degree power law of Faloutsos et al. [8], which for the snapshots that we have used we measured between -2.69 and -2.87[4]. This means that, for low degree vertices, there is strong correlation between total degree and outdegree, expressing the intuitive fact that most of the peering relationships of small ASes is due to their providers.

We applied the algorithm of Section IV.A adding peering relationships to the eight snapshots of NLANR data discussed in Section IV.B. We compared the in-degree and out-degree plots of the generated directed graphs, to the corresponding plots of directed graphs obtained by running Gao's script. The slopes were nearly indistinguishable and the Pearson coefficients comparable. In addition, the peering relationships inferred by the algorithm of Section IV.A agreed, in all cases, with at least 85% of the peering relationships inferred by Gao's script. This can be thought of as a preliminary favorable evaluation of the approach taken by the algorithm of Section IV.A. More work remains to be done in defining evaluation metrics for directed graphs and suitable to assess peering relationships. We leave the definition of such metrics as an important open question.

[3]Below, note that what we call in this paper "total degree" is called in [8] "outdegree", while what we call in this paper "indegree" and "outdegree" is not examined in [8]

[4]To be precise, we took the union over seven consecutive days.

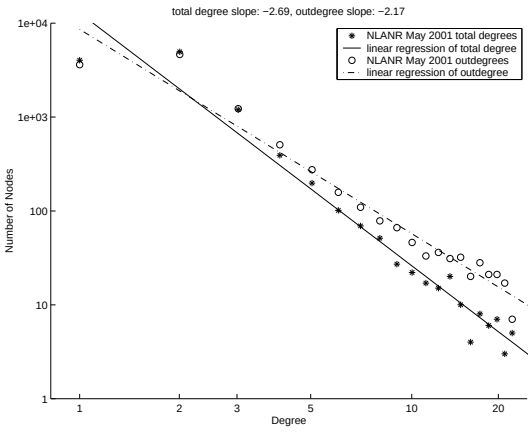

Fig. 3. **The outdegree power law**

C. Remarks

One might envision an alternative method, in the spirit of Inet, and along the extensions discussed in Section II.C. Specifically, since the indegrees and outdegrees of the real data can be measured using [10] or [17], generate two sequences of integers one representing indegrees and one representing outdegrees (extrapolating from the real data like Inet), and interpolate a graph that simultaneously satisfies both (as in Section II.C). The problem with this approach is that even though power laws hold on both ends of total degrees (high degree nodes following rank and low degree nodes following frequency), they only hold on the high end of indegrees (rank) and the low end of outdegrees (frequency). One can verify from total degree summation principles that power laws cannot hold on high and low ends of indegrees and outdegrees simultaneously. Hence it is not obvious how to generate distinct indegree and outdegree sequences separately.

V. AS Graph Semantics: Clustering

In this section we initiate a study of clustering in AS topologies. We use adaptations of spectral filtering techniques as efficient heuristics to identify groups of ASs with good clustering properties.

Firstly, we have identified a variety of clusters in real AS topologies. In addition to passing clustering metrics, these clusters express natural semantics such as geographic proximity, or similarity of interests such as business or research. Thus, the identified clusters are of their own interest, since they capture coarse-grain characteristics. They may be able also valuable for assigning semantics to links and for simulation purposes(intra-group links may be of higher bandwidth or experience higher demand of inter-group links).

Next, we applied the same clustering techniques to synthetic data sets. The clustering properties of topologies produced by all generators driven purely by degree sequence, namely, Inet and its derivatives discussed in Section II, and Brite without setting on the geography parameter, are clearly substantially weaker than those of real data. Indeed, this was expected, since there appears to be no intuitive reason why meeting a particular skewed (or any) degree sequence is correlated with clustering. On the other hand, Brite, with the geography parameter turned on, can produce topologies with stronger clustering properties. We hence conclude that topology generators driven purely by the degree sequence are not sufficient to capture all global properties of real Internet topologies.

We should stress that finding the most suitable adaptation of the generic spectral filtering approach to the specific application is an involved technical problem in its own right. We discuss this in Sections V.C and V.D.

A. Hierarchy and Semantics as Global Semantics

Degree-sequence and peering relationships are *local* properties and express local characteristics of the ASs and their connections. These AS connections (adjacencies with or without peering relationships) can be also thought of as explicit efficient descriptions of these local semantics. What are important *global* characteristics of the AS topology? Following long standing practices of several fields that have studied semantics and scaling in large data sets [21], [29], [2], [18], [11], [26], [27], [3], [25], we claim that the two fundamental global characteristics are *hierarchy* and *clustering*. In the context of Internet topologies, this has been first stated in [39], [5] and it forms the core of GT-ITM, and it is intuitively also clear that hierarchy and clustering are particularly relevant for all modeling and simulation purposes. Note that, given the adjacencies of a graph, hierarchy and clustering are not explicitly described (if fact, they are not even well defined) and, typically, have to be inferred in non-trivial ways.

Hierarchy for Internet topologies at the AS level has been studied in [38], [17], [36]. [38] obtain a hierarchy of several levels where, roughly speaking, the "level" or "size" or "class" of an AS is one step below the level of the "smallest" provider of this AS. Using more detailed methods, data, measurements and validation arguments, [17] obtain a hierarchy of dense core (level 1), transit core (level 2), outer core (level 3), small regional ISPs (level 4) and customers (level 5). Most importantly and relevant to our work, [28], [36] introduce several new metrics aiming to capture, among other, hierarchical properties, and conclude that *topology generators meeting the local characteristic of degree sequence also yield good results with respect to the global characteristic of hierarchy.*

Here we initiate the study of the other fundamental global property, namely clustering. First of all, let us note that hierarchy and clustering need not be correlated (they follow mostly independent primitives). For example, a cluster driven from geography may contain ASes of different "size" (as we shall see, this is indeed the case), and a hierarchical class may contain parts of several different clusters (there is also some natural correlation, for example, as expected, the dense core is also a cluster). In addition, while hierarchy is a strict partitioning of the ASs, in the sense that each AS belongs to exactly one level of the hierarchy, clustering is *not* a partitioning. Not every AS needs to belong to a good cluster, and perhaps some ASs belong to

more than one clusters (for example, a very big European ISP provider may belong to a cluster expressing the very big providers around the globe, and to cluster expressing Europe).

B. Preliminaries on Clustering and Spectral Filtering

How shall we define clusters, and how shall we find them? To define and quantify clustering properties, we draw on graph theoretic metrics for *relative density* [6]. Roughly speaking, for a graph $G(V, E)$ and for a set of nodes $S \subset V$, the relative density of S is the ratio of links that are entirely inside S, namely both endpoints belong to S, divided by the total number of links incident to S, namely links inside S plus "crossing" links with one endpoint in S and the other in $V \setminus S$:

$$\frac{|\{\{v,u\} \in E \ : \ v \in S \ \text{ and } \ u \in V \setminus S\}|}{|\{\{v,u\} \in E \ : \ v \in S \ \text{ or } \ u \in S\}|}$$

Intuitively, S is a good cluster if the relative density of S is *large*[5].

To efficiently isolate groups of ASs with good clustering properties we draw on spectral filtering heuristics from the field of Information Retrieval and data-mining. These are fairly sophisticated technical methods. They are also strikingly strong and robust, in the sense that they have been successful in a wide range of applications, such as bibliometrics, digital libraries, data-mining in massive datasets, the Genome project, and inferring authorities and communities in the WWW [21], [29], [2], [18], [11], [26], [27], [3], [25].

Spectral filtering is spectral analysis of graphs. The "spectrum" of a graph is a listing of the eigenvalues of (a suitable modification of) its adjacency matrix in non increasing order. Look at Figure 4. For a graph consisting of two (dense) connected components A and B, we will get a spectrum whose first two largest eigenvalues are $\lambda_1 = \lambda_2 = 1$. The corresponding eigenvectors assign weights $+x$ on A and 0 on B, and $+y$ on B and 0 on A respectively. Each connected component can be thought of as an extreme case of a cluster. The two largest eigenvalues and the corresponding eigenvectors, precisely indicate these two clusters. Look at Figure 4. For a graph consisting of two connected components connected with a "small" number of edges, we observe a slight shift of the previous values. In this case the second largest eigenvalue is $1 - \epsilon$ (for some positive ϵ very close to 0) and, for x' and y' distinct and positive, the corresponding eigenvector assigns values approximately equal to x' to all vertices in A and approximately equal to $-y'$ to all vertices in B. The second eigenvector again indicates the clusters A and B; let us call such a vector $e_{A,B}$ Similar remarks hold also for the clusters C and D. However, the most interesting case is when we consider the union of the clusters A, B, C, and D

[5]For several technical and aesthetic reasons beyond the current scope, graph theorists mostly use metrics related to the inverse of relative density for which they use terms like "expansion", "magnification", "conductance", or "isoperimetries" [6].

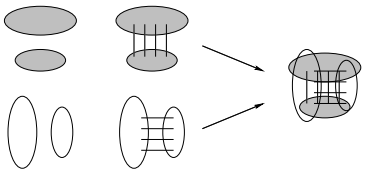

Fig. 4.

over the same set of vertices. We would now get the second and third eigenvalues $\lambda_2 = 1 - \epsilon'$ and $\lambda_3 = 1 - \epsilon''$ with corresponding eigenvectors very close to $e_{A,B}$ and $e_{C,D}$, still indicating the clusters A, B, C, and D! In general, when the graph contains many overlapping clusters, this is a general purpose heuristic for isolating the strongest such clusters. It isolates "latent", "hidden", in our case global and not explicitly expressed clustering semantics of the graph (hence also the name Latent Semantic Indexing).

Of course, this general purpose method requires several adjustments to each specific application. In some sense, despite the fact that Faloutsos et al had given a first spectral snapshot of the AS-level topology by their eigenvalue power-law [8], they did not perform any adjustments to the raw data and thus were not able to apply the general method.

C. Adaptation of Spectral Filtering to AS Topologies

In this section we will describe the process of constructing an appropriate matrix for the AS topology on which we will perform spectral filtering to identify clusters.

Let A be the adjacency matrix of a graph. If all edges are bidirectional, then A will be symmetric ($A = A^T$). Symmetric matrices have the property that all eigenvalues and eigenvectors are real. This makes the analysis of the results possible. Assume that e_i and λ_i are the i-th eigenvector and eigenvalue. The maximum value of λ_i can be as high as the maximum degree in the graph [19]. The intuition behind spectral analysis is that the best clusters will be identified by the eigenvectors for which their corresponding eigenvalue is large enough. Also, a graph has better clusters when there are many large eigenvalues.

We will not use matrix A directly, since working with arbitrarily large λ_i's is not suitable. (In fact, [1] explain that for the AS topology, the first eigenvalues express the "stars" of small clients connected to the largest ISPs and even the eigenvalue power law measured in [8] is a re-statement of the rank-degree power law with half the slope.) Instead we define the stochastic matrix A_{RW}. For each entry (i, j) of the matrix we have: $A_{RW}[i, j] = A[i,]/d_i$, where d_i is the degree of node i. For technical reasons beyond the scope of this article, spectral analysis will be performed on the following variation of A_{RW}: $A'_{RW} = \frac{1}{2}(I + A_{RW})$. The eigenvalues of this matrix are in $[0, 1]$, the largest being 1, or in other words: $1 = \lambda_0 \geq \lambda_1 \geq \lambda_2 \geq \ldots \geq 0$.

The AS topology matrix is undirected if we do not take into account the relationships between the ASes. If, on the other hand, we label the edges as customer-provider

and peer-to-peer, as was explained in Section IV, then the resulting graph is directed. An edge from i to j exists if and only if i is a customer of j, or i and j are peers. We used the data provided in [17] to create the AS-topology graph and label the edges. This graph is not symmetric. Thus, we define $A^* = A \cdot A^T$. The interesting property of this matrix is that each entry (i, j) is equal to the number of ASes that are children of both AS i and j, or peers of both of them (plus 1 if there is an edge between i and j). In doing so, we put more weight on pairs of ASes that have many customers and peers in common and in this way are related to each other stronger. For example, if there are many customers that peer with two ASes, then these two ASes have some common properties (like located in the same area, or targeting the same group of customers) and we are able to identify them even if they do not directly peer with each other. This technique is also known as co-reference or bibliometric coupling, and was used by Kleinberg in his seminal work of computing semantics on the WWW [18].

A primary additional modification that we have made was to not use all the ASes to construct the matrix A. We have pruned the ASes that were assigned to level 5 of the data [17] and which correspond to small customers. In this way we are examining only the approximately 2200 more important ASes. This is a crucial modification. It avoids the complications created by the stars which exist in the topology when many small customers use the same ISP. Theoretically, this has been explained in [1]. It turns out that in the heuristic method of spectral filtering, stars and small ISPs overwhelm all other clusters and we have to avoid them.

Finally, let us stress that all spectral filtering methods are *heuristics*. Thus, give *approximate* results and require additional processing at each step [24] [33].

D. Results

We used the techniques discussed above to investigate the existence of clusters in the AS topology and to identify them.

We will give three concrete examples of the clusters that were identified using this technique. These can be found in Tables VII ,VIII, and IX. In these tables, we list some of the ASes that belong to each cluster. The first of them is a cluster of big ISPs in the US. This cluster corresponds to the eight larger eigenvalue (there are technical but well-understood reasons why most important cluster corresponds only to the eighth eigenvalue; roughly speaking, the very large number of non-trivial customers of these ISPs decrease its relative density metric). The cluster was created because big providers tend to peer with each other Similar results have been already reported in ([38], [17]). This confirms our method which correctly identifies the dense core as a good cluster. However our methods are generic and can identify clusters that are difficult to infer with other techniques.

A sample of the nodes in the second cluster can be found in Table VIII. In this case the cluster is the result of an effort to connect national research and educational networks in Europe and elsewhere. This cluster has different characteristics from the previous one, since the ASes do not tend to peer a lot with each other. Instead they connect to backbone networks like Ten-155 and Abilene and use them as ISPs. Most of the ASes connect to one of them or both, and there are examples like SingaREN that connects to SURFnet.

Another example of a cluster we identified using the eigenvector that corresponds to the fourth larger eigenvalue is shown in Table IX. As we can see, it is composed of ASes from the eastern part of Europe (mostly Ukraine and Bulgaria). The interesting thing that links most of them is that they receive service from the Satellite Media Services ISP, which has its headquarters in UK.

By examining more eigenvectors, it was possible to discover more clusters, like national research and educational networks, clusters of providers in different continents and countries and so on. Also, by altering the basic method by which we construct the matrix (for example by taking into account all ASes), we were able to identify clusters with different characteristics. As a concrete example the eigenvector which corresponds to the largest eigenvalue (excluding the trivial one) for the full graph except the nodes that have degree 1 gave around 150 ASes from South Korea. It is an area of future research to understand what are the different kinds of clusters and which method is more suitable in identifying them.

In the discussion above, it was not very clear how we picked the clusters and selected the results we have presented. Our approach was to examine the ASes that have the highest negative and positive weights in the eigenvector. We assumed that the first n of them were part of the cluster. The size of n, which is the size of the cluster, is also technical to find. Roughly speaking, we order the ASes according to the weights assigned by the eigenvectors, and look for points where there is a big drop in the size of the weight. This is only one way of identifying candidate groups. Having more criteria is an important open problem. We believe that several more criteria exist.

E. Clusters and topology generators

In the previous paragraphs, we observed that there are clusters in the AS topology and that spectral graph analysis can be used to identify them. Since clusters are an integral part of the Internet, the next question in mind is whether topology generators generate graphs with clusters and how close they are in capturing this property of the graph.

To measure the quality of the clusters, we computed the first 30 eigenvalues of two random topologies generated with Brite and three with Inet and compared them to the real data. The topologies generated where for 11000 nodes. From the generated graphs, we isolated the 2200 ones with the highest degree. We used this heuristic to isolate the core of the network. The eigenvalues were computed for the induced subgraph. The reason that these many nodes were picked was to be able to compare the results with the ones presented in the previous section for the real data. However, we got similar results when we used the Algo-

AS number	Description	Weight	level
2828	XO Communications Inc.	-1.3089e-02	1
7018	AT&T	-1.3085e-02	1
1239	SprintLink	-1.2687e-02	1
2548	Digex	-1.2571e-02	1
3967	Exodus Communications	-1.2382e-02	1
1	GTE Internetworking	-1.2282e-02	1
701	Alternet	-1.2204e-02	1
2685	AT&T Americas	-1.2024e-02	2
2914	Verio	-1.1851e-02	1
209	Qwest	-1.1773e-02	1
293	ESnet	-1.1642e-02	2
3549	Global crossing	-1.1465e-02	1
3561	Cable & Wireless	-1.1369e-02	1
4006	NetRail, Inc.	-1.1209e-02	1
	Four entries missing		3, 4, 2, 2
174	Performance Systems International	-1.0862e-02	1
4200	Telia Internet	-1.0801e-02	1
	One entry is missing		4
1833	TeliaNet USA	-1.0779e-02	1
	Three entries missing		3, 3, 4
3356	Level 3 Communications	-1.0645e-02	1
6453	Teleglobe Canada Inc	-1.0582e-02	1

TABLE VII

Sample of a cluster that shows big providers in the North America

rithm of Section IV.A to add peering relationships on the Inet output and pruned ASes that had no customers and either (a) a single provider or (b) at most two providers. We obtained the same qualitative results. As it can be seen in Fig.5, the degree-driven topology generator Inet is not good in creating clusters.

A natural question to ask metrics of "goodness" of clustering. (Note that the "clustering coefficient" of [36] is not adequate, in the sense that Inet and real data have similar clustering coefficients, but substantially different spectra of the type of Table 5; it is generally accepted in graph theory that spectral images are strong indicators of clustering [6]). From the first 20 eigenvectors computed from the real topology, we measured the value of the relative density for the groups composed of the first 1, 2, ..., 200, nodes. The choice of 200 is ad-hoc. For each cluster size, we computed the mean relative density among all groups. We have also computed the 10th and 90th percentiles. The same process was repeated for the first 20 eigenvectors produced by Inet. Finally, we created 20 random sets of 200 nodes each and calculated the same metrics for cluster size from 1 to 200. The results are depicted in 6. In that graph higher values are better. As we can readily observe the clusters produced by Inet are better than the random ones, but not nearly as good as the ones found in the real topology. This gives more evidence that topologies generated by Inet and other degree-sequence generators cannot generate topologies with good clusters.

AS number	Description	Weight	level
	Four entries missing (part of RENATER)		4, 4, 4, 4
11537	Abilene	2.4370e-01	2
	Four entries missing (part of RENATER)		4, 4, 4, 4
8933	Ten-155 (Europe's Research and Education backbone network)	2.1987e-01	2
293	Energy Sciences Network (in US)	2.0666e-01	2
2200	Renater 2 (in France)	2.0470e-01	2
10764	Science, Technology and Research Transit Access Point	2.8422e-02	3
2603	NORDUnet (Nordic Internet highway)	1.3707e-02	2
1103	SURFnet (in Netherlands)	1.3384e-02	3
6509	NTN BELL2 MBONE Service	1.3348e-02	3
378	ILAN (Tel Aviv University, Israel)	1.2593e-02	3
680	DFN-IP (in Germany)	1.2444e-02	2
5408	Greek Research and Academic Network	1.0021e-02	4
513	CERN	9.6932e-03	3
786	JANET IP Service (in UK)	9.3910e-03	3
7610	SingaREN (in Singapore)	9.0508e-03	3
3343	RUNNet (in Russia)	9.0268e-03	4
766	RedIRIS (in Spane)	8.5164e-03	4
1930	Portuguese Academic and Research Network	8.4781e-03	4
1955	HungarNet (in Hungary)	8.4354e-03	3
559	SWITCH, Swiss Academic and Research Network	8.3855e-03	4

TABLE VIII

Sample of a cluster that shows Research and Education Networks mostly in Europe,but also elsewhere

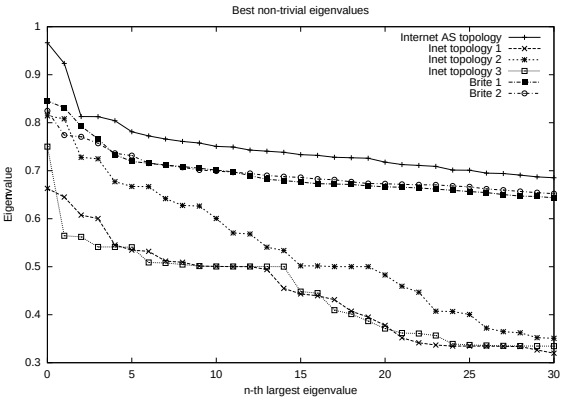

Fig. 5. **The largest 30 eigenvalues for the real AS topology and for topologies generated by Inet and Brite.**

AS number	Description	Weight	level
13228	DG ISP, Ukraine	-3.2116e-01	3
3252	Relcom-Ukraine ISP	-3.0050e-01	3
5415	Global Ukraine Ltd.	-2.4365e-01	3
12963	UA.LDC, Ukraine	-2.2718e-01	3
12369	UKRSAT, Ukraine	-2.0871e-01	4
12294	Technological Systems CJVC, Ukraine	-2.0750e-01	4
8343	DOnbass Regional Information System, Ukraine	-1.9464e-01	4
6846	UKRPACK.NET, Ukraine	-1.8437e-01	3
8717	Spectrum Net, Bulgaria	-1.8017e-01	4
9184	NetPlus, Bulgaria	-1.7580e-01	3
9154	Internet Bulgaria Ltd.	-1.7149e-01	4
9127	NETISSAT, Bulgaria	-1.7149e-01	4
12304	Yerevan Physics Initute, Armenia	-1.6856e-01	3
6886	INTS, Ukraine	-1.6246e-01	3
9000	ESER, Turkey	-1.5361e-01	3
12358	Minsk, Belarus	-1.4905e-01	4
8788	Adamant ISP, Ukraine	-1.4440e-01	4
6702	Apex NCC, Ukraine	-1.3972e-01	4
13249	PFTS, Ukraine	-1.3972e-01	4
6876	TeNeT, Ukraine	-1.3707e-01	3
13126	Satellite Media Service, UK	-1.3234e-01	3

TABLE IX
Sample of a cluster of networks in Eastern Europe

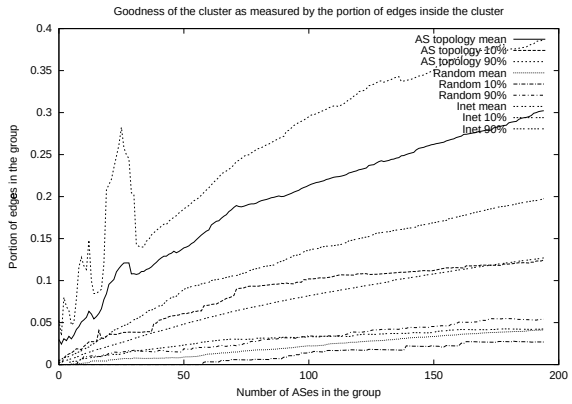

Fig. 6. **Relative density for groups found in the AS-topology, for groups generated by Inet and random groups.**

VI. Related Work

n the area of topology generators, we have already mentioned the Inet degree-based generator and the Brite evolutionary generator. There are a number of further examples in the evolutionary category, though these do not specifically aim to model AS-level topology. Further, the evolutionary models generally do not agree as closely with real data as Inet. We are unaware of any work beyond Inet that *directly* uses the degree sequence, perhaps because Inet has generally done well for a set of metrics.

Some of the early router-level topology generators do incorporate some notion of semantics. For example, one form of Waxman graphs use the Euclidean distance between two nodes to affect the probability of an edge. One could use the Euclidean distance as a weight to reflect, for example, link delay. The transit-stub model in the GT-ITM suite assigns routing policy weights to links so that shortest paths constructed using the policy weights have certain properties. For example, the path between two nodes in a domain will remain in the domain; the path between two nodes in different domains will traverse a transit domain unless there is a stub-stub edge that gives a shorter path. Transit-stub graphs also contain a form of higher-level semantics by grouping routers into specific domains.

Several projects have aimed to extract semantic information from real datasets. We have already mentioned [10][17] who associate peering relationships with edges in the AS-level topology, and the work in [10][17] who aim to characterize hierarchy. Chang et al. describe a method for inferring AS-level topology from router-level path traces [15]. Such a technique has the potential to allow one to bring semantic information from the router-level data "up" to the AS-level topology (e.g., estimates of the number of routers in an AS), however the Chang work primarily focuses on the important task of identifying border routers and not on issues of semantics.

Also closely related to our work are attempts to understand which evaluation metrics are most important in assessing the quality of topologies. This is a difficult problem, since the answer may be influenced by the particular use of the synthetic topology. Perhaps the earliest work in this area is the work by Zegura et al. to evaluate several topology generators using a variety of measures including metrics related to multicast routing [40]. More recently, several papers propose the use of "large-scale" or "overall" properties, based on the intuition that these properties are more important than local properties. For example,metrics such as expansion, resilience and distortion have been examined. We agree with the intuition that large-scale properties are important, particularly if they can reflect semantic structure in network topology (rather than more abstract notions derived from basic graph theory).

We mention one other important area in topology modeling. Significant progress has been made recently in mapping both the router-level topology of the Internet as well as the AS-level topology. Improvements in understanding of the real topology obviously lead to improved models.

VII. Summary

We have provided evidence that clusters exist in the AS topology and described a general method for finding them. We have compared the clustering properties of Inet to the ones found in the real data and concluded substantial differences. So, extra work is needed to create topology generators that effectively capture both the degree-sequence found in the real network and the existence of clusters.

As a side issue, we want to investigate the different types of clusters that exist in real networks and try to incorporate them in topology generators. In order to do so, more work is needed in quantifying what is a good cluster, how to measure it, and how to systematically extract clusters from the eigenvectors.

In addition, we found that degree-sequence alone (even for very skewed degree sequences like the ones of AS-level Internet topologies) can result in a variety of topologies with strikingly different characteristics. On the flip side, such extreme cases can be used as stress test in simulation.

We have indicated foundations in graph theory that can make future design of topology generators substantially richer and more formal.

References

[1] Milena Mihail Christos H. Papadimitriou Alex Fabrikant, Elias Koutsoupias. *DIMACS Workshop on Internet and WWW Measurement, Mapping and Modeling*, February 13-15, http://dimacs.rutgers.edu/Workshops/Internet/program.html 2002.

[2] Yossi Azar, Amos Fiat, Anna R. Karlin, Frank McSherry, and Jared Saia. Spectral analysis of data. In *ACM Symposium on Theory of Computing*, pages 619–626, 2001.

[3] Per Bak. *How nature works: The science of self-organized criticality*. Springer-Verlag, New York, 1996.

[4] Claude Berge. Graphs and hypergraphs. *North Holland Publishing Company*, 1973.

[5] K. Calvert, M. Doar, and E. Zegura. Modeling internet topology. *IEEE Communications Magazine*, June 1997.

[6] Fan R.K. Chung. Spectral graph theory. *American Mathematical Society Book Series*, 1997.

[7] P. Erdos and T. Gallai. Graphs with prescribed degrees of vertices. *Mat. Lapok, 11*, 1960.

[8] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *Proc. of ACM SIGCOMM*, 1999.

[9] National Laboratory for Applied Retwork Research. Route views archive. *http://moat.nlanr.net/Routing/rawdata*.

[10] L. Gao. On inferring autonomous system relationships in the internet. *IEEE Glogal Internet*, 2000.

[11] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring web communities from link topology. *UK Conference on Hypertext*, 1998.

[12] S.L. Hakimi. On the realizability of a set of integers as degrees of the vertices of a graph. *SIAM J. Appl. Math. 10*, 1962.

[13] V. Havel. A remark on the existence of finite graphs. *Caposis Pest. Mat. 80*, 1955.

[14] Qian Chen Hyunseok. The origin of power laws in internet topologies revisited.

[15] Hyunseok Chang Sugih Jamin. Inferring as-level internet topology from router-level path traces.

[16] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. *CSE-TR-433-00, U. Michigan, Ann Arbor*, 2000.

[17] L. Subramanian S. Agarwal J. Rexford R.H. Katz. Characterizing the internet hierarchy from multiple vantage points. *U.C. Berkeley Tech. Report, August 2001*, 2001 (Infocom 2002, to appear).

[18] J. Kleinberg. Hubs, authorities, and communities on the www. *ACM Computing Surveys, 31(4es):5*, 1999.

[19] L. Lovasz. Combinatorial problems and exercises. *North Holland, Amsterdam*, 1979.

[20] L. Lovasz and M. Plummer. Matching theory. *North-Holland Mathematics Studies, No 121*, 1986.

[21] G. O'Brien M. Berry, S. Dumais. Using linear algebra for intelligent information retrieval. *SIAM review, 37(4)*, 1995.

[22] A. Medina, I. Matta, and J. Byers. On the origin of power-laws in internet topologies. *ACM Computer Communications Review*, April 2000.

[23] R. Motwani and P. Raghavan. Randomized algorithms. *Cambridge University Press*, 1995.

[24] R. Motwani and P. Raghavan. Randomized algorithms. *Cambridge University Press*, 1995.

[25] Patrick O. Brown Orly Alter and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA, Vol. 97, Issue 18*, 2000.

[26] L. Page. Pagerank: Bringing order to the web. *http://hci.stanford.edu/ page/papers/pagerank*, 1998.

[27] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *PODS 98*, 1998.

[28] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On characterizing network topologies and analyzing their impact on protocol design. *USC-CS-TR-00-731, http://citeseer.nj.nec.com/radoslavov00characterizing.html*, March 2000.

[29] P. Raghavan. Information retrieval algorithms: A survey. *Proc. 8th SIAM Symposium on Discrete Algorithms (SODA)*, 1997.

[30] A. Sinclair. Algorithms for random generation and counting: A markov chain approach. *Springer-Verlag*, 1997.

[31] R. Taylor. Constrained switchings in graphs. *SIAM J. ALG. DISC. METH., Vol. 3, No 1*, 1982.

[32] Traceroute.org. Public route server and looking glass site list. *http://www.traceroute.org*.

[33] V.V. Vazirani. Approximation algorithms. *Springer-Verlag*, 2001.

[34] B. M. Waxman. Routing of multipoint connections. *IEEE Jour. Selected Areas in Communications (Special Issue: Broadband Packet Communications)*, 6(9):1617–1622, 1988.

[35] D. Williamson and M. Goemans. Computational experience with an approximation algorithm on large-scale euclidean matching instances. *ORSA J. Computing, 8*, 1996.

[36] H. Tangmunarunkit R. Govindan S. Jamin S. Shenker W. Willinger. Network topologies, powerlaws and hierarchy. *U.S.C. Tech. Report http://www.isi.edu/ hongsuda/resume.html*, May 2001.

[37] Q. Chen H. Chang R. Govindan S. Jamin S. Shenker W. Willinger. The origin of power laws in internet topologies revisited. *Infocom 2002, to appear (also in http://topology.eecs.umich.edu)*.

[38] S. Jaiwal L.Gao Z. Ge, D.R. Figueiredo. On the hierarchical structure of the logical internet graph. *ITCOM 2001*.

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

[40] E. Zegura, K. Calvert, and M. Donahoo. A quantitative comparison of graph-based models for Internet topology. *To appear in IEEE/ACM Transactions on Networking*, 1997.