

Generating Representative ISP Topologies From First-Principles

Chong Wang
Computer Science Department
Boston University
Boston, MA 02215
wangch@cs.bu.edu

John W. Byers
Computer Science Department
Boston University
Boston, MA 02215
byers@cs.bu.edu

ABSTRACT

Understanding and modeling the factors that underlie the growth and evolution of network topologies are basic questions that impact capacity planning, forecasting, and protocol research. Early topology generation work focused on generating network-wide connectivity maps, either at the AS-level or the router-level, typically with an eye towards reproducing abstract properties of observed topologies. But recently, advocates of an alternative “first-principles” approach question the feasibility of realizing representative topologies with simple generative models that do not explicitly incorporate real-world constraints, such as the relative costs of router configurations, into the model. Our work synthesizes these two lines by designing a topology generation mechanism that incorporates first-principles constraints. Our goal is more modest than that of constructing an Internet-wide topology: we aim to generate representative topologies for single ISPs. However, our methods also go well beyond previous work, as we annotate these topologies with representative capacity and latency information. Taking only demand for network services over a given region as input, we propose a natural cost model for building and interconnecting PoPs and formulate the resulting optimization problem faced by an ISP. We devise hill-climbing heuristics for this problem and demonstrate that the solutions we obtain are quantitatively similar to those in measured router-level ISP topologies, with respect to both topological properties and fault-tolerance.

General Terms

Network topology modeling, network design, optimization.

1. INTRODUCTION

Topology modeling has now emerged as a central area of network research, due to its significant impact on other aspects of network research. This is evidenced by recent work on topology discovery and inference [25, 10], topology modeling and generation [29, 19, 11] and the impact of network topology on protocol design [24, 4]. But to date, network topology research has only addressed a limited subset of the key questions in the problem space. In particular, there has been extensive focus on modeling the network connectivity, i.e. the underlying graph, but this meets only some of

the community’s needs. For a topology model to be more broadly applicable and useful, it should be able to describe multiple aspects of the network, including connectivity, performance characteristics (bandwidth, latency, etc.), reliability properties (loss probability, fault tolerance, etc.), the growth of the network, and so on.

A prevalent aim of topology research has been to match observed graph-theoretic properties of the underlying network. For instance, researchers developed mathematical models to generate certain degree distributions observed from the AS-level topology [20, 7, 5], as well as evolutionary models that attempt to explain how certain degree distributions and other graph-theoretic properties arise [6, 27]. But in evaluating network protocols and for performing other network research, simple connectivity maps are inadequate; moreover, quite different topologies can have similar values of some graph theoretical metrics [15].

Recently, a first-principles approach [15] was proposed for understanding the router-level topology. In a first-principles approach, technological constraints and economic tradeoffs are considered as important factors that contribute to an ISP’s router-level topology. This first-principles theory attempts to model the engineering issues faced when a network is designed and therefore can both accurately reflect certain properties found in the real network and provide an explanatory model for the process by which a network evolved.

In this paper we build a topology generation model that starts from first-principles explanations, and use it to produce topologies for single ISPs. In our framework, the topology of an ISP can be viewed as the result of a concrete optimization problem: namely, a network design problem in which costs of deploying equipment (links, various router configurations, various PoP configurations) are known and the objective is to build a capacitated network to satisfy an initial customer demand specified as a traffic demand matrix. There have been a number of works studying related network design problems, either from the perspective of a motivating framework, such as HOT [2, 8], or focusing on the algorithmic aspects of a specific optimization problem in a theoretical model [22, 23], or from the game-theoretic process of network creation [3, 9]. Our work differs from these lines as we are not so much interested in theoretical properties of the constructed graphs, so much as realism and representativeness for a concrete problem of interest. We show that with a very simple and natural optimization framework with relatively few inputs and parameters, networks quantitatively similar to measured ISP topologies can

be heuristically generated. Surprisingly, although our optimization framework does not explicitly reward the designer for building fault-tolerant configurations, the networks we generate do appear to contain substantial route diversity.

An open issue with our approach is validation of the underlying assumptions in our model. The price models of ISPs may be complex and contain confidential information, and as such, it is not currently possible for us to quantify discrepancies between our model and those used in reality. Nevertheless, we argue that our model, while surely imperfect, captures many of the scaling costs present in large-scale ISP design.

The remainder of the paper is organized as follows: In Section 2 we present and justify our model, and in Section 3 we discuss some facts from both the observation from the real topologies and the first-principles approach that simplifies the optimization. A hill-climbing heuristic is described in Section 4, and in Section 5 we describe several mechanisms to improve the performance of the heuristics on problems of large scale. Experimental results are shown in Section 6, followed by our conclusions and directions for future work in Section 7.

2. PROBLEM STATEMENT

We assume we are interested in building a network that connects users that lie on a two-dimensional plane. In this setting, we are given a set of client locations $C = \{c_1, c_2, \dots, c_n\}$ and a set of candidate facility locations $F = \{f_1, f_2, \dots, f_m\}$. Each location L has a pair of coordinates: (x_L, y_L) on the plane, and we consider the Euclidean distance between locations L_1 and L_2 : $\sqrt{(x_{L_1} - x_{L_2})^2 + (y_{L_1} - y_{L_2})^2}$.

We are also given a traffic demand matrix $T_{n \times n}$, in which $t_{i,j}$ is determined by $C \times C \rightarrow R : t_{i,j} = t_{j,i} \geq 0$.

The objective is to select a set of facilities $S \subseteq F$, a set of edges $E \subseteq (S \times C) \cup (S \times S)$ and to build an undirected graph $G = (S \cup C, E)$ with the minimal value of $cost(G) = \sum_{v \in S \cup C} cost(v) + \sum_{e \in E} cost(e)$.

We let the function $B : E \rightarrow R^+$, denote the capacities of edges. Now we define the costs of nodes and edges. At present, there is no publicly available and widely agreed upon cost model for the router level network. Our model offers a simple but natural model that captures several of the essential networking considerations. For a node (a router), the cost is due to routing the traffic going through this node to different interfaces. Though the exchange of traffic may introduce additional cost that is super-linear in the (sum of the) adjoining link bandwidths, for simplicity and for lack of concrete evidence to the contrary, we model the cost of a node v as linear in this quantity:

$$cost(v) = a \sum_{(u,v) \in E} B(u,v), a > 0.$$

For an edge (a link), a commonly used cost metric is the bandwidth-delay product of the link. From an economic point of view, there is also the additionally one-time cost of building a physical link, and thus the marginal cost per unit length or unit bandwidth unit decreases as the length or bandwidth of the link increases. So we model the cost of an edge as

$$cost(e) = (b + B(e)^{d_1})(l(e))^{d_2}, b > 0, d_1, d_2 \in (0, 1], e \in E,$$

where $l(e)$ is the length of e , i.e. the Euclidean distance

between the two nodes it connects.

Finally, we subject the graph to the following constraints:

1. Clients must be single-homed: $\forall c \in C, \text{degree}(c) = 1$. We note that in reality, clients are multi-homed for fault-tolerance. However, in this model we do not include this feature of the network.
2. The shortest path $P(i, j)$ is well-defined between all pairs of nodes i and j , and is used to route all traffic between nodes i and j .
3. The network has sufficient capacity to carry the offered load across the shortest paths:

$$\forall k, \sum_{(i,j) | e_k \in P(i,j)} t_{i,j} \leq B(e_k).$$

3. DISCUSSION OF PRACTICAL CONSIDERATIONS

In reality, an AS may contain thousands of routers, and solving an NP-hard network design problem of this scale is obviously intractable. However, in a typical network, a large fraction of network demand, and by extension, a large fraction of routers, are clustered in certain areas. In the remainder of the paper we consider all the routers in the same city as a PoP, though there may be more than one physical PoP in a city. A natural approach is to optimize the problem at the PoP level and then design each PoP. The assumption is that both intra-PoP structure and inter-PoP connections follow a certain rule. In this section, we justify this assumption by empirical evidence and by a first-principles explanation.

From the router-level ISP topologies measured by RocketFuel [25], we observed a structural similarity among PoPs: a typical basic structure of a PoP is transit-stub, where transit routers are interconnected and stub routers are connected to one of the transit routers (though there are extra connections). For instance, consider the PoPs from AS 7018 (AT&T) measured by RocketFuel. We filter out too small PoPs (those with less than 25 routers) and for each PoP, we calculate P_1 , the probability that the router's degree is more than 1. The average value of P_1 is 6.32% (with a standard deviation of 6.9%). So the majority of the routers inside PoPs are stub routers. Inter-PoP connections are typically connections between transit routers of different PoPs.

This can also be explained by the first-principles theory: aggregating the customer traffic demands at PoPs instead of long-haul connections between small routers is a reasonable heuristic solution to avoid the high expense of such links. The reason for the divide-and-conquer nature of a transit-stub structure inside a PoP is similarly motivated. Consequently, the problem of designing a PoP just becomes the problem of satisfying the traffic and connection requirements by different types of routers, each identified with prices and technological specifications, with the minimum cost. This is a multiple-dimensional knapsack problem [18].

In the remainder of the paper we consider only the PoP level topology, i.e. each node represents a PoP and each edge represents all links between a pair of PoPs.

4. APPROXIMATION ALGORITHM

In Section 3 we argued that aggregating the traffic at PoPs and solving the optimization problem at the PoP level is a

reasonable solution to the NP-hard network creation problem stated in Section 2. So we take a 2-phase approach: first, we generate a set of PoPs which satisfy the customer traffic demands, then we build the optimized PoP-level network.

4.1 Generating Base PoPs

The purpose of this phase is to connect every customer with a facility, in other words, to build an annotated sub-graph $G' = (V', E')$ where $E' \subseteq S \times C$ with the same objective function and cost model as stated in Section 2. However, G' is not necessarily connected, so the constraint of the existence of the shortest path between every customer applies to every connected component of G' . This is an instance of the universal facility location problem where we are given a set of locations each of which has a known demand, and a set of locations where we can open facilities. There is a cost of shipping demands from one location to another, and a cost of opening a facility. The objective is to open a set of facilities such that the sum of shipping costs and facility costs is minimum. Depending on the properties of the cost functions, there are some variants of this problem. There are two types of solutions to this problem, local search heuristic [14, 16] and linear programming approach [17], each suitable to solve some variants of the problem.

In the following we will use the term “base PoPs” to denote the PoPs to serve the customer traffic demands. In addition to generating a set of base PoPs, we want to estimate a reasonable value for a in our model by comparing the result of this phase with the real network, for instance, whether or not the number of base PoPs opened is close to the number of PoPs in the real network. When solving the facility location problem, we make the following approximations: first, in practice a physical PoP is built at some cost to serve a certain amount of customer traffic demands, furthermore, the customer traffic demands each PoP serves is a very small portion of the traffic it carries in the global network, so we approximate the cost of a PoP by a fixed value; second, as the links are expected to be short, we omit the link construction cost, i.e. $b = 0$. Then the problem is reduced to a classical uncapacitated facility location problem. There is a simple local search heuristic to solve it known as Hamburger’s algorithm [14]: from an initial feasible solution, in each iteration we greedily optimize the solution by adding a new facility, dropping a facility, or substituting an existing facility with a new facility, until no further improvement can be done. Korupolu et al. proved that this achieves a constant-factor approximation in polynomial time [12].

4.2 Optimizing the PoP-level network

Starting from the set of base PoPs each with a traffic demand aggregated from the customers, the second phase aims to build an optimized PoP-level network.

We employ the following greedy heuristic: starting from an initial feasible solution, i.e. a connected graph in which each edge is annotated with the capacity determined by the shortest path routing algorithm and the aggregated traffic demand at each base PoP, we attempt the following operations in each iteration to optimize the network:

1. **Add an edge:** Add a capacitated edge (u, v) to the current version of graph $G^{(i)} = (V^{(i)}, E^{(i)})$, where the capacity of the edge is determined by the shortest-path routing algorithm and the all-pairs traffic demand.

2. **Remove an edge:** Remove an existing edge $e \in E$ in the current version of the graph G , such that removal of e does not disconnect the graph.
3. **Add a PoP.** The motivation of this operation is to make traffic demands between different PoPs share links instead of using separate links. To do so, we need to choose a PoP to add, connect it to some neighboring PoPs and remove some existing edges in the neighborhood to reroute traffic. To reduce the potentially vast search space with each insertion, we use additional heuristics. For a proposed facility f , we let r_f be the minimum radius for which exactly T_1 existing facilities have distance at most r_f from f . Denote those T_1 facilities by N_f . We then consider all edges (u, v) such that $u, v \in N_f$, and let $c(u, v)$ be the closest point to f along the line segment joining u and v . If the distance between f and $c(u, v)$ is below a second threshold T_2 , we would delete (u, v) when building f , using the intuition that it is more economical to route traffic through f than maintain a (long) link closely bypassing f . So for every proposed facility, there is a deterministic, polynomial-time procedure to remove redundant edges, so we can readily evaluate the set of graphs that would result from adding each facility.
4. **Remove a PoP** which was added by a previous operation. Note that in our 2-phase optimization framework a base PoP can not be removed and we do not redistribute the customer traffic demands. Note that this procedure cannot directly reverse the Add a Pop operation, since it does not add edges.

Let the set $\{G_t^{(i+1)}\} = \{(V_t^{(i+1)}, E_t^{(i+1)})\}$ comprise all the graphs to be evaluated at iteration $i + 1$, which is the union of the sets of graphs to be evaluated due to the operations described above. For each graph $G_t^{(i+1)}$, we compute all-pairs shortest paths and calculate the capacities of edges as well as the cost of the graph. Denoting E_o the set of edges whose capacities are changed by an operation o , then

$$E_o = \{e_o \in E^{(i)} \cup E_t^{(i+1)} \mid \exists u, v, s, t. e_o \in P^{(i)}(u, v) \ominus P^{(i+1)}(u, v)\}.$$

Here, \ominus denotes the symmetric difference between two sets and $P^{(i)}(u, v)$ denotes the shortest path between u and v in $G^{(i)}$.

Let V_o denote the set of nodes whose traffic is changed by an operation o : $V_o = \{v_o \mid \exists v_1((v_o, v_1) \in E_o)\}$. Let $c^{(i)}(x)$ be the cost of x calculated on $G^{(i)}$. We only need to recalculate the costs of edges and nodes in E_o and V_o , and the construction costs of the edges that are effectively removed by an operation, so

$$\begin{aligned} c^{(i+1)}(G_t^{(i+1)}) = & \sum_{e_o \in E_o} cost(e_o) + \sum_{v_o \in V_o} cost(v_o) \\ & + \sum_{e_s \in E_t^{(i+1)} - E_o} c^{(i)}(e_s) + \sum_{v_s \in V_t^{(i+1)} - V_o} c^{(i)}(v_s) \\ & - \sum_{e_r \in E^{(i+1)} \mid \forall i, j (e_r \notin P^{(i+1)}(i, j))} cost(e_r). \end{aligned}$$

Definition 1. Denote the **benefit** of an operation t in iteration i to be

$$c^{(i)}(G^{(i)}) - c^{(i+1)}(G_t^{(i+1)}).$$

In each iteration we choose the operation with highest **benefit**, apply the operation to $G^{(i)}$ to realize an improved version of the network $G^{(i+1)}$, until no further improvement can be made.

4.3 Analysis of the Approximation Algorithm

In [12] a detailed performance analysis of the local search heuristic of the facility location problem was presented. Here we analyze the second phase of our approximation algorithm.

To evaluate each operation, we have to compute all-pairs shortest paths and update the traffic on edges in E_c and nodes in V_c , so the complexity of evaluating one operation is $O(s^3 \log s)$, where s is the number of PoPs in the current version of the network. In each iteration, every pair of node is evaluated either by the operation of adding an edge or removing an edge, and every possible facility location is evaluated either by the operation of adding a PoP or removing a PoP, so we need to evaluate $O(s^2 + m)$ operations in each iteration.

5. IMPROVING THE PERFORMANCE OF THE ALGORITHM

As analyzed in the previous section, the time complexity of the algorithm is high. We now focus on making our algorithm practically usable in generating large topologies and predicting future topologies.

5.1 Employing a Priority Queue

In practice, re-evaluating the cost of every operation at each iteration is overkill. The cost of many operations does not change, and the overall rank of most operations does not change dramatically. Therefore, we manage the operations in a priority queue. The operations are sorted by their **benefit**. In each iteration we evaluate only a small fraction of the operations in the priority queue and thus reduce the total number of operations to be evaluated during the optimization, while sacrificing the guarantee of making the optimal greedy move with full information.

In the priority queue, the operations with a higher **benefit** are more likely to be evaluated than those with a lower one. On the one hand, in a large scale network, a single local optimization will not change the network dramatically. So intuitively, with high probability an operation with a low **benefit** in an iteration will still have a low **benefit** in the next iteration. On the other hand, the network may change greatly after a number of iterations, which makes an operation with formerly a low **benefit**, or even an operation which was not previously applicable, a good operation. Considering these two facts, in the next section we experimentally evaluate two strategies of choosing operations to be evaluated. The first is a weighted probability strategy, whereby an operation whose position in the priority queue is j is re-evaluated with probability proportional to $\frac{1}{j}$ in a given iteration, and thus the total number of operations evaluated in each iteration is $\sum_{j=1}^N \frac{1}{j} = O(\log N)$, where N is the total number of feasible operations. The other strategy, the uniform probability strategy, is less aggressive: in each iteration we evaluate the top c_u items in the priority queue and sample the other items each with a probability p_b . Here, the total number of operations to be evaluated in each iteration is $c_u + p_b(N - c_u)$. In practice, we assign

a probability rp_b for operations with a positive **benefit**, where $r > 1$ is a small constant.

5.2 Incremental Optimization

Another method we found to be beneficial in ordering PoP-level operations is an *incremental* strategy to the network design problem that does not operate in the strict phased order described earlier. Briefly, we insert the base PoPs into the network in batches, ordered by the amount of customer traffic demands that they serve. Between each such insertion, we re-optimize the intermediate network to minimize the cost. This approach has two benefits: first, the time to converge to an optimized solution after each insertion is faster, and summing over all insertions, leads to faster convergence. Second, this approach mimics aspects of the historical evolution that takes place inside real networks. In the experimental section, we demonstrate that the topologies output using this incremental approach are indeed comparable to the strict two-phase approach described earlier.

We note that yet another alternative approach to incremental optimization is to simulate the growth of the network by increasing customer demands and iteratively recomputing both the configuration of base PoPs and the intermediate network as time elapses. Implementation of this step would move us closer to a faithful model of a time-evolving, growing network.

6. EXPERIMENTAL RESULTS

In this section we show our experiment results from the following two perspectives: one is the evaluation of the prioritization strategies mentioned in Section 5, the other is the comparison of the topologies generated by our approach with RocketFuel data (we note the ambiguities of RocketFuel data reported in [26], however it is the best data set for the ISP topologies up to date). Before showing the results, we first describe the details of experiments.

6.1 Description of the Experiments

We have shown our model and heuristics. However, some aspects remain unspecified from previous sections: the input traffic demand matrix and some parameters in our model. Our methodology is to leverage the real-world datasets to seek reasonable settings of them. In this sections we use 2 well-known ISPs as demonstration. We first discuss why we use these ISPs, then describe the detailed experimental settings.

6.1.1 ISP Selection

In reality, some ISPs are regional while others are global but with emphasis on a certain region. Determining representative customer traffic demands for a global ISP is complex and is beyond the scope of this paper. Instead, we focus on an ISP spanning a region that is large, but is relatively homogeneous in terms of per capita demand for network traffic, namely the continental US. So in this section we will show 2 representative ISPs: one is a regional ISP (AS 7018 [AT&T]), and one is a global ISP (AS 1239 [SprintLink]), which has some links from the United States to some positions all over the world. We do not exhaust all global ISPs by guessing the traffic demands or making an extremely complicated traffic model. However, by these two examples of representative and well-known ISPs, using a simple traffic

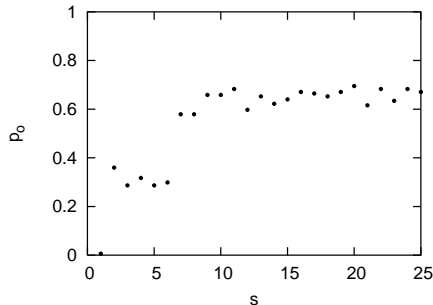


Figure 1: p_o of the weighted probability strategy

model, we show that our approach can generate topologies reasonably similar to reality.

6.1.2 Traffic Demand Matrix

Traffic demand matrix is an important input in our approach. Note that in our problem we need an estimation of static potential traffic demands over a network yet to be built instead of real traffic over the Internet. So an intuitive approach is to use the products of population at two locations as the estimation of traffic demands between them. A similar method [13] was used to estimate the volume in a telecommunication network. In our experiments, we use the population grid [1] which divides a region into areas of equal size and records the population in each area. Each area in the population grid serves as a client location as well as a candidate facility location mentioned in the problem statement. The traffic matrix is calculated in the following way: let p_i be the population obtained from the population grid at candidate location i , then $t_{i,j} \propto p_i p_j$.

This traffic model is also a simple form of gravity model. Note that the general gravity model is given by the following equation: $X_{ij} = \frac{R_i A_j}{f_{ij}}$, where X_{ij} represents the force from i to j , which can be interpreted as the amount of traffic amounts between i and j ; R_i represents the repulsive factor entering i ; A_j represents the attractive factor leaving j ; and f_{ij} is a friction factor from i to j . In our calculation, both the repulsive factor and the attractive factor are simply populations and the friction factor is a constant.

Recently there are advanced approaches [30, 21] to calculate traffic matrix. The approach used in our experiments is an approximation compared to those approaches, however, for an ISP without any information on the real traffic, population might be a data set with good availability and precision to take consideration into their marketing models.

6.1.3 Setting of Parameters

In our model, the values of d_1 and d_2 are set to 0.96 and 0.9 respectively. This setting might not represent the real price curve in the industry field, however, we claim that with this simplified setting, our results of both the first phase and the second phase of optimization share similarities to the reality. In section 4 we mentioned that the value of a is estimated by the number of base PoPs generated by the facility location algorithm. Though the number of PoPs generated depends on the traffic demand matrix, experimentally for the traffic demand matrix we use, the value of a is a deciding factor of the number of base PoPs. Then b is the only free parameter of our experiments. Note that both a and b are relative

values, so b might be different among situations. There are 2 minor unspecified parameters from our add PoP operation: we choose 8 for T_1 and 1 degree of latitude/longitude for T_2 .

6.1.4 Other Experimental Details

Resolution: In the real network, some PoPs are in towns which are geographically proximate, and they have either very similar roles or a master-slave relation in the network. In our experiments we view them as a single PoP. In our optimization framework, we explain the geographically proximate PoPs by the resolution during the process of optimization. In our experiments the resolution of AS 1239 is 2.5 degrees of latitude by 2.5 degrees of longitude, while the resolution of AS 7018 is 1 degree of latitude by 1 degree of longitude. We also fit the RocketFuel data to this resolution by combining multiple PoPs within a unit of resolution to a single PoP.

Base PoP locations: For a regional network, the locations and the customer traffic demands aggregated at the base PoPs are generated by solving the facility location problem. For a global network, the main region is considered as a regional network. However, we need to specify the base PoPs outside the main region. In our experiments on AS 1239, the locations of base PoPs outside the main region are manually set according to the real network measured by RocketFuel, and the customer traffic demands are set to reasonable values.

In reality, the positions of base PoPs may be determined by many factors in conjunction with the customer traffic demands, for instance, the convenience of building a physical PoP. Intuitively the different layouts of base PoPs will influence the generated topologies. For a comprehensive comparison, we will also show the results of our second phase optimization starting from the positions of PoPs from the real network by RocketFuel.

Incremental approach: Consider the incremental approach discussed in Section 5. A network evolves from a small size to a larger one as more and more base PoPs join the network. We use the term **period** to refer to the process of optimization based on the current network and the newly added base PoPs. So **period** i starts from the result of **period** $i - 1$, and the number of base PoPs added during **period** i is proportional to 2^i .

For ISPs with a small number of PoPs, the advantage of the incremental approach is not apparent; and for a global ISP, the imprecision of traffic demands at outliers will introduce a bias to the network. So here we only show the results of the incremental approach for AS 7018, which is both a very large AS and a regional AS.

Initial feasible solutions: For the strict two-phase optimization, the initial solution is simply a star topology, in which all the other nodes are connected to the node with the largest customer traffic demand; for the incremental approach, the initial solution for **period** 0 is a star topology, while for future **period**, the initial solution is to connect each new base PoP to the geographically nearest PoP in the intermediate network.

6.2 Evaluating Prioritization Strategies

In Section 5 we discussed two prioritization strategies. Theoretical evaluation of these is outside the scope of this paper; here we experimentally evaluate them. An important metric is p_o , the proportion of the operations applied

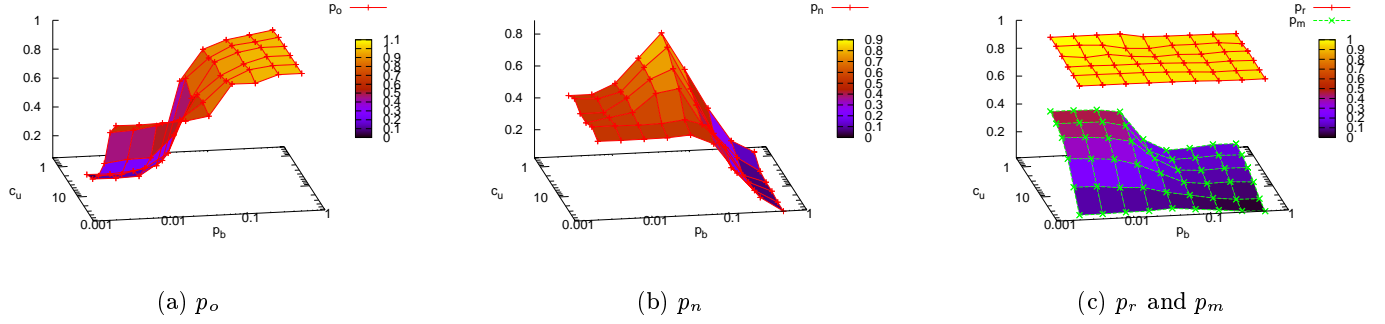


Figure 2: Performance metrics associated with the uniform probability strategy.

AS	network type	density	clustering coefficient	avg. distance	diameter	betweenness centrality	closeness centrality
7018	RocketFuel	0.035	0.325	3.32	6	35.64%	32.82%
7018	OPT-RealPoP	0.0358(0.0006)	0.138(0.04)	3.44(0.09)	7.13(0.35)	45.61%(5.57%)	34.42%(3.13%)
7018	OPT-FLPoP	0.0376(0.0007)	0.181(0.048)	3.58(0.09)	7.63(0.52)	57.67%(7.97%)	38.73%(3.57%)
7018	OPT-I-RealPoP	0.0348(0.0004)	0.159(0.024)	3.61(0.02)	7(0)	30.99%(0.57%)	27.39%(0.20%)
7018	OPT-I-FLPoP	0.0366(0.0004)	0.132(0.012)	3.86(0.01)	8(0)	61.05%(0.48%)	36.78%(0.31%)
1239	RocketFuel	0.5333	0.707	1.486	3	8.16%	40.82%
1239	OPT-RealPoP	0.5179(0.01)	0.788(0.013)	1.482(0.01)	2(0)	17.82%(0.63%)	43.20%(3.46%)
1239	OPT-FLPoP	0.5714(0.015)	0.821(0.021)	1.429(0.015)	2(0)	19.15%(2.5%)	63.13%(2.17%)

Table 1: Graph-theoretic metrics of networks. For AS 7018, $a = 0.2797, b = 825000$; for AS 1239, $a = 0.34965, b = 41000$. Experimentally computed values depict the average of 8 experiments followed by the standard deviation.

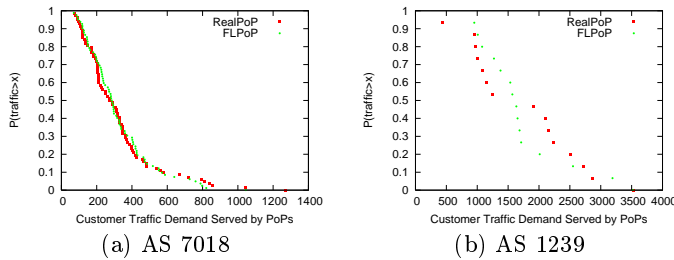


Figure 3: Modeled distributions of customer traffic demands served by PoPs

using the strategy compared to the operations applied if all items in the priority queue were evaluated in each iteration. For the weighted probability strategy, we study the evolution of p_o when the scalar s changes, i.e. when the operation at position j in the priority queue is evaluated with probability $\frac{s}{j}$. In figure 1 we show p_o during our experiments to generate the topology for AS 7018. From the figure we can see that the weighted probability strategy can give a reasonable performance, however, the performance does not scale as s grows. This can be explained by the fact mentioned in Section 5: when the priority queue is large, an initially bad operation is not even likely to be re-evaluated before the optimization terminates. So in the following we evaluate the less aggressive uniform probability strategy.

In addition to p_o , which evaluates the strategy from the

aspect of the final effect, we study the following metrics: p_n , the probability that the best operation in an iteration is not actually evaluated; p_r , the probability that an operation in position of the first c_u items in the priority queue still has a positive benefit in the next iteration; p_m , the probability that an unevaluated operation has a positive benefit if all operations were evaluated. Figure 2 shows the four metrics mentioned above during our experiments to generate the topology for AS 1239. From p_r and p_m we can see that the benefit of the operations are slowly changing, so it makes sense to utilize the priority queue instead of evaluating every operation; unless we evaluate a large portion of the operations, p_n is not small. However, a good overall effect may not have a small p_n , instead it can be a trade off between applying the very best operation and applying a reasonably good operation in each iteration.

6.3 Evaluating the Topologies Generated by our Approach

In this section we compare the topologies generated by our approach with the RocketFuel data. We first introduce the notation that will be used throughout this section, then outline the properties that we concern before showing the experimental results.

In this section, “RocketFuel” denotes the real network obtained from RocketFuel and fitted to our resolution, “OPT” denotes the network generated by our strict two-phase optimization approach, “I” stands for the incremental approach. “RealPoP” and “FLPoP” are the set of base PoPs used as the input to the second optimization phase, in which “Re-

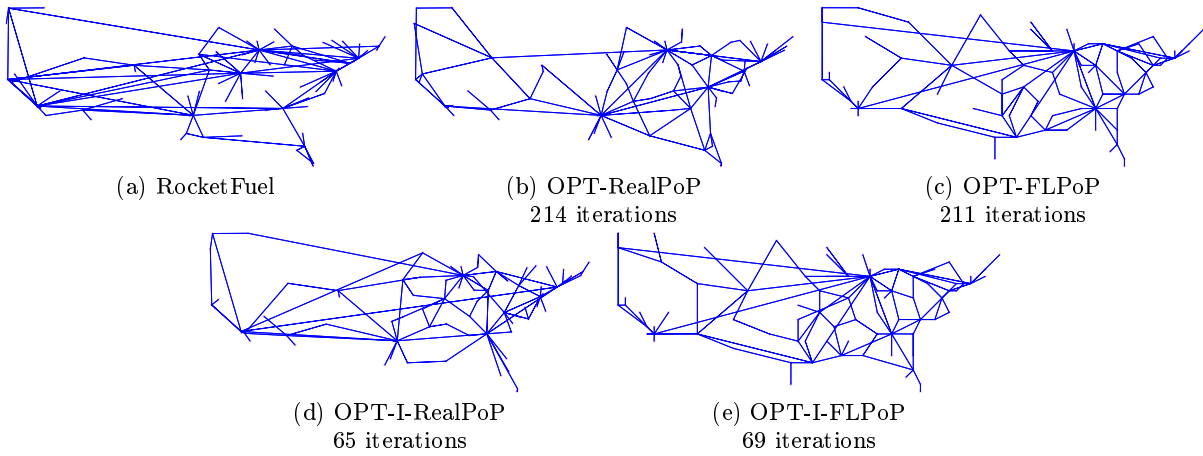


Figure 4: Measured and modeled network topologies for AS 7018 (82 PoPs)

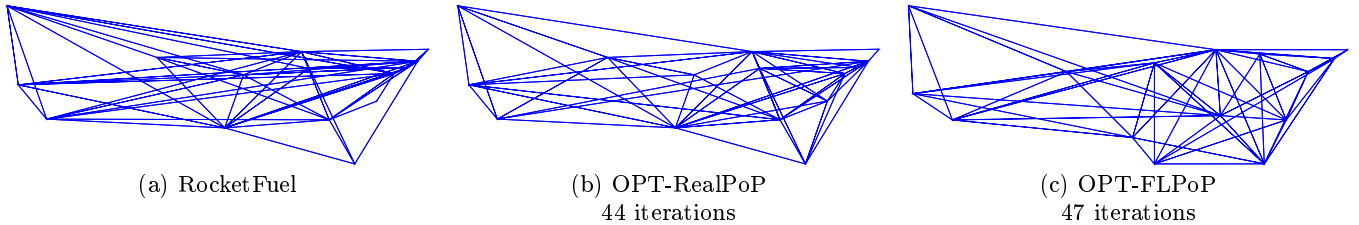


Figure 5: Measured and modeled network topologies for AS 1239 (16 PoPs)

alPoP” means the positions of PoPs from real networks, each annotated by the amount of customer traffic demands calculated according to our model, while “FLPoP” means the PoPs generated by the facility location process. We use a combination of acronyms for every situation, for instance, “OPT-I-FLPoP” means the result of the network generated by the incremental optimization approach, in which the positions of base PoPs are generated by facility location.

Our first question is: Are “RealPoP” and “FLPoP” significantly different in term of the customer traffic demands served by them? This can be answered by comparing the results of the first-phase optimization. Then moving forward to the second-phase optimization, we are ready to compare the topologies. First we compare the traditional graph-theoretic metrics, including degree distributions and multiple numerical summaries of the graph. Note that different from previous topology generation work, we generate annotated graphs, so traffic on edges and nodes are also of interest. Finally we discuss fault-tolerance properties.

6.3.1 Customer Traffic Demands Served by Base PoPs

From RocketFuel we get the locations of PoPs and distribute the customer traffic demands to those PoPs (the traffic demands are determined by the gridded population data as stated above). We compare the customer demands served by those PoPs against the ones obtained by solving the facility location problem. The comparison is shown in figure 3. From the figure we can see that the distribution of AS 7018 shows a reasonable similarity to the reality under our model while that of AS 1239 shows some differences. One plausible explanation is that the traffic demands out-

side the main region may influence the design of base PoP positions (note that the demands outside the main region are manually set and may be imprecise). Differences also likely reflect economic incentives or other design considerations not explicitly captured by our model.

6.3.2 Comparison of Graph-theoretic Properties

The generated topologies as well as the real networks are shown in Figures 4 and 5 for AS 7018 and AS 1239 respectively. To show the results clearly on a comparable scale, only the US-based region of each ISP is depicted in the figure. In each figure we also show the size of the network and the number of iterations for our optimization algorithm to converge. We can see that our results share similarities with the real networks in terms of the overall structures and the local layouts. However, there are more long links connecting large base PoPs in the real networks than in our results. Figures 6, 7 are comparisons of degree distributions. For the ease of comparison, in the plots of this section we do not show the results of the incremental optimization approach, and each plot shows the result of a single representative experiment. We can see that most data points differ from the real networks within a small range. There are two notable differences: one is in AS 7018, the number of 1-degree nodes in the real network is considerably larger than that in our results; the other is in our network of AS 1239 optimized from base PoPs generated by facility location, many nodes have more connections than the real network as well as the network optimized from PoP positions in the real network.

In Table 1 we compare some graph-theoretic metrics among real networks and our results. We can see that our topolo-

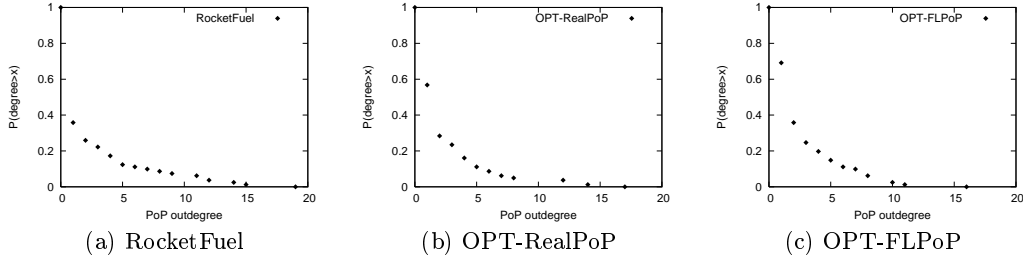


Figure 6: Degree distributions of topologies for AS 7018

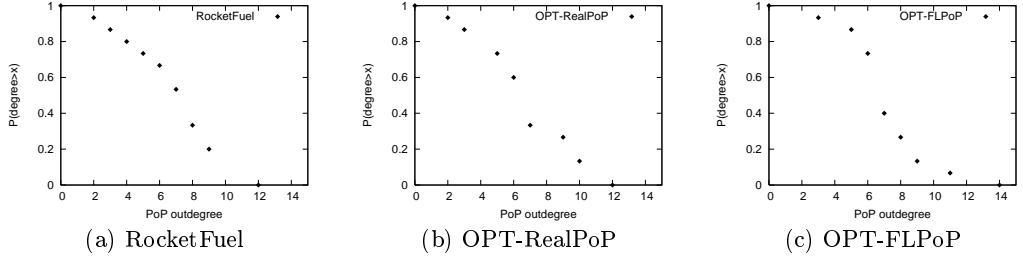


Figure 7: Degree distributions of topologies for AS 1239

AS	Network Type	Cost of G
7018	RocketFuel	45651.10
7018	OPT-RealPoP	40836.42(49.21)
7018	OPT-FLPoP	41782.82(168.37)
7018	OPT-I-RealPoP	41569.57(36.16)
7018	OPT-I-FLPoP	42798.22(29.09)
1239	RocketFuel	30789.76
1239	OPT-RealPoP	30061.48(34.01)
1239	OPT-FLPoP	30076.84(98.61)

Table 2: Network costs, as computed by our model.

gies have very similar density and distance metrics with the real networks. Furthermore, in table 2 we show the costs of the networks under our model. We can see that the costs of the real networks are higher than those of our networks. But for AS 7018, the difference is within 11%, and for AS 1239 the difference is within 3%.

6.3.3 Comparison of Traffic on Edges and PoPs

Another important metric of the annotated networks is the distribution of traffic on edges and PoPs. For every network we compute the all-pair shortest paths to get the amount of traffic going through each link and each PoP. In Figures 8, 9, 10 and 11 we show the traffic distribution on edges and on PoPs of AS 7018 and AS 1239 respectively. We can see that from the traffic distributions on edges, several edges in our optimized networks carry more traffic than the largest traffic an edge carries in the real network; while from the traffic distributions on nodes we can see that in the real networks the proportion of the nodes carrying relatively small amount of traffic is higher than that in our optimized networks.

6.3.4 Comparison of Fault-tolerance

Our optimization approach does not take fault-tolerance into consideration. However, the network generated by our approach has some routing redundancy. Here we also show some preliminary comparisons with respect to fault-tolerance, comparing our results to that of the RocketFuel dataset. In particular, in this section we consider two types of failures: link failure and node failure.

Based on our experiments, we observe two types of redundancy for tolerance of link failure: one is the physical redundancy, where there may be duplication of links between a pair of nodes; the other is structural redundancy which describes the fault tolerance even if all the links between a pair of nodes are disconnected. Here we study the structural redundancy. Let $DC(E)$ denote the set of edges of a graph $G = (V, E)$, such that removal of any one edge in $DC(E)$ disconnects the graph. There are potentially two types of edges in $DC(E)$ of an ISP topology: those that are at the periphery of the network, whose removal disconnects a single node; and those that are bridges, whose removal breaks the graph. The latter class of edges rarely arise in measured and modeled ISP topologies of significant size.

We study the following metrics that we define next: *worst-case edge cost* and *penalty of weighted latency*.

Definition 2. For a network $G = (V, E)$, we define the *worst-case edge cost* to be

$$p_c(G) = \frac{\max_{e' \in E \setminus DC(E)} \text{cost}(V, E - e') - \text{cost}(V, E)}{\text{cost}(V, E)}$$

Definition 3. The *weighted latency* of PoP v in graph G is defined as

$$w(G, v) = \sum_{v' \in V - v} d(v, v') \times t_{v, v'}$$

in which $d(v, v')$ is the length of the shortest path between v and v' , and $t_{v, v'}$ is an element in the traffic demand matrix as defined in section 2.

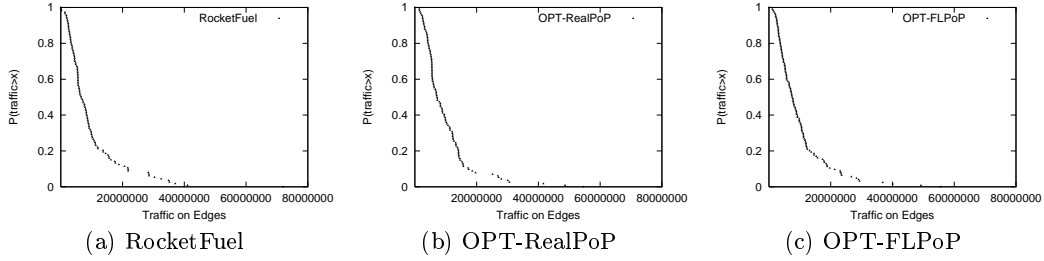


Figure 8: Modeled traffic distributions on edges of AS 7018

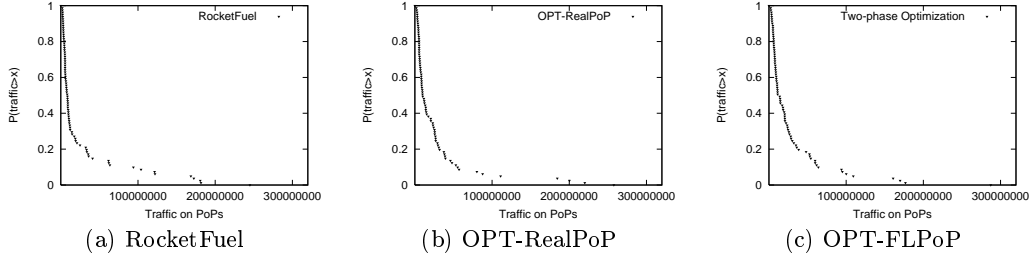


Figure 9: Modeled traffic distributions on PoPs of AS 7018

AS	network type	Number of Bridge Edges	$p_c(G)$
7018	RocketFuel	0	0.0357
7018	OPT-RealPoP	2	0.022
7018	OPT-FLPoP	2	0.023
7018	OPT-I-RealPoP	1	0.023
7018	OPT-I-FLPoP	2	0.034
1239	RocketFuel	0	0.0412
1239	OPT-RealPoP	0	0.0317
1239	OPT-FLPoP	0	0.0179

Table 3: Worst-case edge cost from link failure. As $p_c(G)$ on the basis of different number of bridge edges can not be aggregated, each value is for the single experiment which we used to show the distributions.

Definition 4. The weighted latency penalty for PoP v in graph G is

$$p_w(G, v) = \frac{\max_{e' \in E \setminus DC(E)} w((V, E - e'), v) - w(G, v)}{w(G, v)}$$

Of course, to study the metrics above, we need to consider the number of bridge edges as well. We show $p_c(G)$ along with the number of bridge edges in table 3. In figures 12 and 13 we show the distributions of weighted latency penalty.

For node failure, because of the change of traffic, it is hard to summarize the penalty of a node failure in term of cost or weighted latency. In our preliminary study, we define the worst-case traffic loss $p_t(G)$, which measures the percentage of all-pair traffic lost due to the node failure. The results for $p_t(G)$ are reported in table 4.

Definition 5. For a network $G = (V, E)$, we define the all-pair traffic $T(G)$ to be $\sum_{v \in V, v' \in V - v} t_{v, v'}$.

Definition 6. The worst-case traffic loss $p_t(G)$ is defined

as

$$p_t(G) = \frac{T(G) - \max_{v \in V} T(V - v, E - (v, v'))}{T(G)} \quad (v, v') \in E.$$

We can see that in the real network, if an arbitrary link is removed, all traffic demands can still be served while in our optimized network for AS 7018 this is not true, though the worst-case edge cost of our optimized network is lower; if an arbitrary node fails, the worst-case traffic loss of the real network is significantly smaller than that of our optimized network for AS 7018. For AS 1239 the fault-tolerance metrics of our results and the real networks show a good match — this may be the result of the rich connections of the networks.

We provide several possible explanations to the differences mentioned above. For the difference in the number of 1-degree nodes and traffic distributions on nodes, a possible reason is that in our model, the cost of a node is proportional to the sum of the traffic on the edges to which it connects, however, in practice, a PoP might have to accommodate traffic for exchange, so the real cost may be more than the one in our model. For the difference in traffic distributions on edges, a possible reason is the technological constraints of links: if the capacity of a link is limited, a duplicated link may be built, which introduces marginal cost. Also, in the design of real networks, many other factors have to be considered, for instance, fault tolerance and consideration of links built by other ISPs, and so on. Also, the networks optimized from the positions of real PoPs show a better similarity than those optimized from base PoPs generated by facility location. In fact, placing base PoPs to serve end users' traffic demands may need more considerations from multiple aspects other than just traffic demands. Nevertheless, even with this simplified model, we have shown that our approach can generate networks reasonably similar to the real networks.

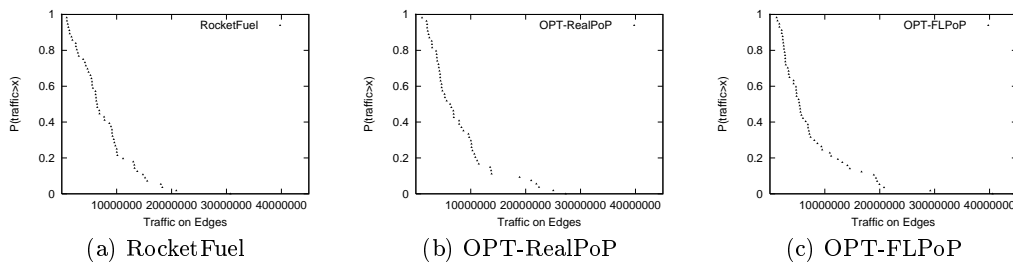


Figure 10: Modeled traffic distributions on edges of AS 1239

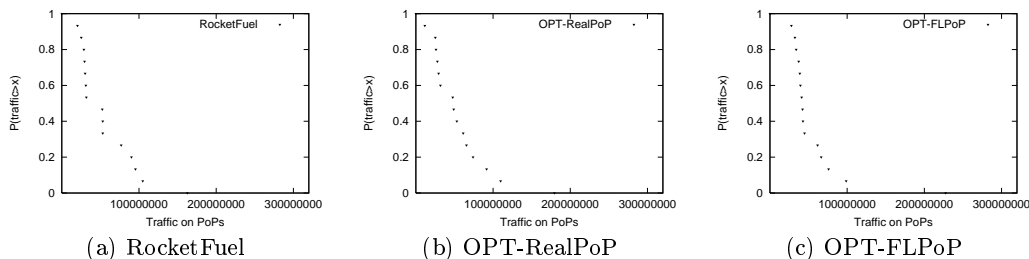


Figure 11: Modeled traffic distributions on PoPs of AS 1239

AS	Network Type	$p_t(G)$
7018	RocketFuel	0.1622
7018	OPT-RealPoP	0.316(0.026)
7018	OPT-FLPoP	0.2101(0.04)
7018	OPT-I-RealPoP	0.2996(0)
7018	OPT-I-FLPoP	0.1897(0)
1239	RocketFuel	0.2486
1239	OPT-RealPoP	0.2486(0)
1239	OPT-FLPoP	0.2479(0)

Table 4: Worst-case traffic loss from node failure

7. CONCLUSION

Our work treats the problem of building an ISP topology as a natural network design problem that attempts to capture the key practical constraints associated with deploying routers, high-speed links, and network PoPs. While the resulting optimization problem is NP-hard, we demonstrate that it is possible to apply results from the approximation algorithms literature along with various heuristics to prune the search space that can readily generate moderately large topologies. We demonstrate that the topologies are visually and quantitatively similar to topologies measured by RocketFuel, where our quantitative analysis considers network cost, topological characteristics of the network, as well as performance via sensitivity to failures.

We claim that with our approach we can generate and explain the Internet topologies better than those currently available. Compared with the generators trying to generate scale-free graphs, our explanatory model captures more practical aspects in network creation. Furthermore, the results in [28] suggest that the router-level node degrees are not consistent with the claims of scaling distributions found at the AS-level. With our model, annotated topolo-

gies can be generated, which is crucial to network research but not available in current topology generators and measured topologies. Another power of our model is that it can readily predict the topologies subject to the change of certain conditions.

Our future work will consider extending our model to expose more features of an ISP topology, and will consider game theoretic aspects of the more complex optimization process that arise when multiple competing ISPs design their networks in parallel.

8. REFERENCES

- [1] Gridded population of the world. <http://sedac.ciesin.columbia.edu/gpw>.
- [2] D. Alderson, J. Doyle, R. Govindan, and W. Willinger. Toward an optimization-driven framework for designing and generating realistic Internet topologies. *ACM Computer Communication Review*, 33(1), 2003.
- [3] V. Bala and S. Goyal. A non-cooperative model of network formation. *Econometrica*, 68:1181–1229, 2000.
- [4] A. Basu and J. G. Riecke. Stability issues in OSPF routing. In *Proceedings of ACM SIGCOMM*, San Diego, CA, 2001.
- [5] T. Bu and D. Towsley. On distinguishing between Internet power law topology generators. In *Proceedings of IEEE INFOCOM '02*, New York, NY, 2002.
- [6] H. Chang, S. Jamin, and W. Willinger. To peer or not to Peer: modeling the evolution of the Internet’s AS-level topology. In *Proceedings of IEEE INFOCOM '06*, Barcelona, Spain, 2006.
- [7] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. J. Shenker, and W. Willinger. The origin of power laws in Internet topologies revisited. In *Proceedings of IEEE INFOCOM '02*, New York, NY, 2002.
- [8] A. Fabrikant, E. Koutsoupias, and C. H.

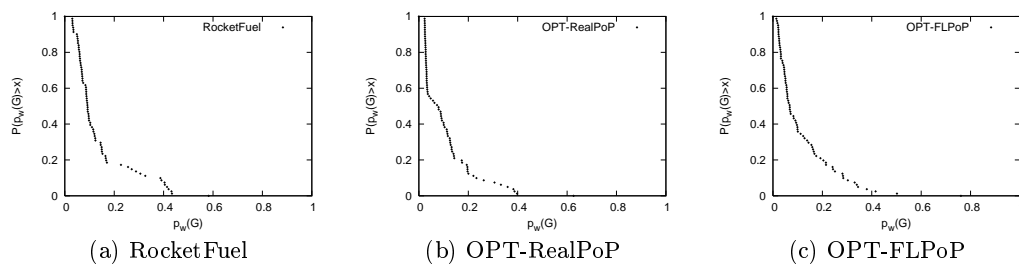


Figure 12: Weighted latency penalty for link failure in AS 7018

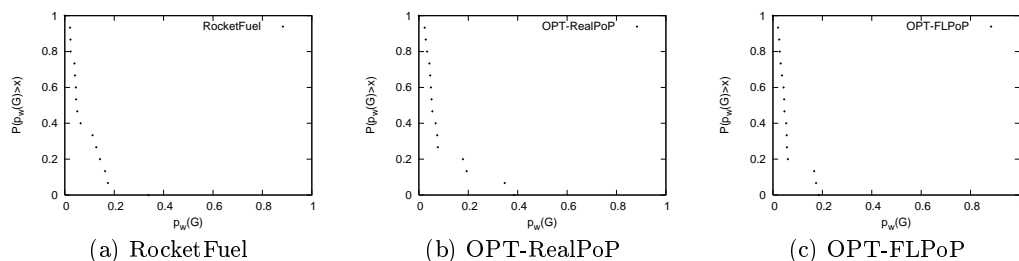


Figure 13: Weighted latency penalty for link failure in AS 1239

- Papadimitriou. Heuristically Optimized Trade-offs: a new paradigm for power laws in the Internet. In *Proceedings of ICALP '02: the 29th International Colloquium on Automata, Languages and Programming*, M laga, Spain, 2002.
- [9] A. Fabrikant, A. Luthra, E. Maneva, C. H. Papadimitriou, and S. Shenker. On a network creation game. In *Proceedings of PODC '03: the twenty-second annual symposium on Principles of distributed computing*, Boston, MA, 2003.
- [10] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, 2000.
- [11] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. Technical Report CSE-TR443-00, Department of EECS, University of Michigan, 2000.
- [12] M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. In *Proceedings of SODA '98: the Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, San Francisco, CA, 1998.
- [13] J. Kowalski and B. Warfield. Modeling traffic demand between nodes in telecommunications network. In *Proceedings of ATNAC '95: Australian Telecommunications networks and Applications conference*, Sydney, Australia, 1995.
- [14] A. A. Kuehn and M. J. Hamburger. A heuristic program for locating warehouses. *Management Science*, 9(4):643–666, 1963.
- [15] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first principles approach to understanding the internet's router-level topology. In *Proceedings of ACM SIGCOMM*, Portland, OR, 2004.
- [16] M. Mahdian and M. Pal. Universal facility location. In *Proceedings of ESA '03: the 11th Annual European Symposium on Algorithms*, Budapest, Hungary, 2003.
- [17] M. Mahdian, Y. Ye, and J. Zhang. Improved approximation algorithms for metric facility location problems. In *Proceedings of APPROX '02: Fifth International Workshop on Approximation Algorithms for Combinatorial Optimization*, Rome, Italy, 2002.
- [18] S. Martello and P. Toth. *Knapsack Problems: Algorithms and Computer Implementation*. Wiley, Chichester, England, 1990.
- [19] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITe: an approach to universal topology generation. In *Proceedings of MASCOTS '01: the 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, Cincinnati, OH, 2001.
- [20] A. Medina, I. Matta, and J. Byers. On the origin of power laws in Internet topologies. *ACM Computer Communication Review*, 30(2):18–28, 2000.
- [21] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: existing techniques and new directions. In *Proceedings of SIGCOMM*, Pittsburgh, 2002.
- [22] S. Melkote and M. S. Daskin. An integrated model of facility location and transportation network design. *Transportation Research*, 35(6):515–538, 2001.
- [23] H. Podnar and J. Skorin-Kapov. Genetic algorithm for network cost minimization using threshold based discounting. *Journal of Applied Mathematics and Decision Sciences*, 7(4):207–228, 2003.
- [24] P. Radoslavov, H. Tangmunarunkit, H. Yu, R. Govindan, S. Shenker, and D. Estrin. On characterizing network topologies and analyzing their impact on protocol design. Technical Report USC-CS-TR-00-731, Computer Science Department, University of Southern California, 2000.

- [25] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking*, 12(1):2–16, 2004.
- [26] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. In search of path diversity in ISP networks. In *Proceedings of IMC '03: the 3rd ACM SIGCOMM conference on Internet measurement*, Miami Beach, FL, 2003.
- [27] X. Wang and D. Loguinov. Wealth-based evolution model for the Internet AS-Level topology. In *Proceedings of IEEE INFOCOM '06*, Barcelona, Spain, 2006.
- [28] W. Willinger, D. Alderson, and L. Li. A pragmatic approach to dealing with high-variability in network measurements. In *Proceedings of IMC '04: the 4th ACM SIGCOMM conference on Internet measurement*, Taormina, Sicily, Italy, 2004.
- [29] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee. How to model an Internetwork. In *Proceedings of IEEE INFOCOM '96*, San Francisco, CA, 1996.
- [30] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. In *Proceedings of ACM SIGMETRICS '03: international conference on Measurement and modeling of computer systems*, San Diego, CA, 2003.