Modeling Autonomous–System Relationships

Xenofontas Dimitropoulos
Georgia Tech
fontas@ece.gatech.edu

George Riley
Georgia Tech
riley@ece.gatech.edu

Abstract

*The development of realistic topology generators that produce faithful replicas of Internet topologies is critical for conducting realistic simulation studies of Internet protocols. Despite the volume of research in this area the last several years, current topology generators fail to capture an inherent aspect of the autonomous–system (*AS*) topology of the Internet, namely the fact that* AS *links reflect business agreements between competing entities, which impose restrictions on how traffic is routed between* ASs*. These restrictions result in inflated* AS *paths and generally in suboptimal routing in the Internet. In this work, we first evaluate the importance of modeling* AS *relationships when conducting accurate and realistic simulation studies. We demonstrate that ignoring* AS *relationships produces different simulation results than modeling AS relationships based on known relationships between Internet* Internet Service Providers *(*ISPs*). Then, we introduce a framework for generating synthetic* AS *topologies annotated with realistic relationships. In addition to modeling the degree distribution of a network, which is the property that most existing topology generators model, our framework also models new properties that capture the characteristics of* AS *relationships. Finally, we propose a novel algorithm for generating synthetic graphs, annotated with* AS *relationships, that reproduce these* AS *relationships-aware properties.*

1 Introduction

In recent years several efforts have focused on developing topology models and topology generators that produce synthetic topologies with characteristics that accurately reflect properties of real Internet networks. Accurate topologies are essential for performing realistic simulations of new protocols, routing, and architectures. They are especially important in areas such as multicasting, routing and overlay networks, where protocol performance is strongly coupled with the structure of the underlying topology. For example, in the case of multicasting, the study in [18] demonstrated that the performance of a well-studied protocol changed drastically after deploying more accurate topology models. For this reason, accurate and realistic topology generators are of paramount importance in conducting reliable performance evaluation experiments.

Here, we take at a completely new approach to topology generation, which is based on the idea of modeling different node relationships. Node relationships are an inherent aspect of many real networks. Links of *AS* topologies represent different types of business relationships, like customer-to-provider (c2p) peer-to-peer (p2p) and sibling-to-sibling (s2s) relationships [10]. Links in social networks represent different types of social relationships while links in protein networks represent different types of protein interactions. However, current network topology generators overlook the diversity of node relationships by modeling networks as abstract undirected graphs. Such graphs identify all the links of a network as equivalent, missing the different types of node relationships. Yet, knowing the types of relationships between network nodes and having realistic models of these relationships is very important for several applications.

The main application that motivates this work is determining routing *AS* paths in synthetic *AS* topologies. Routing paths between *ASs* are determined by *AS* relationships. These relationships result in the valley-free routing model which states that every *AS* path has a hierarchical structure [10]. Given an *AS* topology annotated with inferred *AS* relationships, we can compute the policy-compliant *AS* path between any two *ASs* using a modified version of Dijkstra's algorithm [14]. On the other hand, without knowing *AS* relationships we are forced to assume shortest path routing which leads to unrealistic results. It is well known that actual *AS* paths in the Internet are substantially longer [22, 21, 11, 20] than the shortest path. Unfortunately, the shortest path routing assumption is made by default in most simulation studies without further investigation. The reason is that all existing topology generators do not model *AS* relationships, which makes it impossible to simulate path inflation effects.

The second reason for which modeling of node relationships is important is that it enables us to produce more accurate synthetic topologies. Different types of links are likely to exhibit different topological properties. For example, borrowing the terminology of [13], c2p links are more *radial*, in that they connect small degree to large degree *ASs*. However, p2p links are more *tangential* in that they connect *ASs* of similar degree. To capture this diversity of properties, it is necessary to build a topology generator that takes into account the existence of different types of links that may have different topological characteristics. Then, we can effectively model a wider range of topological properties than can currently available generators.

In this work, we first focus on the importance of modeling *AS* relationships in conducting accurate and realistic network simulations. We identify and discuss the following three shortcoming of ignoring *AS* relationships: 1) *AS* paths are substantially shorter than in reality, 2) the traffic load on *AS* links and on individual *ASs* is substantially lower than in reality, and 3) the number of alternative *AS* paths available to an *AS* is substantially larger than in reality. We use simulation experiments to demonstrate these shortcoming and to show how they can effect commonly used performance evaluation metrics.

Next, we introduce a framework for modeling *AS* relationships and for generating realistic *AS* topologies annotated with realistic *AS* relationships. We start by identifying topological properties that capture important *AS* relationships characteristics. Then, we use statistical tools to model these properties in real *AS* topologies annotated with inferred *AS* relationships. Finally, we introduce an algorithm for reproducing these properties in synthetic *AS* topologies.

In the next section we briefly review related work in the area of topology modeling and topology generation. Then, in section 3 we discuss and demonstrate shortcomings of ignoring *AS* relationships in conducting realistic network simulations. In section 4 we introduce the topological properties that enable us to model *AS* relationships. In section 5 we outline our framework for modeling these properties and for generating synthetic *AS* graphs. Finally, in section 6 we conclude our paper and discuss future research directions.

2 Related Work

A large number of published works have focused on modeling Internet topologies and on developing realistic topology generators. The first topology generator that became widely known was introduced by Waxman [23]. Waxman generator is a variation of the classical Erdos-Renyi random graphs. Later, after it became evident that networks do not have a random structure, new generators like GT-ITM [25] and Tiers [9] emphasized the hierarchical structure of networks. Consequently, these topology generators were characterized as *structural*. In 1999, Faloutsos et al. discovered that the degree distributions of router- and AS-level topologies of the Internet follow a power-law. Structural generators failed to reproduce this power-law, which triggered a number of new topology generators that tried to achieve this goal. These newer topology generators can be classified into causality-aware and causality-oblivious. The first class includes the Barabasi-Albert (BA) [2] preferential attachment model and the model by Chang *et al*. [4] based on the idea of *highly optimized tolerance* [3]. These models grow a network by incrementally adding nodes and links into a graph based on some evolution process so that the resulting graph follows a power-law degree distribution. In the same family belongs the BRITE [15] topology generator, which employs the BA model to generate synthetic Internet topologies. On the other hand, causality-oblivious generators like PLRG [1], Inet [24] and the model by Gkantsidis *et al*. [12] try to match the power-law degree distribution of the Internet without accounting for different rules that might drive the evolution of the topology.

3 *AS* relationships on simulations

AS relationships reflect business agreements between *ASs* and can be classified in three categories. In the c2p category, a customer *AS* pays a provider *AS* for transiting traffic from the customer and also for delivering traffic to the customer. In the p2p category, two *ASs* exchange traffic between their customers but do not exchange traffic from or to their providers or peers. Two sibling *ASs* exchange traffic between their providers, customers, peers or other siblings. Sibling *ASs* usually belong to the same organization or to strongly affiliated organizations. For example, the relationship between the European and North American divisions of a global ISP would be s2s. To honor these agreements network administrators configure export policies on BGP routers according to the following rules:

- Exporting to a provider: When exporting routes to a provider, an *AS* advertises routes received from customer *ASs* and local routes. It does not advertise routes received from peer and provider *ASs*.
- Exporting to a customer: When exporting routes to a customer, an *AS* advertises all its routes, i.e., local routes and routes received from customer, provider, peer and sibling *ASs*.
- Exporting to a peer: When exporting routes to a peer, an *AS* advertises routes received from customer *ASs* and local routes. It does not advertise routes received from peer and provider *ASs*.
- Exporting to a sibling: When exporting routes to a sibling, an *AS* advertises all its routes, i.e., local routes and routes received from customer, provider, peer and sibling *ASs*.

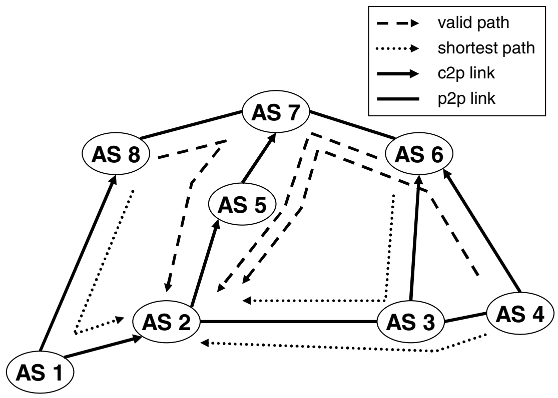

Figure 1. Example AS topology annotated with AS relationships. The topology is extracted from a real AS topology and the AS relationships are inferred using the heuristics in [7]. The dotted lines represent shortest paths between ASs 4, 6 and 8 to AS 2. The dashed lines represent policy compliant paths from the same sources to the same destination.

If all *ASs* strictly adhere to these export policies, then every *AS* path must comply with the following hierarchical pattern: an uphill segment of zero or more c2p or s2s links, followed by zero or one p2p link, followed by a downhill segment of zero or more provider-to-customer (p2c) or s2s links. The paths that follow this hierarchical structure are called *valley-free* [10] or *valid*.

In addition to export policies, network administrators also configure route selection policies. The most widely used route selection policy is that *ASs* prefer customer routes over routes through peers or providers. This is because *ASs* do not have to pay for sending traffic to a customer and also because they tend to avoid congestion at peering exchange points. This route selection policy is referred as *prefer-customer* routing [10].

Routing policies reflect business agreements and economic incentives, and for this reason they are deemed more important than quality of service criteria and thus they take precedence in the route selection process. Consequently, suboptimal routing and inflated *AS* paths often occur. The study by Gao and Wang [11] used BGP data to measure the extent of *AS* path inflation due to valley-free and prefer-customer routing in the Internet. They found that at least 45% of the *AS* paths observed in BGP data are inflated by at least one *AS* hop and that *AS* paths can be inflated by as long as 9 *AS* hops.

Taking into account such inflation effects is important for conducting meaningful and realistic simulation studies. Consider, for example the *AS* topology in Figure 1 that we extracted from a real topology[1]. Directed links represent c2p relationships that point towards the provider and undirected links represent p2p relationships. If we ignore *AS* relationships then the shortest paths from *ASs* 4, 6 and 8 to *AS* 2 are shown with dotted lines. On the other hand, if we account for *AS* relationships these paths are no longer valid. In particular, the path 4→3→2 transverses two p2p links; the path 6→3→2 transverses a p2c link followed by a p2p link; and the path 8→1→2 transverses a c2p link after having gone through a p2c link. All these paths violate the hierarchical structure of the valley-free model and thus are not used in practice. The paths actually used are the policy compliant paths marked with dashed lines.

The first effect of taking *AS* relationships into account is that paths become longer than the corresponding shortest paths. From a performance perspective, longer paths can affect metrics such as end-to-end (e2e) delay, server response time, jitter, convergence time and others. To demonstrate this we simulated the topology in Figure 1 using BGP++ [8, 5]. BGP++ is a BGP simulation module based on Zebra routing software. We use a single router for each *AS* and configured appropriate export rules between *ASs* according the guidelines discussed above. We set the delay of each link to 10 milliseconds and the bandwidth to 400kbps. Then, we configured exponential on/off sources at *ASs* 4, 6 and 8 that send traffic to *AS* 2 at a rate of 500kbps. We run the simulation for 120 seconds; for the first 100 seconds we wait for routers to converge[2] and at the 100th second we start the traffic sources. We first measure the e2e delay between the sources and the destination under the following two configuration scenarios: 1) *AS* relationships disabled, and 2) AS relationships enabled.

In Figure 2 we depict the cumulative distribution function (CDF) of the e2e delays for the two scenarios. First, notice that the CDF corresponding to simulating *AS* relationships is skewed to the right, which means that there is a significant increase in the e2e delay. In particular, the average e2e delay with *AS* relationships enabled is 0.853 seconds whereas without *AS* relationships it drops to 0.389 seconds. Besides this decrease in the e2e delay, note that in Figure 2 the CDF corresponding to simulating *AS* relationships is much smoother than the second CDF, which exhibits a step-wise increase. This difference shows that the e2e delay with *AS* relationships enabled exhibits a much higher variability compared to ignoring *AS* relationships. This variability is likely to affect other performance metrics like jitter and buffer occupancy.

[1] *AS* numbers have been anonymized since *AS* relationships are considered sensitive information by the ISPs.

[2] Typically routers take much less than 100 seconds to converge, but to be conservative we used a longer period.

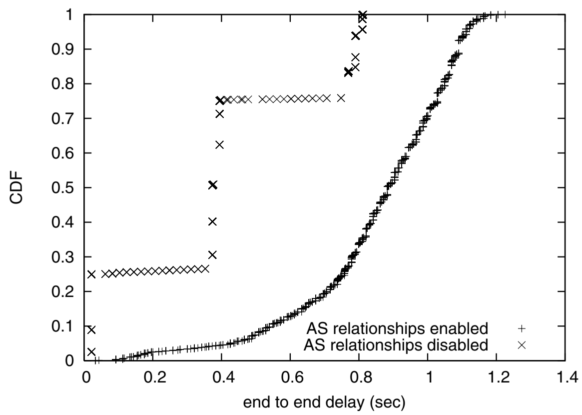

Figure 2. CDF of e2e delay between traffic sources and destination.

Table 1. Total number of paths for each AS with AS relationships enabled and AS relationships disabled.

AS number	1	2	3	4	5	6	7	8
AS relationships enabled	12	9	10	8	8	7	9	6
AS relationships disabled	12	13	16	15	13	15	15	13

A second implication of policy routing is that *ASs* have fewer alternative *AS* paths. For example, in Figure 1 when ignoring *AS* relationships *AS* 7 has three (one through each neighbor) disjoint paths to reach destination 2. One the other hand, with *AS* relationships enabled, *AS* 7 has only one possible path through *AS* 5, since the other two paths are not valley-free. In Table 1, we show the total number of paths we find in the BGP tables of the 8 simulated *ASs*. The consistent decrease in the number of paths when *AS* relationships are enabled highlights that *ignoring AS relationships increases the path diversity* of the *ASs* in a simulation. Path diversity is a property that can play an important role in simulations measuring such properties as network resilience, vulnerability to attacks, links and router failures, load balancing, multi-path routing, convergence of routing protocols and others.

An additional implication of policy routing is that due to the smaller number of available *AS* paths as compared to shortest path routing, some *ASs* or *AS* links are likely to receive greater load than when assuming shortest path routing. For example, in Figure 1 the dashed paths share the links from *AS* 7 to *AS* 5. On the other hand, when assuming shortest path routing the three paths are mostly disjoint, with only the link between *AS* 3 and *AS* 2 being shared by two flows. Thus, *AS* links and *ASs* will receive greater load than when ignoring *AS* relationships, which is likely to produce more packet drops, increased delay, congestion, router failures and other important events. In our simulations, we find that because of the increased load on the links between *AS* 7 and *AS* 5 the average bandwidth of the three flows decreases substantially. In Table 2, we list the average bandwidth for each of the three flows with and without *AS* relationships enabled.

Table 2. Average bandwidth per flow with AS relationships enabled or disabled.

Flow	$4 \rightarrow 2$	$6 \rightarrow 2$	$8 \rightarrow 2$
AS relationships enabled Bandwidth (Kbps)	113	164	121
AS relationships disabled Bandwidth (Kbps)	202	196	397

In summary, we highlight that ignoring *AS* relationships produces the following important artifacts:

- *AS* paths are substantially shorter than in reality.
- The number of alternative *AS* paths available to an *AS* is substantially larger than in reality.
- The traffic load on *AS* links and on individual *ASs* is substantially lower than in reality.

4 *AS* Relationships-aware topological properties

To represent *AS* topologies annotated with *AS* relationships, we use a graph G with edges annotated as c2p or p2p. c2p edges are *directed* from the customer *AS* to the provider *AS*, while p2p edges are *undirected*. We call such graphs G *annotated graphs*. Annotated graphs can also be used to represent other link characteristics, like link bandwidths, link latencies, or node characteristics, like router vendor models or router locations. In this study, we focus on using annotated graphs to model *AS* topologies annotated with *AS* relationships.

The topological property that most state-of-the-art topology generators reproduce is the degree distribution of a network. For our topology generator we choose to reproduce the following three properties.

***AS*-degree distribution.** Along the lines of existing topology generators, we reproduce the degree distribution of the *AS* topology of the Internet. The degree distribution tells us how many nodes of each degree are in the network.

Annotation-degree distributions. The degree distribution of an *AS* topology does not convey any information about the different types of *AS* relationships in a topology.

To take into account *AS* relationships we look at the number of customers, providers and peers each *AS* has. We define the *customer-degree* d_{p2c} of an *AS* as the number of its customers, the *provider-degree* d_{c2p} as the number of its providers and the *peer-degree* d_{p2p} as the number of its peers. We collectively refer to d_{p2c}, d_{c2p} and d_{p2p} as *annotation degrees*. The second property we select to reproduce is the annotation-degree distributions of an *AS* topology. These distributions are a natural generalization of the degree distribution of a network that take into account the presence of different types of *AS* relationships. The customer-degree distribution tells us how many nodes with a specific number of customers are in the network. Similarly, the provider- and peer-degree distributions tell us how many nodes with a specific number of providers and peers, respectively, are in the network.

Annotation-degree correlations. The annotation-degree distributions do not tell us anything about the correlations between these degrees, i.e., how many customers, providers and peers a specific *AS* has. Correlations between different annotation degrees appear often in the Internet. For example, large tier-1 *ASs* typically have a large number of customers, i.e., large d_{p2c}, no providers, i.e., zero d_{c2p}, and a small number of peers, i.e., small d_{p2p}. On the other hand, medium size ISPs have a small set of customers, several peers, and few providers. Note, that simply ignoring these correlations can lead to graphs that follow the previous two properties, but have artifacts, like high degree nodes with many providers.

The exact correlations between the annotation degrees of an *AS* are captured in the joint distribution $P(d_{p2c}, d_{c2p}, d_{p2p})$, which is defined as the number $n(d_{p2c}, d_{c2p}, d_{p2p})$ of nodes in the network with d_{p2c} customers, d_{c2p} providers and d_{p2p} peers over the total number of nodes n:

$$P(d_{p2c}, d_{c2p}, d_{p2p}) = n(d_{p2c}, d_{c2p}, d_{p2p})/n.$$

We call this distribution the *joint annotation-degree distribution* (JADD). JADD is a multivariate distribution and its marginals[3] are the annotation-degree distributions, i.e., our second property. From JADD we can also derive the *AS*-degree distribution simply by summing the annotation degrees of a node. Consequently, JADD is a union of the three properties we have discussed thus far.

5 *AS* Topology Generator

In this section we outline our framework for modeling and reproducing the JADD of real *AS* topologies. Our topology generation scheme proceeds in two phases. In the first phase, given the number of nodes N in the target graph we produce N degree triplets d^i_{p2c}, d^i_{c2p} and d^i_{p2p}, $1 \leqslant i \leqslant N$, such that the JADD of these triplets follows the JADD of real *AS* topologies. In the second phase given the N degree triplets we contract the annotated graph.

5.1 Modeling JADD

We model JADD using *copulas* [17], an powerful statistical tool that fully quantifies the dependence among multiple random variables. In contrast to other well-known correlation metrics, like Pearson's coefficient, Kendall's tau or Spearman's rho, copulas do not provide a single scalar value but a function that can capture complex correlations and fine-grained details of the dependence structure.

According to Sklar's theorem [19], any continuous[4] 3-dimensional multivariate cumulative distribution function (CDF) F can be written in the form:

$$F(x_1, x_2, x_3) = C(F_1(x_1), F_2(x_2), F_2(x_3)), \quad (1)$$

where F_1, F_2 and F_3 denote the marginal CDFs. The function C is called a copula and has uniform distributed marginals in $[0, 1]^3$. Given the copula function and the marginal CDFs F_1, F_2 and F_3, we can determine the joint distribution F using equation 1. Thus, copulas have two important properties: 1) given the marginals they *fully* describe the joint distribution F, and most importantly, 2) they enable the practitioner to model the dependence structure *independently* of the marginal distributions.

Modeling marginal distributions is a fairly easy task, since there exist a wealth of statistical methods and distributions for matching univariate samples. To find the appropriate marginal distributions we constructed an *AS* topology from RouteViews [16] data. We downloaded a BGP table from the collector `route-views2.oregon-ix.net` on 07/18/2005 and extracted *AS* links, ignoring private *AS* numbers and *AS* sets. We inferred c2p and p2p relationships using the heuristics in [7, 6]. This way, we derived a real *AS* topology annotated with c2p and p2p relationships. From this topology, we extracted the customer-, provider- and peer-degree distributions and evaluated alternative fitting models. We find that the customer-degree distribution can be well approximated using a generalized Pareto distribution (GPD). Moreover, the peer-degree distribution can be accurately modeled with a pair of GDPs, one for the body and one for the tail of the distribution. For the provider-degree distribution, we were not able to fit a parametric model. For this reason, we model the distribution by

[3] Given three jointly distributed random variables X, Y and Z, the marginal distribution of X is the probability distribution of X ignoring information about Y and Z, typically calculated by summing or integrating the joint probability distribution over Y and Z.

[4] Degree distributions are inherently discrete distributions. Nevertheless, they can be turned into continuous by adding a random uniform noise $U(-0.5, 0.5)$ to each degree sample.

treating its six highest quantiles as invariant. This approximation results in underestimating the degree of the nodes in the tail of the distribution. However, the tail accounts for only 2% of the nodes and the maximum provider-degree (17) is relatively small. Thus our approximation is not expected to induce significant bias. The first step to reproduce JADD is to generate N customer, N provider and N peer degrees from the corresponding fitted models.

Next, we model the copula by resampling historical correlation data. We first construct a set with all the degree triplets of the collected *AS* topology. Then, we sample N degree triplets from this set. These N triplets include information on both the actual annotation degrees and the correlations between them. We extract the correlation information by mapping each triplet into the $[0,1]^3$ space. To do so, we replace each annotation degree with its rank normalized by $1/N$. The resulting triplets $(u^i_{p2c}, u^i_{c2p}, u^i_{p2p})$ reflect the correlations between the annotation degrees in the original *AS* topology and are independent of the actual annotation degrees. Each of the u^i_{p2c}, u^i_{c2p} and u^i_{p2p} is uniformly distributed in $[0,1]$.

Finally, we combine the $(u^i_{p2c}, u^i_{c2p}, u^i_{p2p})$ triplets with the generated annotation degrees to derive the final degree triplets that follow the JADD of the original topology. Each $(u^i_{p2c}, u^i_{c2p}, u^i_{p2p})$ triplet is resolved into a degree triplet $(d^i_{p2c}, d^i_{c2p}, d^i_{p2p})$ by mapping each u^i_{p2c}, u^i_{c2p}, u^i_{p2c} into the inverse CDF of the corresponding annotation degrees. For example, we map u^i_{p2c} into d^i_{p2c}, where d^i_{p2c} is the value of the inverse CDF of the generated customer degrees at the point u^i_{p2c}. Thus, we derive the N annotation-degree triplets $(d^i_{p2c}, d^i_{c2p}, d^i_{p2p})$, $1 \leqslant i \leqslant N$, that follow the JADD of the original topology.

5.2 Generating annotated AS topologies

Given the N annotation-degree triplets, we construct a random annotated graph using the following algorithm:

1. For each of the generated triplets, we introduce a node with d^i_{p2c} customer stubs, d^i_{c2p} provider stubs and d^i_{p2p} peer stubs[5].

2. We connect stubs by performing one random matching between p2p stubs and a second random matching between c2p and p2c stubs. If the number of p2p stubs is odd or if the number of c2p stubs is not equal to the number of p2c stubs, then some stubs will remain unmatched. We ignore such stubs.

3. Random matchings can lead to self-loops and multi-edges. We extract the final graph by removing self-loops and multi-edges.

[5]A stub is a half edge that is adjacent to a single node. By connecting two stubs we get a regular edge.

This algorithm for constructing random graphs is a generalization of the algorithm used by the PLRG topology generator [1]. The PLRG algorithm uses random matching to create an undirected graph ignoring *AS* relationships. We extend this algorithm by using different types of stubs to account for customer, provider, and peer edges. Then, we perform two random matchings[6] between stubs of the same type and of compatible direction. A limitation of the PLRG topology generator is that it produces graphs that contain self-loops and multi-edges. Self-loops and multi-edges usually appear on or between large degree *ASs*. This is because large degree *ASs* have many stubs and thus it is quite likely that the random matching will match two stubs that belong to the same *AS* or more then two stubs between two *ASs*. In our generalization this two problems are diminished. This is because edges of high degree *ASs* are mainly customer edges that can only connect to customer *ASs*, which usually are of small degree, and not to other high degree *ASs*.

To make a first evaluation of the accuracy of the resulting synthetic graphs, we generate a topology with 20,305 *ASs*, which is the number of *ASs* we found in the *AS* topology we constructed from RouteViews. Then, we compare the customer-, provider-, and peer-degree distributions of the synthetic topology with the corresponding distributions of the real topology. In Figure 3 we plot the CDF of the customer, provider, and peer degrees. The empty points show the distributions observed in the real *AS* topology, whereas the solid points depict the same distributions in the synthetic topology. We first observe that the customer distribution exhibits the longest tail, followed by the peer distribution, followed by the provider distribution, which has a rather short tail. The maximum number of customers is 2,384, the maximum number of peers is 434 and the maximum number of providers is 17. These distributions confirm that different types of relationships can have radically different properties as we argued in the introduction. Next, we see that the generated degree distributions follow closely the real degree distributions, which highlights the effectiveness of our marginal models.

6 Conclusions

We highlighted the problems that the Internet community is facing due to the lack of *AS* relationship models and discussed its implications on conducting realistic and reliable simulation studies. We used simulation experiments to demonstrate that ignoring *AS* relationships can change a wide range of performance metrics, which are typically used by researchers in performance evaluation studies. We

[6]More generally, we need to perform as many random matchings as the number of different link annotations. Thus, if we also want to model s2s links, then we can add s2s stubs and perform a random matching between them.

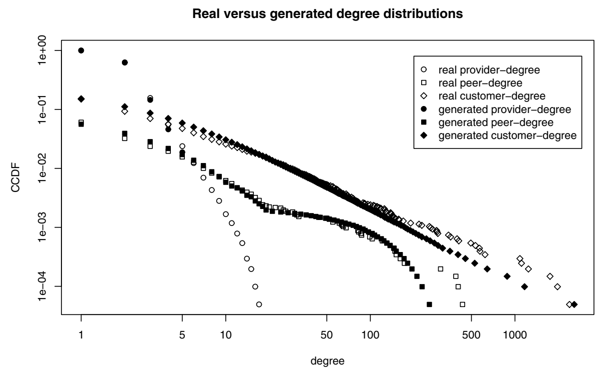

Figure 3. Customer, peer, and provider degree distributions of real graph and of synthetic graph of the same size.

draw motivation from our findings first to note that shortest path routing is a questionable assumption that should be used with great care in simulation studies and secondly to introduce a novel topology generation framework. Our framework improves the state-of-the-art by producing *AS* graphs that follow the degree distribution of the Internet as well as two new properties: 1) the annotation-degree distributions, and 2) the joint annotation-degree distribution (JADD). These two properties extract information about the number of customers, providers and peers of *ASs* in the Internet and enable us to create synthetic *AS* graphs with realistic customer, provider and peer assignments. We use powerful statistical tools to model these properties on real *AS* topologies and, finally, we introduce an algorithm to reproduce these properties in synthetic *AS* topologies.

Soon, we intend to supplement our framework with a comprehensive evaluation of the properties of the resulting graphs. An additional promising venue is to use our annotated graphs to model router level topologies and other interesting network characteristics, like link bandwidths, link latencies, router vendor models or router locations. Finally, we will make publicly available a new topology generator capable of modeling and generating annotated graphs.

References

[1] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 171–180. ACM Press, 2000.

[2] R. Albert and A.-L. Barabási. Topology of evolving networks: Local events and universality. *Physical Review Letters*, 85(24):5234–5237, 2000.

[3] J. M. Carlson and J. Doyle. Highly optimized tolerance: A mechanism for power-laws in designed systems. *Physical Review E*, 1999.

[4] H. Chang, S. Jamin, and W. Willinger. Internet connectivity at the AS level: An optimization driven modeling approach. In *Proceedings of MoMeTools*, 2003.

[5] X. Dimitropoulos, S. Beeram, and G. Riley. BGP++ Home Page. `http://www.ece.gatech.edu/research/labs/MANIACS/BGP++/`.

[6] X. Dimitropoulos, D. Krioukov, M. Fomenkova, B. Huffaker, kc claffy, and G. Riley. AS relationships: Inference and Validation, 2006. Under Submission.

[7] X. Dimitropoulos, D. Krioukov, B. Huffaker, kc claffy, and G. Riley. Inferring AS relationships: Dead end or lively beginning? In *Proceedings of 4th Workshop on Efficient and Experimental Algorithms (WEA' 05)*, May 2005.

[8] X. Dimitropoulos and G. Riley. Creating realistic BGP models. In *Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'03)*, October 2003.

[9] M. Doar. A better model for generating test networks. In *IEEE GLOBECOM*, 1996.

[10] L. Gao. On inferring Autonomous System Relationships in the Internet. In *IEEE/ACM Transactions on Networking*, December 2001.

[11] L. Gao and F. Wang. The extent of AS path inflation by routing policies. In *IEEE Global Internet Symposium*, 2002.

[12] C. Gkantsidis, M. Mihail, and E. Zegura. The markov chain simulation method for generating connected power law random graphs. In *SIAM Workshop on Algorithm Engineering and Experiments (ALENEX)*, 2003.

[13] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, kc claffy, and A. Vahdat. Lesson from three views of the Internet topology. Technical Report TR-2005-02, CAIDA, 2005. `http://www.caida.org/outreach/papers/2005/tr-2005-02/`.

[14] Z. M. Mao, L. Qiu, J. Wang, and Y. Zhang. On AS-level path inference. In *SIGMETRICS*, 2005.

[15] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *MASCOTS*, 2001.

[16] D. Meyer. University of Oregon Route Views Project, 2004.

[17] R. B. Nelson. An introduction to copulas. *Springer-Verlag Lecture Notes in Statistics*, 139:216, 1999.

[18] C. Palmer and G. Steffan. Generating network topologies that obey power laws. In *GLOBECOM*, November 2000.

[19] A. Sklar. Fonctions de repartition a n dimensions et leurs marges, 1959.

[20] N. Spring, R. Mahajan, and T. Anderson. Quantifying the causes of path inflation. In *ACM SIGCOMM*, 2003.

[21] H. Tangmunarunkit, R. Govindan, and S. Shenker. Internet path inflation due to policy routing. In *SPIE ITCom*, 2001.

[22] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin. The impact of routing policy on Internet paths. In *IEEE INFOCOM*, 2001.

[23] B. M. Waxman. Routing of multipoint connections. *IEEE JSAC*, 1988.

[24] J. Winick and S. Jamin. Inet-3.0: Internet topology generator. Technical Report UM-CSE-TR-456-02, University of Michigan, 2002.

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