INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & MANAGEMENT

A SURVEY OF MODELLING BGP ROUTE SELECTION FOR EVALUATING STRUCTURAL CHANGE IN A NETWORK

Ananthaneni Venkata Naga Surendra,Dr.Syed Umar*, Pillarisetty Vivek, B.Kiran Sai
B.Tech Student, ECE dept., K L University, vaddeswaram, Guntur, AP, India
*Assoc.Professor, CSE dept., K L University, Vaddeswaram, Guntur, AP, India

ABSTRACT

The Internet has quickly evolved into a vast global network owned and operated by thousands of different administrative entities. During this time, it became apparent that vanilla shortest-path routing would be insufficient to handle the myriad operational, economic, and political factors involved in routing. ISPs began to modify routing configurations to support routing policies, i.e. goals held by the router's owner that controlled which routes were chosen and which routes were propagated to neighbours. The performance of IP networks depends on a wide variety of dynamic conditions. Traffic shifts, equipment failures, planned maintenance, and topology changes in other parts of the Internet can all degrade performance. To maintain good performance, network operators must continually reconfigure the routing protocols. Operators configure BGP to control how traffic flows to neighbouring Autonomous Systems (ASes), as well as how traffic traverses their networks. However, because BGP route selection is distributed, indirectly controlled by configurable policies, and influenced by complex interactions with intradomain routing protocols, operators cannot predict how a particular BGP structure would behave in practice. To avoid inadvertently degrading network performance, operators need to evaluate the effects of structure changes before deploying them on a live network. We propose an algorithm that computes the outcome of the BGP route selection process for each router in a single AS, given only a static snapshot of the network state, without simulating the complex details of BGP message passing.anage them differently. We show that this can be achieved by utilising knowledge bases and new software solutions.

Keywords: BGP Route Selection.

INTRODUCTION

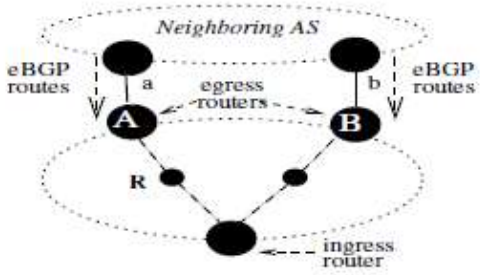
The delivery of IP packets through the Internet depends on a large collection of routers that compute end-to-end paths in a distributed fashion, using standardized routing protocols. Providing low latency, high throughput, and high reliability requires network operators to adjust routing protocol structure when performance problems arise or network conditions change. For example, an operator might adjust the structure to respond to network congestion or equipment failures or to prepare for planned maintenance. However, the complexity of the protocols, the large number of tuneable parameters, and the size of the network make it extremely difficult for operators to reason about the effects of their actions. The common approach of .tweak and pray. is no longer acceptable in an environment where users have high expectations for performance and reliability [1]. To avoid costly debugging time and catastrophic mistakes, operators must be able to make predictions quickly based on an accurate model of the routing protocols.

Previous work in this area has focused on Interior Gateway Protocols (IGPs), such as Open Shortest Path First (OSPF) and Intermediate System-Intermediate System (IS-IS), that operate within a single Autonomous System (AS). In these protocols, each link has a structure is able integer .cost. that is used to compute the shortest path (smallest cost path) through the network. Tuning these link weights gives operators a way to modify the paths inside the AS to satisfy network and user performance goals [2]. Several existing tools [3,4, 5] capture how changes to the link weights would affect the flow of the offered traffic over the new paths and also propose good candidate settings of the link weights. Although these tools are extremely valuable to network operators, the flow of traffic in large service provider backbones ultimately depends on the interdomain routing protocol as well. A provider uses the Border Gateway Protocol (BGP) to exchange reachability information with neighbouring domains and to propagate these routes within its own network. Unfortunately, several subtleties make modelling BGP route selection in an AS much more challenging than emulating IGP:

- BGP's distributed, path vector operation means that every router may have a different view of network state. In contrast to link state protocols that good complete information throughout the network, a BGP-speaking router sends incremental reachability information only to its immediate neighbours. In BGP, a router sends an advertisement to notify its neighbour of a new route to a destination prefix and a withdrawal to revoke the route when it is no longer available. Each BGP-speaking router locally computes its own best BGP route from the best routes announced by its neighbours. A change in the best

path at one router can affect the selection of the best route at another router, and no single router has a complete view of the available BGP routes in the AS

- BGP route selection depends on interactions with intradomain routing protocols. Whereas the IGP can be modelled in isolation, the selection of the best BGP route at each router also depends on the IGP path cost to the BGP next hop announcing the route. .Hot potato. routing, where a router prefers the route with the shortest IGP path (the closest exit point), introduces a complex coupling between BGP and the underlying IGP



Network engineering terminology

- Hierarchical iBGP structure affects the routing choicesthat are available at each router. Internal BGP (iBGP) is used to distribute BGP-learned routes throughout an AS. Rather than having an iBGP session between each pair of routers (.full meshiBGP.), large provider networks typically distribute routes in a hierarchical fashion. This makes the choices available at one routerdependent on the decisions made at other routers and may prevent the protocol from converging.

The Internet consists of thousands of Autonomous Systems(ASes)—networks that are each owned and operated by a single institution. BGP is the routing protocol used to exchange reachability information across ASes. Usually each ISP operates one AS, though some ISPs may operate multiple ASes for business reasons (e.g. to provide more autonomy to administrators of an ISP's backbones in the United States and Europe) or historical reasons (e.g. a recent merger of two ISPs). Non-ISP businesses (enterprises) may also operate their own ASes so as to gain the additional routing flexibility that arises from participating in the BGP protocol. Compared to enterprise networks, ISPs usually have more complex policies arising from the fact that they often have several downstream customers, connect to certain customers in multiple geographic locations, have complex traffic engineering goals, and run BGP on internal routers (rather than just border routers as enterprises often do). Although some of the observations we make apply to enterprise networks, our core focus in this paper is on ISP networks. In this section, we describe BGP from the standpoint of a single AS, describing first the protocol that transmits routes from one AS to another, then the decision process used to choose routes, and finally the mechanisms used at routers to implement policy.

NETWORK ENGINEERING

In this section, we discuss network engineering problems that operators commonly face. These examples demonstrate the need to systematically model BGP's route selection process and move beyond today's .tweak and pray. techniques. We then discuss why a practical model of BGP is useful; in particular, we discuss the advantages a model has over simulation and live testing.

Network Engineering Problems

Network operators must adjust routing protocol structure to respond to the following common changes in network conditions: Changes in traffic load. Because traffic is dynamic, the amount
of traffic to any destination may suddenly change, causing changes in traffic distribution across network links. For example, a Web site sometimes experiences a traffic surge due to a flash crowd (i.e., the
.Slashdot effect.); a network that was routing traffic to that destination through a neighbouring AS could experience congestion on an outbound link to that destination. A network operator must rare structure routing policy to alleviate congestion

- Changes in link capacity. Network links are frequently upgraded to higher capacity. In response, network operators may wish to adjust configuration to route additional traffic through recently upgraded links. For example, if link _ in Figure 1 were upgraded from an OC12 to an OC48, a network operator would want to shift traffic from b to a.
- Changes in available routes. Due to protocol dynamics and changing commercial agreements, an AS might suddenly receive an alternate route to a destination that could cause traffic flow to shift dramatically, alternatively, an existing route could suddenly disappear. These events may require a

network operator to rebalance the flow of traffic. Planned maintenance. Network operators commonly perform routine maintenance on portions of their network, adjusting interior routing link weights to divert traf_c away from the part of the network that is undergoing maintenance. If router were undergoing an upgrade, the network operator could adjust both the IGP link weights and the BGP con_ guration to divert traf_c away from without overloading links that border neighbouring domains.

- Failure and disaster planning. An operator may wish to evaluate the robustness of the network by examining the effects of failures on routing and traffic flow. Studying the effects of a component or link failure or even a more serious catastrophe (e.g.,flber cut, tunnel are, or terrorist attack) helps network operators and planners make design decisions. Tools exist to help network operators adjust interior routing protocol parameters to rebalance traffic within an AS [3, 4, 5], but operators must also rebalance traffic across its external links to other networks by restructure BGP. A network operator who wants to shift traffic from link a to b in Figure 1 typically would adjust the import policy for a set of routes at router , commonly by decreasing he .local preference. attribute that router _ assigns to these routes [8]. This would cause the routes learned at router _ for these destinations to appear more attractive than those at _ , causing the traffic to those destinations to exit the AS via.

Selecting A Route At A Router

A BGP router in an ISP may have several alternate routes to reach a particular destination. In the absence of policy, the router would choose the route with the minimum path length, with some arbitrary way to break ties between routes with the same path length. However, in order to give operators greater Table 1: Steps in the BGP decision process.

Step Attribute Controlled by local or neighbour AS?

1. Highest LocalPref local
2. Lowest AS path length neighbour
3. Lowest origin type neither
4. Lowest MED neighbour
5. eBGP-learned over iBGP-learned neither
6. Lowest IGP cost to border router local
7. Lowest router ID (to break ties) neither

control over route selection, several additional attributes were added to advertisements, allowing a router to alter its decisions based on the values of these attributes. The end result is the BGP decision process, consisting of an ordered list of attributes across which routes are compared, as shown in Table 1.

Step	Attribute	Controlled by local or neighbor AS?
1.	Highest LocalPref	local
2.	Lowest AS path length	neighbor
3.	Lowest origin type	neither
4.	Lowest MED	neighbor
5.	eBGP-learned over iBGP-learned	neither
6.	Lowest IGP cost to border router	local
7.	Lowest router ID (to break ties)	neither

The router goes down the list, comparing each attribute in the list across the two routes. If the routes have different values for the attribute, the router chooses the one that has the more desirable attribute, otherwise it moves on to compare the next attribute in the list. The route that is chosen is used by the router to forward packets. The ordering of attributes allows the operator to influence various stages of the decision process. For example, the Local Preference (LocalPref) is the first step in the decision process. By changing LocalPref, an operator can force aroute with a longer AS path to be chosen over a shorter one. As another example, the Multi- first four steps of the decision process that compare BGP attributes. Each internal router then chooses the router in that set that is closest according to the Interior Gateway Protocol (IGP) path cost to reach that border router. For example in Figure1, suppose prefix 6.0.1.0/24 is reachable to B via both A and C, but B's Local Pref is set higher for routes through A. The set of equally good border routers would then contain R1 and R2, and each router in B would select the route that was closest exit point (lowest IGP cost): ra and R1 would choose the route through R1, and all other routers would choose the route through R2.There are three steps a router uses to process route advertisements. First import policy is applied to determine which routes should be filtered and hence eliminated from consideration, and may append or modify attributes. Next, the router applies the decision process to select

the most desirable route. Finally, an export policy is applied which determines which neighbours the chosen route will be exported to. An ISP may implement its policy by controlling any of these three steps, i.e., by modifying import policy to filter routes it doesn't want to use, modifying route attributes to prefer some routes over others, or by modifying export policy to avoid providing routes for certain neighbours to use. In addition, an ISP can modify attributes of routes it advertises, which can influence how its neighbours perform route selection. Exit Discriminator (MED) is typically used by two ASes connected by multiple links to indicate which peering link should be used to reach the AS advertising the attribute. MED was placed lower in the decision process as this allows an ISP to override these suggestions, e.g. by setting LocalPref. Using a strict ordering of attributes in the decision process simplifies policy expression and makes it easiest predict the outcome of making configuration changes. While some vendors allow operators to disable certain steps in the decision process, they typically do not permit the operators to put the steps in a different order. Hence some policies that violate this ordering (e.g. ignore AS path length, or first choose lowest MED then highest LocalPref) may require various hacks which can complicate router configuration and lead to unforeseen side effects. There are different locations where a route attribute can be setby policy: (a) Locally, for example LocalPref is an integer value set at and propagated throughout the local AS and filtered before sending to neighbouring ISPs. (b) Neighbour, for example the MED attribute is typically used by two ASes connected by multiple links to indicate which peering link should be used to reach the AS advertising the MED attribute, and is not used to compare routes through two different next-hop ASes. (c)Neither: some attributes, for example whether the route was learned through an external BGP (eBGP) neighbour or from an internal router speaking BGP (iBGP), are set by the protocol and cannot be changed. The collective results of the decision process across routers isto produce a set of equally good border routers for each prefix, where each router in the set is equivalent according to the

Network Engineering Requirements

Motivated by these practical network engineering examples, we now present several requirements for a network engineering tool that predicts the outcome of BGP route selection. (Throughout the paper, we use the term .prediction. to describe the process of determining the outcome of BGP route selection offline.)

- Operators need to be able to predict the effects of a structure change before deployment in the live network. Operators typically handle network engineering tasks by tuning the existing structure of the live network, witnessing the effects of the change, and reverting to the previous con_guration if the desired effects are not achieved. This method is time consuming and can lead to unnecessary performance degradation. The complexity of interdomain routing makes it essentially impossible to compute back-of the-envelope estimates of the effects of con_guration changes.
- Because network engineering involves exploring a large search space, operators need to predict the effects of each candidate structure change as quickly as possible. Operators usually need to experiment with many possible structure changes before arriving at an acceptable solution. This requires techniques for quickly evaluating the effects of a proposed structure change. In practice, the candidate structure changes typically are just small modifications to the existing structure , which are less likely to cause significant unexpected changes in routes or the offered traffic load. To be useful, a route prediction tool should be fast at evaluating incremental changes.
- Network engineering tasks require an accurate prediction of the outcome of the BGP path selection process but do not require detailed simulation of the protocol dynamics. Network simulators(e.g., SSFNet [7]) help operators understand dynamic routing protocol behavior, but simulation represents network behavior in terms of message passing and protocol dynamics over a certain period of time. In contrast, network engineers usually just need to know the outcome of the path-selection process and not the low-level timing details. Furthermore, existing simulators do not capture some of the relevant protocol interactions that can affect the outcome of the decision process. Simulating every detailed interaction is hard to do without a higher level model of BGP in the first place.

In summary, BGP path selection is a massive, highly tunable, distributed computation. To maintain good end-to-end performance, operators need a way to predict the outcome of this computation under various structure
(1) efficiently,
(2) in a way that facilitates evaluating incremental changes, and
(3) without deploying Structure changes on a live network. In the following sections, we describe a way to model the BGP decision process across every router in an AS; our algorithm correctly and efficiently computes
the outcome of the BGP decision process at each router.

MODELING FRAMEWORK

At a glance, modelling the routing decisions in an AS seems as simple as applying the BGP decision process at each router. However, the BGP routing system inside an AS is not guaranteed to converge to a unique solution where routers make consistent decisions. Even when the system converges, the decision made at one router can affect the options available to other routers. Applying theoretical results from previous work, we impose three

practical constraints on the routing system that make modeling BGP routes election possible. We then show that route selection can be modelled in three distinct phases when these constraints are satisfied.

Constraints on the Routing System

Given the eBGP-learned routes to a destination prefix, we would like to determine which route each router would ultimately select. Unfortunately, certain network configurations do not lend themselves to efficient route prediction. This is a fundamental problem underlying BGP. Some configurations do not converge, and determining whether a system converges to a unique solution is computationally intractable [9]. To make modeling BGP route selection feasible,

ROUTE PREDICTION ALGORITHM

Drawing on the modeling framework from Section 3, this section proposes an algorithm that computes the best route at each router in an AS. Since the path selection process is independent for each prefix, we focus on the prediction problem for a single destination prefix. To ensure that the algorithm is ef_cient, the best route that the algorithm predicts for each router should not change the predictions made earlier for other routers. In other words, *we design the algorithm to emulate a BGP message ordering that does not require any backtracking*. Satisfying this constraint for the first phase of the algorithm is straightforward because applying the import policy is a purely local operation at each router. However, the second and third phases of the algorithm are more complicated because of subtle interactions in the BGP decision process:

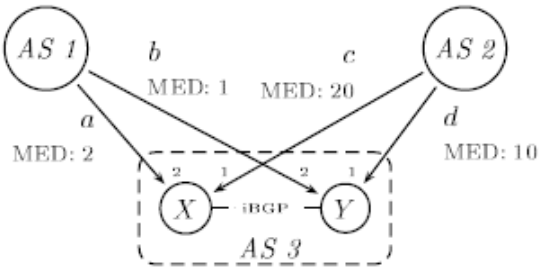
- *The interaction between MED and router ID* prevents the second phase of the algorithm from simply selecting the locallybestroute at each router. This complication arises because the BGP decision process only compares MED values for routes with the *same next-hop AS*. As such, the MED value of an eBGP route learned at one router may affect the *local* ranking of eBGP-learned routes at another router.
- *The interaction between iBGP and the IGP* prevents the third phase of the algorithm from visiting each router in the ASin an arbitrary order. This complication arises because the best BGP route at one router (chosen based on the IGP path costs and router IDs from its vantage point) affects the options available to its iBGP neighbors. The remainder of this section describes how the second and third phases of the algorithm arrive at the correct prediction without back tracking.

4.1 Computing the Set of Best eBGP Routes

The second phase of the algorithm begins with a complete set off eBGP routes (after medication by the import policies) and produces the set of candidate best eBGP routes; the best route that each router selects is an element from this set. Thus, the second phase of the algorithm must have the following properties:

- It must eliminate any eBGP-learned routes that could not be the best route at any router in the AS. We impose this constraint to simplify later stages of the algorithm. Impose three constraints and explain why these constraints are reasonable to assume in practice. If the eBGP-learned routes change frequently, the internal routing system does not have time to propagate the effects of one eBGP advertisement before the next one arrives. As such, we assume:

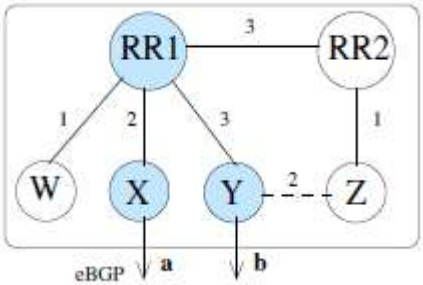
CONSTRAINT 1. The eBGP-learned routes change slowly with respect to the timescale of network engineering decisions. In practice, most BGP routes are stable for days or weeks at a time [10], and the vast majority of traf_c is associated with these table routes [11]. This allows emulation to operate on a static snapshot of the eBGP routes. Any eBGP routing change can be treated as a separate problem instance. Network operators have significant in deciding how to propagate the eBGP-learned routes throughout the AS. In the simplest case, the routers are configured in a full mesh of iBGP sessions. In general, though, the routers may form a more complicated signalling graph of iBGP sessions, as shown in Figure 2. Each edge corresponds to an iBGP session between a pair of routers. A router does not normally forward iBGP-learned routes to its other iBGP neighbors. However, a router can be configured as a route reflector(RR), which forwards routes learned from one of its route-reflector clients to its other clients. Following terminology from previous work [12], we use the term up for the iBGP session from a router to its RR, over for a conventional iBGP session between two routers, and down for the iBGP session from an RR to a client. A valid signalling path consists of zero or more up edges, followed by atmost one over edge, followed by zero or more down edges. Alternatively, routers within an AS can also be grouped into one or more. Confederations.. Because confederations are used much less frequently than route reflectors, we focus on modelling route reflectors and do not model the effects of confederations. Every eBGP-learned route should propagate through the signallinggraph to every other router in the AS.



Interaction between MED and router ID in the BGP decision process. Small numbers are router IDs

 At the completion of the second phase, each eBGP-speaking router must contribute *at most one candidate route*. This property must hold because the BGP protocol speci_cation requires each router to select and propagate a *single* best route. The second phase of the algorithm determines the best route to a destination at every eBGP-speaking router by starting with the complete set of eBGP-learned routes and *systematically eliminating routes from this set of candidate routes*. First, the algorithm eliminates every route from this set that would be eliminated based on the _rst three steps of the BGP decision process (i.e., local preference value, AS path length, and origin type). The algorithm effectively applies the _rst three steps of the BGP decision process *globally* across the set of all candidate eBGP routes. It might seem appealing either to continue applying steps of the BGP decision process globally or to apply the remaining steps of the decision process locally at each router (i.e., eliminate all routes from the set of candidate routes based on MED, router ID, etc.). However, the interaction of the MED and router ID attributes precludes having a consistent ranking of routes at each router. Figure 4 presents a simple example that illustrates why these two approaches are incorrect. *Global route elimination is not correct.* Consider the example in Figure 4 and assume that AS 3 learns eBGP routes that are equally good through step 3 in the BGP decision. Routes and flare learned from AS 1, and routes _and = are learned from AS 2. In a global comparison of the routes, andfl are first eliminated based on MED, and then router A picks route = (since = is preferred to _ based on the router ID comparison applied at router However, this conclusion is incorrect, because A would *always* prefer route _ over route =, since is learned via eBGP (step 5) and _ and = are equally good up through step 4 (recall that a router does not compare the MEDs of routes with different next-hop ASes).
Local route elimination is not correct. It might also seem natural to simply select a best route *locally* at each router and subsequently eliminate routes from this set by comparing routes within this set of locally best routes. This does not work either. Consider Figure 4 again. Applying the decision process locally at each router, route A selects route _ because its router ID is lower than that of _; similarly, router selects route =. This suggests that router A would ultimately select = as its best route, since = is better than due to MED comparison. However, this conclusion is also incorrect, because A will always prefer route _ over route =.To correctly handle the interaction between the MED and router ID attributes, the algorithm determines the effects of steps 4.8 of the BGP decision process by emulating the effects of a *particular* message ordering that correctly propagates the effects of MED on each router's best route without backtracking.

Computing the Best Route at Each Router
The third phase of the algorithm determines the best BGP route at each router, given the IGP path costs to each eBGP speaking router and the iBGP signaling graph. On the surface, this problem has a relatively simple solution: for each router, take the set of best eBGP-learned routes and select the closest egress point, breaking ties based on router ID. This approach would work if the eBGP speaking
Routersre advertised every eBGP-learned route to every router in the AS, as in a full-mesh iBGP configuration. However, the use of route reflectors (RRs) typically violates this assumption. The routing alternatives available to a particular router depends on its position in the route reflector hierarchy. For example, consider



Interaction between iBGP and IGP in the decision process, where small numbers are IGP path costs, solid lines are iBGP sessions, and the dashed line is an IGP path. The algorithm operates by designing two partial orderings

of the routers, based on the *up* edge and the *down* edges, respectively. This is possible because Constraint 3(b) requires that the signalling graph does not have any cycles of these edges. Visiting the routers in the *up* direction ensures that each router selects a *down* route, where possible, as required by Constraint 3(a). Visiting the remaining routers in the *down* direction ensures that each router has all of the *up* and *over* routes available when making a decision. Considering the routers in this particular ordering guarantees This theorem is a restatement of a theorem from earlier work [15] on sufficient conditions for stable BGP routing *at the AS level*. The previous result uses a constructive proof to show that a specific set of export policy and preference relations between ASes can guarantee that BGP will arrive at a stable path assignment: the proof shows the existence of a stable routing _rest by assigning paths to *up* AS neighbours, then to *over* and *down* neighbours. Other previous
work observed that the conditions for BGP stability at the AS level are the same as those for iBGP with route reflection within an AS [12]. Theorem 3 follows directly from applying the constructiveproof for stable global routing to iBGP with route rejection, and the proof is the same.

EMULATOR DESIGN

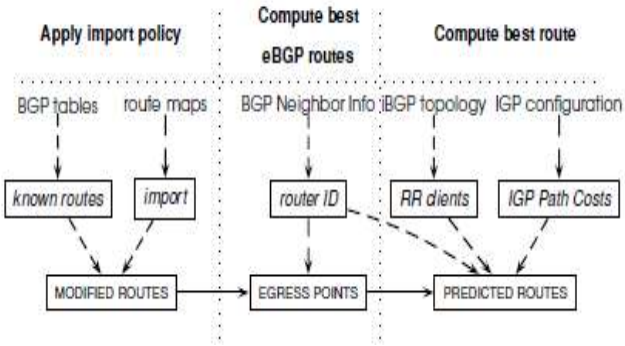
To demonstrate that the algorithm from Section 4 is accurate and practical, we have implemented a BGP emulator that computes the outcome of the BGP decision process for all routers in an AS. In this section, we discuss the input data that the emulator requires and
how network operators can obtain this data in practice. Then we describe the basic design of the emulator, deferring the discussion of performance optimizations to Section 6.3.

Input Data

As shown in Figure 8, the emulator uses three inputs:

- *BGP routing tables:* The BGP tables for the eBGP-speaking routers provide the first phase of the algorithm with a snapshot of the routes advertised by neighbouring ASes. We ignore the router's current view of the best route and the current setting of the local preference attribute, since these relate to the existing network con_guration rather than the scenarios we might want to emulate.
- *Router configuration files:* The configuration files are used to (1) determine the import policies (i.e., route maps) for each eBGP session, (2) determine the iBGPsignaling graph, and (3) compute the IGP path costs between each pair of routers. The import policies are used to manipulate attributes of the eBGP routes in the _rest phase of the algorithm, and the Ibgpand IGP information are needed for the third phase.
- *BGP neighbour information:* Because the BGP decision process depends on the router ID associated with the BGP session announcing the route, our algorithms require knowing the router ID associated with each BGP session. The second phase uses the router IDs of the *eBGP*sessions, while the third phase uses the router IDs for the *iBGP*sessions.

We emphasize several points with regard to the input data. First, a network operator can capture all of the necessary data with telnet orssh access to each router. Second, many aspects of the input data (e.g., the router ID data, routes for prefixes with stable traffic volumes, etc.) do not change very often; as such, the emulator is useful even if all of the input data is collected infrequently (e.g., once a day). Finally, because certain inputs can be approximated (e.g., router ID is typically the loopback IP address of the router) and often do not affect the decision process, the emulator can be effective even with limited input.



Data flow in the emulator. Fonts specify raw inputs, *input tables*, and DERIVED TABLES. In practice, operators might collect raw inputs once a day.

SECURITY

An AS is highly vulnerable to false information in BGP updates. By sending false information, an ISP can subvert a neighbour's routing goals, cause routers to overload and fail, or degrade service quality. False information can

have a significant influence on routing in an AS, even if the source of the information is several AS hops away [11]. Such information is sometimes generated by router bugs and misconfiguration. It could
also be maliciously generated by an IP's neighbour, who may be competing for customers and hence has a vested interest in making the ISP's customers dissatisfied with service.
Hence an ISP may wish to exercise *defensive programming* to protect itself against attacks.

Discarding invalid routes (by import filtering):
ISPs may wish to protect their customers from learning invalid routes by performing sanity checks to ensure update contents are valid before propagating them internally. For example, routes to special-use or private addresses, or address blocks that have not yet been allocated are obviously invalid [12]. Moreover, advertisements from customers for prefixes they do not own should not be propagated. ISPs can also perform certain sanity checks on the AS path; for example a Tier-1 ISP should not accept any routes from its customers that contain another Tier-1
ISP in the AS path. Also, advertisements containing private AS numbers in the AS path may be considered invalid. ISPs may configure its filters based on the contents of public repositories of routing configurations called *routing registries*, other public reports [13], or private disclosures from neighbours.

Protect integrity of routing policies (by rewriting attributes):
An ISP may want to prevent a neighbouring AS from having undue influence over its routing decisions, in violation of their peering agreement. Otherwise, the ISP could be duped
into carrying traffic a longer distance across its backbone on the neighbour's behalf. For example, suppose the ISP peers with a neighbour in both New York and San Francisco. By advertising a prefix with a MED of 0 in New York and a MED of 1 in San Francisco, the peer could trick the ISP into having all of its routers direct traffic for this destination through the New York peering point, even if the San Francisco peering point is closer. The peer could achieve the same goal by configuring its San Francisco router to advertise the route with the next hop attribute wrongly set to the IP address of the New York
router. To defend against violations of peering agreements, the ISP can configure the import policy to delete attributes or overwrite them with the expected values. For example, the import policy could set all MED values to 0, unless the ISP has agreed in advance to honour the neighbour's MEDs. Similarly, the import
policy could set the next-hop attribute to the IP address of the remote end of the BGP session, and remove any unexpected community values. Unfortunately, these techniques are
not sufficient to prevent all violations of peering agreements1.

Securing the network infrastructure (by export filtering):
An ISP may wish to prevent external entities from accessing certain internal resources by configuring its *export policies* that filter BGP advertisements for destinations that should not be externally reachable. For example, the ISP may protect its own backbone infrastructure by filtering the IP addresses used to number the router interfaces. The ISP may also wish to protect certain key internal services, by filtering the addresses of the hosts running network-management software. Finally, as a courtesy to its neighbours, an ISP may also do export filtering of invalid routes (e.g., routes with invalid addresses or contents),as a preventative measure.

Blocking denial-of-service attacks (by filtering and damping):
Denial-of-service attacks can degrade service by overloading the routers with extra BGP update messages or consuming excessive amounts of link bandwidth. For example, the ISP's routers could run out of memory if a neighbour sends route advertisements for a large number of destination prefixes. To protect itself, the ISP can configure each BGP session with a maximum acceptable number of prefixes, tearing down the session when the limit is exceeded; in addition, the import policycould filter prefixes with large mask lengths (e.g., longer than /24). As another example, a neighbor sending an excessive number of BGP update messages can easily deplete the CPU resources on the ISP's routers. Upon detecting the excessive BGP updates, the operators could modify the import policy to discard advertisements for the offending prefixes or disable the BGP session.

CONCLUSION

To perform everyday network engineering tasks effectively, efficiently, and with minimal unnecessary changes to the live network, operators need a way to model how a routing protocol configuration will behave before deploying that configuration. In this paper, we have presented a model that *accurately* and *ef_ciently*predicts the outcome of the BGP route selection process in a single AS using only a snapshot of the network configuration and the eBGP learned routes from neighbouring domains. The algorithm we have presented is the first that models the *outcome* of the BGP decision process across every router in an AS, without simulating protocol dynamics. We have implemented an emulator based on this model to demonstrate that our algorithm is accurate and efficient enough to be used in practice for many network engineering tasks.

Although BGP policies can be highly complex, there are a number of common design patterns that are typically used by ISPs. In this article we discussed several common patterns and how they can be realized using BGP policy mechanisms. We believe that by recognizing these patterns exist we can more efficiently develop tools that directly support them, such as analysis tools that check correctness, languages that preclude errors, or architectures that are designed for common cases.

REFFERENCES

1. Y. Rekhter, T. Li, "A border gateway protocol 4," IETF RFC 1771, March 1995.
2. E. Chen, T. Bates, "An application of the BGP community attribute," IETF RFC 1998, August 1996.
3. B. Quoitin, O. Bonaventure, "A survey of the utilization of the BGP community attribute," expired Internet Draft *draft-quoitinbgp-comm-survey-00.txt*, February 2002.
4. Y. Yang, H. Xie, H. Wang, A. Silberschatz, Y. Liu, L. Li, A.Krishnamurthy, "On route selection for interdomain traffic engineering," *IEEE Network Magazine, Special issue on InterdomainRouting*, Nov
5. N. Feamster, J. Winick, J. Rexford, "A model of BGP routing for network engineering," in *Proc. ACM SIGMETRICS*, June 2004.
6. J. Bartlett, "Optimizing multi-homed connections," in *Business Communications Review*, vol. 32, no. 1, pgs. 22-27, January 2002.
7. D. Chang, R. Govindan, J. Heidemann, "An empirical study of router response to large BGP routing table load" in *Proc. Internet Measurement Workshop*, November 2002.
8. S. Bellovin, R. Bush, T. Griffin, J. Rexford, "Slowing routing table growth by filtering based on address allocation policies," unpublished, June 2001, http://www.cs.princeton.edu/~jrex/papers/filter.pdf
9. T. Grif_n, F. B. Shepherd, and G. ilfong, .The stable pathsproblem and interdomain routing,.*IEEE/ACM Trans.Networking*, vol. 10, no. 1, pp. 232.243, 2002.
10. C. Labovitz, A. Ahuja, and F. Jahanian, .Experimental study of Internet stability and wide-area network failures,.in*Proc.Fault Tolerant Computing Symposium*, June 1999.
11. J. Rexford, J. Wang, Z. Xiao, and Y. hang, .BGP routingstability of popular destinations,. in*Proc. InternetMeasurement Workshop*, November 2002.
12. T. G. Grif_n and G. Wilfong, .On the correctness of IBGPcon_guration,.in*Proc. ACM SIGCOMM*, August 2002.
13. N. Feamster and J. Rexford, .Network-wide BGP route prediction for traf_cngineering,.in*Proc. Workshop onScalability and Traf_c Control in IP Networks, SPIE ITCOMConference*, August 2002.
14. A. Feldmann, A. Greenberg, C. Lund, N. Reingold,J. Rexford, and F. True, .Deriving traffic demands foroperational IP networks: Methodology and experience,.*IEEE/ACM Trans. Networking*, vol. 9, June 2001.
15. L. Gao and J. Rexford, .Stable Internet routing withoutglobal coordination,.*IEEE/ACM Trans. Networking*, vol. 9, pp. 681.692, December 2001.
16. T. Ye, H. T. Kaur, and S. Kalyanaraman, .A recursive random search algorithm for large-scale network parametercon_guration,.in*Proc. ACM SIGMETRICS*, June 2003.
17. T. G. Grif_n and G. Wilfong, .Analysis of the MEDoscillation problem in BGP,.in*Proc. InternationalConference on Network Protocols*, November 2002.
18. N. Feamster, .Practical veri_cation techniques for wide-arearouting,.in*2nd ACM Workshop on Hot Topics in Networks*, November 2000.