Network-Wide BGP Route Prediction for Traffic Engineering

Nick Feamster[a] and Jennifer Rexford[b]

[a] Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, MA, USA
[b] Internet and Networking Systems, AT&T Labs–Research, Florham Park, NJ, USA

ABSTRACT

The Internet consists of about 13,000 Autonomous Systems (AS's) that exchange routing information using the Border Gateway Protocol (BGP). The operators of each AS must have control over the flow of traffic through their network and between neighboring AS's. However, BGP is a complicated, policy-based protocol that does not include any direct support for traffic engineering. In previous work, we have demonstrated that network operators can adapt the flow of traffic in an efficient and predictable fashion through careful adjustments to the BGP policies running on their edge routers.[1] Nevertheless, many details of the BGP protocol and decision process make predicting the effects of these policy changes difficult. In this paper, we describe a tool that predicts traffic flow at network exit points based on the network topology, the import policy associated with each BGP session, and the routing advertisements received from neighboring AS's. We present a linear-time algorithm that computes a network-wide view of the best BGP routes for each destination prefix given a static snapshot of the network state, without simulating the complex details of BGP message passing. We describe how to construct this snapshot using the BGP routing tables and router configuration files available from operational routers. We verify the accuracy of our algorithm by applying our tool to routing and configuration data from AT&T's commercial IP network. Our route prediction techniques help support the operation of large IP backbone networks, where interdomain routing is an important aspect of traffic engineering.

Keywords: Traffic engineering, IP routing, BGP, router configuration

1. INTRODUCTION

Traffic engineering involves adapting the operation of a network according to the prevailing traffic conditions in order to improve performance and use resources efficiently. In practice, traffic engineering involves adjusting the resource allocation policies for path selection, buffer management, and link scheduling at the individual routers in the network. For example, if some traffic is experiencing high delay or packet loss due to a congested link, operators can adjust the configuration of the routing protocol to divert part of the traffic to other paths. Alternatively, an operator may be able to improve performance by reconfiguring the buffer management policy at the router; one approach might be to selectively mark or discard packets (e.g., by tuning the Random Early Detection (RED) parameters) to encourage some of the TCP senders to reduce their transmission rates before the buffer becomes full.[2] If the link carries multiple classes of traffic, the operator can also reconfigure the link-scheduling parameters to devote more bandwidth to some portion of the traffic.

Selecting the appropriate values for these parameters requires an accurate, up-to-date view of the offered traffic, network topology, and router configuration, which a well-designed network monitoring infrastructure can provide. Effective traffic engineering also depends on the ability to predict the outcome of possible changes to the router configuration. Evaluating "what-if" scenarios requires network management tools that simulate the network protocols and mechanisms[3, 4] or explicitly model their effects on the traffic.[5] In some cases, such as capturing the influence of RED parameters on TCP traffic over an entire network, simulation may be the only feasible alternative. In contrast, predicting the effects of routing changes does not require a complex simulation of the messages exchanged in the routing protocol. Nevertheless, deriving a closed-form analytic expression for the *optimal* parameter settings may prove difficult. Instead, we provide a way to explore many different parameter values to allow operators to select a good configuration that makes efficient use of network resources.

Most of the recent research and standards work on traffic engineering has focused on the Interior Gateway Protocols (IGPs), such as OSPF (Open Shortest Path First) or IS-IS (Intermediate System-Intermediate System), which control the selection of paths within a single Autonomous System (AS).[5–8] Because network operators manage *all* of the routers that participate in the IGP for a given network, they have complete control over intradomain routing. For example, network

operators can configure the link weights that control the selection of shortest paths in OSPF or IS-IS routing. However, most of the traffic carried by a large IP backbone network traverses multiple AS's, which makes *inter*domain routing an important aspect of traffic engineering. Additionally, the links between AS's are common points of congestion, largely because the control of these links is shared between two or more (sometimes competing) parties. Careful control over interdomain routing is important for improving end-to-end performance and making efficient use of network resources.

In this paper, we focus on traffic engineering in the context of the *existing* interdomain routing protocol—the Border Gateway Protocol (BGP).[9–11] Thus, our traffic engineering solutions do not require any modifications to the existing IP infrastructure. However, BGP is a complex, policy-based protocol with a large number of configuration options. Because changes to BGP routing policies can affect routing stability and the flow of traffic in the Internet as a whole, network operators should understand the potential impact of changes in routing policy *before* reconfiguring the operational routers. In this paper, we describe how to predict the influence of configuration changes, based on a snapshot of the state of the network. This allows a network operator to evaluate possible changes to BGP policies and compare their impact on the flow of traffic. Specifically, we present three main contributions:

- **Network-wide model:** We propose a model of the network state required to predict the influence of changes in BGP policies on path selection. The model incorporates the BGP routes advertised by neighboring domains and the BGP import policies configured by network operators. The model specifies the inputs to existing tools that capture the influence of the IGP configuration.[5]
- **Route prediction algorithm:** We present a linear-time, centralized algorithm that computes the best BGP routes chosen by the various routers in the AS based on the routing policies and BGP advertisements. We show that such an algorithm can predict routes without simulating the passing of BGP messages between routers. Additionally, we prove that our algorithm accurately represents the BGP decision process implemented on IP routers.
- **Prototype implementation:** We describe how we populated our network model using the data available from the routers in AT&T's commercial IP network. We describe a prototype implementation of our tool that accurately predicts the effects of BGP import policy changes on path selection.

We present these topics in three separate sections after a brief background section that describes the BGP protocol and decision process. The paper concludes with a summary of our approach and a discussion of future research directions.

2. BORDER GATEWAY PROTOCOL

In this section, we first present an overview of the Border Gateway Protocol (BGP) and the attributes associated with BGP advertisements. Next, we describe the BGP decision process, which governs the selection of the best route for each destination prefix at each router. Finally, we briefly explain how a router constructs a forwarding table based on its best BGP route and the IGP parameters.

2.1. BGP Protocol

Internet routing and forwarding operate at the level of *prefixes*, which represent blocks of contiguous IP addresses. A prefix is represented by a 32-bit address and a mask length. For example, $192.0.2.0/24$ specifies 256 addresses ranging from $192.0.2.0$ to $192.0.2.255$. Neighboring AS's exchange routing information by configuring a BGP session between a pair of edge routers. The two routers establish a session and exchange update messages as they acquire new information about how to reach individual destination prefixes. For a given prefix, a router sends an *advertisement* to inform its neighbor of a new route to the destination prefix or a *withdrawal* to indicate that the route to that prefix is no longer available. Each advertisement includes an *AS path* that identifies the list of AS's en route to the origin AS that announced the destination prefix; for this reason, BGP is called a path-vector protocol. Before accepting an advertisement, the receiving router discards any routes that contain its own AS number in the AS path to prevent the formation of routing loops.

Route advertisements include several other attributes. The *next hop* attribute indicates the IP address of the router associated with next hop along the path to the destination. The *origin type* identifies how the origin AS learned about the route—within the AS (e.g., static configuration), EGP (a now-defunct distance-vector protocol), or injection from another routing protocol. A neighbor AS may include a *multiple exit discriminator* (MED) in the route advertisement to

encourage the recipient to select a particular exit point for sending traffic to the neighboring AS; typically, this is done by advertising different MED values at different interconnection points between the two AS's. An internal BGP (iBGP) message may include a *local preference* attribute to aid the recipient in ranking the paths learned from different routers in the AS. The *community* attribute provides a generic mechanism for tagging routes to aid in specifying and applying routing policies. For example, an AS might assign different community values to a path depending on whether it was learned from a customer or a peer.

BGP routing depends heavily on locally-configured policies. A BGP-speaking router may receive multiple routes for the same destination prefix. Upon receiving a route advertisement, the router applies *import policies* to filter unwanted routes (e.g., advertisements for routes to prefixes in the private address space and other so-called "martian" addresses) or to alter the attributes associated with the route. Network operators configure import policies to influence path selection. Ultimately, the router invokes a *decision process* to select exactly one "best" route for each destination prefix among all the routes it hears. The router uses *export policies* to manipulate the attributes of its best route and determine whether to advertise this route to neighboring AS's. Network operators often use export policies to limit the distribution of routes to certain neighboring AS's, based on the commercial relationship between the two institutions. For example, routes learned from a peer or upstream provider should not be readvertised to another peer or upstream provider.[12,13] Network operators specify import and export policies using diverse set of configuration commands.

A large backbone network typically has multiple BGP-speaking routers, multiple BGP sessions with each neighbor AS, and BGP sessions with several different neighboring AS's. For example, two large AS's that exchange routing information might have have BGP sessions with each other at multiple geographic locations, such as the East and West Coasts of the United States. In addition to exchanging BGP messages with neighboring domains, an AS may use iBGP to distribute routing information among its routers. The simplest approach is to have an iBGP session between each pair of routers (i.e., a full iBGP mesh), but most large networks have a hierarchical configuration using route reflectors or confederations to achieve better scalability.[10] An iBGP session operates in the same fashion as an external BGP (eBGP) session, with the exception that routes learned from one iBGP neighbor are not advertised to another iBGP neighbor. Every router must select a *single* best route for each destination prefix among the advertisements from the various eBGP and iBGP neighbors. Because the best route that a router selects is dependent on its location in the network, each router will not necessarily select the same best route.

2.2. BGP Decision Process

A BGP-speaking router may learn multiple paths to the same destination prefix from eBGP and iBGP neighbors. Although the selection of a best path depends on the attributes in the BGP advertisements, the complete details of the decision process are not part of the protocol specification. Nevertheless, router vendors adhere to a *de facto* standard.[14–16] After certain routes are removed from consideration (e.g., because they have a loop in the AS path, have an unreachable next hop, or were filtered by the import policy), the router applies a sequence of steps to narrow the set of candidate routes to a single choice, as follows:

1. *Highest local preference*: Prefer routes with the highest local preference, where local preference is assigned by the import policy and is conveyed via iBGP.
2. *Shortest AS path*: Prefer routes with the shortest AS path length, as conveyed in the BGP advertisement.
3. *Lowest origin type:* Prefer routes with the lowest origin type (IGP is preferable to EGP which is preferable to INCOMPLETE), as conveyed in the BGP advertisement or reset by the import policy.
4. *Lowest MED:* For routes with the same next-hop AS*, prefer routes with the smallest MED value, as conveyed in the BGP advertisement or reset by the import policy.
5. *eBGP over iBGP*: Prefer routes learned via eBGP over routes learned via iBGP, since leaving the AS directly is preferable to forwarding traffic through the AS to another router.

*If the router is configured with the `always-compare-med` directive, the MED value is compared across all advertised routes.

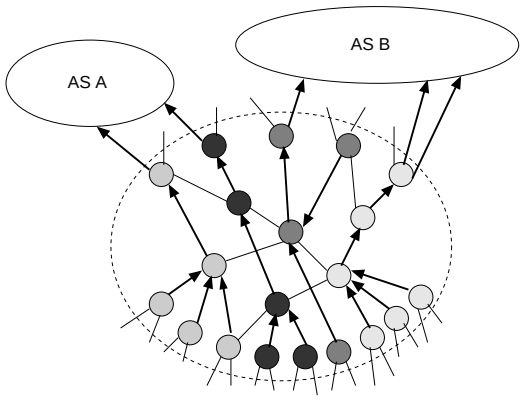

Figure 1. Flow of traffic from ingress routers to the egress links. Each node represents a router within the AS. Routers with the same shading have the same closest egress point.

6. *Lowest IGP metric*: Prefer routes with the smallest intradomain (Interior Gateway Protocol) metric to reach the next hop, since this enables each router to select its "closest" exit point.
7. *Oldest route*: Prefer the route that was received earliest, since this route is more likely to be stable.
8. *Lowest router ID*: Prefer the route learned from a router with the lowest router identifier, as conveyed during establishment of the BGP session.

Router vendors typically provide configuration options to disable one or more of these steps. In our work, we assume that step 7 is disabled to ensure that the BGP decision process does not depend on the order or timing of the update messages.[1] The network operator's configuration of the import policies affects the decision process in several important ways: filtering of unwanted routes, assignment of local preference, and possible resetting of the origin and MED attributes.

Over time, each router receives eBGP messages from neighboring domains, as well as iBGP advertisements for the best routes seen at other routers in the AS. In the meantime, the routers also participate in an IGP that affects the selection of the best path, as well as the path through the domain to reach the BGP next hop. Figure 1 shows a collection of routers that select different routes toward a destination prefix reachable via AS's A and B. Each router selects a route with the "closest" egress point, based on the IGP configuration (in step 6 of the decision process). Each router forwards packets based a combination of information from BGP and the IGP. The forwarding table determines how the router directs an incoming packet to the appropriate outgoing link(s). For example, consider a router that would forward traffic for destination prefix 192.0.2.0/24 to the outgoing link Serial2. The router employs the BGP decision process to select an AS path and next-hop IP address of the border router. The router learns how to reach this next-hop address via the intradomain routing protocol (e.g., OSPF or IS-IS). Based on the IGP weights, the router computes a shortest path to the BGP next hop and identifies the outgoing link, Serial2, along the shortest path. The router combines these two pieces of information to construct the forwarding table entry. When a packet arrives, the router performs a longest-prefix match on the destination address (say, 192.0.2.147) to determine the appropriate outgoing link. The next router repeats the process and directs the traffic to the next step toward the destination.

3. NETWORK-WIDE MODEL

This section presents a model that describes the influence of BGP import policy changes on the flow of traffic based on a static snapshot of the network state. We divide the problem into four modules, as shown in Figure 2: (1) each eBGP session applies import policies to the routes learned from neighboring domains; (2) the BGP decision process determines the set of best routes to each destination prefix; (3) for each ingress router, the IGP configuration and the underlying network topology dictate the selection of the closest egress point and the path through the domain; and (4) the offered traffic is joined with the paths to compute the total load on each link in the domain. The rest of the paper focuses primarily on the first two modules in Figure 2; previous work describes the last two modules.[5]

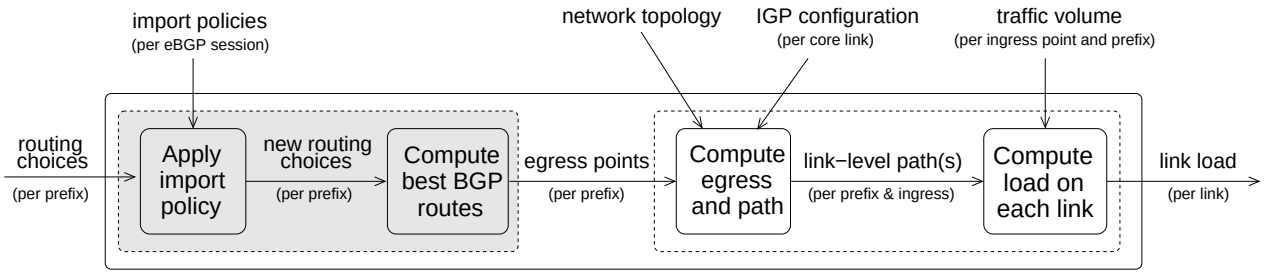

Figure 2. Modeling the impact of BGP policies, IGP weights, and network topology on the flow of traffic. In this paper, we focus on the modules inside the shaded box.

3.1. Routing Choices

The modules inside the shaded box focus on the aspects of BGP routing that do not depend on the IGP configuration or the network topology. The first module considers the routes learned via eBGP sessions with neighboring domains. Two neighboring AS's exchange messages over an eBGP session between two routers. A border router may often have eBGP sessions with many different neighboring AS's, or even multiple eBGP sessions with the same AS. A neighbor AS sends route advertisements over an eBGP session to advertise reachability to a given prefix. The set of all eBGP-learned routes for a given prefix constitutes the *routing choices*, shown as one of the inputs to the first module in Figure 2. Each advertised route has four attributes—the AS path, the origin type, the MED, and the router ID. The router ID is actually associated with the eBGP session; as such, all routes learned via the same eBGP session have the same router ID. For simplicity, we incorporate the router ID as an attribute of the individual routes, since the router ID plays a role in the BGP decision process.

3.2. Import Policy

Each eBGP session has an import policy that applies to all route advertisements heard on that session. Some aspects of import policy, such as route filtering, do not relate directly to traffic engineering. Import policies allow network operators to reassign some route attributes based on a regular expression match on the AS path associated with each route advertisement, or by the advertised prefix itself. Assigning different local preference values to different routes is a convenient way to control the flow of traffic, since the first step of the BGP decision process compares the local preference values of routing advertisements. We model the import policy associated with each eBGP session as a list of mappings that identify which routes should receive a particular value of local preference, origin type, or MED. Each mapping refers to routing advertisements based on the destination prefix or a regular expression on the AS path. For example, the import policy *at one router for a given eBGP neighbor* might be expressed by the following mappings:

$\{192.0.2.0/24, 10.0.0.0/8\} \rightarrow$ local-pref 80
$\{\hat{}65000\$\} \rightarrow$ local-pref 110
$\{*\} \rightarrow$ local-pref 100

This import policy assigns a local preference of 80 to any route for prefixes $192.0.2.0/24$ and $10.0.0.0/8$. Any remaining route with a one-hop AS path of 65000 is assigned a local preference of 110. Any route that does not match these two rules is assigned a local preference value of 100. All routes retain their initial values for origin type and MED.

3.3. New Routing Choices

The first module in Figure 2 captures how the import policy manipulates the routing choices learned from neighboring domains. This produces new routing choices that are a subset of the original routing choices and can be represented in the same fashion. A route in the set of new routing choices includes the local preference attribute and may have new values for the origin type and/or MED. The new routing choices are an input to the second module, which emulates the operation of the BGP decision process at the various routers in the network. This module captures the effects of distributing the eBGP-learned routes to the various routers in the domain via iBGP, without considering the influence of the IGP parameters on the BGP decision process. We discuss this module in more detail in Section 4.

3.4. Egress Points

The output of the second module is the set of egress points associated with each prefix. Each egress point in the set corresponds to the eBGP session responsible for advertising this route. The routers that select this egress point direct their traffic toward the edge router associated with this session. In reality, each packet that uses this eBGP-learned route eventually traverses some egress *link* from the edge router to the router in the neighboring domain. More generally, each eBGP session is associated with one or more egress links at the router. For example, a router may have parallel links connecting to a router in the neighboring domain, as shown by the rightmost router connecting to AS B in Figure 1. In addition, a single egress link may be associated with more than one eBGP session at the router. For example, the router may have a link that connects to a shared medium, such as a FDDI ring or ATM switch, at a public Internet exchange point (IXP), where multiple AS's meet to exchange BGP routes and IP traffic. One of the functions of the third operation in Figure 2 is to associate each egress point (an eBGP session) with a group of egress links (associated with that session), based on the network topology.

3.5. IGP Configuration

The third module in Figure 2 computes the shortest path(s) between each pair of routers in the domain, based on the topology and the settings of the IGP weights and areas. IGP parameters affect both the BGP decision process (in step 6) and the path(s) between each pair of routers in the AS. In the event where multiple route advertisements are equally good through the first five steps of the BGP decision process, the router will select a best route based on which has the shortest IGP cost. Therefore, the third module requires knowledge about IGP costs of internal links. When modeling the selection of the closest egress point, we assume a full-mesh iBGP configuration, where every router receives a copy of the best route from each of its iBGP neighbors. In practice, the use of route reflectors or confederations may limit route advertisement distribution, but we believe our model can be extended to support these configurations. The IGP parameters also determine the shortest path(s) through the network from the ingress router to its closest egress point. For example, the OSPF and IS-IS protocols assign an integer weight to each unidirectional internal link and compute the shortest path route(s) as the sum of the link weights.

3.6. Traffic Volumes

The shortest-path computation determines how traffic that enters at a particular ingress point traverses a path through the domain to a certain egress point en route to the destination prefix. Combining this path information with traffic measurements from the ingress points[17] provides an estimate of the total load on each link in the AS.[5] The last module sums traffic volumes over each of the links. These estimates of link load can be used to evaluate and compare the influence of different configurations of the BGP import policies and IGP parameters on the flow of traffic. The network operator may have an objective function that quantifies the "goodness" of a particular solution. For example, the objective function might reflect the utilization of the most heavily loaded link. An operator or network optimization tool could then experiment with various BGP policies and IGP weights to find a good solution based on this objective function.

4. NETWORK-WIDE ROUTING PREDICTION

In this section, we describe an algorithm for predicting the set of best routes based on a network-wide view of the routing choices, after manipulation by the import policies. First, we present a linear-time algorithm that computes the best routes without simulating the complex exchange of update messages on eBGP and iBGP sessions. For simplicity, we initially focus on the case where the MED attribute is compared across all routes, irrespective of the next-hop AS. Next, we explain how limiting the comparison of the MED attribute to the routes with the same next-hop AS makes route prediction more complicated. Then, we present an extension to our algorithm that captures the influence of MEDs on the selection of best routes. We defer several of the theorems and proofs to the Appendix.

4.1. Centralized Algorithm for Route Prediction

Each router selects its best route based on the BGP advertisements received from its eBGP and iBGP neighbors. A change in the best route may trigger a new update message to other routers in the domain which, in turn, may affect their routing decisions. In this section, we show that a deterministic, centralized algorithm can compute the set of best routes selected by the routers throughout the AS once the internal distribution of routes converges. Assume we have an AS with a set

of n edge routers, $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$, where each router r_i has a set of (eBGP-learned) routes $\mathcal{A}_i$ for the destination prefix d, after the routes have been manipulated by the import policies. We focus on the routing decision for a single d since the path-selection process for each prefix proceeds independently. Our goal is compute the set $\mathcal{B}$ of best routes from the sets $\{\mathcal{A}_i\}$ in a deterministic fashion. First, we show that it is possible to compute $\mathcal{B}$ based only on the sets $\{\mathcal{A}_i\}$. Next, we present an algorithm that first determines the locally-best (eBGP-learned) route $f_i \in \mathcal{A}_i$ at each router r_i and then computes $\mathcal{B}$ by identifying the best of the $\{f_i\}$ across all of the n edge routers. The running time of our algorithm is linear in the total number of eBGP-learned routes.

Since BGP is a message-passing protocol that sends incremental routing updates, we must first establish that the set of best routes $\mathcal{C}$ as determined by the BGP decision process is independent of the ordering of arrivals of the eBGP and iBGP update messages in the network.

THEOREM 4.1. *Given the updates received for a destination prefix up to time t for all edge routers, the best routes as determined by the BGP decision process, $\mathcal{C}$, are independent of the order of the update messages.*

Proof. At each router, the BGP decision process ranks routes based on the attributes for each advertised route. The ranking of routes in the decision process is independent of arrival times and order—*the best route to a prefix will only change if the newly received route is ranked higher than the current best route*. Thus, for any two advertisements, the BGP decision process will rank those two advertisements in the same fashion, regardless of which order they arrived. Each edge router determines the best BGP route locally. If the locally-best route was learned via eBGP, the router readvertises this route to its iBGP neighbors, including the other edge routers. Thus, as well as hearing eBGP advertisements, an egress router hears additional best route advertisements from its iBGP neighbors. However, by the same argument, iBGP advertisements can be interleaved anywhere in the arrival of eBGP advertisements without affecting the best route, since the ranking that the BGP decision process applies is independent of the order in which these messages were received. □

An important consequence of the theorem is that any ordering of the eBGP and iBGP messages would produce the same final selection of the best routes. Our algorithm models a simple ordering that consists of two steps: (i) each router r_i receives and processes all of its eBGP-learned routes and selects the best of these routes f_i using the BGP decision process, and (ii) all routers receive the routes $\{f_i\}$ via iBGP and repeat the BGP decision process to compute their final best route. It is important to note that the route f_i is not necessarily in $\mathcal{B}$ because some routes learned from other routers may be better. That is, some routers may change their best route in the second step. These routers do *not* need to send their new best route to others, since iBGP-learned routes are not exchanged with other iBGP neighbors. Hence, the message passing terminates after the second step under our message ordering. In the end, some routers r_i have a best route $f_i \in \mathcal{B}$ and other routers must select their best route from $\mathcal{B}$ based on the later steps in the decision process (i.e., the IGP metric and the router ID), as captured in the third module in Figure 2.

The first step of the algorithm identifies the route f_i for each edge router r_i. When the MED attribute can be directly compared across all routes, computing f_i for each router r_i is relatively simple. For computing f_i, the local preference, AS path, origin type, MED, and router ID attributes form an ordering on the set of (eBGP-learned) routes $\mathcal{A}_i$ at router r_i. The parts of the BGP decision process based on the "eBGP vs. iBGP" (step 5) and the "lowest IGP metric" (step 6) are not relevant here, since all of the routes in $\mathcal{A}_i$ were learned directly via eBGP at router r_i. Computing f_i involves iterating through the routes in $\mathcal{A}_i$ and comparing each route to the current best route. More formally, for two routes $p, q \in \mathcal{A}_i$, where $p \neq q$, the algorithm eliminates advertisement p if:

$$\begin{aligned}
\text{LOCALPREF}(p) &< \text{LOCALPREF}(q) \quad \text{or} && (1)\\
\text{ASPATHLENGTH}(p) &> \text{ASPATHLENGTH}(q) \quad \text{or} && (2)\\
\text{ORIGIN}(p) &< \text{ORIGIN}(q) \quad \text{or} && (3)\\
\text{MED}(p) &> \text{MED}(q) \quad \text{or} && (4)\\
\text{ROUTER ID}(p) &> \text{ROUTER ID}(q) && (5)
\end{aligned}$$

The second stage of the algorithm uses the locally-best routes $\{f_i\}$ to compute $\mathcal{B}$. This involves iterating through the routes $\{f_i\}$ and comparing each route to the current set of best routes based on the attributes in Equations 1–4; note that

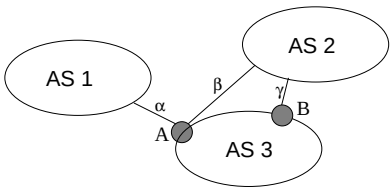

Figure 3. Multi-exit discriminators prevent an edge router from achieving an ordering of routes based only on locally-learned routes.

this does not include the router ID tie-breaking step in Equation 5. If a route p has a better ranking than the existing best routes, then the set of best routes is reset to contain the single route p. If the p has the same ranking, then p joins the existing set of best routes. Otherwise, p is discarded.

Together, the two parts of the algorithm have a running time that is linear in the total number of eBGP-learned routes for the destination prefix. In Appendix A, we present a series of proofs that demonstrate the correctness of our algorithm. In Theorem A.1, we show that eliminating routes from $\mathcal{A}_i$ based on Equations 1–5 for each router r_i does not eliminate any routes that appear in $\mathcal{B}$, as determined by the BGP decision process. Theorem A.2 shows that our algorithm does not eliminate any routes that appear in $\mathcal{B}$ when it eliminates routes $\{f_i\}$ based on route comparison across all edge routers. Finally, Theorem A.3 shows that every route advertisement that is eliminated from $\mathcal{C}$ will also be eliminated from $\mathcal{B}$ by our prediction algorithm. In proving these three theorems, we show that the sets $\mathcal{B}$ and $\mathcal{C}$ are equal, which proves that our prediction algorithm produces the same set of routes that would be produced by the separate application of the BGP decision process at each router based on the arrival of eBGP and iBGP update messages.

4.2. The Trouble with MEDs

The results in the previous subsection apply to the case where the BGP decision process compares all route attributes across all advertised routes. Network operators commonly configure routers to treat MEDs in this way to provide extra control over which routes are chosen by the BGP decision process. For example, operators sometimes reassign MED values using BGP import policies, because MED comparison can be used as a way to prefer one route over another *after* the AS path length comparison step. In other cases, the network operator may configure import policies that reset the MED attribute (e.g., using the `set metric 0` directive) on all eBGP sessions to prevent neighboring domains from influencing the BGP selection process. In these two cases, all attributes can still be compared across all route advertisements. However, in some cases, network operators limit the MED comparison to routes with the same next-hop AS. Operators commonly use this functionality to achieve certain traffic engineering objectives, such as "cold-potato" routing, whereby the neighboring AS uses MEDs to signal which egress point should be used to carry the traffic.

Limiting MED comparison to routes with the same next-hop AS makes the ranking of routes *non-transitive*. A lower-ranked route at router A may be a *better* route than the best route at router B, which may, in turn, be better than the locally-best route at router A. Consider again the example in Figure 3, where AS3 learns routes for a prefix from both AS1 and AS2. AS3 receives the route advertisement α from AS1 for some destination prefix at router A and the routes β and γ from AS2 at two different edge routers A and B. Assume that the three routes have the same local preference, AS path length, and origin type. Suppose that router A prefers route β because it has a smaller router ID than α; since the routes were learned from different next-hop AS's, MEDs do not play a role in the decision. Therefore, locally at A, β is the best eBGP-learned route; locally at B, γ is the best (and only) eBGP-learned route. However, suppose that γ has a smaller MED value than β; then, router A would prefer γ over β. Yet, A would prefer α over γ since α is an eBGP-learned route and γ is an iBGP-learned route. In summary, A prefers β over α (due to router ID), γ over β (due to MED), and α over γ (due to "eBGP vs. iBGP").

Ultimately, the selection of the best route at router A depends on the *order* of the comparisons between the routes. This dependency makes the outcome of the BGP decision process dependent on the order in which these messages are received. To avoid this problem, router vendors recommend enabling the `bgp deterministic-med` feature.[18] This forces a router to repeat the comparison of *all* routes after receiving a new advertisement or withdrawal message, rather than simply comparing against the current best route. This removes the dependency on the order of message arrivals.

However, the non-transitivity of MED comparisons still causes problems for the algorithm we present in Section 4.1. Given a local ranking of the eBGP-learned routes $\mathcal{A}_i$ at each router r_i, we can no longer guarantee that $\mathcal{B}$ will be some subset of the best routes $\{f_i\}$ selected locally at the various edge routers (i.e., Theorem A.1 no longer holds). In the next subsection, we discuss revisions to the algorithm from Section 4.1 such that network-wide route prediction still produces the same set of best routes as the separate application of the BGP decision process at each router to the sequence of BGP update messages.

4.3. Algorithm Revisions

Because the use of MEDs makes route comparison non-transitive, the algorithm must take care not to eliminate any routes that could potentially become the best route after the MED comparison step. The algorithm should account for the fact that local elimination of routes from $\mathcal{A}_i$ should not go past the MED step in the decision process. At this point, the algorithm must determine whether the best route at each router is better than the best route at some other edge router. The best route that remains at a particular router is the one that would would be considered to be the locally-best route. From a high-level, our algorithm for selecting the best route to a prefix at each egress router proceeds in three steps:

1. *Eliminate routing choices locally at each router based on steps up to and including MED (Equations 1–4)*: Construct an initial set of candidates for best routes to a destination prefix at each router based on the highest local preference value heard globally for this route. From this set, eliminate those that do not have the shortest AS path length among all routes in this set, and so forth. At the end, each router r_i has one or more locally-best routes that differ only in their router ID attributes.
2. *Eliminate routes that are always worse than the best routes at other routers*: For each r_i, compare the locally-best route f_i to the locally-best route at other routers. If f_i is worse than one or more of these routes with regard to Equations 1– 3, eliminate all of the routes at router r_i. At the end of this step, all routes that cannot compete with regard to local preference, AS path length, and origin type have been eliminated.
3. *Select a local best route at one edge router and propagate the effects of this choice globally across all edge routers*: While there are still routers with one or more candidate best routes, pick one of these routers r_i and select the best eBGP-learned route f_i at that router (based on the router ID tie break). If this route is "worse" than the locally-best route at one or more other routers, based on the MED comparison (Equation 4), eliminate this route. Otherwise, assign this route to $\mathcal{B}$ and eliminate all other routes at r_i as well as all other routes that have the same next-hop AS that do not have the same MED value; this router r_i does not require further inspection.

The first two steps eliminate routes that could not possibly belong to the set of best routes $\mathcal{B}$. The third step propagates the effects of the other routes one step at a time. At the point where MED comparison is applied, we must determine if there are other potential best routes at other egress routers that are "better" than the locally-best route. Consider the example in Figure 3. The first two steps of the algorithm have already been applied, leaving three routes α, β, and γ. We select router A and its locally-best route β, which is ranked ahead of α based on the router ID. However, when we compare to the locally-best route at router B, we find that γ is a better route; thus, we eliminate β from consideration. Now there are two routers each with one route. We select router A again and propagate the effects of the route α. No other route has the same next-hop AS, so this route does not eliminate any other routes and is not eliminated itself; thus, we include α in $\mathcal{B}$. Then, we select router B and add γ to $\mathcal{B}$. The algorithm terminates with $\mathcal{B} = \{\alpha, \gamma\}$.

Essentially, we have altered the algorithm from Section 4.1 to eliminate routes based on Equations 1–4, rather than on Equations 1–5. Therefore, Theorem A.1 still applies, since any route advertisement q that is better than an advertisement p based on Equations 1–3 would also be eliminated by the BGP decision process. In Appendix A.2, we show that, in the case of the revised algorithm, routes that are eliminated by global comparison of routes must not be in the set of best routes. This is more complicated because we must show that our algorithm never eliminates a route *prematurely*—that is, our algorithm never eliminates a route that could eventually be selected as a globally best route.

5. PROTOTYPE IMPLEMENTATION

In this section, we describe the prototype implementation of a tool that applies import policies to the eBGP-learned routes and computes the set of best routes for each destination prefix. We describe how to obtain import policies and routing choices from a Cisco router and express this data in terms of the model we presented in Section 3. We also describe how we applied this data to verify the correctness of the algorithm we presented in Section 4.

```
   Network          Next Hop            Metric LocPrf Weight Path
*  10.0.0.0/8       192.0.2.10            2130     80      0 65000 183 i
*>i                 192.205.32.162        2130    100      0 65000 183 i
*> 10.23.0.0/16     192.0.2.10               0    110      0 65000 i
* i                 192.205.32.162           0    110      0 65000 i
```

Figure 4. Excerpt from a BGP routing table dump (i.e., output of `show ip bgp`).

5.1. Routing Choices

We extract the routing choices from the BGP routing table, also known as the Routing Information Base (RIB), from the routers that connect the AT&T network to other large providers. A simple script connects to each router and issues a command to dump the RIB (e.g., `show ip bgp` in Cisco IOS[18]). Figure 4 shows example output with two routes for 10.0.0.0/8 and two routes for 10.23.0.0/16. Each entry shows attributes that are learned via a BGP advertisement or assigned by the existing import policy, such as the next-hop IP address (192.0.2.10), MED (2130), local preference (80), AS path (65000 183), and origin type ("i" for IGP); the weight parameter is a Cisco-proprietary attribute. The "i" near the beginning of an entry indicates that the route was learned via iBGP from another router in the AT&T network. The ">" symbol identifies the router's "best" route for this prefix. For example, this router prefers the second route to 10.0.0.0/8 because it has a larger local preference (100 vs. 80). The router favors the eBGP-learned route to 10.23.0.0/16 since the local preference, AS path length, origin type, and MED were the same for both routes.

Using routing table dumps to reconstruct the routing choices for each prefix presents several limitations.[1] The RIB records the routes *after* the application of the current import policy. However, the import policy may have filtered some routing advertisements. Since we do not try to model changes in the filtering policy, this is not a significant limitation. The import policy may also have manipulated the BGP attributes (i.e., local preference, origin type, and MED) of the remaining routes. Thus, we cannot necessarily determine the values of these attributes that accompanied the corresponding BGP advertisement. Nevertheless, because import policies often reassign these attributes for traffic engineering purposes, we can use the data available from the routing tables to evaluate new import policies that assign new local preference values. Similarly, we can evaluate policies that either reset the origin type or MED (based on the prefix or AS path) or retain the existing values. In constructing the routing choices for each prefix, we focus on the routes learned via eBGP; each of the iBGP-learned routes exists as an eBGP-learned route in the table of another router. For each eBGP-learned route, we extract the prefix, AS Path, origin type, and MED attributes.

In order to accurately predict the outcome of the BGP decision process, our algorithm must know the router ID associated with each route. The neighbor's router ID, a 32-bit unsigned integer with a default value that depends on the implementation of the neighbor's router, is transmitted in the OPEN message that initiates the establishment of the BGP session. The router ID could be the router's loopback address or the highest IP address across all of the interfaces on the router. Alternatively, the operator in the neighboring AS might explicitly configure an arbitrary router ID (e.g., using the `bgp router-id` command,[18] in Cisco IOS parlance). Each BGP session is associated with exactly one router ID.

For Cisco routers, the `show ip bgp neighbors` command describes which router ID is associated with each session[18]; this command also provides a variety of information about every BGP session at that router. Figure 5 shows an excerpt of the output. In this example, the `BGP neighbor` is 192.0.2.10 and the `remote router ID` is 71.169.232.8. In the AT&T network, a script archives the output from `show ip bgp neighbors` on a daily basis. For every eBGP session, we obtain the corresponding router ID and include it as an attribute for every route associated with that eBGP session.

5.2. Import Policies

Router vendors offer a wide variety of configuration commands for specifying import policies. The commands applied to a router are preserved in a configuration file that can be archived for backup and analysis (e.g., for Cisco routers, the `show running-config` command outputs this file). A Cisco IOS configuration file is divided into a number of sections that capture the configuration state of various aspects of the router, including the interfaces and the routing protocol. Figure 6 shows an example of a BGP session configuration for a router in AS 7018. The router has a BGP session with IP address 192.0.2.10 in AS 65000. The second `neighbor` statement specifies that the inbound route map called `IMPORT` should

BGP neighbor is 192.0.2.10, remote AS 10, external link
Index 1, Offset 0, Mask 0x2
Inbound soft reconfiguration allowed
BGP version 4, **remote router ID 71.169.232.8**
BGP state = Established, table version = 27, up for 00:06:12
Last read 00:00:12, hold time is 180, keepalive interval is 60 seconds
Minimum time between advertisement runs is 30 seconds
Received 19 messages, 0 notifications, 0 in queue
Sent 17 messages, 0 notifications, 0 in queue
Inbound path policy configured
Route map for incoming advertisements is testing
Connections established 2; dropped 1
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
...

Figure 5. Excerpt from `show ip bgp neighbors` output.

```
router bgp 7018
     neighbor 192.0.2.10 remote-as 65000
     neighbor 192.0.2.10 route-map IMPORT in
!
route-map IMPORT permit 1
     match ip address 199
     set local-preference 80
!
route-map IMPORT permit 2
     match as-path 99
     set local-preference 110
!
ip as-path access-list 99 permit ^65000$
access-list 199 permit ip host 192.0.2.0 host 255.255.255.0
access-list 199 permit ip host 10.0.0.0 host 255.0.0.0
```

Figure 6. Example of a Cisco IOS import policy.

be applied to all advertisements heard on this BGP session. This route map has two clauses that implement the import policy outlined in Section 3.2. The first clause assigns a local preference of 80 for advertised routes to 192.0.2.0/24 and 10.0.0.0/8, as defined in access-list 199. The second clause assigns a local preference of 110 to routes with an AS path of 65000 (i.e., a one-hop path to AS 65000). All remaining routes are assigned the default local preference value, 100.

Our parsing mechanism looks for `neighbor` statements to determine which route map is associated with the BGP session. Each route map consists of one of more clauses that `permit` or `deny` certain route advertisements. Our algorithm ignores the `deny` clauses since these correspond to filtering operations and represents each `permit` clause as a mapping from either a list of prefixes or AS path regular expression to an attribute assignment (as described in Section 3.2). The `match` statement indicates the routes for which a particular mapping is applicable (i.e., the list of prefixes or AS path regular expression), and the `set` statement specifies the attribute assignment that the import policy applies to those routes. For example, the first clause of the `IMPORT` route map in Figure 6 is represented as:

$$\{192.0.2.0/24,\ 10.0.0.0/8\} \rightarrow \text{local-pref } 80$$

After parsing the route map, our algorithm creates an additional mapping that assigns a default local preference of 100 and retains the existing values of the origin type and MED attributes, consistent with the behavior of Cisco routers.

5.3. Route Prediction

For each router, our route prediction tool obtains a set of eBGP-learned routes (described in Section 5.1) from the routing table, as well as a set of mappings that express the import policies for all eBGP sessions (described in Section 5.2). The tool then parses the BGP neighbor information to obtain a mapping from next-hop IP address to router ID for each session, and applies each mapping to the appropriate set of routing choices to produce a new set of routing choices for that prefix. The tool then applies the prediction algorithm described in Section 4 to determine the set of best routes to each destination prefix. We applied the existing import policies for the AT&T network to the routing choices obtained from routing tables of each of AT&T's border routers and verified that our tool produces the same attributes for local preference, origin type, and MED for every eBGP-learned route. We are in the process of testing the implementation of our prediction algorithm on this data.

We can optimize our route prediction tool by taking advantage of the fact that many of the destination prefixes have exactly the same routing choices. In March 2002, we observed that the AT&T BGP tables have over $100,000$ prefixes but just over $20,000$ unique routing choices (we note a similar result in previous work[1]). As such, computing the best route for every unique set of *routing choice*, rather than for every individual prefix, can reduce route prediction overhead by as much as a factor of 5. Additionally, operators typically experiment with small changes to the import policies. For example, an operator might change or add one clause in the route map associated with a single eBGP session. There is

no need to reapply the import policies for routes learned via other sessions or to repeat the prediction algorithm for the unaffected destination prefixes. We plan to extend our prototype to incorporate these enhancements.

6. CONCLUSION

In this paper, we have presented a model and an algorithm for predicting the effects of BGP import policy on path selection. We presented a model (summarized in Figure 2) that expresses the influence of BGP import policies, the BGP decision process, the IGP parameters, and the offered traffic on the distribution of traffic across links in the network. Our route prediction algorithm accurately determines the best routes as determined by the BGP decision process, given only a snapshot of the network state. The algorithm has a running time that is *linear* in the number of eBGP-learned routes and applies when the MED attribute is compared either across all routes or only for routes with the same next-hop AS. Finally, we have described how to obtain the data needed to run our algorithm from production routers and built a prototype that demonstrates that our algorithm correctly predicts the best BGP routes.

In future work, we intend to integrate our prototype with existing tools that capture the influence of the IGP configuration on the path selection process. We are also investigating how to account for the influence of route reflectors on the selection of the best BGP route at each router in the network. In addition, we plan to incorporate traffic measurements from the operational network to demonstrate how a change in import policy affects traffic load on network links. In previous work, we demonstrated ways for operators to change the flow of traffic in an efficient and predictable manner.[1] We envision that, in the future, our tool will generate recommendations for possible modifications import policies according to specified traffic engineering goals. Together, these pieces provide a useful traffic engineering framework for network operators.

APPENDIX A. CORRECTNESS THEOREMS

In this section, we prove the correctness of the network-wide route prediction algorithm from Section 4. In Section A.1, we prove the correctness of our algorithm in the case where the BGP decision process compares MEDs across all advertised routes. Section A.2 presents a proof of correctness in the case where MEDs are compared only across routing advertisements from the same neighboring AS.

A.1. Routing Prediction Without MEDs

We first show that the first stage of our prediction algorithm does not eliminate any local routing choices that would not be eliminated by the BGP decision process. Next, we show that the second stage of our algorithm, which compares across all network-wide routing choices, does not eliminate any routes that the BGP decision process would not eliminate. Furthermore, we show that every route that the BGP decision process eliminates is also be eliminated by our algorithm, thus proving that our algorithm produces the same results as the BGP decision process, given the same set of eBGP-learned routes.

THEOREM A.1. *Local routing prediction never eliminates a route that the BGP decision process would select as a globally best route. Formally,* $\forall p_i \in \mathcal{A}_i, (p_i \neq f_i \Rightarrow p_i \notin \mathcal{C})$.

Proof. The proof is by contradiction. Assume that there exists some $p_i \neq f_i$ that is in the set of globally best routes, $\mathcal{C}$. Then, it must be the case that route p_i is better than f_i according to Equations 1–5. However, if one of these conditions is true, then the BGP decision process would have eliminated p_i in favor of f_i. Thus, p_i cannot be in $\mathcal{C}$. □

Thus, we have shown that the application of local decision rules would *never* eliminate a route that the BGP decision process would have selected as a globally best route to that prefix. We now show that global comparison of the locally-best routes $\{f_i\}$ results in a set $\mathcal{B}$ that contains every route that belongs in the set of best routes $\mathcal{C}$.

THEOREM A.2. *Let* $\mathcal{C}$ *be the set of best routes as determined by the BGP decision process. Then,* $\forall p_i \in \mathcal{A}_i, (p_i \notin \mathcal{B} \Rightarrow p_i \notin \mathcal{C})$. *That is, if our algorithm eliminates a routing choice from some router* r_i*, then that route could not have been a route that BGP would have selected as a globally best route.*

Proof. The proof is by contradiction. Assume that there is some $p_i \in \mathcal{C}$ that is eliminated by our algorithm. Since Theorem A.1 showed that such an eBGP-learned route would not be eliminated locally, p_i must have been eliminated in the second stage of the algorithm by another route q that is better according to Equations 1–4. However, if this is the case, then the BGP decision process would also eliminate p_i in favor of q. Thus, p_i cannot be in $\mathcal{C}$. □

We have now shown that given a static, network wide view of the eBGP-learned routes (after application of the import policies), our algorithm determines a set of best routes $\mathcal{B}$ that entirely contains the set of best routes $\mathcal{C}$ that would be selected by the distributed collection of routers in the AS. To prove that our algorithm produces the *same* set of best routes as the BGP decision process, we must now show the converse—that if a route is eliminated by application of the BGP decision process, then our algorithm does not include this route in $\mathcal{B}$.

THEOREM A.3. *Any route that is eliminated by the BGP decision process must also be eliminated by application of network-wide routing prediction. That is,* $\forall p_i \in \mathcal{A}_i, (p_i \notin \mathcal{C} \Rightarrow p_i \notin \mathcal{B})$.

Proof. The proof is by contradiction. Assume that there exists some $p_i \in \mathcal{B}$ where $p_i \notin \mathcal{C}$. Then it must be the case that, for some router, a different routing choice q was chosen as a best route to a prefix, where our algorithm would have selected p_i. This, however, implies that the routing choice q, which appears in $\mathcal{C}$, was eliminated from $\mathcal{B}$. However, we know from Theorems A.1 and A.2 that this cannot be true. □

We have shown that if BGP is configured using `always-compare-med`, then our network-wide route prediction algorithm produces a set of routes that are consistent with the routes that would have been chosen by the BGP decision process, given some time-ordering of all route advertisement messages. Thus, if BGP is configured to determine the best route to a destination prefix independent of BGP message arrivals, our algorithm produces the same best routes as those which would result from the application of the BGP decision process to a series of BGP messages at each egress router.

A.2. Routing Prediction With MEDs

In this section, we show that the revised algorithm presented in Section 4.3 produces the same results as the BGP decision process in the case where routers are configured to compare MEDs only across the routes learned from the same neighboring AS.

THEOREM A.4. (THEOREM A.2 REVISITED). *Let $\mathcal{C}$ be the set of* best routes *as determined by the BGP decision process. Then,* $\forall p_i \in \mathcal{A}_i, (p_i \notin \mathcal{B} \Rightarrow p_i \notin \mathcal{C})$. *That is, if a routing choice is eliminated from any router r_i, then that route could not have been a route that BGP would have selected as a globally best route.*

Proof. The proof is by contradiction. Assume that there is some $p_i \in \mathcal{C}$ that is eliminated by our algorithm. Then, it must be the case that the route was eliminated by the application of Equations 1–4 for eBGP-learned routes (locally at each router), *an advertisement from another egress router with a lower MED value, based on Equation* 5, or locally based on router ID (Equation 5), or by a route learned from another router.

Theorem A.1 showed that such a route via eBGP could not be eliminated locally and still appear in the selection of best routes; if we consider the same argument for the application of Equations 1–4, then it cannot be the case that a route eliminated based on these equations can ever appear as a best route in $\mathcal{C}$.

If the route was eliminated by a route learned at another router with a lower MED value, then it must be the case that this route would have been eliminated by the BGP decision process when such a route was heard by iBGP. Therefore, a routing choice eliminated in this fashion cannot appear in the set of best routes $\mathcal{C}$.

If the routing choice p_i was eliminated locally by $q \in \mathcal{A}_i$ according to Equation 5, then it must be the case that q was one of the best routes at the global level (i.e., $q \in \mathcal{B}$). A router r_i can contribute at most one globally-best route. As such, p_i cannot be an element in $\mathcal{C}$.

Similarly, suppose p_i was eliminated by a route learned by another router. Then, it must be the case that some routing choice q exists for which Equations 1–4 are true. By the same argument as before, however, the BGP decision process would also eliminate such p_i from $\mathcal{C}$. As such, if a route is eliminated using our algorithm, that route must not be in $\mathcal{C}$. □

Note that Theorem A.3 still holds. Therefore, in the case where `bgp deterministic-med` is enabled, our revised algorithm produces the same results as the BGP decision process under arbitrary message ordering.

Acknowledgments

We would like to thank Jay Borkenhagen and Rich Kwapniewski for providing access to the configuration and routing data from the AT&T network, and for their help in understanding the routing policies. Thanks also to Tim Griffin and Joel Gottlieb for many helpful discussions about BGP policies and router configuration, and to Dave Andersen for helpful comments on a draft of this paper.

REFERENCES

1. N. Feamster, J. Borkenhagen, and J. Rexford, "Controlling the impact of BGP policy changes on IP traffic," Tech. Rep. HA173000-011106-02TM, AT&T Labs – Research, November 2001.
2. S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Networking* **1**, pp. 397–413, August 1993.
3. H. T. Kaur and K. S. Vastola, "The tunability of network routing using online simulation," in *Proc. Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 2000.
4. T. Ye, D. Harrison, B. Sikdar, H. T. Kaur, B.Mo, J.Jiang, S. Kalyanaraman, B. Szymanski, and K. Vastola, "Network management and control using collaborative on-line simulation," in *Proc. International Conference on Communications*, June 2001.
5. A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "NetScope: Traffic engineering for IP networks," *IEEE Network Magazine* , pp. 11–19, March 2000.
6. D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of Internet traffic engineering." Work in progress, Internet Draft draft-ietf-tewg-principles-02.txt. Expires May 2002.
7. D. O. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for traffic engineering over MPLS." Request for Comments 2702, September 1999.
8. D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communication Magazine* , pp. 42–47, December 1999.
9. Y. Rekhter and T. Li, "A Border Gateway Protocol." Request for Comments 1771, March 1995.
10. S. Halabi and D. McPherson, *Internet Routing Architectures*, Cisco Press, second ed., 2001.
11. J. W. Stewart, *BGP4: Inter-Domain Routing in the Internet*, Addison-Wesley, 1998.
12. G. Huston, "Interconnection, peering, and settlements," in *Proc. INET*, June 1999.
13. L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. Networking* **9**, pp. 681–692, December 2001.
14. "BGP Best Path Selection Algorithm." `http://www.cisco.com/warp/public/459/25.shtml`.
15. "How the Active Route Is Determined." `http://arachne3.juniper.net/techpubs/software/junos42/swconfig-routing42/html/protocols-overview4.html#1045417`.
16. "Foundry Switch and Router Installation and Configuration Guide, Chapter 19, Configuring BGP4." `http://www.foundrynet.com/services/documentation/SRguide/FoundryManual_BGP4.html`.
17. A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Trans. Networking* **9**, June 2001.
18. "BGP Commands." `http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/np1_r/1rprt1/1rbgp.htm`.