Building an AS-Topology Model
that Captures Route Diversity

Wolfgang Mühlbauer
Anja Feldmann
TU München

Olaf Maennel
Matthew Roughan
University of Adelaide

Steve Uhlig
Université catholique
de Louvain

ABSTRACT

An understanding of the topological structure of the Internet is needed for quite a number of networking tasks, e.g., making decisions about peering relationships, choice of upstream providers, inter-domain traffic engineering. One essential component of these tasks is the ability to predict routes in the Internet. However, the Internet is composed of a large number of independent autonomous systems (ASes) resulting in complex interactions, and until now no model of the Internet has succeeded in producing predictions of acceptable accuracy.

We demonstrate that there are two limitations of prior models: (i) they have all assumed that an Autonomous System (AS) is an atomic structure — it is not, and (ii) models have tended to over-simplify the relationships between ASes. Our approach uses multiple quasi-routers to capture route diversity within the ASes, and is deliberately agnostic regarding the types of relationships between ASes. The resulting model ensures that its routing is consistent with the observed routes. Exploiting a large number of observation points, we show that our model provides accurate predictions for unobserved routes, a first step towards developing structural models of the Internet that enable real applications.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*; C.2.5 [**Computer-Communication Networks**]: Local and Wide-Area Networks—*Internet (e.g., TCP/IP)*
General Terms: Algorithms, Experimentation, Measurement
Keywords: BGP, inter-domain routing, route diversity, routing policies

1. INTRODUCTION

The Internet is composed of a large number of independently administered Autonomous Systems (ASes) coupled by the Border Gateway Protocol (BGP) into a single globe spanning entity. The structure of this interconnected system has been of some interest for a variety of reasons; most commonly, because its topology plays a significant role in determining the performance of the Internet, though pure scientific interest has played a substantial role in these investigations. Now, we propose that more direct use be made of

this information to predict the behavior of the Internet under specific conditions.

In the past, high-level features of the inter-domain topology have been used to make generic inferences about its behavior, e.g., power-law distributions [1] have been used to imply important "centralized" nodes (see [2] for a discussion of this issue). These types of generic inference are useful in terms of scientific understanding of the Internet as it evolves, but do not allow one to answer specific questions about the current Internet. We seek to be able to answer specific what-if questions, e.g., what if a certain peering link was removed, or what-if we change policies thus? In principle, knowledge of the Internet's inter-domain topology can be used to answer such questions, and the capability would provide great utility for providers. This is particularly true given that the focus for large providers has moved from simply providing connectivity, to maintaining contractual or business relationships that may require resilience despite changing traffic demands or link failures, in addition to supporting customers who demand more control over their traffic flows [3, 4].

Despite the requirements, current practice is quite limited. Often, the only available approach is "tweak and pray" [5, 6]; that is, providers make changes with limited ability to predict the results, and then observe to see if the desired effect occurred. We propose to build an AS-routing model which enables us to predict unobserved Internet paths with good accuracy.

It is known [7], that for the extracted model to be useful in prediction, it must be substantially better than those tested so far. Until now, models of the network structure have been predominantly inter-domain level models that do not worry about the details of the ASes [7–9]. However, ASes are not simple nodes in a graph — they are comprised of routers. The internal structure of an AS *does* matter. It influences inter-domain routing, for instance via hot-potato routing [10, 11]. Furthermore, there are multiple connections between ASes, typically from different routers, and this adds to the diversity of known routes [12]. Even where policy is uniform across an AS, internal features of the AS may result in different route choices for each router — this is a feature of BGP that allows behaviors such as hot-potato routing. Such diversity is commonly observed in public routing databases such as Route-views [13]. An AS which is a single node must always choose a single best path to pass to its neighbors, and therefore cannot represent this type of diversity.

In addition, inter-domain routing is controlled by diverse *policies*, decided locally by each AS, but acting globally across the entire system [14]. Hence the topology of the inter-domain graph is not, in itself, sufficient to make predictions about Internet routing. In addition, policies need to be considered. Many policy relationships may be described as "customer-provider" or "peer-peer",

and in these simple cases policies are enacted using simple, well-known filtering rules [15]. Several papers have discussed inference of these simple policy rules [16–18], but unfortunately, not all policies fit these simple rules: for instance, in some cases of multiple links between two ASes, the policies may vary even between links. Our approach to all of these issues is to remain agnostic about what practices occur or do not occur in the current Internet. We make minimal assumptions about inter-domain routing, and let the data speak for itself.

Of course, it is impossible to infer all of the internal details of an AS's policies. We are not seeking to reverse engineer the Internet. Our model does not necessarily correspond to the policies actually used by the ASes. Rather the results are analogous to the IGP (Interior Gateway Protocol) link weights inferred by Rocketfuel [19,20], which do not correspond to those of the real networks investigated, but are nevertheless useful in understanding intra-domain topologies. In this paper, we introduce policies into our AS-routing model with the goal of making predictions about the behavior of unobserved paths.

Likewise, we do not seek to reproduce a Rocketfuel-like detailed intra- and inter-domain connectivity map [20], as a significant part of this information is not used in determining routes. Rather, we shall build topological models incorporating intra- and inter-domain information at the minimum level of detail needed to explain the observed routing in the Internet. The resulting simplicity allows us to derive insight into the relationship between routing policies, path diversity, and the actual choice of the paths propagated across the Internet without having to model the complexity of the routing inside an AS [21]. It gives us the ability to determine precisely where internal details matter, and how much.

Our approach is based on the idea of building a topology and policy model that is consistent with the observed routing in the Internet. To this end we exploit BGP observations from more than thirteen hundred observations points (including Routeviews [13], RIPE [22], and a number of other sources). We separate these into two datasets: a training dataset, and a validation dataset. The training set is used to build a topology and policies consistent with observed routing. We do so using a set of simulation-based iterative refinement heuristics (described in Section 4) that introduce a minimal set of topology and policy changes required to match observed routing. We accommodate path diversity by creating multiple quasi-routers within each AS. A quasi-router represents a group of routers all making the same choice about best route, and so the "quasi-router topology" does not represent the physical router topology of a network, but rather the logical partitioning of its policy rules. Importantly, we try to minimize the assumptions we make about "likely" policies, e.g., we do not assume that relationships fall into neat categories. We find that we can build an AS-routing model that matches the training set *exactly*. However, remember this is not the real topology, and the fact that it can match the training set exactly is not sufficient to show that the results are of practical use. We test the usefulness of the model by making a set of predictions about routes, and validating them with the data excluded from training. We find that we can match the predictions down to the final BGP tie break in more than 80% of the test cases, see Section 5.

This paper is not (principally) concerned with modeling Internet routing dynamics. The dynamics are clearly important, but considerable effort has already gone into such modeling e.g., [14, 23–27]. In our first prototype for predicting Internet behavior we model the equilibrium behavior of this system, for the (vastly) predominant case that a stable routing solution exists. It is these equilibrium behaviors that are of most interest for the questions posed earlier. At-
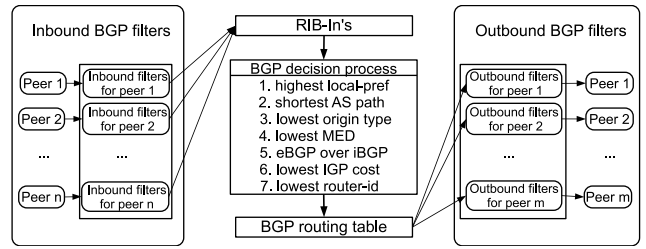

Figure 1: Operation of a BGP router.

tempting to model, and incorporate dynamic information into our predictions is a worthwhile goal (for example see [28]), but beyond the scope of this paper.

To summarize our contributions: We present a methodology for deriving an AS-routing model that can reproduce all observed AS-paths, and predict unobserved routes with reasonable accuracy. Furthermore we show the importance of considering more than one router per AS and accommodate a wide range of policies. Another major distinction of our work is that we use simulations to refine our model based on a large set of BGP data from diverse vantage points but evaluate the results using a separate set of BGP vantage points.

2. REVIEW OF INTER-DOMAIN ROUTING

BGP routers exchange routing information over BGP sessions. External BGP (eBGP) sessions are established over inter-domain links, i.e., links between two different ASes (BGP peers), while internal BGP (iBGP) sessions are established between the routers within an AS. Through its BGP sessions, each router receives and propagates BGP routes for destination prefixes. A BGP router processes and generates route advertisements as shown in Figure 1. Administrators specify input filters per BGP peer, which are used to discard unacceptable incoming BGP advertisements. Once a route advertisement is accepted by the input filter, it is placed together with the routes originated at this router in the incoming Routing Information Base (RIB-In) for the peer, possibly after some of the route attributes have been modified according to the local routing policies. Next, the BGP decision process is used to select the *best route* for each prefix from among the available routes. This route is then placed into the BGP routing table, which we will also refer to as the *RIB-Out*. Finally, administrators may specify output filters for each peer, which are used to decide which best routes to propagate to a BGP neighbor.

The BGP decision process consists of a sequence of elimination steps. Its final goal is to select a single best route for any given prefix. For this purpose the BGP decision process considers several of the BGP routes attributes. One of the first attributes is *local-preference* (in short, *local-pref*). As local-pref is a non-transitive attribute, it can be used to locally rank routes. The next BGP attribute examined by the BGP decision process is the *AS-path*. An *AS-path* contains the sequence of ASes that a route crossed to reach the current AS. Routes with shorter AS-paths are preferred. Next in the evaluation process is the *multi-exit-discriminator* (in short, *med*). This attribute is used to rank routes received from the same neighbor AS, but it can also be used across neighbors. Then the decision process ranks routes according to the IGP cost of the intra-domain path towards the *next-hop*, preferring routes with smaller IGP cost. This rule implements hot-potato routing [29]. Finally, if there is still more than a single route left, the router breaks ties, for example by selecting the route to the neighbor which has the lowest router-id (typically one of its IP addresses).

196

Given a set of filters and policies, it is possible to simulate the propagation of BGP routes using simulators such as C-BGP [30]. C-BGP's model for the inter-domain routing protocol relies on the computation of the paths that routers know once the BGP routing has converged [23]. For this purpose, it models the propagation of BGP messages and reproduces the selection performed by each router [31].

3. DOMAINS AS SIMPLE NODES

In this section, we use measured routing data to illustrate the need to go beyond treating ASes as simple nodes in a graph. We first analyze the degree of route diversity present in the current Internet and then examine the limitations of single-node AS models for predicting path choices throughout the Internet accurately. The data shows that one must have a way to capture some internal details of routing at least for a subset of ASes.

3.1 BGP data set

There are many different techniques for collecting BGP feeds from an AS. One of the most common technique is to rely on a dedicated workstation running a software router that peers with a BGP router inside the AS. We refer to each peering session from which we can gather BGP data as an *observation point*, and the AS to which we peer as the *observation AS*.

We use BGP data from more than $1,300$ BGP observation points including those provided by RIPE NCC [22], Routeviews [13], GEANT [32], and Abilene [33]. The observation points are connected to more than 700 ASes, and in 30% of these ASes we have feeds from multiple different locations. As we are currently not yet interested in the dynamics of BGP we use a static view of the routes at a particular point in time. The table dumps provided by the route monitors are each taken at slightly different times. We use the information provided in these dumps regarding when a route was learned to extract those routes that were valid table entries on Sun, Nov., 13, 2005, at 7:30am UTC, and that were stable in the sense that they have not changed for at least one hour. In the future we are planning to also incorporate the AS-path information from BGP updates. Our dataset contains routes with $4,730,222$ different AS-paths[1] between $3,271,351$ different AS-pairs. We derive an AS-level topology from the AS-paths. If two ASes are next to each other on a path we assume that they have an agreement to exchange data and are therefore neighbors in the AS-topology graph. We are able to identify $58,903$ such edges. We identify level-1 providers by starting with a small list of providers that are known to be tier-1. An AS is added to the list of level-1 providers if the resulting AS-subgraph between level-1 providers is complete, that is, we derive the AS-subgraph to be the largest clique of ASes including our seed ASes. This means that the AS-graph contains edges for all level-1 AS-pairs. This results in the following 10 ASes being referred to as level-1 providers (174, 209, 701, 1239, 2914, 3356, 3549, 3561, 5511, 7018). Note, this list is not complete. However, all found ASes are well-known tier-1 provider. There are $7,994$ ASes that are neighbors of a `level-1` provider in the BGP graph. We refer to these as `level-2`. All other $13,174$ ASes are grouped together into the class `other`. Of the $21,178$ ASes $3,486$ provide transit for some prefixes in the sense that they appear at least once in the middle of an AS-path. Among those ASes that do not provide transit, called stub-ASes, we distinguish between those that are observed to have a single upstream provider (are single-homed) and those that have multiple providers (are multi-homed). We find

---

[1]We removed AS-path prepending to prevent distraction from the task of route propagation.
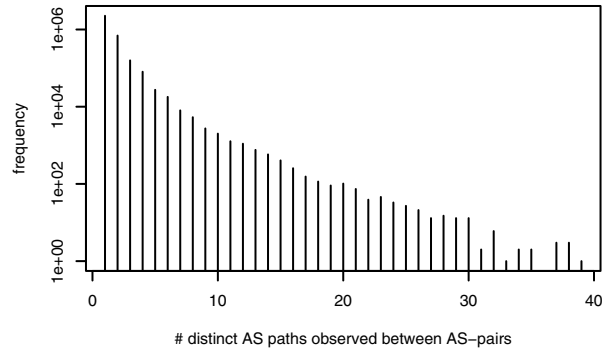

Figure 2: Histogram of # of distinct AS-paths.

that there are $6,611$ single-homed and $11,077$ multi-homed ASes. Single-homed ASes that do not provide transit only add limited information about the AS-topology as long as any path information gathered from prefixes originated at such stub-ASes is transfered to a prefix originated at its AS neighbor. Removing single-homed stub-ASes and AS-paths with loops from the AS-topology results in a graph with $14,563$ nodes and $52,288$ edges. Note, our data does not cover the complete AS topology [34] since not all AS relationships are observable in our data. There are relatively more observation points in the level-1 and level-2 ASes than in the other ASes. Therefore it is likely that AS-relationships involving level-2 providers are missing. Yet, their impact with regards to routing can be expected to be less significant.

3.2 Route diversity in the Internet

To investigate the significance of route diversity in the Internet we examine how many different routes can be seen for each originating and observation AS pair (over all prefixes advertised by the origin). Figure 2 plots a histogram of the number of distinct AS-paths using a logarithmic y-axis. Note, that for more than 30% of the AS-pairs we see more than one AS-path. Indeed, there are more than $5,000$ pairs with more than 10 different paths.

Each AS may originate multiple prefixes and an AS-path may be used by many prefixes. Indeed, we find that there are very popular AS-paths used by more than $1,000$ different prefixes while the number of AS-paths that are only used by a single prefix is less than 50%. When plotting the histogram of how many prefixes are propagated along an AS-path on a log-log plot, one can see a linear relationship (plot not shown). In terms of route diversity, we observe that most prefixes are only propagated through a single AS-path. Yet, there are quite a number of prefixes whose propagation samples the full path diversity between two ASes.

Obviously, one router per AS is not sufficient to capture the full diversity imposed by intra-domain routing. A single router can only propagate the route it chooses as best. With multiple routers each router within the AS can select its own best route and propagate it.

To motivate the need for modeling ASes with several routers, let us consider a concrete example from our data for the prefix `202.94.48.0/20` at AS 5511 shown on Figure 3.

AS 24249, which originates this prefix, is multi-homed to two ASes: AS 4694 and AS 4716. From these two providers the route is propagated to five level-1 providers: AS 2914, AS 3356, AS 3549, AS 3561, and AS 7911. Since AS 3356 propagates multiple AS-paths to AS 3356 it needs to be modeled by at least two different routers. Which route is propagated can depend on the specific setup within the AS. Yet, path diversity within the ASes is only partially responsible for the route diversity. Another reason is the large in-
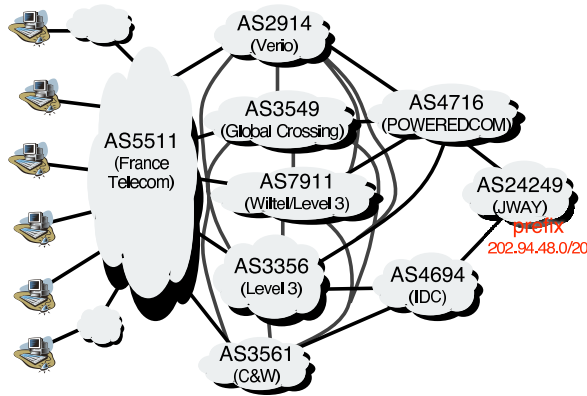
Figure 3: Example of path diversity.

Percentile	25	50	75	90	95	98	99	100
max # of unique AS-path	1	2	4	5	7	10	10	23

Table 1: Maximum route diversity received for all ASes.

terconnectivity in the core of the Internet; in this case 5 out of 8 AS-paths. Still AS 3356 needs eight routers to propagate all paths further downstream.

To judge how much of the path diversity is due to multiple routes per ASes rather than multiple routes from different ASes we determine the distribution of the maximum number of distinct unique paths each AS receives towards any destination prefix. This value is a lower bound on how many routers are needed inside an AS to propagate all these paths to downstream ASes. Table 1 shows the larger quantiles of this distribution. We observe that more than 50% of the ASes receive two unique AS-paths for at least one destination prefix, 10% more than 5, and 2% more than 10, respectively. This highlights the importance of not loosing such path diversity.

3.3 Route diversity in single router models

In the past, large-scale models of routing in the Internet frequently assume that each AS consists of a single router, e.g., [7]. To judge how appropriate this is for answering practical questions we now examine how accurately it can predict AS-path choices throughout the Internet.

We use the BGP simulator C-BGP [30] to compute AS-level paths on the AS-level graph after eliminating the stub-ASes. We originate one prefix per AS, resulting in 14,563 prefixes. Originating multiple prefixes per AS does not provide more information since at this point we do not consider per-prefix specific policies. To evaluate the quality of the model we compare the predicted and observed AS-paths. Table 2 summarizes the results. Not surprisingly we have agreement for only 23.5% of the AS-paths. The main problem is again that for slightly less than 50% of the prefix/observation point combinations the observing AS does not even learn the "correct" AS-path. For the remaining 50.6% only 4.7% of the incorrect decisions occur due to the shortest-path step of the BGP-decision process (Figure 1). If a router learns the "correct" route it seems to be able to choose the "correct" one in roughly 50% of the cases.

Today's Internet does not use shortest-AS-path routing as we assumed above. Most BGP peerings come with routing policies of which the most common ones can be classified as customer-provider and/or peering relationships. Relying on the BGP data we use a simple heuristic for inferring customer-provider relation-

Criteria	Shortest Path	Customer/ Peering Policies
AS-Paths which agree	23.5%	12.5%
AS-Paths which disagree due to	76.4%	87.5%
AS-path not available	49.4%	54.5%
shorter AS-path exist	4.7%	5.7%
lowest neighbor ID	22.2%	27.3%

Table 2: Agreement between predicted and observed AS-paths (single router per AS).

ship utilizing the valley-free assumption [15, 16, 18]. We start by declaring all links between the level-1 ASes as peering and then iteratively infer customer-provider relationships. We verify our classification by using data from several ASes whose peering policy we have access to. This results in 34,087 customer-provider peers, 7,290 peering relationships, and 640 siblings. All other edges cannot be classified. We then realized appropriate policies based on the local-pref BGP attribute and route filters[2] in the simulator and rerun the simulations. The results are fairly discouraging with only 12.5% agreement on the AS-paths. The main problem is that for a lot of the prefix/observation point combinations the observing AS does not learn the "correct" path. Overall, this indicates a low accuracy for AS-path prediction, if an AS-routing model is solely based on AS-relationship inference.

Unfortunately, an agreement of less than 1/4 for the selected best AS-paths and just above 1/2 for the available AS-path, while not too bad, is not sufficient to answer, e.g., what-if questions such as how the routing in the Internet would change if a peering is added or de-peering of some provider occurs. Accordingly, we in this paper tackle the task of deriving more accurate models. In order to account for route diversity and to predict unobserved Internet paths, we allow for *routing policies* as well as for *multiple routers* inside ASes.

4. METHODOLOGY

The goal of this section is to propose a methodology for building an AS-routing topology model that captures the outcome of the routing policies and the internal structure of all ASes from observed BGP data in order to answer practical questions about routing. The example question we use to highlight the capabilities of our model concerns predicting Internet path choices for previously unobserved AS-paths.

We consciously choose an approach which allows for multiple routers, so called quasi-routers, within an AS, and that is agnostic about inferred relationships such as customer-provider and/or peering relationships. After all, the real world knows many variants of such relationships [35]. We take the approach of modeling what we actually observe. In this manner we can avoid many potential pitfalls that arise from incomplete assumptions or trying to press BGP into some fixed schema.

In the following we first introduce the components of our AS-routing model and then show how one can evaluate its predictive capabilities. Next, we introduce our principle approach and then give an example of how to use it for deriving an AS-routing model from gathered BGP data. Finally, we discuss how to use the model for predicting previously unconsidered AS-paths, and how to improve it for previously unconsidered prefixes.

---

[2]We treat siblings in the same manner as peerings relationships and set the same local-preference for unknown AS edges as for peerings.

4.1 Components of the AS-routing model

The AS-routing model should be capable of predicting AS-level paths, as used in the Internet, and so it needs to have a notion of inter-domain connectivity. Since it should capture the impact of intra-domain routing it needs to account for the diversity and connectivity within each AS. Furthermore, as BGP is used to implement policies, we must accommodate this in our model.

Based on these criteria and the fact that we do not yet consider BGP dynamics, we propose to use a class of topology models that can also be used as input to the C-BGP simulator [21, 30]. C-BGP is designed for studying the propagation of routing information along a topology model that consists of multiple ASes. It allows multiple routers within an AS, the setup of BGP sessions between any pair of routers, and supports iBGP as well as eBGP. To propagate routing information, C-BGP models the propagation of BGP messages and executes the BGP decision process based on routing policies. Hence, C-BGP's routing model addresses all our requirements. Since C-BGP only computes the steady-state choice of the BGP routers after the exchange of the BGP messages has converged and not the whole state machine of the BGP routing protocol, it is thus possible to perform large-scale simulations for single prefixes on topologies with more than $16,500$ routers split among $14,500$ ASes in $2 - 45$ minutes with 200 MB $- 2$ GB memory consumption depending on the complexity of the routing policies. C-BGP's capability of simulating large-scale propagation of BGP routes not only allows us to test how accurately the model can answer our example question, it also enables us to refine an AS-routing model incrementally.

While deriving the model we make the simplification that we only originate one prefix per AS. This allows us to address questions regarding path diversity while keeping the model manageable. For similar reasons we again exclude stub-ASes but keep their AS-path to ensure that we do not loose any path information.

We capture the inter-domain connectivity via an AS-topology graph as extracted from the BGP data. In order to represent the intra-domain routing diversity we allow each AS to consist of multiple quasi-routers. A *quasi-router* represents a group of routers within an AS all making the same choice about best route, and so the "quasi-router topology" does not represent the physical router topology of a network, but rather the logical partitioning of its policy rules. Each edge (AS 1, AS 2) of the AS-topology is realized by establishing a BGP session between one or more quasi-routers from AS 1 to one or more quasi-routers from AS 2. Propagation of routes can be restricted by applying route filters and/or by introducing other routing policies.

4.2 Evaluating prediction

C-BGP enables us to predict, using an AS-routing model as input, the AS-path along which the routing information for any prefix, originated at any node, is propagated to any other node.

For a fair evaluation we need one dataset to derive the AS-routing model, called `training`, and another separate one, called `validation`, to evaluate the quality of the AS-routing model. We divide the available BGP data randomly into two subsets by assigning observation points to either subset. This places *all* paths, observed at an observation point, into one of the two subsets. The `training` set is then used to derive the AS-routing model while the `validation` set is used for evaluation purposes.

An alternative way of slicing the data is to split the set of AS-paths according to the originating ASes into two subsets. One can then compare how well an AS-routing model derived from a subset of the prefixes predicts the AS-paths for another set of prefixes. Furthermore, one can combine both approaches and partition the
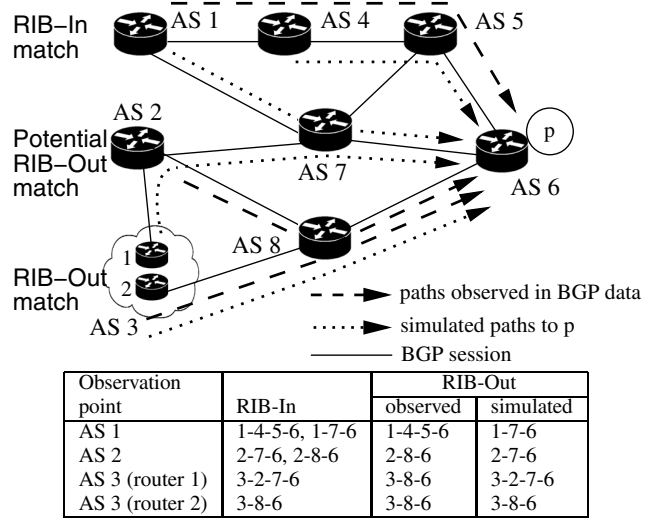

Observation		RIB-Out	
point	RIB-In	observed	simulated
AS 1	1-4-5-6, 1-7-6	1-4-5-6	1-7-6
AS 2	2-7-6, 2-8-6	2-8-6	2-7-6
AS 3 (router 1)	3-2-7-6	3-8-6	3-2-7-6
AS 3 (router 2)	3-8-6	3-8-6	3-8-6

Figure 4: Metrics - Example.

obtained `training` or/and `validation` subsets according to the originating AS.

The evaluation proceeds by executing a C-BGP simulation for each prefix and then comparing the predicted AS-path according to the AS-routing model with the actual observed AS-path in the Internet. In this manner we can evaluate the predictive capabilities of the model. Since routing decisions are determined independently for each prefix we run a separate simulation for each prefix.

After the simulation runs one has access to the routing information base (RIB) of all quasi-routers. Therefore, we can now compare for each AS the AS-path that is recorded in the BGP data to the AS-paths chosen in the simulation. Some mismatches have to be expected. We measure the degree of mismatch by determining if a route with the AS-path is received by a quasi-router within an AS (RIB-In), if it is selected by a quasi-router (RIB-Out), or if it could have been selected but was not due to an "unlucky" decision in the last step of the BGP decision process, the tie-breaker (potential RIB-Out). More precisely we use the following metrics:

RIB-In match: The observed route at an observation point is contained in the simulated RIB-In for at least one quasi-router in the observed AS. Note, this does not say that the simulated and observed RIB-Ins are the same. as the observation point only sees the best routes advertised by the monitored AS. The metric provides an upper bound on the prediction accuracy — we can only expect a RIB-Out match if we have a RIB-In match. A RIB-In match is a necessary but not sufficient condition for a RIB-Out match.

Potential RIB-Out match: A RIB-In match where in the process of choosing a best route the observed route is eliminated in the last tie-breaking step of the BGP decision process in the simulation ("Lowest Neighbor IP address").

RIB-Out match: At least one quasi-router in the AS has selected the route with the observed AS-path as its best route and propagates it to its neighbors.

Furthermore we count for how many prefixes we find RIB-Out matches for at least 50%, 90%, or 100% of their respective unique AS-paths.

To visualize the various possibilities Figure 4 shows a toy example with 8 ASes, three observation points (at AS 1, AS 2, and

AS 3) and one prefix p originated at AS 6. The dashed arrows[3] indicate the traffic flows along the observed AS-paths while the dotted arrows indicate the paths chosen by the simulation. Consider first AS 1 — its RIB-In contains the learned routes `1-7-6`, and `1-4-5-6` to reach AS 6. The path `1-7-6` is chosen instead of `1-4-5-6`, which has been observed in BGP data. This represents a RIB-In match, but no RIB-Out match. Since the observed AS-path is longer than the simulated path, the used policies are clearly wrong. Next, consider AS 2. Once again, we see that there is a RIB-In match (neighbor AS 8 propagates the "correct" suffix path to AS 2). But there is no RIB-Out match. In this case, the best path is chosen "wrongly" in the final BGP tie-break. We call this a potential RIB-Out match, because the choice is made based on the tie-breaker. This mismatch is due to an unlucky decision in the simulation, rather than using incorrect policies. In real routing IGP weights, etc., are also used to break these ties. Finally, AS 3 has a RIB-Out match: simulation and observation agree for router 2 of AS 3.

4.3 Deriving an AS-routing model

In this section, we introduce the details of our iterative approach for constructing an AS-routing model based on a `training` set of BGP data from multiple vantage points in the Internet.

We start from the simplest AS-model possible. It consists of one quasi-router per AS and contains one edge between any two connected ASes of the AS-level graph. Accordingly, this model only includes information that is easy to derive from the input data set. Then we determine for the `training` set, where the AS-paths predicted by the current AS-routing model differ from those observed in the Internet (those in the `training` set). This can be due to two reasons: First, the model prefers the shortest AS-path in the absence of more complex policies. Second, the quasi-routers inside an AS do not suffice to capture the required route diversity.

To reduce the discrepancies between the observed AS-paths and those predicted by the model, we *alter* the model iteratively by either adding routing policies or quasi-routers. Adding quasi-routers enables us to propagate more than one best route to the next AS, a necessity as the data analysis shows (see Section 3.2).[4] By adding policy rules we ensure that the appropriate AS-path is selected and can be propagated, even though it may not be the shortest one.

We do not aim at inferring the actual policies used by the ASes. Rather, it is our goal to derive an AS-routing model where the simulated AS-paths correspond to the observed AS-paths for the `training` set. By doing so we hope to, and indeed do, improve the predictive capabilities of the AS-routing model over the models discussed in Section 3.3. We are in this way capable of removing the limitations of the "one router per AS" model of the Internet.

In effect each iteration of the heuristic, see Figure 6, consists of comparing the AS-paths predicted by the model to those in the `training` data. Based on the results, changes to the model (new quasi-routers or changes to the policies) are determined and the path propagation is re-simulated for all prefixes that are effected by the changes. This cycle is repeated until the desired level of agreement for the `training` set is achieved. In the following, we present more details about the initial model and how the iterative refinement proceeds.

4.4 Example: refining an AS-routing model

Since any simple AS-routing model with just one quasi-router per AS is unlikely to match reality, we now illustrate with an ex-

ample how to use routing policies and topology diversification to improve the model. Suppose there are five ASes, interconnected as shown in Figure 5 (a), with two prefixes $p1$ originated at AS 3 and $p2$ at AS 4, and one observation point at AS 1 which observes a route with AS-path `1-4-3` for $p1$ and routes with paths `1-4` and `1-5-4` for $p2$. These AS-paths are visualized via dashed lines. The AS-paths currently chosen after a simulation run are paths `1-2-3` for $p1$ and `1-4` for $p2$ (dotted lines).

Starting with prefix $p1$, the heuristic detects that in the simulations the path `1-2-3` is chosen instead of the path `1-4-3` at AS 1. This mismatch is due to the fact that in our setup the quasi-router of AS 2 has a lower IP address than the quasi-router at AS 4. To correct this "wrong" tie-break decision, our heuristic sets up a policy at the quasi-router in AS 1 to prefer routes learned from AS 4 for prefix $p1$. We re-simulate, and now the path `1-4-3` is selected instead of the path `1-2-3` (see Figure 5(b)).

Next, consider the two AS-paths observed for prefix $p2$ at AS 1. A route with the shorter AS-path `1-4` is already selected by the quasi-router in AS 1; therefore no changes are required. Yet, in order to account for the AS-path `1-5-4`, a second quasi-router inside AS 1 is needed. Therefore, a new quasi-router b is created as an identical copy of the existing quasi-router a with the same neighbors as quasi-router a (see Figure 5(c)). Thus, quasi-router b will have a RIB-In match for a route with AS-path `1-5-4`, but does not select it as best route (the AS-path `1-4` is shorter). In order to correct this at router b of AS 1, two policy rules are used. A filter at AS 4 prevents routes for prefix $p2$ from being propagated to quasi-router b of AS 1 and a ranking policy is set to prefer routes for $p2$ announced by AS 4. This ensures that quasi-router b of AS 1 can select the route with AS-path `1-5-4` as its best route.

4.5 Initial model

To derive the initial model we use *all* available BGP feeds, `training` as well as `validation`, to derive an AS-graph from the AS-path information. Such an AS graph is likely to be incomplete, as it is probable that there are other peerings that are not used by any of the AS-paths recorded at our vantage points. It is possible to further improve the coverage of the AS-graph by adding additional observation points or information from the routing policy database or traceroute data. Yet, as these additional data sources come with some uncertainties [36], we only focus on data from our observation points.

Initially, all ASes consist of a single quasi-router, and peerings are established according to the edges of the AS graph. Next, we assign IP address to each quasi-router. This choice is important as the IP address is used as the final tie-breaker in the BGP decision process. (In case of a tie a quasi-router prefers the AS-paths announced by the quasi-router with the lower IP address.) Therefore, this choice can directly influence the quality of the prediction process. We choose to use IP addresses such that the high order 16 bits are set to the AS number and the low order bits are a unique ID for each quasi-router within the AS.

4.6 Iterative refinement

The goal of the iterative refinement process, see Figure 6, is to modify the AS-routing model until one achieves the desired level of agreement between the predicted AS-paths and the observed AS-paths. Accordingly, we now introduce our *refinement heuristic* which, by adding quasi-routers and BGP policy rules, reduces the discrepancies between the simulated and observed AS-paths for a set of prefixes.

We have two main reasons for using an iterative process instead of trying to correct the discrepancies in a single step. First,
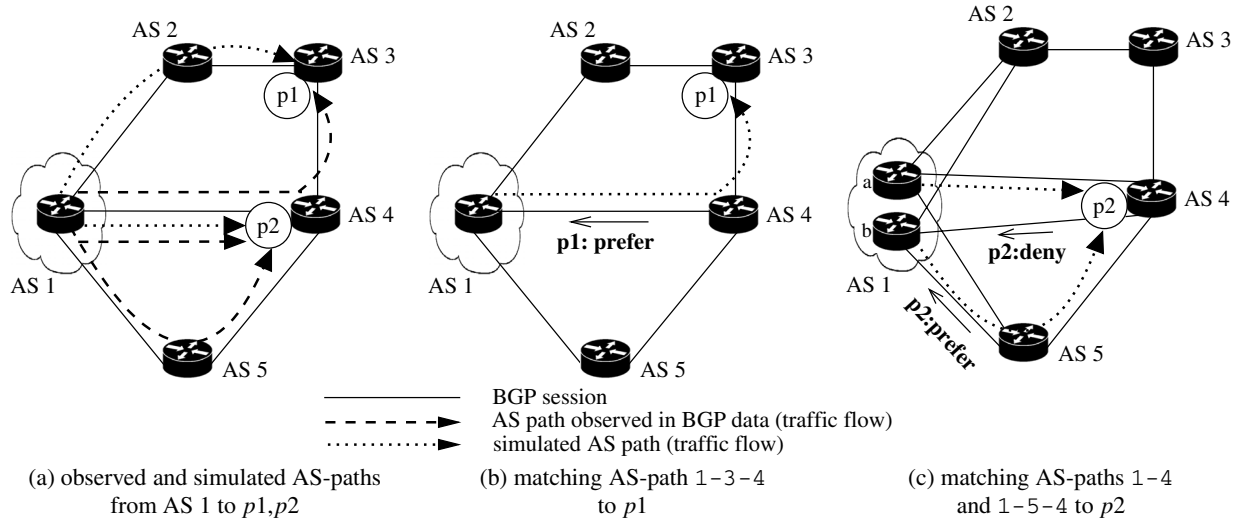
---

[3]In all figures routes are directed according to the flow of traffic.

[4]Keep in mind that a quasi-router does not have to correspond to an actual router. It is just an entity responsible for routes.

(a) observed and simulated AS-paths
from AS 1 to p1,p2

(b) matching AS-path `1-3-4`
to p1

(c) matching AS-paths `1-4`
and `1-5-4` to p2

Figure 5: Heuristic example: applying changes at AS 1 for prefixes $p1$**,** $p2$**.**


```
ITERATIVE REFINEMENT
run simulations for prefixes p (initial model)
repeat until RIB-out match for all observed AS-paths
    apply heuristic, compute changes
    restart simulations

HEURISTIC (1 ITERATION)
foreach prefix p
    foreach AS a
        O = (suffixes of) AS-paths for p observed at AS a
        foreach o ∈ O
            if RIB-Out match
                no change
                mark the quasi-router as used
            else if RIB-In match
                duplicate a quasi-router if necessary
                add policies: filtering, ranking (MED)
                mark the quasi-router as used
            else skip
```

Figure 6: Model refinement – methodology.

route propagation itself is an iterative process. For the AS-path of `1-2-3-4` from the origin (AS 4) to be observable at the observation point (AS 1), AS 3 first has to select an appropriate route and propagate it to AS 2. Then AS 2 has to select this route as its best one and propagate it to the observation point. To reproduce this step-by-step process in the AS-routing model we move from the origin of the route towards the observation points and change the policies or the topology at the AS where the path chosen in the model differs from the one observed in the `training` set. The change ensures that the desired route is propagated one AS further towards the observation point in the next iteration. This is reasonable since this is a local decision and one does not have to determine how the changes influence the overall route propagation beyond the local changes. This task is delegated to C-BGP. Accordingly, our second motivation for the iterative approach is that we do not have to reimplement the full routing logic of C-BGP to determine the necessary changes to the AS-routing model. Note that it is not necessary to proceed AS-hop by AS-hop. Rather in each iteration one determines the AS which is closest to the originating AS with a discrepancy between the observed AS-path and the selected best route and fixes this discrepancy at this AS.

In the following we first introduce our principle approach; then explain how the policies are adjusted; and finally how they may have to be corrected.

Refinement heuristic – principle approach:
The heuristic proceeds prefix-wise starting with the results of all C-BGP simulations runs for all prefixes of the respective `training` set based on the initial or previous AS-routing model. For each prefix p with AS-path P of the `training` set and each AS a on the path it checks the following conditions and if necessary takes appropriate actions:

RIB-Out match:

> **Condition:** The observed path up to this AS (the suffix up to a) is selected as best route by at least one quasi-router inside the AS.

> **Action:** We choose among this set of quasi-routers the one with the lowest quasi-router ID and mark/reserve this quasi-router as being responsible for this AS-path and not available for matching another observed AS-path for the same prefix.

RIB-In match but no RIB-Out match:

> **Condition:** There is at least one quasi-router which learns the observed AS-path up to this AS. But none of the quasi-routers has selected it as best route and none of these quasi-routers are already reserved for other routes for this prefix.

> **Action:** We choose among this set of quasi-routers the one with the lowest router ID and mark/reserve it as being responsible for this AS-path. Then we adjust this prefix' BGP policy at this quasi-router by either adding filters or setting MED values as described below.

> **Condition:** Same as above but all quasi-routers are already reserved for other routes for this prefix.

> **Action:** In this case we choose to "duplicate" one quasi-router with a RIB-In match. The new quasi-router has the same neighbors and policies as the copied one to ensure that it also has a RIB-In match for the prefix p. Then the BGP policy for this prefix is adjusted as in the previous case.

No RIB-In match:

> **Condition:** No quasi-router at the current AS has learned a route with the observed AS-path.

> **Action:** No action as a route with an appropriate AS-path first has to be propagated to this AS.

Refinement heuristic – policy adjustment:

Two ideas are central to our refinement process: First, new quasi-routers are added to account for path diversity. Yet, contrary to the routers in the Internet we do not establish iBGP sessions between the quasi-routers within an AS. Experiments with such an approach have shown that it is extremely difficult to control route selection, in particular to install different routes at neighboring iBGP routers. Therefore, we choose to use quasi-routers instead of routers. Each new quasi-router receives the appropriate routes by duplicating the BGP sessions to the neighboring ASes but remains isolated from other quasi-routers inside the AS. In effect we short-circuit the intra-AS route propagation process. As a result each AS can consist of multiple separate quasi-routers which do not exchange their reachability information.

Second, we use policy rules on a per-prefix basis to filter and rank routes at each selected quasi-router such that the route with the desired AS-path can be selected as the best route. Suppose that a quasi-router learns a route with the correct (suffix) path for a certain prefix, yet it does not select it as its best route (RIB-In match but no RIB-Out match). This can happen at any one of the steps in the BGP decision process, see Figure 1. At the same time this multi-step decision process provides us with many different ways in which we can change the decision: by either adding a policy at the current quasi-router or through a filter at the announcing neighbor which ensures that a route is no longer available at the current quasi-router. At this point our goal is not to infer the specific routing policy used by the AS. Rather we want to account for all possible weird routing policies.

The first step in the decision process is based on the BGP attribute *local-pref*. It has been shown in [37] that the preference of routes with longer AS-paths over those with shorter ones can lead to divergence. Attempts to use *local-pref* for building our routing model resulted in divergence problems which are very hard to debug. Therefore, we choose to not rely on this attribute. Rather we use BGP filters to ensure that routes with shorter AS-paths than the route we are looking for are not propagated to the current quasi-router. This is achieved by setting a filter policy for this prefix at the announcing neighbor. To avoid further reduction of route diversity we do not filter those routes that have the same AS-path length as the one we are looking for. Instead, we take advantage of the next step in the BGP decision process that relies on the *MED* attribute. If two routes have the same local-pref and the same AS-path length the one with the lower MED value is selected. We assign a lower MED value to routes announced by the AS from which the observed AS-path is learned. We require that MED values are always compared during the BGP decision process, even for routes learned from different neighbor ASes. Since quasi-routers inside an AS are not connected in our model, no iBGP divergence can arise [38]. Simply changing the ID of the router does not work as this would affect all routes.

It should be noted that our choice of BGP policies - filtering and MED values - is arbitrary and in general does not correspond to the policies actually used in the Internet. Infering the actual policies will be addressed in future work. In the Internet, *local-pref* is often used to implement business relationships and for traffic engineering. Yet, prioritizing AS-paths via MED is also not uncommon, as
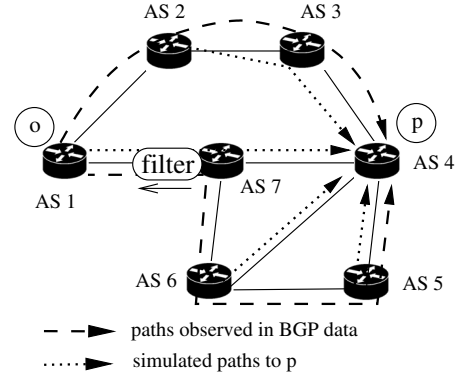

Figure 7: Necessity of filter deletion.

MED allows the realization of cold-potato routing [39]. However, as noted above we are not concerned about reverse engineering real policies: rather we aim at understanding the impact of routing policy on route diversity.

Refinement heuristic – filter deletion:

If one could process all AS-paths in a single step it would be easy to determine when an AS needs multiple quasi-routers to propagate AS-paths of different length. Using our iterative process this is not possible. It can happen that a filter is set while processing the "shorter AS-path" which stops the "longer AS-path" from being propagated. This filter has thus to be removed in a later iteration.

In Figure 7 observation point AS 1 observes two routes with AS-paths `1-2-3-4` and `1-7-6-5-4` for prefix *p*, originated by AS 4. Neither of the two AS-paths is selected as best route when simulating the initial model. The quasi-router at AS 1 chooses a route with AS-path `1-7-4` to reach prefix *p* (dotted arrow from AS 1 to AS 4). However, the heuristic detects a RIB-In match for `1-2-3-4` at AS 1 during the first iteration. To prevent the shorter AS-path `1-7-4` from being propagated to AS 1, a filter at the egress of AS 7 to AS 1 is set. Restarting the simulations results in a RIB-Out match for AS-path `1-2-3-4`.

With regards to the second AS-path `1-7-6-5-4`, the quasi-router at AS 7 does not select the correct suffix as best path until a later iteration. However, when it does select it as best route, it cannot propagate it to its neighbor AS 1 due to the egress filter set during the first iteration. As a consequence, AS 1 does not learn the observed AS-path `1-7-6-5-4`. When we do not find a RIB-Out or RIB-In match for a suffix of an observed AS-path, we check for a RIB-Out match at all announcing neighbor ASes. Provided that there is a RIB-Out match at this AS we remove any filter rule that prevents the propagation of the observed AS-path towards the observation point.

The removal of the filter in Figure 7 leads to the creation of a new quasi-router at AS 1 for a route with AS-path `1-7-6-5-4`. After the next iteration the route with this path is selected as best route by AS 1 and the above problem is circumvented and progress is ensured and no cycles will occur. Perfect RIB-Out matches are achieved after a total number of iterations that is a multiple of the maximum AS-path length.

4.7 Using the AS-routing model for predictions for other prefixes

At this point we can use the AS-routing model derived from a `training` set as input to the C-BGP simulator and predict likely AS-path choices for the prefixes of the `training` set to previously not considered observation points.
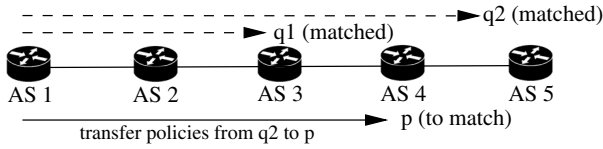
Figure 8: Transfer of policies across prefixes.

But as the policies are determined on a per-prefix basis it is unclear so far how to take advantage of the AS-routing model for predicting AS-paths for prefixes that are *not part* of the `training` set but for which we have AS-path information for some observation points. One approach is to use multiple iterations of the refinement heuristic with the drawback of ignoring the routing policy information accumulated in the AS-routing model derived from the `training` set.

To overcome this limitation we introduce the *reuse policy heuristic*. The key assumption behind this heuristic is that most ASes specify their policy rules on a per-peer basis — reflecting the economic relationship between peering ASes — and not on a per-prefix basis. Accordingly, independent of the success of this heuristic, we can improve our understanding of the correlation between routes for different prefixes, i.e., whether different prefixes are treated equally or differently by the policies within an AS. In the following, we explain *from where* and *which* policy rules are reused.

In order to determine from where policy rules are transferred we again proceed prefix by prefix. For each of the new prefixes and observation points we have an observed AS-path o. For each such AS-path the *reuse policy heuristic* tries to find an AS-path a that satisfies the following conditions:

1. The AS-path a is part of the `training` set, i.e., the input to the refinement heuristic and the AS-routing model shows a RIB-Out match for a.

2. Both AS-paths end at the same observation points, i.e., the first ASes of both AS-paths are identical. Furthermore we require that both AS-paths share at least the first two edges. The underlying assumption is that the policies applied for the "new" prefix are the same as for the "old" prefix.

3. There is no other AS-path x that satisfies the first two conditions that is longer than a.

The example shown in Figure 8 illustrates this process for a simple topology that consists of five ASes. AS 1 is again our observation point. Prefix p is originated by AS 4, $q1$ by AS 3, and $q2$ by AS 5. We assume that the AS-paths for prefixes $q1$ (`1-2-3`) and $q2$ (`1-2-3-4-5`) result in RIB-Out matches after using the refinement heuristic to derive an AS-routing model. The goal is to find a sensible policy for prefix p with AS-path `1-2-3-4`. Since both AS-paths (`1-2-3` and `1-2-3-4-5`) satisfy the first two conditions, the longer path is selected. In the absence of this AS-path the shorter one would have been chosen.

Policies, that allow the propagation of AS-path `1-2-3-4-5`, are likely to ensure the propagation of the similar path `1-2-3-4`, too. The underlying assumption is that policies in the Internet are in general specified for complete BGP sessions (neighbor-basis) and not on a per-prefix basis.

In the example of Figure 8, we transfer policies from the subpath `1-2-3-4` of `1-2-3-4-5` to the current one for p. If there is a policy (MED, filter) for prefix $q2$ along `1-2-3-4`, it is converted into a policy rule for prefix p. In contrast to the refinement heuristic, no new quasi-routers are added.

5. RESULTS

In this section we evaluate the *refinement* and *reuse policy heuristics* by using them to derive an AS-routing model for various sets of `training` data, and evaluate their effectiveness using separate `validation` data.

Data:
Of the $1,300$ BGP observation points, see Section 3.1, we randomly assign $2/3$ to the `training` set and the remainder ones to the `validation` set. We sub-select the AS-path information from $1,000$ ASes and their corresponding paths from both the `training` and the `validation` sets to derive our base AS-routing model. In order to ensure a reasonable coverage of the AS-graph we include all level-1 ASes as well as randomly selected ASes of the groups level-2 and other. We refer to this set of prefixes and their AS-paths as `psetA`. To evaluate the effectiveness of the *reuse policy heuristic* we select two other disjoint sets of prefixes and their AS-paths in a similar manner. These sets, referred to as `psetB` and `psetC`, again consist of $1,000$ randomly chosen prefixes.
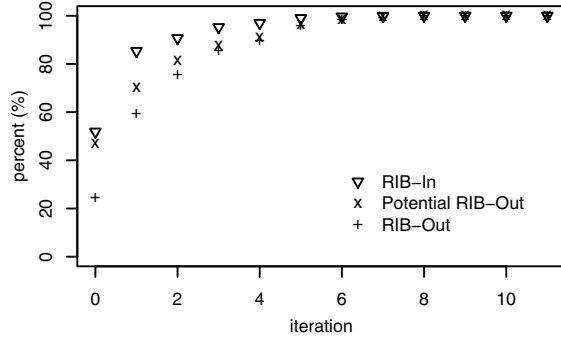
Training:
The inference of the AS-routing model uses an iterative process that incrementally refines the model with the goal of achieving an exact match between the AS-paths predicted by the model and the `training` set. Figure 9(a) shows the progress of the heuristic with each iteration as measured in terms of RIB-In matches, potential RIB-Out matches and RIB-Out matches. The length of the longest AS-path is 10, and 11 iterations happen to suffice to achieve our goal of perfect RIB-Out matches. Notice that the early progress of the heuristic is excellent. Just one iteration more than doubles the percentage of RIB-Out matches from 24.5% to 59.3%, and increases the potential RIB-Out matches and RIB-In matches to more than 70% and 85% respectively. Given that the average length of the AS-paths is about 4.3 it is not surprising that we achieve RIB-Out matches for all but 5% of the AS-paths after five iterations

Further inspection of the data reveals that matching the AS-paths for some prefixes and some observation points requires more policy adjustments than for others. After the fifth iteration we start to see a significant number, 238 out of the 1000 prefixes with RIB-Out matches for all observation points. This number increases with the next iterations via 481 and 683 to 969 after the eighth iteration. This means that at this point we only have a very small percentage of unmatched AS-paths. Note that if we do require RIB-Out matches for 90% of the AS-paths for each prefix, already more than 40% of the prefixes satisfy this condition after two iterations. For the other two subsets of prefixes, `psetB` and `psetC`, even faster improvements are observable
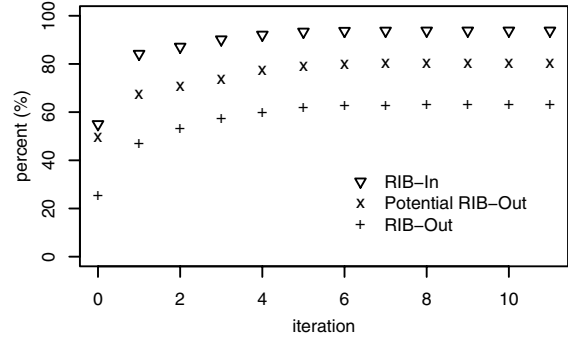
Validation:
Given an AS-routing model we can now evaluate its predictive capabilities for our example question for a different set of observation points. We find, based on the subset of `validation` for `psetA`, that we improve our prediction capabilities from 25.5% (without routing policies) to 63% for RIB-Out matches, and if we ignore the final tie-breaking rule of the decision process, from 50% to more than 80% (see Figure 9(b)). For RIB-In matches we see an improvement from 55% to 93%. Let us point out that the major improvements happen during the first six iterations.

To judge the qualitative improvement of our results vs. those reported by Mao et al. [7] we point out that our results hold across more than 300 observation points rather than 3 ASes and are significantly better. In terms of RIB-Out matches which correspond to *exact matches* we have 63% vs. their 35%, 10%, and 3%; in terms of RIB-In matches which correspond to *matches* we have 93% vs.

(a) Progress for `training` (psetA)

(b) Progress for `validation` (psetA)

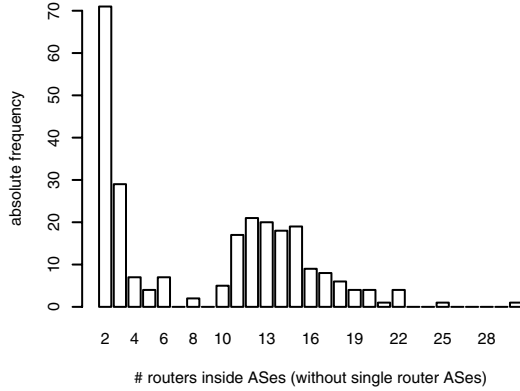**Figure 9: Refinement heuristic– Results**

Figure 10: # of quasi-routers per AS (ignoring 14,305 ASes with one quasi-router)

their 82%, 64%, and 16%. To better compare the results we focus on the same three ASes (7018, 2152, 8121) as Mao et al. moving them from the `training` to the `validation` set and rerunning the refinement heuristic for 200 prefixes of psetA. We find perfect RIB-In matches for AS 7018, almost perfect matches for AS 2152, and 44.2% for AS 8121.

Characteristics of the inferred AS-routing model:
The main reason for the effectiveness of our refinement heuristic is that an AS can, if necessary, consist of multiple quasi-routers. This raises the question how many of these quasi-routers are needed. Figure 10 shows a histogram of the number of quasi-routers per AS for psetA (for those ASes with more than one quasi-router). For almost all ASes (14,305) one quasi-router suffices. For 71 we need two. Yet, there are 138 ASes that need more than 9 quasi-routers. Note, not all of these quasi-routers are needed for all prefixes. Not surprisingly, we find that our level-1 ASes are among the ASes with many quasi-routers. After all, level-1 ASes offer peerings at quite a number of peering locations, peer with and provide service to many other ASes, and have a sizable backbone network. Most of the others are level-2 ASes, but there are also some in the "other" group. The average number of quasi-routers for level-1 ASes is 17.1, while for all other ASes it is 1.03.

Effectiveness of the *reuse policy heuristic*:
Given an AS-routing model, we can now evaluate its predictive capabilities for our example question for a different set of prefixes. We find, based on the subset of `training` for psetB

(psetC), that we improve our prediction capabilities by almost a factor of two from 24.6% (23.7%) (without routing policies) to 46.6% (45.5%) for RIB-Out matches. If we ignore the final tie-breaking rule of the decision process, the number of Potential RIB-Out matches increases from 48.3% (46.7%) to 59.0% (58.3%). This implies that the assumption that policies are well captured by the AS-routing model and that the *reuse policy heuristic* can take advantage of this capability. In addition it shows that some of the policies are applicable on a per-peer basis. Yet, as we do not get perfect matches one should restrain from over-generalizations.

The *reuse policy heuristic* gives an AS-routing model for psetB and psetC which utilizes the results of psetA. If one wants to derive an AS-routing model for psetB/psetC one can either start from the psetA AS-routing model or start from scratch using the refinement heuristic. With regards to potential RIB-Out matches we have for psetB after the *reuse policy heuristic* 46.6% vs. 24.6% for the initial model, after the first iteration 65.8% vs. 55.3%, and after the second iteration 83.2% vs. 81.0%. We note that the additional information from psetA helps both in terms of progress during the early iterations as well as with regards to the predictive capabilities after the initial iterations. Yet, of course the AS-routing model that utilizes the *reuse policy heuristic*, is based on a much larger knowledge base that has needed significant computation time to derive.

Revisiting the effectiveness of the refinement heuristic:
Even though the heuristics are very effective in terms of predicting AS-paths we also need to investigate how and why they may fail. First we point out that it is quite possible for an AS-path to be contained both in the `training` as well as the `validation` set. This can occur since we have multiple observation points in some ASes. We happen to have at least one observation point in both sets for 168 ASes. As a result we find that 22.2% of all AS-paths for psetA are in both. For the subset of these AS-paths in the `validation` set we have RIB-Out matches by design, in this case for 49% of the paths in the `validation` set of psetA.

If the AS-path is not contained in both the `training` and the `validation` sets then, even though we hope to have configured the appropriate policies, the heuristics may have failed to propagate the route along the observed AS-path. A route may not be propagated in the AS-routing model either due to another policy decision or due to a missing policy. Given that we do not have training data from all ASes we have to predict policy choices to some degree, especially if the data used to derive the AS-routing model (`training` set) does not include an AS-path for all AS-edges. We refer to such AS-edges as uncovered.

Path class # uncovered	# paths	RIB-In	Pot. RIB-Out	RIB-Out	Not found
0 AS-edge	9,486	90.0%	59.1%	43.3%	10.0%
1 AS-edge	43,823	86.3%	74.2%	25.9%	13.7%
2 AS-edges	5,624	34.2%	26.3%	6.6%	65.8%
3 AS-edges	45	0.0%	0.0%	0.0%	100.0%

Table 3: Effectiveness of refinement heuristic by uncovered AS-edges.

Table 3 shows the effectiveness of the refinement heuristic by uncovered edges, considering only AS paths contained in the `validation` set but not in the `training` set of `psetA`. We find that the majority of these AS-paths contain at most one uncovered AS-edge. No AS-path includes more than three uncovered AS-edges. Of these uncovered AS-edges almost all, 93.4% to be precise, occur next to the observation point.

As the number of uncovered edges on an AS-path increases, the likelihood of achieving a RIB-In match for these paths decreases. For AS-paths with 0 uncovered edges we have 90.0% RIB-In matches. For paths with 1 uncovered edge we get 86.3% RIB-In matches which further decreases to 34.2% and 0% for 2 and 3 uncovered AS-edges, respectively. Similar observations hold for potential RIB-Out and RIB-Out matches. As shown in Table 3, the percentages of RIB-In matches do not decrease as fast as the ones for RIB-Out matches when there are uncovered AS-edges on the path. This is not surprising as RIB-In matches are not as sensible to the specific policy choices. These results agree with the intuition behind Mao's et al. [7] approach. If one can determine the first hop then the uncertainty about the remaining AS-path is reduced.

In order to predict how good our results can be for arbitrarily chosen observation points, we randomly select a set of 6,000 ASes and compute for all possible originating ASes how many uncovered AS-edges one may have to predict. We find that for 0.7% there is no uncovered edge. For 54.8%, 36.7%, 6.6% there are one, two, and three, respectively. Only 1.2% require more than three. These numbers are a bit more pessimistic than for our data sets. Yet, as shown above we can expect reasonable results, when just one or two AS-edges are uncovered.

6. RELATED WORK

Improving our understanding of routing dynamics has been a topic of huge interest over the last few years, e.g. [23,24,27,40–42]. Most of the attention has been given to the dynamics of the BGP protocol, e.g., to understand why convergence time of BGP can be rather long [23,24,40]. Oscillations in BGP [43] can occur; see [44] for a review of their possible causes. Apart from the aspects related to the time required for BGP to converge, divergence anomalies as defined in [44], are permanent failures of BGP to converge towards a stable path. Divergence anomalies stem from two typical causes: conflicting eBGP policies [14], and iBGP oscillations [43]. The first can be explained within a model that only contains one router per AS, the second cannot.

Recent work has investigated the interactions of routing on traffic within an AS. Based on data gathered from the Sprint network [45], it was shown that the impact of external BGP events on its traffic matrix is limited. Still hot-potato disruptions [10, 11] can have a significant impact on transit ASes. This highlights that routing dynamics can be complex, even when just looking at a single AS.

AS-level topology inference [15,16,18] provides another dimension to the complexity of routing in the Internet. Routing policies are typically partitioned into a few classes which capture the most common practices in use today [46]. Unfortunately it is also known that the reality of routing policies [17] and peering relationships is far more complex than those few typical classes [46, 47]. Current approaches for AS-level topology inference rely on a top-down approach. They first define a set of policies and then try to match those policies with their observations of the system. Yet, policies as used by ISPs have to realize high-level goals [46]. Assuming any kind of consistency of such policies across ASes is questionable, especially as in practice, policies are often configured on a per-router, or per-peering basis [46]. This means that observed BGP routes do not even have to be consistent with the high-level policies of the AS.

Traceroute is one of the most widely used tools for discovering end-to-end paths and can be used in combination with other tools to derive AS-level paths [8,9]. These tools require network access. To overcome this problem Mao et al. [7] proposed a first methodology for inferring AS-level paths based on presumed BGP routing policies. They find that the accuracy of the estimation depends on the precision of the AS relationship inference and the ability to incorporate additional information regarding the first hop.

7. CONCLUSION

This paper makes four fundamental contributions. Firstly, we show that AS topologies using a single router are limited – they cannot hope to capture all kinds of route diversity in the Internet today. This is in contrast to the majority of prior work on modeling the large scale structure of the Internet. Secondly, we show how to use simulation-based heuristics to iteratively build an AS model that is consistent with *all* observed paths. Thirdly, we show that the model can be used to predict previously unobserved AS-paths. The accuracy is in practice limited to around 63% due to policy decisions on parts of the network that we have not previously observed in our data, and tie-break decisions that are inherently hard to predict. Ignoring tie-breaks we reach an accuracy of more than 80%. Finally, our results provide some insights about the structure of the Internet.

Obviously, we shall continue to work on improved heuristics to increase the accuracy of our methods. However, there are many other areas for interesting future work. In particular, we have only considered how well our model answers one type of question. However, there are many questions an operator might wish to ask, for instance, what-if they change a policy, or peering arrangement. We wish to study such questions in future work both to provide practical solutions to operators, and because the results will provide insight into how policies are affecting the current Internet.

Furthermore, the ability to refine instances of our model allows to gain insight into what information has to be present in the real routing system to account for the observed state. This understanding of the relationships between topology and policies and how routing information propagates may be useful input for designing the next inter-domain routing system (e.g., [48]).

8. REFERENCES

[1] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *Proc. of ACM SIGCOMM'99*, 1999.

[2] L. Li, D. Alderson, W. Willinger, and J. Doyle, "A first-principles approach to understanding the internet's router-level topology," in *Proc. ACM SIGCOMM*, 2004.

[3] B. Raghavan and A.C. Snoeren, "A system for authenticated policy-compliant routing," in *Proc. ACM SIGCOMM*, 2004.

[4] O. Bonaventure and B. Quoitin, "Common utilizations of the BGP community attribute," 2003, Internet Draft.

[5] N. Feamster, J. Borkenhagen, and J. Rexford, "Guidelines for interdomain traffic engineering," *ACM CCR*, 2003.

[6] N. Feamster, J. Winick, and J. Rexford, "A model of BGP routing for network engineering," in *Proc. ACM SIGMETRICS*, 2004.

[7] Z.M. Mao, L. Qiu, J. Wang, and Y. Zhang, "On AS-level path inference," in *Proc. ACM SIGMETRICS*, 2005.

[8] Z.M. Mao, J. Rexford, J. Wang, and R.H. Katz, "Towards an accurate as-level traceroute tool," in *Proc. ACM SIGCOMM*, 2003.

[9] Z.M. Mao, D. Johnson, J. Rexford, J. Wang, and R. Katz, "Scalable and accurate identification of AS-level forwarding paths," in *Proc. IEEE INFOCOM*, 2004.

[10] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Dynamics of hot-potato routing in IP networks," in *Proc. ACM SIGMETRICS*, 2004.

[11] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan, "Traffic matrix reloaded: impact of routing changes," in *Proc. ACM PAM*, 2005.

[12] R. Teixeira, K. Marzullo, S. Savage, and G. Voelker, "In search of path diversity in ISP networks," in *Proc. ACM IMC*, 2003.

[13] "University of Oregon RouteViews project," http://www.routeviews.org/.

[14] T. Griffin, F. Bruce Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. Networking*, 2002.

[15] L. Subramanian, S. Agarwal, J. Rexford, and R. Katz, "Characterizing the Internet hierarchy from multiple vantage points," in *Proc. IEEE INFOCOM*, 2002.

[16] L. Gao, "On Inferring Autonomous System Relationships in the Internet," *Proc. IEEE Global Internet*, 2000.

[17] F. Wang and L. Gao, "Inferring and characterizing internet routing policies," in *Proc. ACM IMC*, 2003.

[18] G. Battista, M. Patrignani, and M. Pizzonia, "Computing the types of the relationships between autonomous systems," in *Proc. IEEE INFOCOM*, 2003.

[19] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *Proc. ACM IMW*, 2002.

[20] Neil Spring, Raatul Mahajan, and Thomas Anderson, "Quantifying the causes of path inflation," in *ACM SIGCOMM 2003*, Karlsruhe, Germany, 2003.

[21] B. Quoitin and S. Uhlig, "Modeling the routing of an Autonomous System with C-BGP," *IEEE Network Magazine*, 2005.

[22] "RIPE's routing information service," http://data.ris.ripe.net/.

[23] T. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. ACM SIGCOMM*, 1999.

[24] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "An experimental study of Internet routing convergence," in *Proc. ACM SIGCOMM*, 2000.

[25] C. Labovitz, R. Malan, and F. Jahanian, "Origins of Internet routing instability," in *Proc. IEEE INFOCOM*, 1999.

[26] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The impact of Internet policy on Internet paths," in *Proc. IEEE INFOCOM*, 2001.

[27] A. Feldmann, O Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating internet routing instabilities," in *Proc. ACM SIGCOMM*, 2004.

[28] David G. Andersen, Nick Feamster, Steve Bauer, and Hari Balakrishnan, "Topology inference from BGP routing dynamics," in *Proc. ACM IMW*, 2002.

[29] R. Teixeira, T. Griffin, G. Voelker, and A. Shaikh, "Network sensitivity to hot potato disruptions," in *Proc. ACM SIGCOMM*, 2004.

[30] B. Quoitin, "C-BGP, an efficient BGP simulator," http://cbgp.info.ucl.ac.be/, 2003.

[31] B. Halabi and D. Mc Pherson, *Internet Routing Architectures (2nd Edition)*, Cisco Press, 2001.

[32] Intel-DANTE, "Intel-DANTE monitoring project," http://www.cambridge.intel-research.net/monitoring/dante/.

[33] Abilene, "The Abilene Observatory: Abilene routing data," http://abilene.internet2.edu/observatory/.

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

[35] W.B. Norton, "The art of peering: the peering playbook," 2002.

[36] G. Siganos and M. Faloutsos, "Analyzing BGP Policies: Methodology and Tool," in *Proc. IEEE INFOCOM*, 2004.

[37] T.G. Griffin, F.B. Shepherd, and G. Wilfong, "Policy disputes in path vector protocols," in *Proc. ICNP*, 1999.

[38] T.G. Griffin and G. Wilfong, "On the Correctness of IBGP Configuration," in *Proc. ACM SIGCOMM*, 2002.

[39] N. Feamster, Z. Mao, and J. Rexford, "Borderguard: detecting cold potatoes from peers," in *Proc. ACM IMW*, 2004.

[40] C. Labovitz, G. Malan, and F. Jahanian, "Internet Routing Instability," in *Proc. of ACM SIGCOMM*, September 1997.

[41] G. Siganos and M. Faloutsos, "BGP Routing: A Study at Large Time Scale," in *Proc. IEEE Global Internet*, 2002.

[42] D.-F. Chang, R. Govindan, and J. Heidemann, "The temporal and topological characteristics of BGP path changes," in *Proc. ICNP*, 2003.

[43] A. Basu, C. Ong, A. Rasala, F. Shepherd, and G. Wilfong, "Route oscillations in i-BGP with route reflection," in *Proc. ACM SIGCOMM*, 2002.

[44] R. Musunuri and J. A. Cobb, "An Overview of Solutions to Avoid Persistent BGP Divergence," *IEEE Network Magazine*, 2005.

[45] S. Agarwal, C. Chuah, S. Bhattacharyya, and C. Diot, "The Impact of BGP Dynamics on Intra-Domain Traffic," in *Proc. ACM SIGMETRICS*, 2004.

[46] M. Caesar and J. Rexford, "BGP Routing Policies in ISP Networks," *IEEE Network Magazine*, 2005.

[47] H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "Towards capturing representative AS-level internet topologies," *Computer Networks Journal*, 2004.

[48] T. Griffin and J. Sobrinho, "Metarouting," in *Proc. ACM SIGCOMM*, 2005.