Verifying Policy-based Routing at Internet Scale

Xiaozhe Shao, Lixin Gao
Department of Electrical and Computer Engineering, University of Massachusetts
{xiaozheshao, lgao}@engin.umass.edu

***Abstract*—Routing policy configuration plays a crucial role in determining the path that network traffic takes to reach a destination. Network administrators/operators typically decide the routing policy for their networks/routers independently. The paths/routes resulted from these independently configured routing policies might not necessarily meet the intent of the network administrators/operators. Even the very basic network-wide properties of the routing policies such as reachability between a pair of nodes need to be verified.**

In this paper, we propose a scheme that characterizes routing-policy verification problems into a Satisfiability Modulo Theories (SMT) problems. The key idea is to formulate the SMT model in a policy-aware manner so as to reduce/eliminate the mutual dependencies between variables as much as possible. Further, we reduce the size of the generated SMT model through pruning. We implement and evaluate the policy-aware model through an out-of-box SMT solver. The experimental results show that the policy-aware model can reduce the time it takes to perform verification by as much as 100x even under a modest topology size. It takes only a few minutes to answer a query for a topology containing tens of thousands of nodes.

I. Introduction

Policy-based routing protocols, such as Border Gateway Protocol (BGP), allow network administrators/operators to configure flexible routing policies independently. Each network sets its own routing policies with little or no coordination with other networks. It is therefore challenging to understand how a network's routing policies might impact routes of other networks. The route taken by a packet to traverse in the Internet is determined by the distributed route selection process, and a result of complex interactions controlled by the configured policies.

The paths/routes resulted from these independently configured routing policies might not necessarily meet the intent of the network administrators/operators. For example, a pair of networks might not be able to reach each other due to routing policy misconfiguration. Indeed, the policy misconfiguration of a single network might lead to a widespread outage in the Internet [5]. Further, a network might have a few desirable goals when setting its routing policy. For example, it might want to ensure that its traffic from a specific content provider such as Google not to use a specific provider.

Formal methods can provide a sound and thorough verification to questions whether the intent of a network is satisfied through exhaustive search. Researchers propose a number of formal methods to study and analyze the BGP routing system. To study the safety property of BGP systems, Satisfiable Module Theories (SMT) [31] and Rewriting Logic [30], [32], [33] are used to verify the convergence conditions of the system. Recently, [7], [34] propose to verify network configurations through modeling routing behavior with SMT constraints.

These proposed models for BGP routing verification can typically handle up to hundreds of routers at most. It would be challenging to verify routing properties at Internet scale where there are tens of thousands networks and each network consists of tens to thousands of routers. Even if we verify network routing properties in a confined scope (*e.g.*, within a set of networks owned by a single company), it might require to scale to tens of thousands of routers.

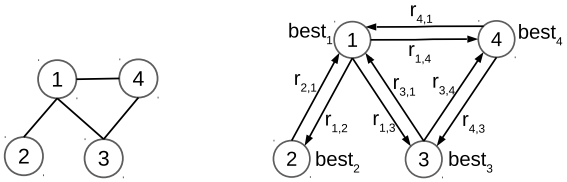
In this paper, we propose policy-aware models that translate the network verification problem into SMT constraints, and perform the network verification through solving the SMT problem. In order to scale the verification system, our policy-aware model reduces the mutual dependencies between variables considering routing policy constraints in the Internet. In other words, we explicitly account for routing policy constraints in constructing the SMT constraints. In particular, we construct a series of policy-aware models. First, we consider routing policies that follow the Gao-Rexford guideline [15]. We then generalize the model to capture routing policies beyond the Gao-Rexford guideline. Furthermore, we propose an approach to prune the generated SMT model.

We implement the verification toolkit based on a SMT solver, Z3 [12], and evaluate the policy-aware model through verifications on network topologies of various sizes. The experimental results show that the policy-aware model can reduce the time it takes to perform verifications by as much as 100x even under a modest topology with a thousand nodes. Further, we show that it is feasible to deploy the model at Internet scale. It takes only a few minutes to answer a query for a topology containing tens of thousands of nodes.

The rest of the paper is organized as follows. Section II describes a basic SMT-based model for BGP. Section III introduces a series of policy-aware models that reduce the mutual dependencies between variables based on routing policy properties. In Section IV, we describe how to specify a query for a verification problem. In Section V, we introduce the optimization technique to eliminate irrelevant variables in the model. We implement the verification toolkits based on the policy-aware model and evaluate the model in Section VI. We discuss related work in Section VII and conclude in Section VIII.

II. A Topology-based Model

To verify policy configuration based on SMT, a system of SMT constraints are generated from the routing policy con-



(a) A topology example. (b) VDG of the topology-based model.

Fig. 1: The topology-based model and its VDG.

TABLE I: Symbolic variables in the model.

Variable	Description	Representation
$r.length$	AS path length for r	[0,2^16)
$r.pref$	Local preference for r	[0,2^16)
$r.nexthop$	The first AS in AS_Path of r	[0,2^32)
$r.origin$	Origin Type of r	[0,2^2)
$r.med$	BGP MED attribute for r	[0,2^32)
$r.distance$	IGP distance of r	[0,2^32)
$r.rid$	Neighbor router ID for r	[0,2^32)
$r.ibgp$	Whether r is learned via IBGP	1 bit
$r.valid$	Whether r has a valid route	1 bit

figuration and the properties of the routing behavior supposed to satisfy. In this section, we introduce a common method to characterize BGP routing behavior.

To model the routing behavior through SMT constraints, we represent the best route of each router and all route announcements between BGP neighbors as variables. Then, for each router, to model the import policy and route selection, the constraints are generated among the best route and all received route announcements while to model the export policy, the constraints are generated among the best route and the route announcements derived from the best route. We refer to these models as *topology-based models* which are proposed in both Bagpipe [34] and Minesweeper [7].

A. Modeling Routes

Symbolic records are used to represent the best routes, the received routes from BGP neighbors and the route announcements to BGP neighbors. We represent the route announced from a router i to its neighbor j, as a record, $r_{i,j}$, which is a collection of variables. In addition, for each router i, $best_i$ represents the best route among routes received from all neighbors. Figure 1(a) illustrates a topology example with four nodes that represent four routers respectively. In the topology, node 1 and node 4 peer with each other and announce routes to each other. We use $r_{1,4}$ to represent the route announced from node 1 to node 4 and $r_{4,1}$ to represent the route announced from node 4 to node 1.

Table I lists the variables of a record that represents the basic information within a route or a route announcement. For node i, that owns the destination d, $best_i$ is the origin route. The constraints are as follows.

$$Origin(i) \iff best_i.length = 0 \wedge \quad (1)$$
$$best_i.nexthop = 0 \wedge best_i.valid \wedge best_i.pref = 2^{16} - 1$$

B. Modeling Import Policy

Following the import policy, each router assigns a local preference to the routes that are received from external BGP neighbors. We model the import policy of node i as the function, $f^i_{in}(r)$. Based on the import policy, constraints are generated for all received routes of a router as follows.

```
if r.valid then
  r.pref = f^i_in(r)
```

C. Modeling Route Selection

The best route of a router is selected from the received routes. The route selection of BGP protocol selects the best route by following a sequence of rules. In general, $best_i$ can be a function of all routes received by node i.

$$best_i = f_{selection}(r_{1,i}, ..., r_{n,i}) \quad (2)$$

where, $f_{selection}$ represents the route selection procedure of BGP. That is, $best_i$ depends on all routes received by node i.

D. Modeling Export Policy

The export policy determines whether a route is announced to neighbors and how to derive the route announcement from the best route. For example, a router announces a route received from an internal BGP neighbor to its external BGP neighbors, but does not announce the route to its internal BGP neighbor. When a best route can be announced to a neighbor according to the export policy, the announced route will be derived from the best route. For example, the length of the announced route will be increased by one when the route is announced to an external BGP neighbor.

In general, all routes that are announced by node i to its neighbors, are functions of $best_i$.

$$r_{i,j} = f^{i,j}_{export}(best_i) \quad (3)$$

where, $f^{i,j}_{export}$ represents the export policy of node i for its neighbor j. Thus, the routes exported by node i depends on $best_i$.

E. Mutual Dependencies between Records

In a topology-based model, as Equation (2) shows, the best route of a router can be determined when all routes received by the router are determined. The change of any received route might lead to the change of the best route. That is, the best route of a router depends on the routes received. In addition, as Equation (3) shows, the routes announced by a router are derived from the best route of the router based on its export policy. That is, the routes announced by a router depend on its best route.

Apparently, the dependency between records is transitive. Suppose that router a announces a route to its neighbor, b. The announced route, $r_{a,b}$, depends on $best_a$ while $best_b$ depends on $r_{a,b}$. As a result, $best_b$ depends on $best_a$. Similarly, $best_a$ also depends on $best_b$. In the topology-based model, the best routes of a pair of BGP neighbors depend on each other through the routes announced between them. Namely, the variable dependency is mutual.

To explore the variable dependency in a model, we represent a model into the *variable dependency graph* (VDG), which is

a directed graph where each node represents the best route of a router and each directed edge represents the route announced from one router to another. Figure 1(b) illustrates the VDG of the topology-based model for the topology in Figure 1(a). We say that a pair of neighboring ASs is *mutual dependent* in the topology-based model, if the pair of nodes representing these two ASs forms a cycle in the VDG of the model.

III. Policy-Aware Models

In this section, we propose a series of *policy-aware models* for routing-policy verification. In Section III-A, we first motivate policy-aware models by discussing how cycles affect the efficiency of the SMT solving process. In Section III-B, we introduce the basic idea of reducing cycles through considering the routing policies that follow the Gao-Rexford guideline. Finally, in Section III-C, we construct a policy-aware model for any routing policy enabled by BGP.

A. Motivation for Reducing Cycles

Given a system of SMT constraints derived from the verification problem, SMT solver tries to determine whether there is a satisfiable assignment for all variables to satisfy all constraints. The verification answer is SAT, if the SMT solver finds a satisfiable assignment for the constraints. The verification answer is UNSAT, if there is no satisfiable assignments for the constraints.

The solving process consists of two iterative steps: deriving the values for variables that can be determined by the constraints and guessing values for the other variables. According to the system of SMT constraints, some variables are determined by the input constraints. For example, as Equation (1) shows, the origin route is always valid. When no more variables can be determined by the constraints, the solver will guess a value for a variable that is not determined as a temporary assignment. Then, the solver derives the values for variables based on the constraints and the current assignments until no more variables can be determined. For example, in the topology of Figure 1(a), when the SMT solver guesses values for $r_{1,3}$ and $r_{4,3}$, $best_3$ is determined by the constraints derived from import policy of node 3. However, the temporary assignments might not satisfy all constraints. If these temporary assignments violate some constraints, the SMT solver needs to try other possible assignments, so that it will not miss any satisfiable assignment before returning UNSAT.

In the topology-based model, only the variables representing the origin route are assigned values through the constraints. After that, the SMT solver is supposed to infer a satisfiable assignment for the other variables. However, the ubiquitous cycles in the model might require the SMT solver to guess numerous potential satisfiable assignments before deriving the result.

B. BiNode Model for Policies following the Gao-Rexford Guideline

To introduce the basic idea of policy-aware models, we first propose a policy-aware model for routing policies that follow the Gao-Rexford guideline. We refer to the model as *biNode model*. For simplicity of exposition, we ignore the internal BGP sessions and model each network as one router/node. We will discuss how to apply the model to iBGP in Section III-B3.

Provider-customer (PC) relationship and peer-peer (PP) relationship are two common agreements between Autonomous Systems (ASs) in the Internet. Within a PC relationship, an AS/node as the customer pays its provider to access the rest of the Internet. Within a PP relationship, two connected ASs/nodes exchange traffic from their own customers for free. There are two rules in the Gao-Rexford guideline.

- GR Preference: an AS/node prefers customer routes over peer and provider routes.
- GR Export: peer and provider routes are exported to customers only.

In the following, we will explore how to transform the routing policies that follow GR Preference and GR Export rules into the biNode model.

1) BiNode Model Construction: We represent the best routes of node i, as two records: $dbest_i$ and $best_i$, where $dbest_i$ represents the best route among all customer routes that are received from the customers and $best_i$ represents the best route among all routes received by node i. The intuition of this separation is that according to GR Preference rule, customer routes always have higher ranking than the other routes. We use a separate variable $dbest_i$ to represent the best route derived from these high-ranking routes.

As a result, the constraints for route selection are divided into two parts for $dbest_i$ and $best_i$ respectively.

$$dbest_i = f_{selection}(r_{k+1,i}, ..., r_{n,i}) \tag{4}$$

$$best_i = f_{selection}(r_{1,i}, ..., r_{k,i}, dbest_i) \tag{5}$$

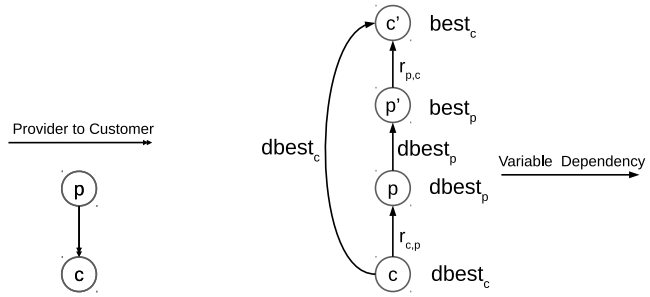
where $r_{1,i}$,..., $r_{k,i}$ are routes received from providers or peers and $r_{k+1,i}$,...,$r_{n,i}$ are routes received from customers. Note, the function $f_{selection}$ is the same as that in the topology-based model, since it is determined by the BGP route selection process. Apparently, $best_i$ depends on $dbest_i$. If there are no variables in the $f_{selection}$ for $dbest_i$ and node i is not the origin node, then $dbest_i$ does not represent a valid route. In that case, $dbest_i$ equals to an invalid route, r_{empty}, in the SMT constraints.

In the VDGs for the biNode model, we use two nodes to represent $dbest$ and $best$ of a node respectively and use a directed line from $dbest$ to $best$ to represent the dependency between $dbest$ and $best$ of the same node.

According to the definition of $dbest$ and $best$ in the biNode model, the constraints derived from export policies that follow GR Export rule are as follows.

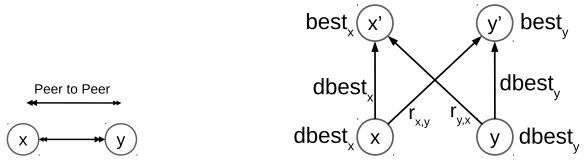
$$r_{i,j} = \begin{cases} f^{i,j}_{export}(best_i) & \text{if } j \in Customers(i) \\ f^{i,j}_{export}(dbest_i) & \text{Otherwise} \end{cases} \tag{6}$$

When the best route of a node is represented by $dbest$ and $best$ in the biNode model, we can avoid cycles between pairs of neighboring nodes with PC relationship or PP relationship.



(a) PC relationship. (b) VDG of the biNode model.

Fig. 2: The biNode model for two ASs with PC relationship.



(a) PP relationship. (b) VDG of the biNode model.

Fig. 3: The biNode model for two ASs with PP relationship.

a) Avoiding cycles for PC relationship: We consider a pair of neighboring nodes, node p and node c, where node p is a provider of node c. Node p announces the best route among all received routes to node c. Therefore, $best_c$ depends on $best_p$. According to GR Export rule, node c will announce only its customer routes to node p. To represent that, $dbest_p$ only depends on $dbest_c$. We illustrate the variable dependency for a pair of nodes with PC relationship in Figure 2.

b) Avoiding cycles for PP relationships: We consider a pair of neighboring nodes, node x and node y, establishing a PP relationship. According to the GR Export rule, node x and node y announce only their customer routes to each other. That is, $best_x$ depends on $dbest_y$ and $best_y$ depends on $dbest_x$. We illustrate the variable dependency for a pair of nodes with PP relationship in Figure 3.

Figure 4 illustrates the biNode model for an AS-level topology and its VDG. There are two nodes in the VDG for each node, where the upside node of node i represents $best_i$ and the downside node of node i represents $dbest_i$.

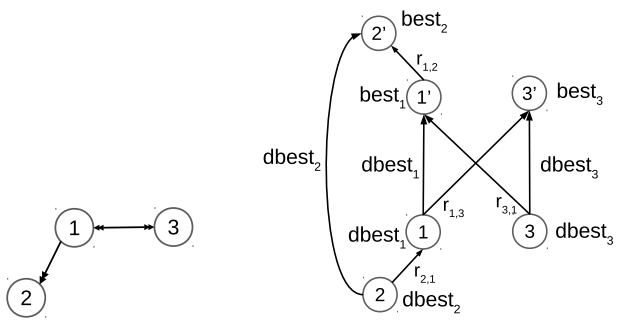
Theorem III.1. *If all nodes follow the Gao-Rexford guideline, and there is no provider-customer cycle, then the biNode model is acyclic*[1].

If the biNode model is acyclic, the SMT solver does not need to guess the satisfiable assignment for any variable.

2) Equivalence Between the BiNode Model and the Topology-based Model: We say that two models are equivalent if the satisfiable assignments for the variables representing the best routes are the same in the two models.

Theorem III.2. *If all nodes follow the Gao-Rexford guideline, the biNode model and the topology-based model are equivalent.*

[1]The technique report includes the proof of Theorems in this paper. Access it through http://rio.ecs.umass.edu/mnilpub/papers/infocom2020-shao-tr.pdf.



(a) A topology with relations. (b) VDG of the biNode model.

Fig. 4: The biNode model for a topology with both PC and PP relationships.

We prove Theorem III.2 by constructing a satisfiable assignment of one model from any satisfiable assignment of the other model. On one hand, the best routes of nodes in the topology-based model are the best routes of nodes in the biNode model. Given the assignments for the best routes in the topology-based model, we can show that there are always satisfiable assignments for the variables that are in the biNode model but not in the topology-based model, such as $dbest$. On the other hand, any satisfiable assignment for variables in the biNode model can be used to derive a satisfiable assignment in the topology-based model.

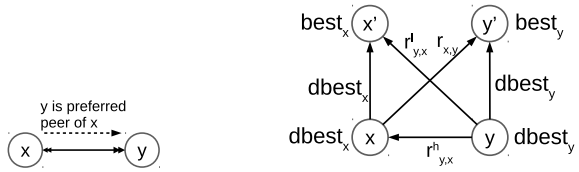
3) Applying to iBGP: The biNode model can be applied to iBGP sessions. BGP routers within each AS also has the hierarchy structure when router reflectors (RRs) are applied. The routing policy of the hierarchy within an AS is similar to that of the hierarchy at AS-level. Within an AS, each RR connects with a set of its clients. All RRs connect with each other in a full mesh. For route announcement, RRs follow two rules:

- if a route is received from non-client peer, announce it to clients only and eBGP peers;
- if a route is received from a client peer, announce it to all peers, except the originator of the route.

For route selection, RRs always prefer routes from their own clients. When we treat the RR as a provider and its clients as its customers, the above routing policy of RRs follows the Gao-Rexford guideline. Then, we can apply the biNode model to verify iBGP within an AS.

C. General BiNode Model for Any Routing Policy

Routing policies of ASs might not follow the Gao-Rexford guideline [18], [3], [25], [26]. Now, we consider how to build the policy-aware model when routing policies of some ASs do not follow the Gao-Rexford guideline. We use the scheme of the biNode model to generate variables that represent the best routes and the route announcements, and the constraints for routing policies of ASs that follow the Gao-Rexford guideline. In the following, we illustrate how to generate constraints for the routing policies of ASs that do not follow the Gao-Rexford guideline. We refer to the policy-aware model that works for any routing policy as *general biNode model*.



(a) Preferred peer. (b) VDG of the general biNode model.

Fig. 5: The general biNode model for two ASs where one is a preferred peer of the other.

1) General BiNode Model Construction: An AS might violate GR preference rule or GR export rule. In the following, we discuss how to generate constraints for these two types of violations respectively. The basic idea is to hold the property that $dbest$ is the best route among a set of high-ranking routes.

a) Preferred Peer/Provider: Violating GR Preference rule means that an AS prefers a route received from its peer or provider over its customer routes. We refer to the peer or the provider as *preferred peer* or *preferred provider* of the AS.

If AS i prefers a route from its preferred peer/provider, p, over any of its customer routes, we make $dbest_i$ depends on the routes with high preference. To do that, we use two records, $r^h_{p,i}$ and $r^l_{p,i}$, to represent the route announcement from the preferred peer/provider, p, to AS i. Any route received from AS p that is preferred over any customer route of AS i is represented as $r^h_{p,i}$ while the other routes from AS p are represented as $r^l_{p,i}$. Therefore, we generate the following constraints.

```
if r_{p,i}.valid then
  if r_{p,i}.pref >= cpref then
    r^h_{p,i} = r_{p,i}; r^l_{p,i}.valid = False
  else
    r^l_{p,i} = r_{p,i}; r^h_{p,i}.valid = False
else
  r^h_{p,i}.valid = False; r^l_{p,i}.valid = False
```

where $cpref$ is the lowest local preference among all customer routes. Then, we make $dbest_i$ to depend on $r^h_{p,i}$ and make $best_i$ to depend on $r^l_{p,i}$ through the following constraints.

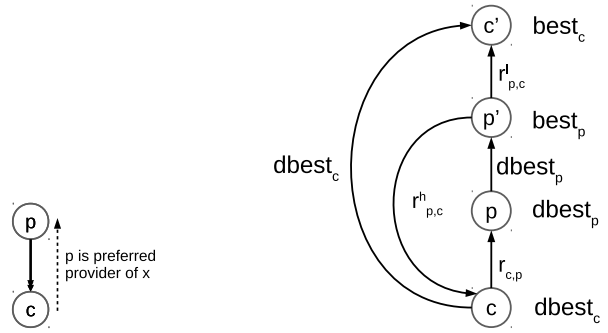
$$dbest_i = f_{selection}(r_{k+1,i}, ..., r_{n,i}, r^h_{p,i}) \quad (7)$$

$$best_i = f_{selection}(r_{1,i}, ..., r^l_{p,i}, ..., r_{k,i}, dbest_i) \quad (8)$$

where $p \in [1, k]$ and AS p is a preferred peer/provider of AS i.

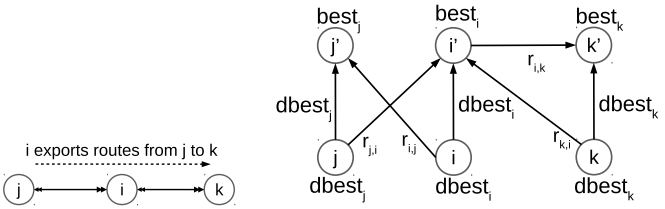
Figure 5 illustrates two ASs, x and y, where AS y is a preferred peer of AS x and the VDG for variables of these two ASs in the general biNode model. Figure 6 illustrates two ASs with PC relationship, where AS p is a preferred provider of AS c and the VDG for variables of these two ASs.

b) Exporting Peer/provider Routes to Peer/provider: An AS violating GR Export rule means that the AS will export a peer route or a provider route to its peer or provider. In that case, the route announcement to the peer/provider should not be derived from $dbest$, but from $best$. Because, $dbest$ does not reflect the routes received from peers or providers. More specifically, if AS i announces routes from a peer/provider,



(a) Preferred provider. (b) VDG of the general biNode model.

Fig. 6: The general biNode model for two ASs where one is preferred provider of the other.



(a) Announcing peer routes to a peer. (b) VDG of the general biNode model.

Fig. 7: The general biNode model for the scenario where an AS exports peer routes to another peer.

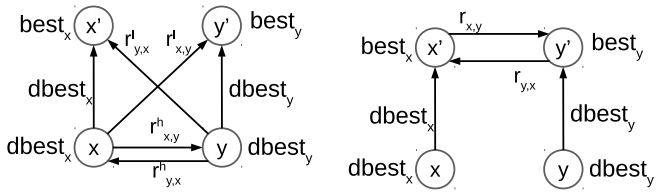
AS j, to another peer/provider, AS k, $r_{i,k}$ should be defined in the following equation:

$$r_{i,k} = f^{i,k}_{export}(best_i) \quad (9)$$

Figure 7 illustrates an example of a topology with three ASs, where AS j and AS k both peer with AS i, and AS i exports routes from AS j to AS k. Figure 7(b) shows the VDG for the special export policy that violates GR Export rule in the general biNode model.

2) Reducing Cycles between Neighboring ASs: The general biNode model accommodates to routing policies that do not follow the Gao-Rexford guideline. However, there might be cycles between the records of two neighboring ASs. For example, a pair of neighboring ASs with a peer-peer relationship might treat each other as a preferred peer and prefer routes from each other over their own customer routes. Then, $dbest$ of these two ASs will depend on each other. Figure 8(a) illustrates the cycles of the general biNode model in this case. As another example, a pair of neighboring ASs with a peer-peer relationship might announce their provider routes to each other. Then, $best$ of these two ASs will depend on each other. Figure 8(b) illustrates the cycles between $best$ of two neighboring ASs when the two ASs announce provider routes to each other.

Typically, the neighboring ASs are not mutual dependent in the general biNode model. In the general biNode model, we say that a pair of neighboring ASs is *mutual dependent*, if any subset of nodes representing these two ASs forms a cycle in the VDG of the model. For a pair of neighboring ASs with a peer-peer relationship, if any of them follows



(a) The general biNode model for two ASs treating each other as preferred peers.

(b) The general biNode model for two ASs announcing provider routes to each other.

Fig. 8: The general biNode model for two ASs with peer-peer relationship violating GR Preference rule or GR Export rule.

the Gao-Rexford guideline, the two neighboring ASs are not mutual dependent. For a pair of neighboring ASs with a provider-customer relationship, if the customer follows the Gao-Rexford guideline, the two neighboring ASs are not mutual dependent. The routing policies of the majority of ASs in the Internet do not violate the Gao-Rexford guideline. Therefore, the majority of neighboring ASs are not mutual dependent in the general biNode model.

Theorem III.3. *Given two ASs with a peer-peer relationship, if one AS prefers customer routes over peer routes from the other AS and announces only customer routes to the other AS, then these two ASs are not mutual dependent in the general biNode model.*

Theorem III.4. *Given two ASs, with a provider-customer relationship, if the customer prefers its customer routes over routes from the provider and announces only its customer routes to the provider, then these two ASs are not mutual dependent in the general biNode model.*

Theorem III.3 and Theorem III.4 can be proved through constructing VDG of the general biNode model for a pair of neighboring ASs, given the routing policy that satisfies the conditions in these Theorems.

According to Theorem III.3 and Theorem III.4, whether there is a cycle between a pair of ASs is only determined by the routing policies between these two ASs. Even an AS does not follow the Gao-Rexford guideline, the AS might not violate it in the routing policies for all its neighbors. Therefore, the cycles in the general biNode model should be rare.

3) Equivalence between the General BiNode Model and the Topology-based Model:

Theorem III.5. *The general biNode model and the topology-based model are equivalent.*

Similar to the proof of Theorem III.2, we can also prove Theorem III.5 by constructing a satisfiable assignment of one model from any satisfiable assignment of the other model.

IV. Query Specification

Given routing policies, the verifier can model the policies and perform a collection of verification queries for the policies. In the following, we use a few verification queries as examples to illustrate the capability of the verifier.

A. Detecting Potential Hijacking Attack

The inter-domain routing system is notoriously vulnerable to hijacking attacks [6], [20], [19]. In a hijacking attack, an attacker AS announces a prefix belonging to another AS in the hope to attract traffic so as to eavesdrop, intercept or blackhole the traffic. Hijacking is an on-going activities and have been observed through a series of analysis and/or monitoring methods [27], [29], [22], [36]. Therefore, it is critical for an AS to realize its vulnerability to hijacking attacks. Before committing to a routing policy, an AS might want to check whether a policy will lead it more vulnerable to hijacking attacks by an attacker. For example, an AS within U.S. might want to make sure its traffic to Google will not be hijacked by an AS from a particular country.

To illustrate the query specification, we consider the scenario where AS I wants to ensure its traffic to a destination, D, will not be hijacked by an attacker, AS A. The verification query is whether the attacker, AS A, can hijack the traffic from AS I to a destination by manipulating its export policy. That is, AS A might export fake routes to its neighbors. We assume that the routing policy of the attacker does not lead to route oscillation, since that is not the goal of the attacker. To detect potential hijacking attack, network operators can use the following query.

$$Q \iff \exists P_A, P_A \wedge Waypoint(I, A) \wedge Origin(D)$$

where, P_A is the policy of AS A, D is the destination AS, and $Waypoint(I, A)$ means that the best route of AS I goes through AS A. To represent the logic of $Waypoint(I, A)$ by SMT constraints, we add a binary variable, $flag_A$, to each route in the model. If $r.flag_A$ is $True$, the route r goes through the AS A. We modify the selection and export constraints (Equation 7, 8 and 9), to include the $flag_A$ variable for each record. For the routes of AS A, $best_A.flag_A$ and $dbest_A.flag_A$ are $True$. For the other routes, the value of $flag_A$ is propagated through the route propagation. Then, we can represent the waypoint property as follows.

$$Waypoint(I, A) \iff r_I.flag_A \wedge r_I.valid$$

If the SMT solver gets a satisfiable assignment, AS A can manipulate its routing policy to attract the traffic from AS I.

B. Inbound Traffic Engineering

As the Internet evolved, it becomes a meshed network [24], [13], [2], [17], [23], where networks directly interconnect with each other through dedicated links or Internet Exchange Points (IXP). As a result, an AS might be accessed by the rest of the Internet through multiple neighbors of the AS. The network operators can perform the inbound traffic engineering to select the neighbor from which the traffic comes. For example, the network operators might want to select one of the providers for the inbound traffic from a specific network, such as Google.

The verifier can help the network operators to set up their routing policies for the inbound traffic engineering. The network operators might want to know whether they can achieve

the purpose if they only change the routing policy of their own network. To specify the query, we can also use the waypoint property. Namely, the best route of the specific source network goes through a selected neighbor of the destination. The query can be represented as follow.

$$Q \iff \exists P_D, P_D \wedge Waypoint(I, A) \wedge Origin(D)$$

where, I is the source network, D is the destination AS and A is the selected neighbor of D for the traffic from I to D. If the SMT solver gets a satisfiable assignment, the network operators can reach the inbound traffic engineering goal through adjusting their own routing policies only.

C. Impact of AS De-Peering

Settlement-free connections are established between a pair of ASs to distribute traffic for their respective customer networks. In contrast, a pair of peering ASs might decide to terminate the settlement-free connections due to the reasons, such as unbalance traffic volume. This operation is known as *De-Peering*. Although, de-peering is not common, it might cause significant impact on the Internet. Especially, when de-peering is over two Tier-1 ASs, it might break the connectivity between their respective customers [35].

Given a de-peering between two neighboring ASs, the verifier can be used to explore its impact. More specifically, it can verify whether some ASs can not reach a destination due to the de-peering. We have the following query.

$$Q \iff \exists I, \neg r_I.valid \wedge Disconnect(A, B) \wedge Origin(D)$$

where, A and B are the two ASs that de-peer with each other and D is the destination. To represent $Disconnect(A, B)$ in SMT constraints, we set $f^{A,B}_{export}$ and $f^{B,A}_{export}$ to return the invalid route, r_{empty}. If the SMT solver returns SAT, AS D can not be reached by at least one AS due to the de-peering.

V. Query Optimizations

In this section, we first analyze the size of policy-aware models to illustrate the difficulty of the verification on the Internet topology. Then, we propose a query-specific pruning method to reduce variables in policy-aware models.

A. Policy-aware Model Size

Although policy-aware models in Section III can accelerate the SMT solving process, it leads to a large number of variables. For the same BGP system, the policy-aware models need more variables than the topology-based model. Given a topology with N nodes and M edges, the number of variables in the policy-aware model is $O((2*N+4*M)*V_{route})$, where V_{route} is the number of variables that represent a single route or route announcement.

Based on the Internet measurement, there are more than $60,000$ ASs and $300,000$ links between ASs [9]. As a result, to model the entire Internet topology, if we treat each AS as one router/node, millions of variables are necessary in the policy-aware models. In the following, we propose a pruning method to reduce the number of variables in the policy-aware models.

B. Model Pruning

We can prune policy-aware models through removing variables irrelevant to the query result. For a specific verification query, variables can be removed due to two facts. Firstly, a verification query is usually associated with a small set of routes which depend only on a subset of records in the models. The other records can be removed from the model. Secondly, according to the construction of the policy-aware models in Section III-B1 and Section III-C1, some records are always invalid routes, r_{empty}. These records do not impact the assignment of the other records. On one hand, in a policy-aware model, if a route is invalid, the associated route announcements depending on the route is invalid. On the other hand, if a route announcement is invalid, the route announcement will not be selected as a best route. Therefore, the records representing invalid routes can be removed from the model.

Given the query, we can construct the VDG of the policy-aware model and identify those irrelevant records. To identify the irrelevant records, we perform both forward and backward graph traversals on the VDG.

a) Backward Traversal: We refer to an route that is regulated in the properties of a verification query as a *target route*. We traverse the VDG starting from a target route in reverse direction and mark all visited nodes as routes that are depended by the target route. The unvisited records do not impact the assignment for the target route.

b) Forward Traversal: We refer to an AS as an *origin AS* for a prefix if the AS originates a route of the prefix without needing a route announcements from its neighbors. We traverse the VDG from $dbest$ of an origin AS and mark all visited nodes as routes that depend on the origin AS. The unvisited records should be invalid.

A record is kept in the model if the record is visited both in the backward traversal and the forward traversal. The other records are removed. A query might be related to multiple target routes. For example, operators might want to ensure that all routers will have valid routes to a destination. That is, we might also perform multiple backward traversals in the second step. In a verification query, there might be multiple origin ASs. For example, a hijacking attacker might be an origin AS in the verification. That is, there are multiple forward traversals in the first step. The time complexity of the pruning algorithm is $O((N_o + N_t) * M)$, where N_o is the number of origin ASs, N_t is the number of target routes and M is the number of edges in the graph.

VI. Evaluation

We implement verification toolkits and evaluate the general biNode model. We introduce the input dataset for the verification system in Section VI-B. In section VI-C, we compare the performance of the general biNode model and the topology-based model on a set of topologies with modest size. In Section VI-D we verify queries on the Internet topologies to show that the general biNode model can verify queries at

Internet scale. Finally, we show the effect of model pruning in Section VI-E.

A. Implementation

We implement the prototype of the verification system which includes both the topology-based model and the general biNode model. The verification system takes as input the network topology with routing policies and the query. To perform the verification, the verification system will encode the query into a SMT problem which is fed into a SMT solver to get the verification result. We use Z3 [12] as the SMT solver to solve the SMT problem.

B. Experiment Setting

To evaluate the topology-based model and the general biNode model for network verification, we use the Internet AS-level topologies in CAIDA AS Relationship Database [9] and generate routing policies based on the relationship. To generate a set of topologies with modest size for model comparison, we extract ASs and the associated links from the real topology. To guarantee that the extracted topology is connected, we select ASs through random walk starting from an origin AS and all links between any pair of visited ASs.

We generate routing policies following the Gao-Rexford guideline and beyond the Gao-Rexford guideline. The majority of ASs have routing policies following the Gao-Rexford guideline. Accordingly to the survey [18], about 30% ASs might violate GR Preference rule or GR Export rule. We generate routing policies with different levels of violation: low violation (*LV*), moderate violation (*MV*) and high violation (*HV*), where 10%, 20% and 30% ASs violate both GR export rules and GR preference rules respectively.

We test the verification system through a set of queries on topologies of various sizes. We test the verification system on a server with an Intel(R) Xeon(R) E5607 2.27GHz CPU and 16GB memory.

C. Comparison between the General BiNode Model and the Topology-based Model

We compare the verification time of the topology-based model and the general biNode model through three types of queries: prefix-hijacking query, inbound traffic engineering query and router reachability query. We perform these queries on the topologies of various sizes. For each topology, we randomly generate 10 queries of the same type and measure the average running time for the comparison.

1) Verifying Routing Policies under the Gao-Rexford Guideline: In Figure 9, we show the running time of the topology-based model and the general biNode model for verifying routing policies under the Gao-Rexford guideline and 99% confidence intervals for the running time. For all three kinds of queries, the general biNode model performs better than the topology-based model when the topology containing hundreds of nodes (ASs/routers).

When pruning optimization is used, the general biNode model can speed the verification by more than 100x for prefix-hijacking queries and inbound traffic engineering queries at topologies of modest sizes. For the router reachability query, the speedup is only 6x. The reason is that in the query, the property regulates the best routes of all routers. Therefore, there are not a lot of irrelevant variables that can be pruned.

As Figure 9 shows, the speedups for all queries increase with the topology size. It indicates that the general biNode model with pruning optimization has much better scalability.

2) Verifying Routing Policies beyond the Gao-Rexford Guideline: We also compare the performance of the topology-based model and the general biNode model for routing policies beyond the Gao-Rexford guideline. Figure 10 shows the average running time with 99% confidence level of prefix-hijacking queries on topologies with routing policies violating the Gao-Rexford guideline. According to the result in Figure 10, routing policies with reasonable violations (low level, moderate level and high level) to the Gao-Rexford guideline do not significantly impact the performance. As Figure 10 shows, the general biNode model with pruning is also 100x faster than the topology-based model at a modest topology size when the routing policies are beyond the Gao-Rexford guideline.

TABLE II: Effect of model pruning (for inbound traffic engineering queries) in the general biNode model.

Year	# of ASs	# of edges	PR (90% confidence level)
2003	15,320	34,720	547.2 ± 53.2
2008	28,411	78,997	1913.3 ± 1541.2
2013	44,326	149,476	4910.8 ± 1105.5
2018	60,874	300,634	2845.7 ± 1024.1

D. Scalability of the General BiNode Model

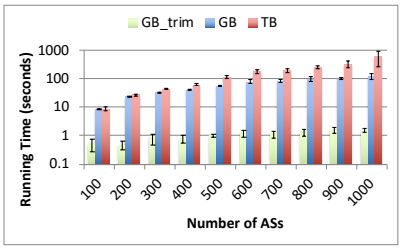
We illustrate the scalability of the general biNode model on the Internet-scale topologies. Figure 11 illustrates the average running time for answering prefix-hijacking and inbound traffic engineering queries on the Internet topologies from Year 2003 to Year 2018. As shown in Figure 11, even for queries on the Internet-scale topology, the verification system can answer the queries within two minutes.

E. Effect of Model Pruning

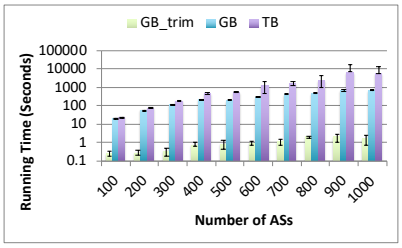
Pruning optimization on the general biNode model can significantly reduce the variables in the model. To measure the variable reduction, we define *pruning ratio (PR)*, as the ratio of variable number in the general biNode model without pruning to variable number after pruning. To evaluate the pruning optimization, we measure PR for queries on the Internet topologies. Table II illustrates the size of Internet topologies from Year 2003 to Year 2018 with five-year intervals and the PR with 90% confidence level. We randomly select 100 inbound traffic engineering queries for each topology and list the average PR in Table II. As shown in Table II, pruning optimization can achieve very high PR and significantly improve the verification performance.

VII. Related Work

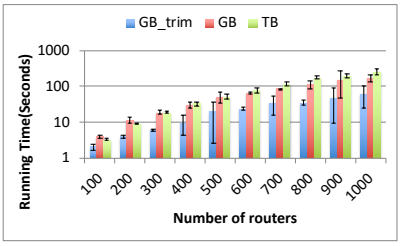
a) Network-wide Verification: A series of network verification techniques [7], [34], [14], [16], [1] have been proposed



(a) Prefix-hijacking queries.



(b) Inbound traffic engineering queries.



(c) Reachability queries.

Fig. 9: Comparison among the verification time of topology-based model (TB), the general biNode (GB) model and the general biNode model with pruning (GB_trim) for routing policies under the Gao-Rexford guideline.

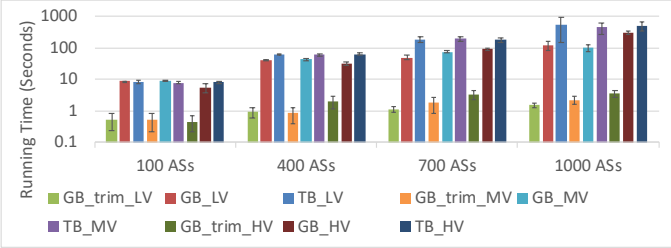


Fig. 10: Prefix-hijacking queries on topologies with routing policies beyond the Gao-Rexford guideline.

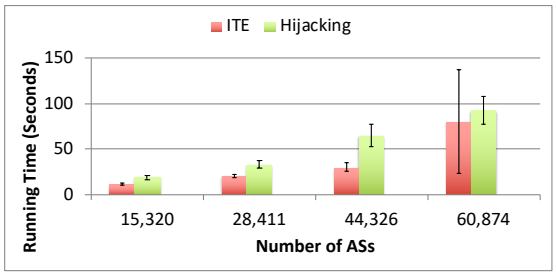


Fig. 11: Running time of the general biNode model with pruning for queries on Internet-scale topologies with 99% confidence level.

to verify reachability at network-wide. These proposed models can typically handle up to hundreds of routers at most. In contrast, our model can scale to Internet-wide queries.

b) Internet-wide Network Verification: A number of formal methods have been exploited to verify properties for inter-domain routing. To study the safety property of BGP systems, Satisfiable Module Theories (SMT) [31] and Rewriting Logic [30], [32], [33] are used to verify the convergence conditions. In our models, we assume the routing convergence and verify a board range of routing properties. A model checking tool is used to search possible attraction attacks on the Internet [28]. The computational complexity of devising a hijacking strategy in different policy-based routing protocols is analyzed in [11]. Both work assume that the Gao-Rexford guideline is followed by all ASs. In contrast, our models can verify queries about the hijacking attack on routing policies beyond the Gao-Rexford guideline.

c) Scaling Verification Through Model Size Reduction: The model size can be reduced to accelerate the network verification [30], [32], [8]. According to the goals of network verification, these techniques preserve different properties. Wang *et al.* [30], [32] propose a network compression method for policy-based routing to accelerate the analysis of convergence behavior through preserving safety property. Recently, Beckett *et al.* [8] propose a compressed model for a broad range of routing protocols to preserve more general path properties, such as reachability, loop freedom and absence of black holes. These methods reduce the model size through exploiting the duplicate router configuration and the symmetry in the topologies. In contrast, our model exploits the intrinsic hierarchy in the routing policy and reflects the hierarchy explicitly in the model to accelerate the formal method.

d) Privacy Preserving of Routing Policy: Recently, a number of research efforts aim at enhancing the inter-domain routing through proposing a logically centralized routing control plane. To preserve policy privacy for domains, Secure Multi-Party Computation (SMPC) methods are proposed for policy-compliant routes computation [4], [10], [21]. In order to preserve the privacy of routing policies, we might implement the proposed model in a logically centralized verification system by exploiting SMPC techniques.

VIII. Conclusion

In this paper, we propose the general biNode model to verify the policy-based routing through characterizing the verification problem into a SMT problem. Through analyzing the intrinsic hierarchy of the routing policy, we can reflect the hierarchy in the constraints to accelerate the SMT solving process. Further, we prune the general biNode model through removing irrelevant variables. We implement the network verification toolkits which include the general biNode model and the topoloy-based model, and evaluate these models. The experimental results show that the general biNode model can reduce the time it takes to perform verification by as much as 100x even under a modest topology size. It takes only a few minutes to answer a query for a topology containing tens of thousands of nodes.

IX. Acknowledgment

The work was supported in part by NSF grants CNS-1900866 and CCF-1918187.

REFERENCES

[1] A. Abhashkumar, A. Gember-Jacobson, and A. Akella. Tiramisu: Fast and general network verification. CoRR, abs/1906.02043, 2019.

[2] B. Ager, N. Chatzis, A. Feldmann, N. Sarrar, S. Uhlig, and W. Willinger. Anatomy of a large european ixp. In Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, SIGCOMM 12, pages 163–174, New York, NY, USA, 2012. Association for Computing Machinery.

[3] R. Anwar, H. Niaz, D. Choffnes, I. Cunha, P. Gill, and E. Katz-Bassett. Investigating interdomain routing policies in the wild. In Proceedings of the 2015 Internet Measurement Conference, IMC 15, pages 71–77, New York, NY, USA, 2015. Association for Computing Machinery.

[4] G. Asharov, D. Demmler, M. Schapira, T. Schneider, G. Segev, S. Shenker, and M. Zohner. Privacy-preserving interdomain routing at internet scale. Proceedings on Privacy Enhancing Technologies, 2017(3):147–167, 2017.

[5] H. Balakrishnan. How youtube was hijacked, 2009.

[6] H. Ballani, P. Francis, and X. Zhang. A study of prefix hijacking and interception in the internet. In Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM 07, pages 265–276, New York, NY, USA, 2007. Association for Computing Machinery.

[7] R. Beckett, A. Gupta, R. Mahajan, and D. Walker. A general approach to network configuration verification. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 17, pages 155–168, New York, NY, USA, 2017. Association for Computing Machinery.

[8] R. Beckett, A. Gupta, R. Mahajan, and D. Walker. Control plane compression. In Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 18, pages 476–489, New York, NY, USA, 2018. Association for Computing Machinery.

[9] CAIDA. The caida as relationships dataset, <05.1998-05.2018>, Oct 2019.

[10] Q. Chen, C. Qian, and S. Zhong. Privacy-preserving cross-domain routing optimization - a cryptographic approach. In 2015 IEEE 23rd International Conference on Network Protocols (ICNP), pages 356–365, Nov 2015.

[11] M. Chiesa, G. Di Battista, T. Erlebach, and M. Patrignani. Computational complexity of traffic hijacking under bgp and s-bgp. Theor. Comput. Sci., 600(C):143–154, Oct. 2015.

[12] L. De Moura and N. Bjørner. Z3: An efficient smt solver. In Proceedings of the Theory and Practice of Software, 14th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS08/ETAPS08, pages 337–340, Berlin, Heidelberg, 2008. Springer-Verlag.

[13] A. Dhamdhere and C. Dovrolis. The internet is flat: Modeling the transition from a transit hierarchy to a peering mesh. In Proceedings of the 6th International COnference, Co-NEXT 10, New York, NY, USA, 2010. Association for Computing Machinery.

[14] S. K. Fayaz, T. Sharma, A. Fogel, R. Mahajan, T. Millstein, V. Sekar, and G. Varghese. Efficient network reachability analysis using a succinct control plane representation. In Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI16, pages 217–232, USA, 2016. USENIX Association.

[15] L. Gao and J. Rexford. Stable internet routing without global coordination. IEEE/ACM Trans. Netw., 9(6):681–692, Dec. 2001.

[16] A. Gember-Jacobson, R. Viswanathan, A. Akella, and R. Mahajan. Fast control plane analysis using an abstract representation. In Proceedings of the 2016 ACM SIGCOMM Conference, SIGCOMM 16, pages 300–313, New York, NY, USA, 2016. Association for Computing Machinery.

[17] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. The flattening internet topology: Natural evolution, unsightly barnacles or contrived collapse? In International Conference on Passive and Active Network Measurement, pages 1–10. Springer, 2008.

[18] P. Gill, M. Schapira, and S. Goldberg. A survey of interdomain routing policies. SIGCOMM Comput. Commun. Rev., 44(1):28–34, Dec. 2013.

[19] S. Goldberg, S. Halevi, A. D. Jaggard, V. Ramachandran, and R. N. Wright. Rationality and traffic attraction: Incentives for honest path announcements in bgp. SIGCOMM Comput. Commun. Rev., 38(4):267–278, Aug. 2008.

[20] S. Goldberg, M. Schapira, P. Hummon, and J. Rexford. How secure are secure interdomain routing protocols. In Proceedings of the ACM SIGCOMM 2010 Conference, SIGCOMM '10, pages 87–98, New York, NY, USA, 2010. ACM.

[21] D. Gupta, A. Segal, A. Panda, G. Segev, M. Schapira, J. Feigenbaum, J. Rexford, and S. Shenker. A new approach to interdomain routing based on secure multi-party computation. In Proceedings of the 11th ACM Workshop on Hot Topics in Networks, HotNets-XI, pages 37–42, New York, NY, USA, 2012. ACM.

[22] S. Hong, J. W. Hong, and H. Ju. Ip prefix hijacking detection using the collection of as characteristics. In 2011 13th Asia-Pacific Network Operations and Management Symposium, pages 1–7, Sep. 2011.

[23] R. Klöti, B. Ager, V. Kotronis, G. Nomikos, and X. Dimitropoulos. A comparative look into public ixp datasets. SIGCOMM Comput. Commun. Rev., 46(1):21–29, Jan. 2016.

[24] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian. Internet inter-domain traffic. In Proceedings of the ACM SIGCOMM 2010 Conference, SIGCOMM 10, pages 75–86, New York, NY, USA, 2010. Association for Computing Machinery.

[25] R. Mazloum, M. O. Buob, J. Augè, B. Baynat, D. Rossi, and T. Friedman. Violation of interdomain routing assumptions. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 8362 LNCS:173–182, 2014.

[26] S. Y. Qiu, P. D. McDaniel, and F. Monrose. Toward valley-free inter-domain routing. In 2007 IEEE International Conference on Communications, pages 2009–2016, June 2007.

[27] P. Sermpezis, V. Kotronis, P. Gigis, X. Dimitropoulos, D. Cicalese, A. King, and A. Dainotti. Artemis: Neutralizing bgp hijacking within a minute. IEEE/ACM Trans. Netw., 26(6):2471–2486, Dec. 2018.

[28] A. Sosnovich, O. Grumberg, and G. Nakibly. Analyzing internet routing security using model checking. In Proceedings of the 20th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning - Volume 9450, LPAR-20 2015, pages 112–129, New York, NY, USA, 2015. Springer-Verlag New York, Inc.

[29] M. Tahara, N. Tateishi, T. Oimatsu, and S. Majima. A method to detect prefix hijacking by using ping tests. In Y. Ma, D. Choi, and S. Ata, editors, Challenges for Next Generation Network Operations and Service Management, pages 390–398, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[30] A. Wang, A. J. T. Gurney, X. Han, J. Cao, B. T. Loo, C. Talcott, and A. Scedrov. A reduction-based approach towards scaling up formal analysis of internet configurations. In IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, pages 637–645, April 2014.

[31] A. Wang, L. Jia, W. Zhou, Y. Ren, B. T. Loo, J. Rexford, V. Nigam, A. Scedrov, and C. Talcott. Fsr: Formal analysis and implementation toolkit for safe interdomain routing. IEEE/ACM Trans. Netw., 20(6):1814–1827, Dec. 2012.

[32] A. Wang, C. Talcott, A. J. T. Gurney, B. T. Loo, and A. Scedrov. Reduction-based formal analysis of bgp instances. In Proceedings of the 18th International Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS'12, pages 283–298, Berlin, Heidelberg, 2012. Springer-Verlag.

[33] A. Wang, C. Talcott, L. Jia, B. T. Loo, and A. Scedrov. Analyzing bgp instances in maude. In Proceedings of the Joint 13th IFIP WG 6.1 and 30th IFIP WG 6.1 International Conference on Formal Techniques for Distributed Systems, FMOODS'11/FORTE'11, pages 334–348, Berlin, Heidelberg, 2011. Springer-Verlag.

[34] K. Weitz, D. Woos, E. Torlak, M. D. Ernst, A. Krishnamurthy, and Z. Tatlock. Scalable verification of border gateway protocol configurations with an smt solver. In Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2016, pages 765–780, New York, NY, USA, 2016. Association for Computing Machinery.

[35] J. Wu, Y. Zhang, Z. M. Mao, and K. G. Shin. Internet routing resilience to failures: Analysis and implications. In Proceedings of the 2007 ACM CoNEXT Conference, CoNEXT 07, New York, NY, USA, 2007. Association for Computing Machinery.

[36] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush. Ispy: Detecting ip prefix hijacking on my own. In Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, SIGCOMM 08, pages 327–338, New York, NY, USA, 2008. Association for Computing Machinery.