Obtaining Provably-Legitimate Internet Topologies

Yihua He Michalis Faloutsos Srikanth V. Krishnamurthy Marek Chrobak
Technical Report
Department of Computer Science and Engineering
University of California, Riverside

ABSTRACT

What topologies should be used to evaluate protocols for inter-domain routing? Using the most current Internet topology is not practical, since its size is prohibitive for detailed, packet-level inter-domain simulations. Besides being of moderate size, the topology should be *policy-aware*, that is, it needs to represent business relationships between adjacent nodes (that represent Autonomous Systems). In this paper, we address this issue by providing a framework to generate small, realistic, and policy-aware topologies. We propose HBR, a novel sampling method, which exploits the inherent hierarchy of the policy-aware Internet topology. We formally prove that our approach generates connected and legitimate topologies, which are compatible with the policy-based routing conventions and rules. Using simulations, we show that HBR generates topologies that: (a) maintain the graph properties of the real topology, (b) provide reasonably realistic inter-domain simulation results while reducing the computational complexity by several orders of magnitude as compared to the initial topology. Our approach provides a permanent solution to the problem of inter-domain routing evaluations: given a more accurate and complete topology, HBR can generate better small topologies in the future.

1. INTRODUCTION

"Which topology should I use to evaluate a new approach for inter-domain routing?"

This practical question lies at the heart of our paper. Inter-domain routing studies need to resort to simulations, since theoretical analysis and experimentation cannot be easily used for many types of BGP performance evaluations. First, BGP and inter-domain interactions are too complex for theoretical analysis, especially when it comes to studying large-scale phenomena, such as cascading failures within BGP [27], where small or canonical topologies may not be adequate. Second, experimentation is also very cumbersome: replicating a medium-size inter-domain network in a lab is not trivial, and experimenting on the Internet itself is not a welcomed proposition to network operators. As a result, simulations are widely used to test and validate new techniques for BGP improvements [39][38][6][8][46] and to study the behavior and performance of BGP with different parametric settings [22][33].

Our goal is to enable feasible and meaningful inter-domain routing studies. We want to be able to conduct simulations on a topology that is: (a) sufficiently small, so that simulations can be conducted and repeated in "human" time (e.g. a few days), and at the same time, (b) appropriately representative, so that the results are reasonable estimates of the performance in the real world. Currently, BGP-related studies are often obliged to choose between these requirements.

First, *routing policies are not considered in many previous and even recent studies*, some as recent as 2005 and 2006 [8][46]. The simulations in these studies model BGP as a pure path vector routing protocol, which chooses the shortest path and each Autonomous System (AS) always advertises the best (shortest) known route to all of its neighbors. However, this is not a realistic behavior due to routing policies. Not considering routing policies may lead to inaccurate or unrealistic conclusions. Here, we use the term **policy-aware** (**no-policy**) to refer to a topology that does (not) represent routing policies. A policy-aware topology has annotated edges, which represent the type of relationship between the corresponding ASes. For example, directed edges are often used to indicate a provider-customer relationship. Note that a realistic policy-aware topology has to be **BGP-connected**: any two ASes must be able to communicate over a path that does not violate any routing policy. Further, we say a topology is **legitimate**, if it is BGP-connected and **relationship loop-free**, which we define in Section 4.

Second, *the community does not have a reasonably-sized policy-aware inter-domain topology for simulations*. The current Internet topology (25,000 ASes) is too large for any detailed simulators, such as SSFNET [3]. Furthermore, BGP simulations are typically repeated a number of times for each combination of parameters. The total number of runs could exceed 200,000 [22]. This will be prohibitive, if the topologies are large and each run requires a great amount of CPU time. As a result, researchers either only use small canonical topologies [22], or rely on topology generators [26][34][7][31] and sampling approaches [29][40] to produce small Internet-like topologies. However, none of these approaches produces policy-aware topologies. A recent work [15] proposes a policy-aware topology generator, but the topology is not

guaranteed to be BGP-connected. In a related but different direction, several recent studies attempt to identify the right level of BGP abstraction [36] [35], but they focus on other protocol issues and not on generating small topologies, as we do here.

As our main contribution, we propose **H**ierarchy-**B**ased **R**eduction (**HBR**), arguably the first sampling method to generate provably legitimate topologies. A key characteristic is that our sampling "follows" the inherent hierarchy of the Internet [45] [44], in a top-down fashion. This enables our method to have a BGP-connected topology at every step. In fact, we formally prove that HBR produces a legitimate topology, as defined above, if the initial topology is legitimate.

A key advantage of our approach is that it will not be outdated any time soon: as the Internet grows or as we measure it more accurately, our approach can always be used to sample the newer, more complete topology. Our work leads to the following key observations:

a. Policy-aware simulations are less computationally intensive. An interesting observation is that simulations on a policy-aware topology only require 1/3rd to 1/20th of the CPU time required for a no-policy topology of the same size. The explanation is that routing policies limit path explorations and thus, reduce the number of events in simulations.

As motivation for the rest of the work, we also examine the effect of considering policies in a simulation study. Going beyond numerical differences, we find that policies have fundamental impact in performance trends and high-level protocol questions, like *"Does MRAI affect the observed performance?"*, as we discuss in Section 3.

b. Our method can reduce a topology successfully to a fraction of its original size. The topology is provably legitimate, and we validate the realism of our topologies using: (a) an extensive list of important graph metrics, and (b) an actual evaluation of BGP performance, such as BGP convergence time.

c. Using our approach, we can reduce the simulation time by 3-4 orders of magnitude. In fact, the simulation time is reduced by the size reduction and, by the use of policy-aware topologies. We claim that this pushes the envelope in our ability to simulate inter-domain routing effectively.

Our work in perspective. Our work can be seen as the first step towards a more realistic usable-in-practice BGP topology model, which is an ambitious and non-trivial goal [35] [36]. Note that we model the topology at the level of ASes, each node represents an AS. Such a model cannot capture intra-AS dynamics or interactions between iBGP and BGP [17]. We leave for the future the enrichment of the model with such considerations. In addition, our work does *not* attempt to explain: how and why the Internet topology has evolved into its current form [10], how the Internet metrics evolve over time [37], or how the graph metrics affect the BGP performance, although we discuss some of these issues later.

The usefulness of our work is attested by the number of requests we have received from research institutions, like CMU and GTech to name a few. A preliminary version of this work appeared as a 6-page mini-paper (whithheld for anonymity). That version did not have: the theoretical analysis and proofs in Section 4.3, the evaluation using BGP performance metrics in Section 5.3, and the simulation speed-up in Section 6, and Section 3 in its entirety. In addition, we use a more extensive list of graph metrics in Section 5.

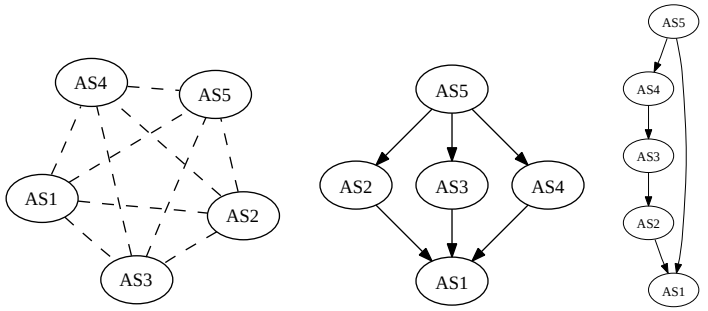
The rest of this paper is organized as follows. In Section 2, we present the related background. In Section 3, we study the effect of routing policies on BGP simulations. In Section 4, we present HBR, which is evaluated in Section 5. In Section 6, we quantify the simulation speed up from using our method.

2. BACKGROUND AND RELATED WORK

The Internet is composed of tens of thousands Autonomous Systems (ASes). The Border Gateway Protocol (BGP) is the *de facto* routing protocol used to exchange reachability information among these ASes and to interconnect them. Simulations have been widely used to study BGP parameters, such as Minimum Route Advertisement Interval (MRAI) [22], which we describe later, and Route Flap Damping [33], and to evaluate new inter-domain protocols [39][38][6][8] [46]. However, simulations in all these studies only use no-policy topologies.

Routing policies are commonly implemented in today's Internet. Policies can be thought of as the rules with which an AS accepts, modifies, and advertises further route information (route updates) that it receives from its neighbors. Although an AS may have specific routing policies for each of its neighbor ASes, general policies are normally determined by its business relationships with its neighbor ASes. For example, a multi-homed AS will not advertise the routes learned from one of its providers to its other providers, since the AS does not want to carry transit traffic between its providers. A set of such commonly-used rules is referred as **No-Valley-Prefer-Customer (NVPC)** routing, and we discuss it more in the next section. AS pairs typically have a provider-customer or peer-to-peer relationship. Such relationships can be inferred from global routing tables [18][48][5][12]. Gao et al. [20][19] study and formalize the model of routing policies that is widely used now. Labovitz et al. [30] measure the impact of topology on BGP performance using data from 200 ISPs, but they do not develop a method to generate topologies.

The challenge of realistic BGP simulations: An Internet scale BGP simulation is very resource consuming and often impossible. The required memory for detailed BGP simulators, such as [3][2][1], increases cubically with the size of the network [14]. In the most popular BGP simulator SSFNET [3], a simulation on a 1000-AS no-policy topology could consume 2GB memory even if each AS only



(a) *Clique* (b) *Focus* (c) *Ring*

Figure 1: Simple AS topologies with relations

Parameters	Values
Link delay	0.10 (sec)
proc_delay	uniform rand.
MIN_proc_time	0.01 (sec)
MAX_proc_time	1.00 (sec)
MRAI jitter	True
Flap Damping	False
WRATE	False
SSLD	False
always_run_dp	True
FCFC	True

Table 1: SSFNET simulation setup

announces one prefix. C-BGP [42] can perform large scale BGP simulations. However, it only implements the BGP decision process, and does not consider details of the protocol, such as timers and BGP messages. A recent simulator, sim-BGP [41], can perform large-scale simulations by ignoring the protocol stack below the application layer, but the number of prefixes in each simulation is very limited. Even if memory requirement were not an issue, large simulations would take months for a single run, as we discuss later in Section 6. In other words, Internet-scale detailed simulations are not currently feasible.

3. THE EFFECT OF POLICIES

As motivation for the rest of the work, we elaborate on the significant impact the use of a policy-aware topology has in a simulation study. We focus on qualitative differences, such as trends and high-level BGP protocol questions, like *"Does MRAI affect the observed performance?"*. In fact, this question was addressed in a real study [22] with topologies similar to the ones we use here except without policies.

Here, we use *small toy* topologies for two reasons: (a) the simulations are computationally feasible, (b) we can understand the reasons for the difference in the performance. In section 5.3, we revisit this issue using larger topologies.

We consider three families of network topologies, *Clique*, *Focus* and *Ring*. These topologies are simple, but they commonly exist embedded in real Internet AS topologies. For example, the top-tier providers form a clique with peer-to-peer relationships as we will see later. (1) *Clique*. A network configuration of size n in the *Clique* family is made up of n ASes in a full mesh. A size-5 policy-aware *Clique* is shown in Fig 1 (a), where a dashed link represents a peer-to-peer relationship between a pair of ASes. (2) *Focus*. A network configuration of size n in the *Focus* family has $n-2$ parallel paths of length two, all terminating at AS n. This type of topology corresponds to a low customer (AS1) multi-homing to $(n-2)$ mid-providers, while AS n might be thought of as a top provider. A size-5 policy-aware *Focus* is shown in Fig 1 (b), where the arrows on the edges point from providers to customers. (3) *Ring*. A network configuration of size n in the *Ring* family has n ASes in a ring. In reality, one possible configuration of a *Ring* is from multi-homing with providers at different levels of hierarchy. For example, in Fig 1 (c), AS1 multi-homes with two providers AS2 and AS5 at different hierarchical levels. AS1 through AS5 form a size-5 policy-aware *Ring*.

We use the most popular BGP simulator SSFNET[3] to study BGP performance with these three topologies. Such topologies have been studied before [22][33][38], but without considering any BGP policy. To configure BGP policies according to AS relationships in SSFNET, we follow the NVPC rules [47]. *Rule 1: paths learned from providers or peers are never advertised to other providers or peers.* We configure this rule in SSFNET by building outbound route export filters according to the source of the routes. *Rule 2: paths learned from customers are preferred to the paths learned from peers and providers, and paths learned from peers are preferred to the paths learned from providers, regardless of path length.* We configure this rule in SSFNET by building inbound route filters to set higher *local_pref* values to the routes learned from customers than the ones from peers or providers, and higher *local_pref* values to the routes from peers than the ones from providers. If there is more than one available path to a destination, the common BGP decision process will first choose the path with the highest *local_pref* value. If there is more than one path with the same *local_pref* value, the path with the shortest path length

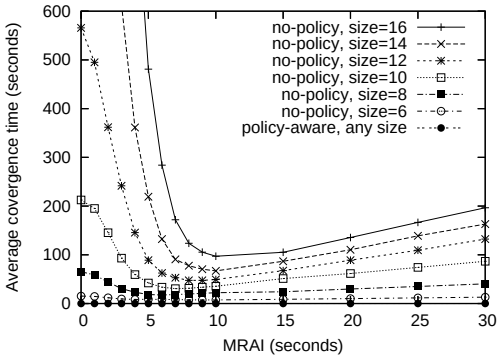


Figure 2: Convergence in *Clique*, *DOWN*.

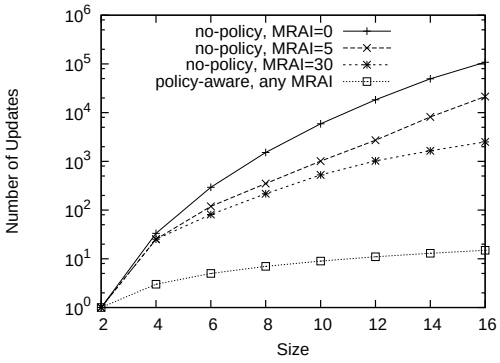
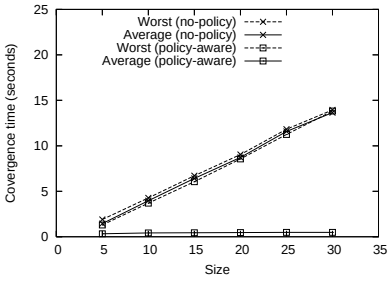
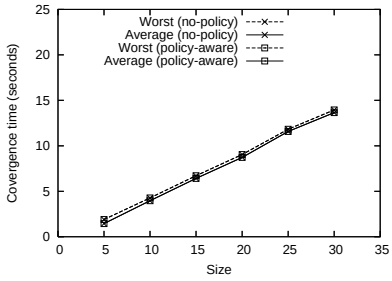


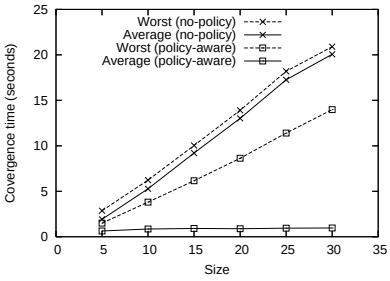
Figure 3: Updates in *Clique*, *DOWN*.



(a) Originated at the top provider



(b) Originated at the low customer



(c) Originated at a mid-provider

Figure 4: Convergence time versus graph size in *Focus* with different prefix origin locations, *DOWN* phase

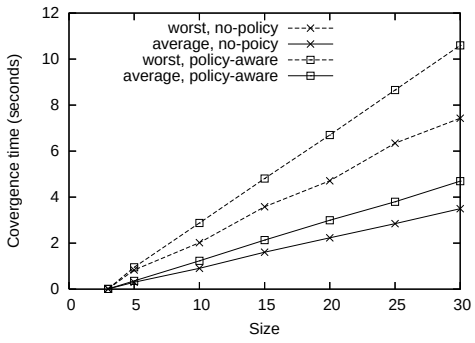
will be chosen.

We measure several performance metrics. We define the *convergence time* of a node D to be the interval between two timestamps T_{send} and T_{stable}, where T_{send} is the time when the origin AS S sends out the update (either advertisement or withdraw) for a prefix p, and T_{stable} is the time when the route to prefix p becomes stable in the forwarding table of node D. The convergence time is an important metric for routing protocols because it quantifies the time needed before a stable path is found and a reliable connection can begin. The *average convergence time* is the average of the convergence times of all nodes in the topology. The *worst convergence time* is the longest convergence time among all nodes in the topology. We also monitor the *number of updates* corresponding to a routing event. This number is an indicator of the efficiency of a routing protocol in propagating a routing event.

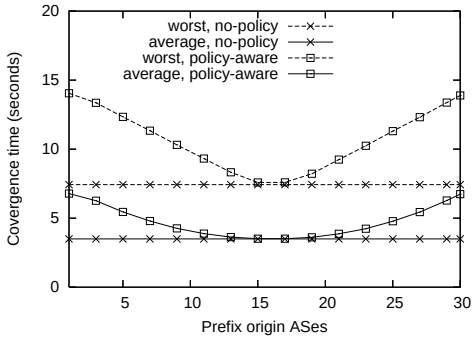
We inject two common BGP routing events in our simulation. (1) *UP*: An origin AS advertises a single prefix. (2) *DOWN*: In a stable state produced by the *UP* experiment, the origin AS withdraws the prefix that was being advertised. We vary several parameters, including the MRAI, the size of the topology and the origin AS. For each combination of the parameters, we have the simulation run 100 times, with different random seeds to get robust results. The configuration of our SSFNET BGP simulations is summarized in Table. 1.

In Fig. 2, we compare the average convergence time with different MRAI values for a *DOWN* event in a *Clique* topology. The results for no-policy Cliques are consistent with [22]: the convergence time increases as the size of the topology increases, and there is an optimal value of MRAI with which there is a smallest average convergence time for a given number of nodes. However, in *Clique* with policies, the result is totally different: the convergence time does not change with topology size or the MRAI value. In fact, the convergence time is a very small constant in all cases. The reason for this difference is that, with the selective export policy, paths learned from peers will not be propagated to other peers. In fact, a router only knows one way to reach an announced prefix (by going directly). When the prefix is withdrawn, no other "false" path is present and no path exploration happens. This is confirmed by examining the total number of updates, which is shown in Fig. 3. With policies, the number of updates is always $n - 1$ if the clique size is n, and it is independent of the MRAI value. On the contrary, when no routing policy is present, the number of updates increases exponentially as the clique size increases, and depends on the MRAI values.

Fig. 4 shows the convergence time for *DOWN* events in *Focus* topologies with different sizes. In Fig. 4 (a), when the prefix is originated from the top provider, we can see that the average convergence time is much shorter than that without policies, and almost independent of the size of the network. However the worst convergence time is the same as in a no-



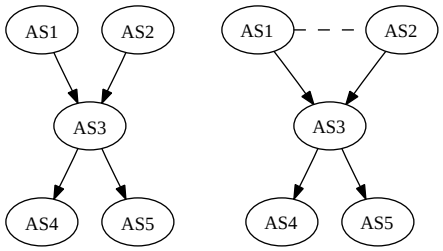
(a) with difference sizes



(b) with difference origin locations

Figure 5: Convergence time in *Ring* topologies, *UP* phase

policy *Focus* network. In fact, the worst case is only experienced by the low customer, because prefix withdrawals do not reach this customer at the same time. Thus, this low customer will keep switching among the mid-providers before it eventually learns that no path to the top provider is available. On the other hand, since the low customer will not transit traffic for its providers, each mid-provider only knows one route to the top provider (by going there directly). Therefore no path exploration will happen at the mid-providers and routes converge quickly. Fig. 4 (b) shows the scenario when a prefix originates from the low customer. In contrast with Fig. 4 (a), the average convergence time is the same as in no-policy *Focus* topologies. This is because each mid-provider and the top provider have multiple routes to the low customer, exactly the same as in no-policy topologies. In other words, we see that routing policies make a huge difference in convergence time even when a prefix originates from topologically symmetric locations. If a prefix originates from a mid-provider, as shown in Fig. 4 (c), routes converge quickly for all nodes except the low customer. Therefore, the average convergence time with policies is almost zero regardless of the *Focus* size. The quick convergence at other nodes also helps reduce the worst convergence time, which occurs at the low customer. This is because there is less interaction and fewer update exchanges between the low customer and the other nodes.



(a) BGP-disconnected (b) BGP-connected

Figure 6: Two topologies and their policy assignments:

Routing policies could also increase convergence times. In Fig. 5 (a), the convergence times, averaged over all prefix origin locations, are shown for the *UP* phase in *Ring* topologies. Both the average and the worst convergence time for a policy-aware *Ring* are longer than those in a no-policy *Ring*. This is because with routing policies, the shortest path is not always the most preferred path. A node would switch to and propagate preferred routes even when shorter, but less preferred routes were learned earlier. In a no-policy *Ring*, such route switches do not happen. Fig. 5 (b) shows the convergence time of a size-30 *Ring*. We can see that the convergence time in a policy-aware *Ring* depends on the origin of the prefix, even though the location of every node is topologically symmetric.

In summary, routing policies have a significant impact on BGP performance measured by simulations. The export filtering policy generally reduces the number of available paths in the topology for certain ASes, and therefore reduces both the convergence time and the number of required updates. On the other hand, the path preference policy may make ASes switch to later-learned longer, but more preferred paths, and therefore could result in an increase in both the convergence time and the number of required updates. Table 2 summarizes the BGP performance comparisons.

Obviously, more complex policies exist in practice (such as set local_pref or export filters by community values). But even this simplified study shows that BGP performance is greatly affected by the routing policies, and thus we need policy-aware topologies for realistic BGP simulations.

4. SAMPLING METHODS

In this section, we tackle the problem of sampling "Internet-like" policy-aware topologies. We start by identifying the essential properties for a legitimate, namely BGP-connected, policy-aware topology. We then propose a sampling method that exploits the inherent Internet structure to guarantee that the sampled topology is BGP-connected. We examine other graph properties of our sampled topologies in Section 5.

4.1 Essential Topological Properties

As discussed earlier, when policies are considered, there are two additional properties that a topological model needs to satisfy. First, **the topology must be "BGP-connected"**.

Table 2: BGP performance comparison with and without policies

	Clique, T_{down}		*Focus*, T_{down}		*Ring*, T_{up}	
	without	with	without	with	without	with
Convergence (avg)	+	−	+	−	−	+
Convergence (wst)	+	−	+	−	−	+
Number of updates	+	−	+	−	−	+
MRAI matters?	Yes	No	No	No	Yes	Yes
Size matters?	Yes	No	Yes	Depends	Yes	Yes
Symmetric Performace?	Yes	Yes	Yes	No	Yes	No

The concept of connectivity is restricted when routing policies are incorporated: having a connected graph is not sufficient. As defined earlier, a topology is **BGP-connected**, if each node has at least one path to every other node without violating the routing policy. For example, in Fig. 6 (a), AS1 and AS2 can not reach each other because AS3, as a customer, will not carry transit traffic between AS1 and AS2. Thus, this topology is not "BGP-connected". In Fig. 6 (b), adding a peer-to-peer edge (doted line) makes the topology BGP-connected. Second, **the topology must be relationship loop-free**. For example, ASes A, B and C form a relationship loop, if A is B's provider, B is C's provider, and C is A's provider. Relationship loops should not occur [24]; otherwise, BGP is not guaranteed to converge [19].

As we defined earlier, a topology is legitimate, if it is BGP-connected and relationship loop-free.

Sampling a topology is more challenging when we consider policies. Most previous efforts [26] [34] [29] [40] do not consider policies. A recent work [15] considered generating a topology with AS relationships by enforcing the joint distribution of provider, customer and peer degrees. However, the resulting topology is not guaranteed to be BGP-connected nor relationship loop-free. We examined if trimming away BGP-disconnected nodes would make the scheme work, but discovered that such a practice deteriorates the quality of the generated topologies with respect to other metrics. In fact, in some cases, trimming could make a topology degenerate into BGP-disconnected pieces. In addition, a careless assignment of AS relationships could introduce relationship loops.

4.2 Hierarchy-Based Reduction

We present Hierarchy-Based Reduction (*HBR*), a method to sample a policy-aware topology. A key property of HBR is that it provably produces a BGP-connected and relationship loop-free topology, as long as the initial topology has these properties.

The intuition behind our approach is to "follow" the Internet hierarchy in a top-down fashion. We start from the clique of top-tier providers at the top of the hierarchy, and go down in the hierarchy. We first present the basic method (*HBR0*), which consists of three stages:

Initialization stage: We first identify and select all ASes that have no providers in the initial topology. As we will see later, if the topology is BGP-connected, these ASes form a clique with peer-to-peer links, which we call the **top-clique**.

Iterative stage: For each AS selected in the Initialization stage, we select randomly its customers, each with probability p. This is Step 1 in this stage. In the next step, we take the chosen ASes from Step 1 and select their customers, again each with probability p. We repeat the process step by step until an iteration does not select any new ASes.

Assembling stage: We construct the smaller topology by keeping all the links between the selected ASes including peer-to-peer links. Naturally, the relationship reflected by a link in the new graph is the same as that in the initial graph. Algorithm 1 provides a pseudo-code description of our method.

Algorithm 1 HBR0 algorithm: $G(V, E) \Rightarrow G_s(V_s, E_s)$

Input: original topology $G(V, E)$, sampling rate $0 < p \leq 1$
Output: smaller topology $G_s(V_s, E_s)$

```
TopASes ⇐ get top clique ASes from G(V, E)
V_s ⇐ TopASes
CurrentLayerASes ⇐ TopASes
while CurrentLayerASes not empty do
    NextLayerASes ⇐ empty
    for all AS in CurentLayerASes do
        for all cust such that cust is a customer of AS do
            if 0 ≤ rand() < p and cust not in V_s then
                add cust into NextLayerASes
            end if
        end for
    end for
    V_s ⇐ V_s ⋃ NextLayerASes
    CurrentLayerASes ⇐ NextLayerASes
end while
E_s ⇐ empty
for all edge in E do
    if both nodes at the two ends of the edge is in V_s then
        add edge into E_s
    end if
end for
return G_s(V_s, E_s)
```

4.3 Provably Legitimate Topologies

HBR0 guarantees that the reduced topology is BGP-connected and relationship loop-free, as long as the initial topology is BGP-connected and relationship loop-free. To support this assertion, we prove the following theorems.

THEOREM 1. *(Characterization of BGP topologies) Let G be a relationship loop-free topology. Then G is BGP-connected if and only if all ASes in G that do not have any providers form a single clique with peer-to-peer relationships.*

We remark here that the loop-free assumption implies that there indeed exist some ASes that have no providers, and thus the clique mentioned in this characterization is not empty. If there is only one AS with no providers, it forms a trivial clique.

THEOREM 2. *If the initial topology G is BGP-connected and relationship loop-free then the topology G_s produced by HBR0 will be BGP-connected and relationship loop-free as well.*

Interestingly, Theorem 1 provides a fundamental guideline for any future sampling algorithms: in any "legal" BGP simulation topology, *all* ASes without providers *must* form a clique with peer-to-peer relationships and every node not in that clique must have a provider.

Before proving these theorems, we need to establish a more rigorous setting for them. We start with the following two assumptions:

Assumption 1: Every pair of adjacent ASes has one of the following two types of relationships: (1) customer-provider, or (2) peer-to-peer.

Assumption 2: An AS will not advertise to its providers or peers routes learned from other providers or peers.

Note that both assumptions are commonly used in BGP research. Some deviations may exist in actual deployments, but typically to a very small extent. In addition, note that Assumption 2 is tightly related to the NVPC routing, which was discussed earlier.

These two assumptions lead to the following definitions. By a *topology* or a *graph* G we mean a mixed graph whose nodes represent ASes and that has edges of two types: directed customer-provider edges, and undirected peer-to-peer edges. We say that G is *relationship loop-free* if it does not contain a directed cycle formed by customer-provider edges. (Other types of cycles are allowed.)

Assume G is relationship loop-free. A *top node* of G is any node that does not have a provider (that is, it has no outgoing customer-to-provider edge.) If A is not a top node, we arbitrarily designate one provider of A as its *parent* and denote it by $P(A)$. More generally, define $P^0(A) = A$ and, if $P^t(A)$ is defined and has a provider then $P^{t+1}(A) = P(P^t(A))$. Thus $P^t(A)$ is simply a node reached from A by following t links to parents.

Note that, since G is relationship loop-free, it must have at least one top node. Further, for any node A, the path $P^0(A), P^1(A), ...$ must end at some node $P^t(A)$ that does not have a provider, and thus is a top node of G. We denote this node by $P^*(A)$.

Based on Assumptions 1 and 2, we define a *BGP-path* to be a path that consists of a segment of edges from customers to providers (possibly empty), followed or not by a single peer-to-peer edge, followed by a segment (possibly empty) of edges from providers to customers. More formally, a BGP path has a form: $A_1, A_2, ..., A_j, A_{j+1}, ..., A_k$ where $(A_1, A_2), ..., (A_{j-1}, A_j)$ and $(A_k, A_{k-1}), ..., (A_{j+2}, A_{j+1})$ are customer-provider edges, and either $A_j = A_{j+1}$ or (A_j, A_{j+1}) is a peer-to-peer edge. (We allow $j = 1$ or $j+1 = k$ to represent the cases when the first segment is empty or the second segment is empty.)

PROOF PROOF OF THEOREM 1. ($\Rightarrow$) Suppose that G is relationship loop-free and BGP-connected. As explained earlier, there is at least one top node in G. We now show that the set of all top nodes forms a clique with peer-to-peer edges. This is trivial if there is only one top node. So assume there are at least two, and let A and B be two different top nodes.

We claim that A and B are connected via a peer-to-peer edge. Towards contradiction, suppose they are not. By BGP-connectivity, A and B must be able to reach each other with a BGP-path, say $A, M_1, M_2...M_k, B$. Since there is no peer-to-peer edge from A to B and A is not a customer of B, we have $k \geq 1$, that is, there must be at least one intermediate node on this path between A and B. It is not possible that both edges (A, M_1) and (M_k, B) are peer-to-peer, because that would violate the definition of BGP-paths. So either (M_1, A) or (M_k, B) must be a customer-provider edge. By symmetry, we can assume that it is (M_1, A), that is, A is a provider of M_1. But then the definition of BGP-paths implies that M_1 is a provider of M_2, M_2 is a provider of M_3, etc., and we conclude that M_k is a provider of B – a contradiction with the assumption that B is a top node.

($\Leftarrow$) Suppose now that the set of all top nodes of G forms a peer-to-peer clique. We claim that this implies that there is a BGP-path between any two nodes A, B of G. Indeed, let $A' = P^*(A)$ and $B' = P^*(B)$. Then A' and B' are top nodes of G, so, by our assumption, either they are equal or they are connected by a peer-to-peer edge. Then the path $A, P(A), ..., P^*(A), P^*(B), ..., P(B), B$ is a BGP-path. We conclude that G is BGP-connected. □

PROOF PROOF OF THEOREM 2. First, it is easy to see that if G is relationship loop-free, then G_s is relationship loop-free, because G_s is a subgraph of G. Second, we show that G_s is BGP-connected. In the iterative stage of HBR0, we only follow provider-to-customer links, and thus, each AS selected in this stage has a provider. Therefore, the only ASes without a provider are the ones we selected in the initialization stage. In addition, they form a clique with peer-to-peer links only, because all peer-to-peer links between the

selected ASes are kept in the assembling stage. According to Theorem 1, the reduced topology is BGP-connected. □

Note that Theorem 2 does not imply that our method preserves specific BGP-paths between nodes. It is possible that a BGP-path from A to B in the original graph went through a node C, and that A and B remained in G_s while C was removed. Nevertheless, Theorem 2 guarantees that there will be another BGP-path from A to B in G_s.

Our characterization may be useful for other purposes. For example, it provides a simple algorithm for verifying whether a given topology is consistent with routing policies, thus allowing one to verify correctness of other topology generators. Give a topology G, one can proceed as follows. First, we verify whether G is relationship loop-free. This amounts to verifying whether the subgraph of G induced by customer-provider edges is acyclic – a task that can be easily accomplished with topological sorting in time $O(m+n)$ (where m and n denote the number of edges and vertices of G). The topological sort will also identify the top nodes of G. To determine if G is BGP-connected, it then only remains to verify whether these top nodes form a clique with peer-to-peer edges. Overall, this yields a simple linear-time algorithm for verifying whether G is consistent with routing policies.

One can also consider a more general question: if we determine that G is not a policy graph, can we "repair" it, say by removing the minimum number of edges. This task, however, is already NP-hard (since it contains the well-known Minimum Feedback Edge Set as a special case [21]), and thus to solve it one needs to resort to heuristics.

4.4 HBR variations and uniform reduction

We recognize that HBR0 is not the only way to reduce a large Internet AS topology to smaller BGP-connected and relationship loop-free graphs. Thus, for each of the three reduction stages introduced in Section 4.2, we consider possible alternatives. However, this is far from an exhaustive list.

HBR1: Here, in the *Initialization stage*, instead of selecting every top-clique AS in the initial topology, we only choose a subset of the tier-1 ASes. The number (s) of the chosen tier-1 ASes has a lower bound $MinS$. s is also related to the initial clique size ($InitS$) and the sampling rate p: $s = MinS + \lceil p * (InitS - MinS) \rceil$. We use $MinS = 1$ in this paper unless otherwise stated. The reason for considering this alternative method is that there are normally fewer tier-1 ASes in a smaller Internet instance, as shown from the history of Internet. HBR1 tries to match the corresponding number of tier-1 ASes when it reduces a large topology to smaller one.

HBR2: Here, in the *Iterative stage*, customers of ASes from an upper tier are considered only once with probability p at a given step. This means, multi-homed ASes have a lower chance of being selected in *HBR2* than in *HBR0*.

HBR3: Here, in the *Assembling stage*, instead of keeping all provider-customer edges among the selected ASes, we only keep the provider-customer edges along which the customer was selected. This variation reduces the number of edges, as well as the presence of multi-homed ASes.

Note that it is easy to prove that Theorem 2 applies to all these HBR variations: they will produce a BGP-connected topology, if they start from one.

For comparison purposes, we use two uniform reduction heuristics: DDRV and DDRE, which are variations of the method for sampling no-policy networks [29].

DDRV: Directed Deletion of Random Vertex. Remove each AS, independently, with probability 1-p, and keep all edges between the remaining ASes. Finally, choose the largest BGP-connected component.

DDRE: Directed Deletion of Random Edges. Remove each edge, independently, with probability 1-p. In the end, choose the largest BGP-connected component.

Note that, in order to improve the performance of these approaches, we do not remove the top-clique ASes in DDRV, or the edges of the top-clique in DDRE.

Other methods and variations. There may be other methods to sample a graph (e.g., using aggregation of nodes or contracting edges[28]), which will be interesting to explore in the future. However, as we will see next, HBR seems to work well in practice. Furthermore, one needs to be careful in adopting sampling methods for undirected graphs [29], since we would like to ensure that it generates BGP connected topologies.

5. EVALUATION

There are at least two possible ways to assess the success of a reduced topology: one can either try to match the properties of real Internet instances in history, or try to match the properties of the initial unreduced instance. If the topological properties do not change with size, these two approaches converge. However, as shown in [16] and later in this section, no strict size-independent Internet topology property seems to exist so far. In fact, some properties *have to* change with size. For example, the average path length between the nodes in a grid topology increases as the size of the network increases. Thus, we decide to compare the properties of the sampled topology with those of historical Internet topologies with approximately the same size as the sampled topology.

We conduct our evaluation using the data from Oregon Routeviews [4]. This is the most frequently used route archival data to infer AS-level Internet topologies. Furthermore, it is the only data archive that has instances dated back to 1997. Although this is not a complete topology, we argue that this is less important in our case: (i) our comparison is consistent, since we start from a large instance of the same data set, and (ii) our sampling does not depend on the completeness of peer-to-peer edges, which are the ones mostly missing from the data [23]. We use snapshots of the Internet topology from Dec 1997 to Dec 2006, a 9-year span during which the size of Internet topology has grown 8-folds, from ap-

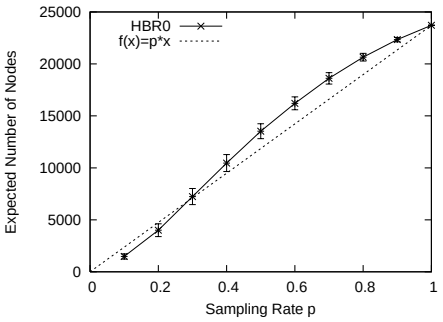


Figure 7: Expected number of nodes

proximately 3,000 ASes in Dec 1997 to nearly 24,000 ASes in Dec 2006.

There are several algorithms for inferring the AS relationships of an AS topology. However, some of them either require additional or seed data, which is not always available for the topology instances we have from 1997 to 2006 [48][12][45], or do not always work for our topology instances. Thus, we use the algorithm described in [18]. A common problem with all inference algorithms is that they sometimes produce a topology that is not BGP-connected. We trim the ASes that are not connected to the top-clique of the topology. The number of the trimmed ASes is very small, typically less than 1% of ASes in the original topology.

5.1 Probabilistic Analysis of the Expected Size

Since HBR follows the AS hierarchy, the probability of a node being chosen depends on its providers. Specifically, in a BGP-connected and relationship loop-free topology, the probability $\mathcal{P}(X)$ of AS X being chosen in HBR0 with sampling rate p can be calculated as:

$$\mathcal{P}(X) = \begin{cases} 1 & X \text{ has no provider} \\ 1 - \prod_i (1 - p\,\mathcal{P}(Y_i)) & Y_i \text{ is } X\text{'s provider} \end{cases}$$

It is easy to see that $1-p\,\mathcal{P}(Y_i)$ is the probability of X *not* being selected through provider Y_i, including the case where Y_i itself is not selected. Thus, the product in the equation is the probability that X is not selected through any of its providers.

The expected size of a sampled topology can be computed by $\sum_X \mathcal{P}(X)$. In Fig. 7, we plot the number of nodes in the sampled topologies by HBR0 with different sampling rates (from 0.1 to 1). From this plot, one can get a sense of the sampling rate needed to create a sampled topology of a desired size. Each point in the plot is the average over 100 runs with different randomization seeds. We also show the 95% confidence interval for each point, which is very close to the average. This is an indication that the sampling is fairly robust, and thus, insensitive to the randomization seed.

HBR captures the evolution of the Internet tiers fairly well. For a macroscopic validation, we investigate whether HBR respects the trend of the Internet instances in terms of the number of nodes per tier. We define tier-1 ASes to be the top-clique ASes, and tier-n ASes to be all ASes that are at least n-1 hops away to any top-clique AS via only customer-to-provider links. In Fig. 8, we plot the number of nodes per tier by our sampling method HBR0 with sampling rates from 0.1 to 1. We can see that the majority of the nodes are in tier-2, tier-3 and tier-4. This matches the case in real Internet instances, which is shown in Fig. 9. The most interesting observation is that, the number of tier-2 and tier-3 ASes are initially similar in the Internet, but the tier-3 ASes gradually out-number tier-2 ASes. Our hierarchy-based reduction captures this evolution very well.

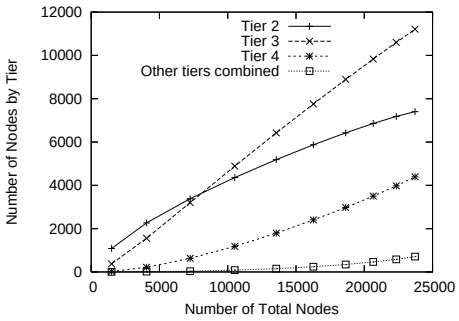


Figure 8: Size per tier: HBR0

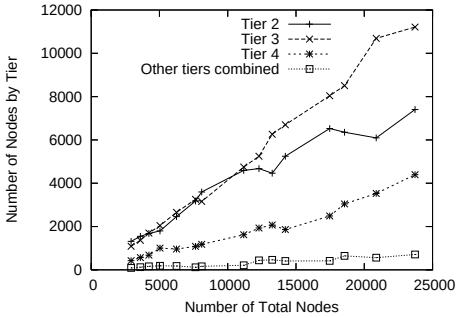


Figure 9: Size per tier: Internet

5.2 Topological properties

We compare a number of topological properties between sampled topologies and the real Internet topologies with the same size. We compare 6 sampling techniques: HBR0, HBR1, HBR2, HBR3, DDRV and DDRE. The starting point is a real Internet instance obtained on Dec 1, 2006 from Oregon Routeviews. We vary the sampling rate p from 0.1 to 1 to get smaller scale topologies with different sizes.

Overview of results: We find that **HBR0 and HBR1 are the best sampling methods**. HBR2 performs adequately, but consistently worse than HBR0. Finally, HBR3, DDRV, and DDRE perform significantly worse with regards to most of the metrics.

We now provide the comparison of these methods in detail, and present some intuition on the performance at the end. Due to space limitations, we can not show all the metrics that were used or provide intuitive explanations for the results in every case.

Number of Edges. The number of edges in a graph of a given size represents the density of a graph. In Fig. 10, we plot the number of edges against the number of nodes in the topologies. According to this figure, through the years from 1997 to 2006, the number of edges grows almost linearly with the number of nodes in the historical Internet instances. Reduction methods HBR0, HBR1 and HBR2 follow the Internet data nicely while HBR3, DDRV and DDRE deviate from the evolution of Internet data.

Degree Distribution. The degree distribution of the AS-level Internet topology is known to follow a power-law with a correlation coefficient larger than 99% [16], especially if we focus on customer-provider edges [23]. We calculate the power-law correlation coefficient for the complementary cumulative distribution (CCDF) function on the node degrees of each topological instance. In Fig. 11, we see that all Internet instances from the Oregon Routeviews follow power-law degree distributions. Topologies sampled with HBR0 and HBR1 follow the Internet instances very well ($\geq$99%) until the sizes drop to 1/8th of the initial size.

Assortativity. The *assortativity coefficient* r of a topology is defined as the Pearson's correlation coefficient of node degrees between all pairs of connected nodes. Intuitively, r captures the tendency of the nodes to attach to nodes with similar (*assortative mixing*, $0 < r \leq 1$) or different degrees (*disassortative mixing*, $-1 \leq r < 0$). In Fig. 12, we plot the r values for all historical Internet instances as well as for the one sampled by our reduction methods. We find that r in the Internet instances is fairly stable at approximately -0.2. Among all the reduction methods, HBR1 works best: the r values from HBR1 graphs follow the Internet values until the size of topology is reduced to 1/8 of the initial size.

Degree Entropy. We define the *degree entropy* $\mathcal{H}$ of a topology as $\mathcal{H} = -\sum_k P(k) ln P(k)$, where $P(k)$ is the probability that a randomly selected node has a degree k in this topology. The degree entropy is a measure of the degree randomness of graphs. In Fig. 13, we plot $\mathcal{H}$ values for all topologies. $\mathcal{H}$ for the Internet instances is fairly stable at about 1.6. $\mathcal{H}$ for topologies produced by HBR0 and HBR1 are very stable and close to that from the Internet.[1] On the other hand, HBR3, DDRE and DDRV perform badly as the degree entropy drops sharply in the sampled topologies that they produce.

Average clustering coefficient. We examine the *clustering coefficient* which has been used to characterize and compare generated and real topologies [25]. Intuitively, the clustering coefficient captures how tightly connected is the one-hop neighborhood of a node. For a node v_i with $n_i > 1$ neighbors, the clustering coefficient of v_i is $\gamma_i = \frac{m}{m_{max}}$, where $m_{max} = \frac{n_i(n_i-1)}{2}$, and m is the number of edges between these neighbors. A clustering coefficient of exactly one means that the neighborhood is a clique. The *average clustering coefficient* $\overline{\gamma}$ is the average γ_i of all nodes in the topology. In Fig. 14, we plot the average clustering coefficient against the number of nodes. For Internet instances before 2001 (there were about 8,000 ASes at that time), $\overline{\gamma}$ grows as the size of the topology grows. However, after 2001, $\overline{\gamma}$ slowly decreases. An investigation of this intriguing trend is outside the scope of this paper. We limit ourselves to observing that HBR0 follows the most recent (2001 to 2006) trend of Internet the best, although HBR1, HBR2, and DDRV are not far behind.

AS Path Length. The AS path length d_{AB} from AS A to AS B in a topology with routing policies is defined as the shortest steady-state AS path length from A to B consistent with the routing policies. Many previous studies only consider the shortest distance without policies, and they may underestimate the AS path length. The average AS path length is the average of d_{AB} for all AS pairs. The number of AS pairs in an n-node topology is $n(n-1)$. Note that in an AS topology with routing policies, d_{AB} is not always the same as d_{BA}. In contrast, d_{AB} is always the same as d_{BA} if no policy is considered. In Fig. 15, we plot the average AS path length for each topology instance. We see that, the average distance between ASes is slowly increasing from 2001 when the size of Internet was about 8,000 ASes. The topologies produced by HBR0, HBR1, HBR2 and DDRV follow this trend very well.

Towards a systematic comparison. To go beyond the visual comparison, we use the *Residual Sum of Squares (RSS)* to quantify the performance of a sampling method over all levels of reduction for each graph metric. First, for each graph metric and each level of reduction, we calculate the difference (or residue) between a reduced topology and a real topology of the same size. We then calculate the sum of the squared distances for each real topology instance that we have from Dec 2000 to Dec 2006. We normalize these sums with the largest one among the different reduction methods. Each number (except from the "Avg" column) in Table 3 denotes one such normalized RSS. In the table, the smaller the value, the better the method is regarding that graph metric. In the "Avg" column, we calculate the average normalized RSSes for each reduction method. As shown in Table 3, HBR0 and HBR1 seem to have better overall performance than all other methods.

We attempt to explain intuitively the observed performance.

a. Among all HBR methods, HBR1 performs best in assortativity. We explain this by recalling that top-clique nodes are high degree nodes which typically connect to many one-degree nodes [44]. By not selecting all the top-clique nodes,

[1] Note that the metric seems to be more sensitive to "noise": even for the real topologies the variation is very large and varies in a non monotonic way from 0.2 to 0.3 and back to 0.2. Thus, a variation of 0.1 for this metric from the sampled topology is considered very small.

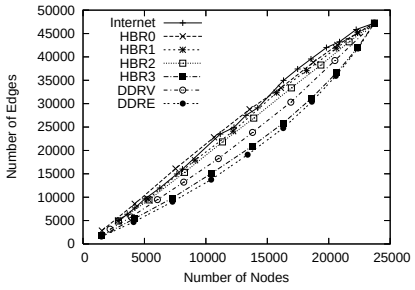


Figure 10: Number of edges

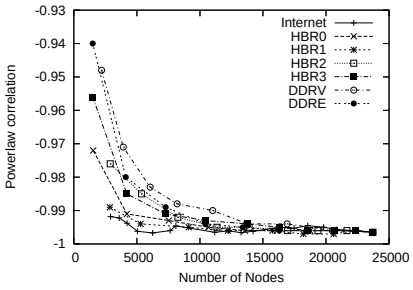


Figure 11: Power-law correlation

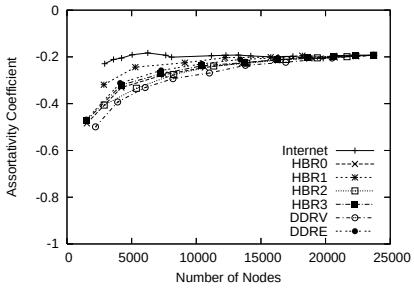


Figure 12: Assortativity Coefficient.

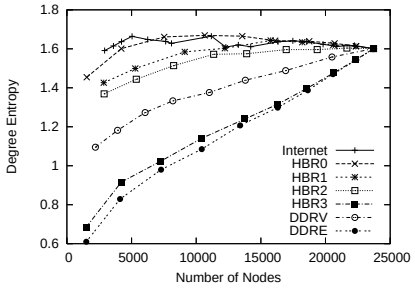


Figure 13: Degree Entropy

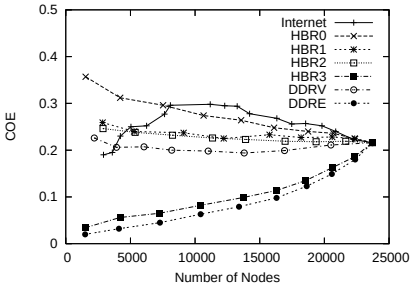


Figure 14: Average Cluster Coefficient

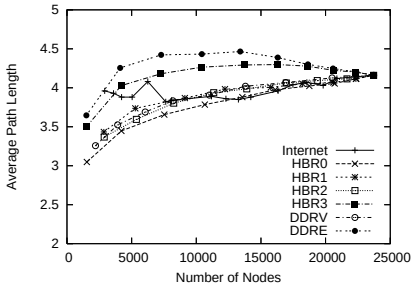


Figure 15: Average Path Length

HBR1 effectively reduces the number of "strongly disassortative" edges, and thus increases to an assortativity value closer to that of the Internet.

b. HBR3 performs poorly because it "under-samples" the number of edges. Recall that HBR3 keeps only the customer-provider edges along which a provider selected a customer: the edge is not included in the reduced graph, if the customer was selected through other providers. As we see in Fig. 10, the reduced graph has fewer edges compared the real instance.

c. DDRE and DDRV perform poorly. We attribute their poor performance to the fact they do not produce a BGP-connected topology "naturally". In those two methods, we are forced to pick the largest BGP-connected component, which may contribute to the observed deviation.

Due to space limitations, we show a subset of the graph metrics that we investigated. The set of metrics shown has been widely used to establish graph similarity in topology generation studies [7] [32]. With these metrics, we hope to capture the essence of the graph, which ultimately affect the routing protocol behavior. We discuss the relationship between the graph metrics and the protocol performance in Section 5.4.

5.3 Evaluation Using BGP Performance Metrics

We evaluate our reduction methods by conducting BGP performance studies. All our experiments show that the BPG performance on reduced topologies faithfully reflects its performance on real topologies. Due to space limitations, we can only show a few of our experiments. We first obtain a real Internet instance from Dec 1, 2006 with 23,718 ASes. To explore the limits of our approach, we target a reduction to approximately 20% of the initial number of nodes. Thus, we use HBR1 with a sampling rate $p = 0.25$ and parameter $MinS = 2$. The sampled topology contains 4,577 ASes, 19% of the size of the original topology. We also obtain four Internet topologies in history with similar size: *OIX19981201* with 4,233 ASes, *OIX19990215* with 4,536 ASes, *OIX19990301* with 4,627 ASes and *OIX19990401* with 4,761 ASes. The yyyymmdd after "OIX" indicates the data collection date. We then configure our SSFNET simulator with policies as in Section 3. Each AS advertises a prefix to the network. After convergence, the origin AS will withdraw the same prefix. For each prefix, we calculate its *average convergence time* (ACT) for both the *UP* and *DOWN* phases as in Section 3. Finally, the distribution of ACT of all ASes is plotted in Fig. 16 for the UP phase and in Fig. 17 for the DOWN phase.

BGP simulations on HBR topologies give similar results compared to real Internet topologies. We see that the ACT distribution of the sampled topology follows that of the real topologies reasonably well. For both the Internet instances and the sampled instance, ACT ranges from about 0.3 to 10 seconds, and about 1 to 5 seconds for the *DOWN* and *UP* phases, respectively. Note that the worst convergence time can be several minutes in our experiments.

Revisiting the importance of policy-aware BGP simulations. Interestingly, the upper bound on ACT is higher in *UP* than in *DOWN*, in our policy-aware simulations. This is

Table 3: Reduction Performance Comparison. Smaller number means better performance.

	Edges	PLcor	Assor	DegEn	Clust	Len	Avg
HBR0	0.018	0.080	0.476	0.007	0.023	0.050	0.109
HBR1	0.029	0.138	0.076	0.008	0.083	0.070	0.067
HBR2	0.082	0.173	0.541	0.028	0.102	0.074	0.167
HBR3	0.792	0.337	0.389	0.873	0.836	0.567	0.632
DDRV	0.319	1.000	1.000	0.234	0.194	0.087	0.472
DDRE	1.000	0.279	0.207	1.000	1.000	1.000	0.748

because the majority of updates in an *UP* phase are advertisements, and the majority of updates in a *DOWN* phase are withdrawals since path exploration is limited when policies are considered. Note that MRAI only applies to advertisements but not withdrawals. Therefore, the convergence is delayed in an *UP* phase, but not as much in an *DOWN* phase.

The situation would change a lot in a no-policy simulation: the average convergence time would be much higher in a *DOWN* phase than in a *UP* phase, because a *DOWN* event would result in excessive path exploration with many "stale" BGP advertisements, if policies are not considered. MRAI would delay these advertisements as well. This qualitative difference confirms the importance of considering routing policies for realistic BGP simulations.

5.4 Discussion

How small can we go? Ideally, we would like to run our simulations with the smallest topology that represents the Internet. At the same time, the more we reduce the topology, the more likely it is to deviate from the real topology. This does not only have to do with the abilities of the sampling method: sampling a topology has some inherent limitations. The randomness of the process becomes more evident in a small size graphs. It probably does not make sense to talk about a power-law degree distribution, when a topology has less than a few hundred nodes. Or pushing this to the extreme: can we have an Internet-like graph with three nodes? Thus, we leave the decision of choosing the right size to the researchers that will use our methods.

Which should be the starting topology? This is, in fact, a key desirable property of our work: it can start from the most complete and current topology at the time of the study. This is especially useful since obtaining the complete topology remains a moving target despite a flurry of efforts [9][49] [43][13] [32] [10] [11] [23] [50].

Mapping graph metrics to network performance metrics. In the previous two sub-sections, we have shown the quality of our sampling methods in terms of both graph metrics and BGP performance metrics. An interesting question is whether we can establish a mapping between these two types of metrics, namely identifying how graph metrics affect the BGP performance metrics. Since graph metrics are generally much easier to calculate than BGP performance metrics on large topologies, such a mapping would be very useful. Although some simple relationships may be easier to establish, an exhaustive and systematic mapping is far from trivial. First, neither the graph metrics, nor the BGP performance metrics are independent: changing one graph metric very often changes other graph metrics as well. For example, increasing the average degree in a given topology is bound to affect the diameter. Second, such a mapping does not seem to be one-to-one: multiple graph metrics can affect a single BGP metric, and a single graph metric can affect multiple BGP metrics. Thus, we leave this non-trivial question as future work.

6. SIMULATION TIME REDUCTION

The ultimate goal for using a smaller-size topology is the reduction of the computing resources for simulations. In this section, we examine the simulation time with different sizes of topologies and compute the CPU time savings. We also make an interesting observation: **taking routing policies into account not only makes BGP simulations more realistic, but also computationally less expensive.**

We use here, the same simulation configurations that were used in SSFNET as in Section 5.3. We take a number of real topologies with sizes from 3,000 ASes to 12,000 ASes. We also have a few sampled topologies from HBR1 with sizes from 1,000 ASes to 3,000 ASes. Each AS is allowed to announce one prefix and, after the network converges, withdraw it. We distribute the simulation work to about 100 workstations with Pentium IV 3.0GHz dual core CPU and measure the total simulation time.

In Fig. 18, we plot the required CPU time against the number of ASes in the topology. An interesting finding here is that BGP simulations with routing policies are computationally less expensive. A policy-aware topology often only requires 1/3rd to 1/20th of the CPU time compared to a no-policy topology of the same size. The larger the network, the larger the difference. This is because routing policies limit the path exploration in the *DOWN* phases and thus, reduce the number of events in the simulation. Policies also reduce the required memory in a simulation. With a 2GB memory, we are able to simulate on a 12,000-AS policy-aware topology, but only a 4,000-AS no-policy topology.

The increase of the simulation time as a function of the network size can be well approximated by a cubic function for both types of topology. However, notice that there is a huge difference in the constant parameters of the related cubic functions, see Fig. 18. Thus, if we reduce the size

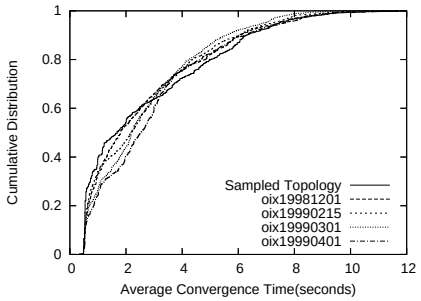


Figure 16: Convergence Time, *UP*

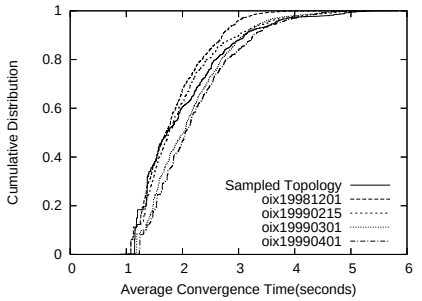


Figure 17: Convergence Time, *DOWN*

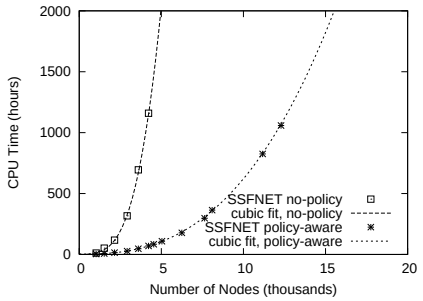


Figure 18: Simulation execution time

to 20% of the original topology as we did in Section 5, we only need 0.8% of the original simulation time. Combining with the savings from using a policy-aware model, we can effectively cut down the required CPU time by 3-4 orders of magnitude.

How large can a BGP simulations be? One may ask why we need a small topology after all, since we seem to be able to simulate on a 12,000-AS policy-aware topology. Let us discuss this more carefully. *First*, a single run takes a very long time for a large topology. For example, had we not distributed our experiment to about 100 workstations, it would have taken **roughly 42 days for a single run** on a 12,000-AS policy-aware topology! This can be prohibitive when a study requires a large *total* number of runs, which can be as much as 200K runs [22]. *Second*, in our simulations, we were forced to limit the number of advertised prefixes in large topologies. In Section 5.3 and Section 6, we only allow an AS advertise its prefix after the previous AS has withdrawn its prefix. Therefore, at any given time, no more than one prefix is "alive". This practice reduces the memory requirements and allows us to distribute the simulation work to multiple machines, but, we are not able to simulate interesting BGP phenomena. If each AS advertises its prefix at the same time, detailed simulations cannot be run even on the smallest measured Internet topology (3,000 ASes). We recommend the following solution: take a real Internet instance of medium size (say 5,000 ASes), and reduce it with HBR0 to 20% or 1,000 ASes, and use this topology.

7. CONCLUSION

As our key contribution, we develop a sampling method to generate provably legitimate, namely BGP-connected and relationship loop-free, topologies. We formally prove that our method produces a legitimate topology, if the initial topology is legitimate. To our best knowledge, this is the first method with such a hard guarantee.

a. Our approach can reduce a topology successfully to a fraction of its initial size. We validate the realism of our sampled topologies using: (a) an extensive list of graph metrics, and (b) actual BGP performance evaluations.

b. With our approach, we can reduce the simulation time by 3-4 orders of magnitude. This is a significant speed-up, especially if we consider the total computational complexity of an in-depth BGP study.

c. Policy-aware simulations are usually less computationally intensive. In other words, using policy-aware topologies is more realistic and less time-consuming at the same time.

Our work enables an important capability for evaluating inter-domain routing protocols effectively. Towards that goal, we intend to release: (a) our HBR tool, and (b) a series of sampled graphs, in an effort to establish a badly-needed community-wide simulation benchmark.

8. REFERENCES

[1] Genesis project, www.genesis-sim.org/genesis.

[2] http://www.ece.gatech.edu/research/labs/maniacs/bgp++/.

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

[4] Oregon routeviews project, http://www.routeviews.org.

[5] G.D. Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between Autonomous Systems, 2003.

[6] A. Bremler-Barr, Y. Afek, and S. Schewarz. Improved bgp convergence via ghost flushing. In *Infocom*, 2003.

[7] T. Bu and D. Towsley. On distinguish between internet power law topology generators. In *Infocom*, 2002.

[8] J. Chandrashekar, Z. Duan, Z. Zhang, and J. Krasky. Limiting path exploration in bgp. In *Infocom*, 2005.

[9] H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger. Towards capturing representative AS-level Internet topologies. *Computer Networks*, 44(6):737–755, 2004.

[10] H. Chang, S. Jamin, and W. Willinger. To Peer or not to Peer: Modeling the Evolution of the Internet's AS Topology. In *Infocom*, 2006.

[11] R. Cohen and D. Raz. The Internet Dark Matter – on the Missing Links in the AS Connectivity Map. In *Infocom*, 2006.

[12] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, kc claffy, and G. Riley. As relationships:inference and validation. *ACM SIGCOMM CCR*, January 2007.

[13] X. Dimitropoulos, D. Krioukov, and G. Riley.

Revisiting Internet AS-Level Topology Discovery. In *PAM*, 2005.

[14] X. Dimitropoulos and G. Riley. Efficient large-scale bgp simulations. *Computer Networks*, (12):2013–2027, 2006.

[15] X. Dimitropoulos and G. Riley. Modeling autonomous-system relationships. In *Workshop on PADS*, 2006.

[16] M. Faloutsos, P. Faloutsos, C. Faloutsos, and G. Siganos. Power-laws of the Internet topology. *IEEE/ACM Trans. on Networking*, 1(4):514–524, 2003.

[17] N. Feamster, H. Balakrishnan, and J. Rexford. Some foundational problems in interdomain routing. In *Hotnet*, 2004.

[18] L. Gao. On inferring autonomous system relationships in the Internet. In *IEEE Global Internet*, 2000.

[19] L. Gao, T. Griffin, and J. Rexford. Inherently safe backup routing with bgp. In *IEEE INFOCOM*, 2001.

[20] L. Gao and J. Rexford. Stable internet routing without global coordination. In *ACM Sigmetrics*, 2000.

[21] M.R. Garey and D.S Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Co., 1979.

[22] T. Griffin and B. Premore. An Experimental Analysis of BGP Convergence Time. In *IEEE ICNP*, 2001.

[23] Y. He, G. Siganos, M. Faloutsos, and S. Krishnamurthy. A Systematic Framework for Unearthing the Missing Links: Measurements and Impact. In *USENIX NSDI*, 2007.

[24] B. Hummel and S. Kosub. Acyclic type-of-relationship problems on the internet: An experiemental analysis. In *ACM IMC*, 2007.

[25] S. Jaiswal, A. Rosenberg, and D. Towsley. Comparing the structure of power law graphs and the Internet AS graph. In *ICNP*, 2004.

[26] C. Jin, Q. Chen, and S. Jamin. Inet: Internet topology generator. Technical report, 2000.

[27] E.G. Coffman Jr, Z. Ge, V. Misra, and D. Towsley. Network resislience: Exploring cascading failures within bgp. In *Infocom*, 2002.

[28] G. Karypis and V. Kumar. A fast and high quality scheme for partitioning irregular graphs. *Technical Report, Department of Computer Science, University of Minnesota: 95-035*, 1995.

[29] V. Krishnamurthy, M. Faloutsos, M. Chrobak, L. Lao, J-H. Cui, and A.G. Percus. Reducing large internet topologies for faster simulations. In *Networking*, 2005.

[30] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary. The impact of internet policy and topology on delayed routing convergence. In *Infocom*, 2001.

[31] P. Mahadevan, D. Krioukov, K. Fall, and A. Vahdat. Systematic topology analysis and generation using degree correlations. In *ACM Sigcomm*, 2006.

[32] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, kc claffy, and A. Vahdat. The Internet AS-Level Topology: Three Data Sources and One Definitive Metric. *ACM SIGCOMM CCR*, January 2006.

[33] Z. Mao, R. Govindan, G. Varghese, and R. Katz. Route flap damping exacerbates internet routing convergence. In *ACM Sigcomm*, 2002.

[34] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite:an approach to universal topology generation. *MASCOTS*, 2001.

[35] W. Muhlbauer, A. Feldmann, O. Maennel, M Roughan, and S. Uhlig. Building an AS-Topology Model that Captures Route Diversity. In *ACM Sigcomm*, 2006.

[36] W. Muhlbauer, S. Uhlig, B. Fu, M. Meulle, and O. Maennel. In Search for an Appropriate Granularity to Model Routing Policies. In *ACM Sigcomm*, 2007.

[37] R. Oliveira, B. Zhang, and L. Zhang. Observing the evolution of internet as topology. In *ACM Sigcomm*, 2007.

[38] D. Pei, M. Azuma, D. Massey, and L. Zhang. Bgp-rcn: Improving bgp convergence through root cause notification. *Computer Networks*, 48(2):175–194, 2005.

[39] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, F. Wu, and L. Zhang. Improving bgp convergence through assertions approach. In *Infocom*, 2002.

[40] BJ Premore. http://www.ssfnet.org/Exchange/gallery/asgraph/index.html.

[41] J. Qiu. http://www.bgpvista.com/simbgp.php.

[42] B. Quoitin and S. Uhlig. Modeling the routing of an autonomous system with c-bgp. *IEEE Networks*, 19(6), 2005.

[43] Y. Shavitt and E. Shir. DIMES: Let the Internet Measure Itself. *ACM SIGCOMM CCR*, October 2005.

[44] G. Siganos, S. Tauro, and M. Faloutsos. Jellyfish: A conceptual model for the as internet topology. *Journal of Comm. and Networks*, 8(3):339–350, 2006.

[45] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz. Characterizing the internet hierarchy from multiple vantage points. In *Infocom*, 2002.

[46] W. Sun, Z. Mao, and K. Shin. Differentiated bgp update processing for improved routing convergence. In *IEEE ICNP*, 2006.

[47] F. Wang and L. Gao. Inferring and characterizing internet routing policies. In *ACM IMW*, 2003.

[48] J. Xia and L. Gao. On the evaluation of as relationship inferences. In *IEEE Globecom*, November 2004.

[49] B. Zhang, R. Liu, D. Massey, and L. Zhang. Collecting the Internet AS-level Topology. *ACM Sigcomm CCR*, January 2005.

[50] Y. Zhang, Z. Zhang, Z. Mao, Y. Hu, and B. Maggs. On the impact of route monitor selection. In *ACM IMC*, 2007.