SIMROT: A Scalable Inter-domain Routing Toolbox

Ahmed Elmokashfi
Simula Research Laboratory

Amund Kvalbein
Simula Research Laboratory

Constantine Dovrolis
Georgia Institute of Technology

ABSTRACT

BGP plays a crucial role in the global communications infrastructure, and there is a strong need for tools that can be used for analyzing its performance under different scenarios. The size and complexity of the inter-domain routing system often makes simulation the only viable method for such studies. This work addresses the lack of a comprehensive toolbox to simulate BGP. It proposes a flexible topology generator that produces AS-level graphs which are annotated with business relationships. The second component of the toolbox is a light-weight BGP simulator that is capable of capturing routing dynamics, while scaling to network sizes of thousands of nodes. We employ our framework to investigate a set of what-if questions concerning the impact of different topology parameters on BGP dynamics. This investigation shows how our proposed framework can help in gaining important insights on inter-domain routing dynamics.

1. INTRODUCTION

The inter-domain routing system is arguably the largest deployed distributed system in the world, consisting of over 37000 Autonomous Systems (ASes) and 354k routable network prefixes [4]. This system is critically important for the global communications infrastructure; if it breaks down, hosts attached to different networks are no longer able to communicate with each other. BGP, the only deployed inter-domain routing protocol, has sustained tremendous growth over the past two decades. BGP is a simple path vector protocol, which gives operators a large degree of freedom in defining the policies that govern the best-path selection process. Operators are free to define their own specialized rules and filters, supporting complex relationships between ASes. This flexibility is arguably one of the main factors behind BGP's success. At the same time, the flexibility makes BGP challenging to configure and manage. There is a rich body of research showing how conflicting policies and protocol configurations can lead to delayed convergence, nondeterministic behavior, or even permanent oscillations in the routing plane [16, 17].

The importance of BGP and the complexity involved in its operation, make it important to have tools that can predict the behavior in scenarios involving different policies, protocol configurations and topologies. Efforts in this direction have different goals, ranging from characterizing the infrastructure to understanding the impact of failures, routing changes, and attacks on its stability and survivability.

BGP churn can be studied through measurements, modeling, or simulations. Simulations are often the only viable option for investigating different what-if scenarios related to the impact of topology growth on routing, new routing enhancements, and radical architectural changes. Simulations of inter-domain routing, however, require good models for the used Internet topology, routing event models, and the operation of the routing protocol. Since scale is one of the most important properties of inter-domain routing, it becomes important to be able to simulate networks of comparable size as the current Internet topology. Currently, there is a lack of an integrated framework that can accurately facilitate investigating BGP routing dynamics.

In this paper we propose SIMROT, a toolbox that consists of a topology generator (SIMROT-top) and a scalable routing simulator (SIMROT-sim) for studying BGP routing dynamics. SIMROT-top improves over the existing tools by generating AS-level topologies that are annotated with business relationships. The knobs of this generator are parameters with operational relevance in practice, such as the multihoming degree (MHD) of stubs versus transit providers, instead of abstract measures such as betweenness or assortativity. The properties of topologies generated using our model match reasonably those of inferred Internet AS-level topologies. SIMROT-sim, on the other hand, makes several simplifying assumptions by focusing only on capturing the control plane functionality of BGP, and leaving out the operation of the underlying TCP sessions. Hence, It is capable of capturing the exchange of routing updates, while scaling to network sizes of thousands of nodes. A benchmarking against the widely used SSFNET simulator [3] illustrates that our simulator produces similar results, while performing significantly better in terms of memory usage and simulation time. Finally, we illustrate the flexibility of our toolbox by investigating a set of what-if questions concerning the impact of different topology parameters on BGP dynamics. These investigations show that our proposed toolbox can give important insights on inter-domain routing dynamics[1].

The rest of the paper is organized as follows. In the next section, we discuss the different approaches for studying BGP, review the related work, and describe briefly our toolbox. In Sec. 3, we describe and validate SIMROT-top. In Sec. 4, we present and benchmark SIMROT-sim. In Sec. 5 we use our toolbox to investigate the impact of several topology parameters on BGP churn, and draw our conclusions in Sec. 6.

[1] SIMROT-top and SIMROT-sim are available at http://simula.no/department/netsys/software/

2. BACKGROUND AND RELATED WORK

Different approaches have been used to study the inter-domain routing system. Active measurements can be performed through injecting changes (e.g. announcing and withdrawing network prefixes) at a certain location in the network and observing their impact at a set of vantage points. The BGP beacons project [27] and RIPE RIS beacons [1] are examples of such approach. However, the fact that the Internet is a mission critical infrastructure limits the flexibility and extent of active measurements. For example, experimenting with operational prefixes (i.e. withdrawing or re-announcing them) will inevitably affect traffic and thus users' experience. On the other hand, passive measurements can be performed by logging BGP routing tables, updates and logs from operational routers. The RouteViews [2] and RIPE RIS [33] projects are pioneering efforts in this direction. Passive measurements can assess and measure the current status of the routing system. However, we can not use it for evaluating new enhancements and protocol extensions. Furthermore, the number and location of used vantage points can influence insights gained through passive measurements and make them only representative to the studied setup. Note that this limitation is inherent, since it is extremely difficult to place a vantage point at each network in the Internet.

Mathematical modeling can also be used for studying and characterizing BGP. Nevertheless, the complexity of BGP and large Internet-like topologies make it difficult to create a tractable and useful mathematical model. Attempts in this direction have been limited to regular topologies with a simplified BGP operation model [37].

Simulating BGP is sometimes the only option to circumvent the limitations of measurements and mathematical modeling. However, there are several pitfalls in making realistic BGP simulations. First, a representative topology model that captures the properties of the Internet AS-level graphs is needed. The model should be able to capture reasonably well the observed properties of the AS-level graph (e.g. power law degree distribution, strong clustering, constant average path length). In addition, it should be able to annotate inter-AS links with business relationships. Second, a reasonable implementation of BGP and a clear description of routing changes and events are crucial for the correctness of the results. Essentially, there is a trade-off between the computational and storage complexity of the simulation model and the level of details that is captured. The computational complexity is a function in the size of the simulated network, the level of topological details, and protocol implementation. Ideally, a simulation model should be able to simulate topologies that are comparable to the current Internet in terms of size.

Topology Generators: Generating graphs that capture the observed properties of the AS-level topology has been a subject of much research in the past decade. In general, the proposed topology generators focused on capturing the abstract properties of the AS-level graph.

Early topology generators such as BRITE [28], Inet [36], and PLRG [5] tried to reproduce the node degree distribution of the AS-level graph. These generators managed to reproduce the node degree distribution reasonably; however, they did not succeed in capturing other abstract properties such as the clustering coefficient and the joint node degree distribution. These limitations are expected since this class of generators assumes that the node degree distribution is a one dimensional independent variable. Consequently, the degree of a node in such topologies is unrelated to the properties of its neighboring nodes. Later, Mahdevan et al. proposed another approach to overcome these limitations by using a group of distributions that capture the correlations of degrees among a set of connected nodes (i.e. a subgraph of the AS-level topology) [26]. They further employed this approach to generate re-scaled topologies of different sizes [24].

The above mentioned topology generators fulfilled reasonably their design purposes. Still, an important limitation is that they consider the AS-level topology as a generic collection of links and nodes. In fact, nodes and links in the Internet are far from being generic. The geographical presence, size, and connectivity of ASes differ depending on their role and business model (e.g. transit providers, content providers). More importantly, inter-AS links are different based on business relationships between the involved ASes. These relationships control and regulate inter-domain routing, and therefore, generating topologies that are annotated with them is crucial.

Several other efforts focused on generating topologies that are annotated with business relationships. The GHITLE topology generator [9] used a set of simple design heuristics to produce such topologies. However, it did not account for the subtle differences between different node types and did not model the number of settlement-free peering (p2p) links in a realistic way. Furthermore, the impact of geographic presence on both peer and provider selection was not captured. The work by Dimitropoulos et al. [11] proposed generating re-scaled annotated topologies by generalizing the work in [24]. He et al. proposed HBR [19] as a method for generating annotated graphs of various sizes through sampling them from larger inferred AS-level topologies. The last two approaches generate topologies in a top-down fashion by starting from an inferred AS-level topology and work to reproduce the measured abstract graph properties. Therefore, they do not provide any flexibility for controlling different topological characteristics.

Most of the existing topology generators focused on generating AS-level topologies rather than router-level topologies. An important reason behind this is that we have a better understanding for the Internet topology at the AS level. In fact, router connectivity varies across networks and is dependent on choices taken locally at each network. In addition, it is constrained by many factors such as the maximum possible degree per router (i.e. maximum number of interfaces), the number of the points of presence a network has, and different engineering decisions taken in order to optimize the topology. An extensive discussion about this can be found in [23]. Several measurement studies aimed at inferring router-level topologies in different networks using traceroutes. Notable example of such approaches are Rocketfuel [34] and ARK [8]. However, inference techniques suffer from limitations that are caused by their sampling nature [22] and traceroutes failure in resolving router aliases [35].

For addressing the lack of router-level topology generators, Quoitin et al. [30] proposed IGen as a generator that builds topologies through considering network design heuristics. IGen is a promising step in the direction of building realistic router-level topologies. However, since it depends on many heuristics, it is not clear how well the generated topologies are able to match known properties of the AS-level graph.

Simulators: Existing inter-domain routing simulators fall into two broad categories. Either they only calculate steady state routes, and do not capture routing dynamics [31], or they include a detailed model of each BGP session (e.g. underlying TCP connections). [3, 12] simulators that belong to the second category are suitable for studying questions that involve both routing and data forwarding. However, since the level of detail limits scalability; they do not scale to network sizes in the order of todays AS-level Internet topology.

In this paper we present SIMROT as a flexible toolbox for simulating BGP. SIMROT consists of a topology generator that captures business relations, and an event-driven simulator that scales to network sizes comparable to the AS-level topology. In the subsequent sections we describe and evaluate SIMROT.

3. TOPOLOGY MODEL

In this section, we first describe some key properties that characterize the AS-level Internet topology. These properties have been stable for the last decade [10], and we believe that they will remain valid in the foreseeable future. We then describe a model that allows us to construct topologies with different configurable properties while still capturing these key properties. The current version of our topology generation model is limited to AS-level graphs. However, it can be extended to produce router-level graphs if there exist reasonable models. A possible scenario would be combining approaches such as IGen [30] and our model for achieving this goal.

3.1 Stable topological properties

The AS-level Internet topology is far from a random graph. Over the past decade it has experienced tremendous growth, but the following key characteristics have remained constant:

1. *Hierarchical structure.* On a large scale, the nodes in the Internet graph form a hierarchical structure. By hierarchical we mean that customer-provider relationships are formed so that there are normally no provider loops, where A is the provider of B who is the provider of C who again is the provider of A.

2. *Power-law degree distribution.* The degree distribution in the Internet topology has been shown to follow a truncated power-law, with a few very well-connected nodes, while the majority of nodes have only few connections [13]. The well connected nodes typically reside at the top of the hierarchy.

3. *Strong clustering.* The ASes in the Internet are grouped together in clusters, with ASes in the same cluster more likely to be connected to each other. One reason for this clustering is that networks operate in different geographical areas.

4. *Constant average path length.* Measurements show that in spite of a tremendous growth in the number of nodes, the AS-level path length has stayed virtually constant at about 4 hops for the last 10 years [10].

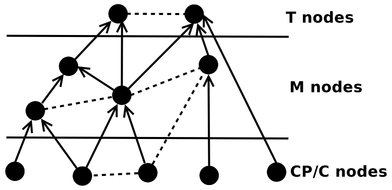


Figure 1: Illustration of network based on our topology model.

3.2 SIMROT-top

Next, we describe a flexible model for generating topologies that captures the above properties of the AS-level graph. Several design choices and parameters in our topology generator were guided by the measurements in [10].

A node in our model corresponds to an AS, and we use four types of nodes. At the top of the hierarchy are the tier-1 (T) nodes. T nodes do not have providers, and all T nodes are connected in a clique using peering links. Below the T nodes, we have the mid-level (M) nodes. All M nodes have one or more providers, which can be either T nodes or other M nodes. In addition, M nodes can have peering links with other M nodes. At the bottom of the hierarchy, we have two different types of stub nodes. We distinguish between customer networks (C) and content providers (CP). In this context, CP nodes would include content provider networks, but also networks providing Internet access or hosting services to non-BGP speaking customers. In our model, the difference between C and CP nodes is that CP nodes can enter peering agreements with M nodes or CP nodes, while C nodes do not have peering links. Figure 1 shows a generic network of the type described above. Transit links are represented as solid lines with arrowheads pointing towards providers, while peer-to-peer links are dotted.

To capture clustering in our model, we introduce the notion of *regions*. The purpose of regions is to model geographical constraints; networks that are only present in one region are not allowed to connect with networks that are not present in the same region. In our model T nodes are present in all regions. 20% of M nodes and 5% of CP nodes are present in two regions, the rest are present in only one region. C nodes are only present in one region.

We generate topologies top-down in two steps. First we add nodes and transit links, then we add peering links. The input parameters n_T, n_M, n_{CP} and n_C decide how many of the n nodes belong to each node type, respectively. First, we create a clique of T nodes. Next, we add M nodes one at a time. Each M node connects to an average of d_M providers, uniformly distributed between one and twice the specified average. M nodes can have providers among both T and M nodes, and we use a parameter t_M to decide the fraction of providers that are T nodes. M nodes can only select providers that are present in the same region. M nodes select their providers using preferential attachment, which gives a power-law degree distribution [7].

We then add the CP and C nodes, which have an average number of providers d_{CP} or d_C, respectively. CP and C nodes can select T nodes as providers with a probability t_{CP} and t_C, respectively. Just like the M nodes, C and CP nodes select their providers using preferential attachment.

When all nodes have been added to the topology, we add peering links. We start by adding p_M peering links to each

	Meaning	Example
n_T	Number of T nodes	$4 - 6$
n_M	Number of M nodes	$0.15n$
n_{CP}	Number of CP nodes	$0.05n$
n_C	Number of C nodes	$0.80n$
d_M	Avg M node MHD	$2 + 2.5n/10000$
d_{CP}	Avg CP node MHD	$2 + 1.5n/10000$
d_C	Avg C node MHD	$1 + 5n/100000$
p_M	Avg M-M peering degree	$1 + 2n/10000$
p_{CP-M}	Avg CP-M peering degree	$0.2 + 2n/10000$
p_{CP-CP}	Avg CP-CP peering degree	$0.05 + 5n/100000$
t_M	Prob. that M's provider is T	0.375
t_{CP}	Prob. that CP's provider is T	0.375
t_C	Prob. that C's provider is T	0.125

Table 1: Topology parameters

M node. As for the provider links, p_M is uniformly distributed between zero and twice the specified average. M nodes select their peers using preferential attachment, considering only the peering degree of each potential peer. Each CP node adds p_{CP-M} peering links terminating at M nodes, and p_{CP-CP} peering links terminating at other CP nodes. CP nodes select their peers among nodes in the same region with uniform probability. Importantly, we enforce the invariant that a node must not peer with another node in its customer tree. Such peering would prey on the revenue the node gets from its customer traffic, and hence such peering agreements are not likely in practice.

3.3 Internet-like topologies

Next, we illustrate how to configure our topology generator for producing graphs that resemble the growth of the Internet over the last decade. The sample configuration parameters are inspired by measurements of the evolution of the Internet topology over the last decade [10]. The growth is characterized by a slow increase in the MHD of stub nodes, and a faster growth in the MHD and the number of peering links at middle nodes. In this sample configuration we use 5 regions, containing one fifth of all nodes each. Table. 1 gives the parameter values for the sample configuration. Note that n in Tab. 1 is the total number of nodes in the graph.
Validation: We validate that the generated topologies capture the four stable properties of the Internet topology discussed in Sec 3.1, and compare some properties of the generated graphs to inferred Internet topologies. We generate topologies of sizes 5000 and 10000 nodes respectively, and compare against two inferred AS-level topologies of sizes 3247 and 17446 nodes. The smaller topology is provided by Dhamdhere and Dovrolis [10] and it is based on RouteViews [2] and RIPE [33] BGP routing tables from January to March 1998. The second inferred topology is provided by Mahadevan et al. [25] and based on RouteViews BGP routing tables from March 2004. Note that the inferred topologies miss a large fraction of peering links, which distorts their characteristics quantitatively [10]. Therefore, our aim is that our topologies match the major topological properties of the Internet qualitatively rather than quantitatively.

Hierarchical structure. This is trivially fulfilled through the way we construct the topologies.

Power-law degree distribution. Figure. 2(a) shows the CCDF of the node degree on a log-log scale. We observe that our model captures the power-law scaling of the node degrees reasonably well, and is comparable to that of the inferred Internet topologies. The use of preferential attachment when selecting which nodes to connect to gives the observed power-law degree distribution [6].

Strong clustering. We measure the local clustering (or clustering coefficient) of each node in a topology. The local clustering of a node is defined as the ratio of the number of links between a node's neighbors to the maximum possible number of such links (i.e. a full clique). Hence, the local clustering measures how well connected a node's neighborhood is. Figure. 2(b) reports the average local clustering, across all nodes of the same degree, as a function of node degree. To keep the figure readable, we plot results for only two topologies (the other pair of topologies show similar results). Our model matches qualitatively the trends seen in the inferred topologies: first, local clustering decreases with the node's degree, and second, the clustering versus degree relation follows a power-law. It should be noted however that our model produces lower clustering than the inferred Internet topologies.

Constant average path length. The average path length in topologies produced by our model is constant at around four hops as the network grows from 1000 nodes to 10000 nodes. This matches closely the average path length in the inferred Internet topology at least since 1998 [10].

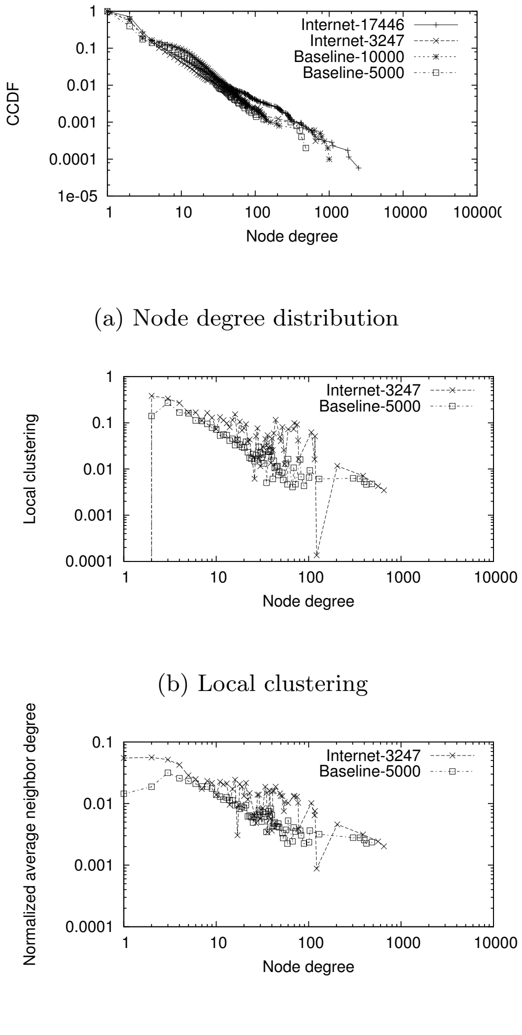
We also investigate the *average neighbor connectivity* [25], which has been difficult to capture by existing topology generators [18]. The average neighbor connectivity of a node is simply the average degree of its neighbors. This metric relates to the assortativity of a graph. It measures whether a node of a certain degree prefers to connect with higher or lower degree nodes. Figure. 2(c) shows the average neighbor connectivity as a function of the node degree. We normalize the average neighbor connectivity by the maximum possible value (the total number of nodes in the graph - 1), in order to compare topologies of different sizes. Our model gives an average neighbor connectivity that matches well the inferred Internet topologies, with smaller degree nodes having a higher average local connectivity than the higher degree nodes (referred to as negative assortativity).

The aforementioned validations illustrate that our topology generation model can reasonably produce graphs that match several important properties of the measured AS-level topology. In the next section, we present our BGP simulation model.

4. SIMROT-SIM

Simulations of any system of the size and complexity of inter-domain routing require several simplifying assumptions based on the goals of the simulations. In this section, we present our simulator "SIMROT-sim" and describe the assumptions and choices we make in its development.

SIMROT-sim is a discrete event simulator that is capable of capturing the exchange of routing updates and hence, simulate BGP dynamics. Furthermore, it is able to scale to network sizes of several thousands ASes.



(a) Node degree distribution

(b) Local clustering

(c) Average neighbor connectivity

Figure 2: Validating SIMROT-top

In order to realize the aforementioned scalability we make two key simplifying assumptions. Firstly, we model a BGP session between two nodes as a logical variable that is either established or not, and thus ignore the underlying TCP nature of the session. This choice enhances the scalability of our simulator since we abstract the TCP details and all involved overhead and signaling (e.g. sessions KEEPALIVE messages). We argue that this simplification does not have an impact since we only simulate the operation of BGP. The details of BGP sessions are important only when studying the interaction between data plane and control plane (e.g. the impact of data traffic on the stability of BGP sessions).

The second assumption is that we model each AS as a single node, and connections between two neighboring ASes as a single logical link. This implies that we do not capture routing effects within an AS, introduced by iBGP or interactions with IGP routing protocols (e.g., hot-potato routing). We have made this simplification to reduce the complexity of our simulations. In addition, intra-AS topologies vary between ASes, and there is no reasonable model that can be used to reproduce them. The lack of this model is partly caused by the reluctance of network operators about revealing internal details of their networks. Note that one possibility to make the simulation of iBGP topologies computationally feasible is to simulate them at a PoP level rather than at a router level. However, this also requires a reasonable model for intra-AS topologies.

SIMROT-sim simulates policy-based routing, with the use of MRAI timers to limit the frequency with which a node sends updates to a neighbor. By “policies”, we refer to a configuration where relationships between neighboring ASes are either peer-to-peer or customer-provider. We use normal “no-valley” and “prefer-customer” policies. Routes learned from customers are announced to all neighbors, while routes learned from peers or providers are only announced to customers. A node prefers a route learned from a customer over a route learned from a peer, over a route learned from a provider. Ties among routes with the same local preference are broken by selecting the route with the shortest AS path, then based on a hashed value of the node IDs.

By “MRAI” or “rate-limiting”, we refer to a configuration where two route announcements from an AS to the same neighbor must be separated in time by at least one MRAI timer interval. We use a default MRAI timer value of 30 seconds. To avoid synchronization, we jitter the timer as specified in the BGP-4 standard. According to the BGP-4 standard [32], the MRAI timer should be implemented on a per-prefix basis. However, for efficiency reasons, router vendors typically implement it on a per-interface basis. We adopt this approach in our model. We follow the MRAI implementation recommended in the most recent RFC (RFC4271) [32], which specifies that both announcements and explicit withdrawals should be rate-limited. Note that the value of the MRAI is configurable in SIMROT-sim.

Figure. 3 shows the structure of a node in our simulator. A node exchanges routing messages with its neighbors. Incoming messages are placed in a FIFO queue and processed sequentially by a single processor. The time it takes to process an update message is uniformly distributed in a user defined range. Each node maintains a table with the routes learned from each neighbor, we call these tables Adjacent-RIB Ins (Adj-RIB-Ins). Upon receiving an update from a neighbor, a node will update this table, and re-run its decision process to select a new best route. The new preferred route is then installed in the forwarding table and announced to its neighbors, the forwarding table is called the Local-RIB (Loc-RIB). For each neighbor, we maintain an export filter that blocks the propagation of some updates according to the policies installed in the network. Outgoing messages are stored in an output queue until the MRAI timer for that queue expires. If a queued update becomes invalid by a new update, the former will be removed from the output queue. We further introduce a set of interrupt messages to signal events such as failures and restorations of various components (e.g. links, nodes, sessions) to the affected nodes, which consequently trigger BGP updates as a response to the signaled change.

There are two factors that determine the scalability of SIMROT-sim. The first one is the state that each node maintains (i.e. routing table), which decides memory requirements. We design a scalable data structure for storing routing tables in SIMROT-sim that minimizes memory requirements by removing the redundancy in the RIBs entries shared by many nodes. For example if a node X and a node Y share the path $\{A, D, F\}$ to a certain destination prefix p,

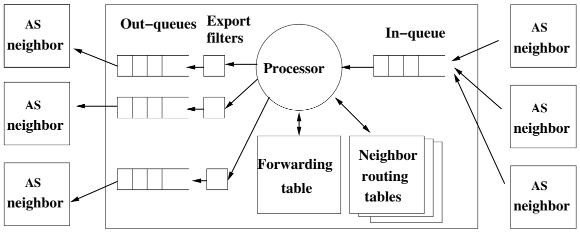


Figure 3: Model for a node representing an AS.

it will be more efficient to store a single entry for this path in the memory and keep two pointers at X and Y to it.

This approach is implemented by maintaining a global tree data structure that represents a global routing information base shared by all nodes in the network (Global-RIB). The tree has a single root which is used to maintain the structure of the tree. The root has n children (i.e. n is the number of the ASes in the network) each one represents a node in the network and is labeled with the corresponding AS number. When an AS X announces a prefix p, it sends the AS_PATH information as a pointer to the tree node labeled as X at level-1 of the Global-RIB. A neighbor of X performs two tasks when receiving this pointer. First it determines the actual AS_PATH by backtracking from the tree node that the pointer refers to in an upward direction until it reaches the root of the Global-RIB. Second it creates a new node that is labeled with its AS number, and will be added as a child to the tree node that the pointer refers to. Furthermore, it will include a pointer to the newly added node above instead of the full AS_PATH when it announces p to its neighbors. This means that the number of children a tree node has is equal to the number of distinct ASes that appear in the AS_PATHs that lead to it.

For the example network shown in the left panel of Fig. 4, assume that AS 1 is announcing a destination prefix p and it has AS2 as an immediate neighbor. When AS 1 announces p to AS 2 it sends the AS_PATH information as a pointer to node 1 in the tree. AS 2 keeps this reference in its routing table and adds a new child for node 1 in the tree labeled with its AS number (i.e. 2). When AS 2 announces p to its neighbors 3 and 5, it just sends the reachability information as a reference to the newly added tree node (i.e. tree node 2). Furthermore, 3 and 5 will back track from the tree node 2 upwards until they reach the root in order to extract the corresponding AS_PATH(i.e. {2,1}). The corresponding GRIB data-structure is illustrated in the right panel of Fig. 4.

To sum up, every AS in SIMROT-sim keeps a table of pointers instead of full AS_PATHs. Each entry in this table points to a node in the Global-RIB. The respective AS_PATH is determined by backtracking from that node towards the root of the tree. In addition, if multiple ASes share the same next hop to a prefix, they will all keep pointers to the same node in the Global-RIB, and thus remove redundancy.

The second factor that can potentially limit the scalability of our simulator is the number of enqueued BGP updates for processing. The impact of this factor is more evident during the initial convergence phase of the simulation. In this phase all prefixes that are part of the simulation are announced by their owners, which results in exchanging a large number of updates, and thus performing many decision process operations.

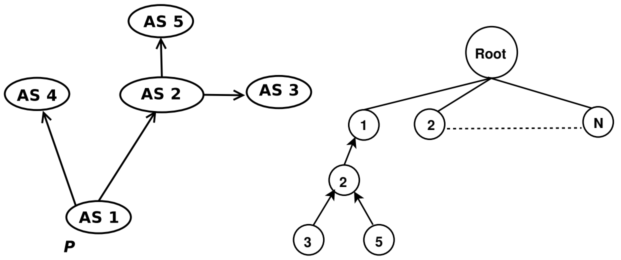


Figure 4: Example topology (left), and RIB data structure (Right)

Instead of simulating the exchange of BGP updates during the initial convergence phase, we implement a routing solver that computes for each node its steady-state reachability information and installs the computed entries in the respective routing tables. Our routing solver working principle is similar to that of C-BGP [31]. This optimization allows us to reduce the required resources for the initial convergence phase. After performing the initial convergence, one can choose to proceed with simulating various routing events (e.g. a link failure).

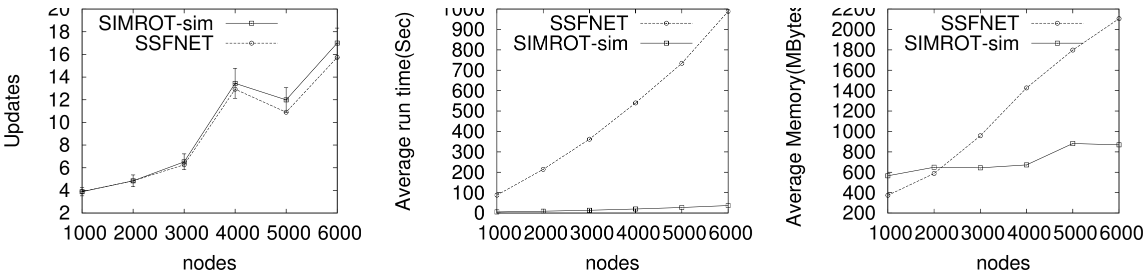
In the rest of this section we validate and evaluate the performance of SIMROT-sim and compare it to that of SSFNET. SSFNET is a large scale discrete-event simulator that is often regarded as the state-of-the-art tool for simulating BGP.

4.1 Performance Evaluation

For validating the operation of SIMROT-sim and comparing its performance with SSFNET, we generate six topologies in the range between 1000 and 6000 nodes using the example configuration described in Sec. 3.3. We limit ourselves to the aforementioned network sizes to ensure that SSFNET can handle our simulations in a scalable way. Then in each topology we simulate the withdrawal of a prefix from a C-type node. The experiment is repeated for 100 different C nodes, and the number of received updates is measured at every node in the network. We record the execution time and memory requirements of each simulation run. We have performed these experiments on a Dell machine (quad core Intel Xeon CPU 3.00 GHZ, 4GB RAM).

Simulation results. The goal of this comparison is to determine whether SIMROT-sim is able to simulate BGP dynamics in an accurate manner. Our main metric in the average number of updates received at a T node after withdrawing a prefix from a C-type node. Figure 5(a) shows the results of SIMROT-sim and SSFNET. The vertical bars the width of the confidence interval for SIMROT-sim results at a 99% confidence level. We observe that the results of the two simulators match well. The slight differences can be explained by the fact that each simulator uses a different random number generator. The deviations, however, are still within the calculated confidence intervals. Further the results of both simulators do not show a monotonic increase in the average number of updates due to differences between topologies; the topologies are generated as snapshots of certain sizes rather than in an evolving manner.

Execution time. We record the time each simulator takes to simulate one prefix failure event. We then average over the 100 runs. Figure 5(b) shows the average execution time. The measurement reflects clearly that SIMROT-sim execu-



(a) Simulation results (b) Average time per simulation run (c) Memory requirements

Figure 5: Evaluating SIMROT-sim

	Meaning	Value	Variation
d_M	Avg M node MHD	4.5	1 to 7.5
d_{CP}	Avg CP node MHD	3.5	-
d_C	Avg C node MHD	1.5	1 to 7.5
p_M	Avg M-M peering degree	3	-
p_{CP-M}	Avg CP-M peering degree	2.2	-
p_{CP-CP}	Avg CP-CP peering degree	0.55	-
t_M	Prob. that M's provider is T	0.375	0.125 to 0.75
t_{CP}	Prob. that CP's provider is T	0.375	0.125 to 0.75
t_C	Prob. that C's provider is T	0.125	0 to 0.3125

Table 2: Experiment parameters

tion time is significantly lower than that of SSFNET. The difference can reach up to two orders of magnitude. For example SSFNET takes about 1000 seconds to simulate the above described event in a topology of 6000 nodes, while SIMROT-sim takes around 35 seconds. The large difference can be attributed to the simplifying choices we have made. By avoiding simulating the initial convergence phase and ignoring the underlying TCP nature of the session and BGP KEEPALIVE messages we reduce the overall number of simulation events. However, SSFNET captures these details which means that the number of simulation events that are executed by our simulator is lower. Furthermore, update messages in SIMROT-sim do not include IP and TCP headers. The presence of these headers increases the memory requirements and the execution time; headers need to be created, added, and processed.

Memory requirements. We also measure the memory requirements of each simulator per each run. We then average over the 100 runs. The average memory requirements is illustrated in Fig. 5(c). The memory requirements in SIMROT-sim is characterized by a slow increase (600 to 800 MBytes). On the contrary, the memory requirements of SSFNET has increased significantly between 400 MBytes and 2.1 GBytes (i.e. an increase of 400%). The slow increase in SIMROT-sim can be attributed to the simple BGP model that it uses, and the Global-RIB data structure explained above. This data structure minimizes memory requirements by removing the redundancy in the RIBs entries shared by many nodes.

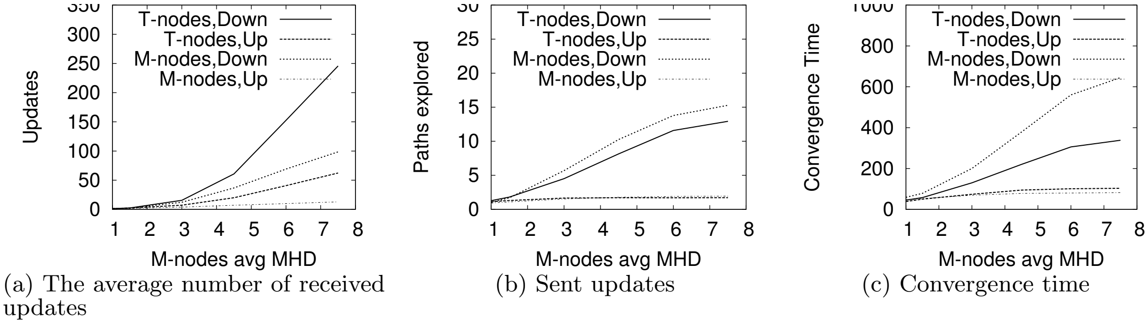
The above presented validation and performance evaluation show that our simulation model can accurately simulate BGP dynamics in a scalable way. In the next section we employ this BGP simulation framework in exploring a set of what-if questions concerning the impact of different topological parameters on BGP scalability with respect to churn.

5. SIMULATING WHAT-IF SCENARIOS

The evolution of the Internet AS-level topology is a complex process that is driven by many factors (e.g. economy, traffic demands, geography, technology). Topology evolution influences routing scalability and stability since it impacts BGP dynamics and convergence time. The long term impact of different topological growth models can only be studied using simulations. In this section, we look at the impact of two different topological properties on BGP dynamics and convergence. We consider a simple event where a network prefix of a stub AS is withdrawn and re-announced. This is the most basic routing event that can take place in the Internet, and at the same time the most radical; these changes must be communicated to all other nodes. The experiment is repeated for 100 different C nodes.

First, we study the impact of multi-homing degree at the core and the periphery of the network on routing. Measurements have shown a rapid increase in multihoming at the core of the Internet [10]. This increase comes from networks' desire to increase resilience through having a diverse set of upstream providers. Second, we investigate how the depth of the hierarchy can affect routing. Several measurement studies lately observed a decrease in the depth of the AS-level hierarchy [21, 15]. The observed change is attributed to the rise of content providers (e.g. Google) which are willing to enter in a large number of settlement-free peering agreements.

We perform our investigation by varying the corresponding parameters in our topology model. The network size n is fixed at 5000 nodes. Furthermore, the values of the remaining parameters are set to match our current knowledge of the AS-level topology [10], Tab. 2 gives the parameter values. For studying the effect of core multihoming, we keep all parameters fixed and vary d_M in the range between 1 and 7.5. The same is done when investigating the impact of edge multihoming, but here we vary d_C in the same range. On the other hand, we look into the depth of the hierarchy by varying t_M , t_{CP}, and t_C in the ranges [0.125,0.75], [0.125,0.75], and [0,0.3125] respectively. These settings reflect an increasing probability in choosing a T node as a provider by other types of nodes, and consequently reducing the depth of the hierarchy.

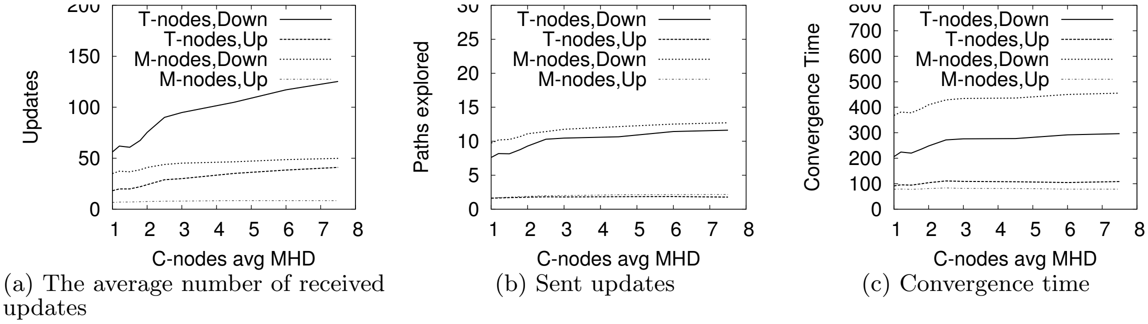


(a) The average number of received updates

(b) Sent updates

(c) Convergence time

Figure 6: Growing core multihoming



(a) The average number of received updates

(b) Sent updates

(c) Convergence time

Figure 7: Growing edge multihoming

We use three metrics to study the interaction between the above presented topology properties and BGP under the specified failure scenario. The first two metrics are *the average number of received and sent updates* by a T or an M node during the down phase (i.e. the phase the follows the prefix withdrawal) and up phase respectively. These metrics reflect the number of routing updates a node needs to process and the actual number of routing changes. The third metric is the average convergence time of the network, which is defined as the period between the occurrence of the event and the last seen update.

5.1 The effect of core multihoming

The plots in Fig. 6 illustrate our three metrics when varying the average multihoming degree at the core. All three metrics increase at both T and M nodes as we increase the average multihoming degree of M nodes from 1 to 7.5. The growth is significantly higher during the down phase. For instance, the number of received updates during the down phase grows by a factor 174.5 at T nodes, while it grows by a factor 45 during the up phase. During the down phase nodes explore a number of paths before they receive the final withdrawal, and hence prolonging the convergence time. During the up phase, on the other hand, nodes stop sending updates after receiving the most preferred path.

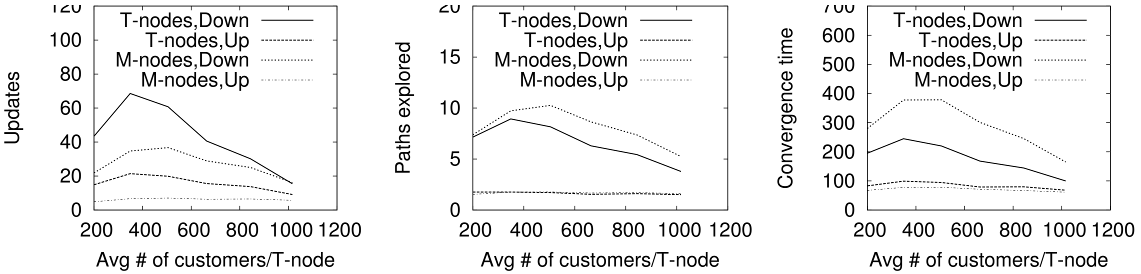
The number of updates a T node receives after a C-event depends on three factors. First, the number of customers a T node has (i.e. potential update sources). Second, the probability that a customer of a T-node has a customer route to the affected prefix. This factor translates in the graph theory terminology to M nodes' betweenness centrality. Finally, the average number of messages a customer sends during the convergence process, this factor is tightly coupled with the convergence time. The three factors mentioned above increase as we increase the core multihoming degree. The number of customers a T node has increases because M nodes have more providers. Furthermore, M nodes become more interconnected and thus, the likelihood that an M node has a customer route to the failing prefix increases. This is also reflected in that the betweenness centrality of M nodes grows by a factor 2.2. Besides, the convergence delay also increases denoting an increase in the number of messages sent by a T node's customer

The increase in the convergence time can be attributed to two factors: the high path diversity which results in additional alternative paths to explore, and more importantly the growth in the depth of the hierarchy [20]. The high multihoming degree of M nodes creates more interconnections among transit providers. The dense connectivity keeps the average path length stable around 4 hops. However, the high path diversity also results in longer preferred paths (i.e. a longer path through a customer is preferred over a shorter path through a provider). The length of the longest preferred path (i.e. the diameter of the network) increases from 5 to 15 hops. Note that such path inflation has been observed and shown to be prevalent in the Internet [14, 29].

5.2 The effect of edge multihoming

The plots in Fig. 7 illustrate our three metrics when varying the average multihoming degree at the edge. When increasing the average multihoming degree from 1 to 7.5, we observe an increase in all three metrics, but at a much slower rate than in the former scenario. For example, the number of received updates during the down phase grows by a factor 2.23 at T nodes; recall from Sec. 5.1 that the same metric has grown by a factor 174.5. In the following we examine the differences between the two scenarios.

The three factors mentioned above increase following the growth of edge multihoming but at a much slower rate than for increased multihoming in the core. The growth in the number of customers per a T node is not important here since it reflects the addition of more stub customers, which



(a) The average number of received updates

(b) Sent updates

(c) Convergence time

Figure 8: Decreasing hierarchy depth

does not have an impact on the number of received updates. Furthermore, the importance of M nodes also increases because more M nodes will have the instability creator in their customer tree. However, the depth of this tree is lower than in the previous case (i.e. no more interconnections between M nodes). Therefore, the main factor behind the observed growth is the slight increase in the importance of M nodes.

5.3 The effect of topology flattening

The plots in Fig. 8 illustrate our three metrics when manipulating the depth of the hierarchy. The x-axis shows the average number of customers a T node has, which corresponds to an increasingly flat topology as it grows.

We observe that all the three metric exhibit initial growth followed by a decreasing trend. Recalling the factors described in Sec. 5.1, it is clear that the average number of customers a T node increases since more nodes prefer to choose T nodes as providers. However, the importance of M nodes decreases, because stub nodes tend to connect with T nodes directly. We measure a decrease in M nodes betweenness centrality by a factor 0.47. Thus, the initial surge in all three metrics is caused by the increase in the average number of customers. This effect is later offset by the decrease in the importance of M nodes. The later observation is reflected in a monotonic downward trend in the average path length (a decrease from 4.5 to 3.8 hops). Further the diminishing role of M nodes contributes also to reducing path diversity and consequently the convergence delay.

5.4 Discussion

The previous subsections demonstrate that our topology and simulation model can help in gaining insights on the interaction between topology parameters and routing. A main observation is that the impact of withdrawals of edge prefixes on routing scalability depends on the level connectivity, and in particular on path diversity at the core of the network. Another insight is that, *the scalability is strongly determined by the hierarchical structure of the topology.* The depth of the hierarchy decides the importance of transit providers and the convergence delay. We also conclude that, *the number of explored paths does not necessarily reflect the number of received updates.*

Our analysis suggests that the impact of topology growth on churn that is generated following a C-event can be understood by measuring few metrics. In the case of Tier-1s, we need to measure: the number of their customers; the betweenness centrality of transit providers; and the convergence time. In our future work, we want to measure the evolution of these metrics in the global routing system using publicly available BGP traces (e.g. RouteViews data). We also want to employ SIMROT in investigating other types of routing events.

6. CONCLUSION

This work addresses the lack of comprehensive framework to simulate BGP. It proposes a flexible topology generator that produces AS-level graphs which are annotated with business relationships. The second component of the framework is a light-weight BGP simulator that is capable of capturing routing dynamics and scaling to network sizes of thousands of nodes.

We validate the topology generator and illustrate its ability in reproducing various known properties of AS-level topology. Besides, we have compared the performance and correctness of SIMROT-sim when simulating BGP dynamics, with that of the widely used SSFNET simulator. This benchmarking confirms that our simulator significantly outperforms the SSFNET simulator in terms of processing time and memory requirements, while producing similar results.

We further employ our framework in investigating a set of what-if questions concerning the impact of different topology parameters on BGP churn triggered by a C-event. The investigations show that BGP scalability with respect to churn under the simulated scenario depends on the densification at the core of the network, and the hierarchical structure of the topology. We also highlight significant differences between the incoming and outgoing churn. Our findings interestingly suggest that the evolution of the Internet routing scalability with respect to stub failures can be understood by measuring few metrics. In our future work, we aim to employ our topology and simulation model to investigate other types of events and relate them to the measured BGP data.

7. REFERENCES

[1] RIS Routing Beacons. www.ripe.net/ris/docs/beacon.html.

[2] Routeviews project page. http://www.routeviews.org.

[3] SSFNet website. http://www.ssfnet.org/.

[4] CIDR report. http://www.cidr-report.org/, March 2011.

[5] W. Aiello, F. Chung, and L. Lu. A random graph model for massive graphs. In *STOC '00: Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 171–180, New York, NY, USA, 2000. ACM.

[6] R. Albert and A. L. Barabasi. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47, 2002.

[7] A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, October 1999.

[8] CAIDA. Archipelago measurement infrastructure. http://www.caida.org/projects/ark, 2010. "Online; accessed 15-Dec-2010".

[9] C. de Launois. GHITLE: Generator of Hierarchical Internet Topologies using LEvels. http://ghitle.info.ucl.ac.be/.

[10] A. Dhamdhere and C. Dovrolis. Ten years in the evolution of the Internet ecosystem. In *IMC 2008*, 2008.

[11] X. Dimitropoulos, D. Krioukov, A. Vahdat, and G. Riley. Graph annotations in modeling complex network topologies. *ACM Trans. Model. Comput. Simul.*, 19(4):1–29, 2009.

[12] X. Dimitropoulos and G. Riley. Efficient large-scale BGP simulations. *Elsevier Computer Networks, Special Issue on Network Modeling and Simulation*, 50(12):2013–2027, 2006.

[13] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *ACM SIGCOMM*, pages 251–262, 1999.

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

[15] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. The flattening Internet topology: natural evolution, unsightly barnacles or contrived collapse? In *PAM'08: Proceedings of the 9th international conference on Passive and active network measurement*, pages 1–10, Berlin, Heidelberg, 2008. Springer-Verlag.

[16] T. Griffin, B. Shepherd, and G. T. Wilfong. Policy disputes in path-vector protocols. In *Proceedings of the Seventh Annual International Conference on Network Protocols*, ICNP '99, Washington, DC, USA, 1999. IEEE Computer Society.

[17] T. Griffin and G. T. Wilfong. Analysis of the MED oscillation problem in BGP. In *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, pages 90–99, Washington, DC, USA, 2002. IEEE Computer Society.

[18] H. Haddadi, D. Fay, A. Jamakovic, O. Maennel, A. W. Moore, R. Mortier, and S. Uhlig. On the importance of local connectivity for Internet topology models. In *the 21st International Teletraffic Congress*, pages 1–8, September 2009.

[19] Y. He, S. V. Krishnamurthy, M. Faloutsos, and M. Chrobak. Policy-aware topologies for efficient inter-domain routing evaluations. In *IEEE INFOCOM 2008 Mini-Conference*, Phoenix, AZ, USA, April 2008.

[20] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *ACM SIGCOMM*, pages 175–187, August 2000.

[21] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. In *SIGCOMM '10: Proceedings of the ACM SIGCOMM 2010 conference on SIGCOMM*, pages 75–86, New York, NY, USA, 2010. ACM.

[22] A. Lakhina, J. W. Byers, M. Crovella, and P. Xie. Sampling biases in IP topology measurements. In *In IEEE INFOCOM*, pages 332–341, 2002.

[23] L. Li, D. Alderson, W. Willinger, and J. Doyle. A first-principles approach to understanding the internet's router-level topology. In *ACM SIGCOMM*, pages 3–14, Portland, OR, 2004.

[24] P. Mahadevan, C. Hubble, D. V. Krioukov, B. Huffaker, and A. Vahdat. Orbis: rescaling degree correlations to generate annotated internet topologies. In *SIGCOMM*, pages 325–336, 2007.

[25] P. Mahadevan, D. Krioukov, M. Fomenkov, X. Dimitropoulos, kc. claffy, and A. Vahdat. The Internet AS-level topology: three data sources and one definitive metric. *SIGCOMM Comput. Commun. Rev.*, 36(1):17–26, 2006.

[26] P. Mahadevan, D. V. Krioukov, K. R. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. In *SIGCOMM*, pages 135–146, 2006.

[27] Z. M. Mao, R. Bush, T. G. Griffin, and M. Roughan. BGP beacons. In *IMC '03: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement*, pages 1–14, New York, NY, USA, 2003. ACM.

[28] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: An approach to universal topology generation. In *Proceedings of IEEE MASCOTS*, pages 346–353, 2001.

[29] W. Mühlbauer, S. Uhlig, A. Feldmann, O. Maennel, B. Quoitin, and B. Fu. Impact of routing parameters on route diversity and path inflation. *Comput. Netw.*, 54(14):2506–2518, 2010.

[30] B. Quoitin, V. Van den Schrieck, P. Franois, and O. Bonaventure. IGen: Generation of router-level internet topologies through network design heuristics. In *Proceedings of the 21st International Teletraffic Congress*, September 2009.

[31] B. Quoitin and S. Uhlig. Modeling the routing of an autonomous system with C-BGP. *IEEE Network*, 19(6), November 2005.

[32] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (BGP-4). RFC4271, January 2006.

[33] RIPE's Routing Information Service. http://www.ripe.net/ris/.

[34] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with rocketfuel. *IEEE/ACM Trans. Netw.*, 12:2–16, February 2004.

[35] R. Teixeira, K. Marzullo, S. Savage, and G. M. Voelker. Characterizing and measuring path diversity of internet topologies. In *Proceedings of the 2003 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '03, pages 304–305, New York, NY, USA, 2003. ACM.

[36] J. Winick and S. Jamin. Inet-3.0: Internet Topology Generator. Technical Report UM-CSE-TR-456-02, EECS, University of Michigan, 2002.

[37] X. Zhao, B. Zhang, A. Terzis, D. Massey, and L. Zhang. The impact of link failure location on routing dynamics: A formal analysis. In *ACM SIGCOMM Asia Workshop*, April 2005.