Stable Internet Routing Without Global Coordination

Lixin Gao *
Computer Science
Smith College
Northampton, MA 01060
gao@cs.smith.edu

Jennifer Rexford
Internet and Networking Systems
AT&T Labs – Research
Florham Park, NJ 07932
jrex@research.att.com

ABSTRACT

The Border Gateway Protocol (BGP) allows an autonomous system (AS) to apply diverse local policies for selecting routes and propagating reachability information to other domains. However, BGP permits ASes to have conflicting policies that can lead to routing instability. This paper proposes a set of guidelines for an AS to follow in setting its routing policies, without requiring coordination with other ASes. Our approach exploits the Internet's hierarchical structure and the commercial relationships between ASes to impose a partial order on the set of routes to each destination. The guidelines conform to conventional traffic-engineering practices of ISPs, and provide each AS with significant flexibility in selecting its local policies. Furthermore, the guidelines ensure route convergence even under changes in the topology and routing policies. Drawing on a formal model of BGP, we prove that following our proposed policy guidelines guarantees route convergence. We also describe how our methodology can be applied to new types of relationships between ASes, how to verify the hierarchical AS relationships, and how to realize our policy guidelines. Our approach has significant practical value since it preserves the ability of each AS to apply complex local policies without divulging its BGP configurations to others.

1. INTRODUCTION

The Internet connects thousands of Autonomous Systems (ASes) operated by different institutions, such as Internet Service Providers (ISPs), companies, and universities. Routing within an AS is controlled by intradomain protocols such as OSPF, IS-IS, and RIP. ASes interconnect via dedicated links and public network access points, and exchange reachability information using the Border Gateway Protocol (BGP) [18; 19]. BGP is an interdomain routing protocol that allows ASes to apply local policies for selecting routes and propagating routing information, without revealing their policies or internal topology to others. However, recent studies have shown that a collection of ASes may have conflicting BGP policies that lead to route divergence [10; 20]. Route divergence can result in route oscillation, which can significantly degrade the end-to-end performance of the Internet. Avoiding these conflicting BGP policies is crucial for the stability of the Internet routing infrastructure. Yet, to be practical, any technique for ensuring convergence should not sacrifice the ability of each AS to apply complex local policies.

A natural approach to the route convergence problem involves the use of an Internet Routing Registry (IRR), a repository of routing policies specified in a standard language [16]. A complete and up-to-date registry could check if the set of routing policies has any potential convergence problems. However, this global coordination effort faces several impediments. First, many ISPs may be unwilling to reveal their local policies to others, and may not keep the registry up-to-date. Second, and perhaps more importantly, even if ISPs decide to reveal their local polices, recent work has shown that statically checking for convergence properties is an NP-complete problem [10]. Third, even if the registry could ensure convergent routes under a given topology, BGP still might not converge under router or link failures, or a policy change. Hence, rather than requiring global coordination, we believe that convergence should be achieved by restricting the set of policies that each AS can apply.

In this paper, we propose a set of guidelines for an AS to follow in setting its routing policies, without requiring coordination with other ASes. Our approach capitalizes on the Internet's hierarchical structure and the commercial relationships between ASes. These relationships include customer-provider, peer-to-peer, and backup. A customer pays its provider for connectivity to the rest of the Internet, whereas peers agree to exchange traffic between their respective customers free of charge; an AS may also provide backup connectivity to the Internet in the event of a failure. Under our guidelines, routing via a peer or a provider is never preferable to routing via a customer link; furthermore, routes via backup links have the lowest preference. An AS is free to apply any local policies to the routes learned from neighbors within each preference class. These guidelines conform to

*The first author was supported in part by NSF grant NCR-9977555, and NSF CAREER Award grant ANI-9875513. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

conventional traffic-engineering practices of ISPs, and this might well explain why Internet routing divergence has not occurred yet. However, it is crucial to make these guidelines explicit since BGP itself does not constrain routing policies to ensure convergence. Based on our results, we propose a simple routing registry that requires each AS to disclose only its *relationship* with each of its neighbors, rather than the entire set of routing policies.

The remainder of the paper is structured as follows. Section 2 presents an overview of interdomain routing and discusses previous work on BGP protocol dynamics. Then, Section 3 presents a formal model of BGP that includes ASes with multiple BGP speakers, both interior BGP (iBGP) and exterior BGP (eBGP), and additional BGP attributes. We define the types of relationships between ASes and describe the hierarchical structure of the AS graph in Section 4. In Section 5, we present our policy guidelines and formally prove that adherence to these guidelines guarantees convergence for all possible initial states. We show how to permit additional flexibility in choosing between routes through customers and routes through peers by making realistic assumptions about peer-to-peer relationships. Then, Section 6 discusses the robustness of our guidelines to changes in network topology, routing policies, and relationships between ASes. We describe how to apply our methodology to new types of relationships that can arise between ASes, and how an AS pair can transition to a new relationship while preserving BGP stability. Section 7 concludes the paper with a discussion of future research directions.

2. INTERDOMAIN ROUTING

In this section, we present background material on the Internet architecture [11] and the use of BGP for interdomain routing [18; 19]. We also summarize previous work on the protocol dynamics of BGP.

2.1 Internet Architecture

The Internet consists of a large collection of hosts interconnected by networks of links and routers. The Internet is divided into thousands of distinct regions of administrative control, referred to as *autonomous systems* (ASes). Examples range from college campuses and corporate networks to large Internet Service Providers (ISPs). An AS has its own routers and routing policies, and connects to other ASes to exchange traffic with remote hosts. A router typically has very detailed knowledge of the topology within its AS, and limited reachability information about other ASes. ASes interconnect at public Internet exchange points (IXPs) or dedicated point-to-point links. Public exchange points typically consist of a shared medium, such as a FDDI ring or an ATM switch, that interconnects routers from several different ASes. Physical connectivity at the IXP does not necessarily imply that every pair of ASes exchanges traffic with each other. AS pairs negotiate contractual agreements that control the exchange of traffic. These relationships include customer-provider, peer-to-peer, and backup, and are discussed in more detail in Section 4.

Each AS has responsibility for carrying traffic to and from a set of customer IP addresses. The scalability of the Internet routing infrastructure depends on the aggregation of IP addresses in contiguous blocks, called *prefixes*, each consisting of a 32-bit IP address and a mask length (e.g., 1.2.3.0/24). An AS employs an *intradomain* routing protocol (such as OSPF or IS-IS) to determine how to reach each customer prefix, and employs an *interdomain* routing protocol (BGP) to advertise the reachability of these prefixes to neighboring ASes. BGP is a distance-vector protocol that constructs paths by successively propagating advertisements between pairs of routers that are configured as *BGP peers* [18; 19]. Each advertisement concerns a particular prefix and includes the list of the ASes along the path (the *AS path*). Upon receiving an advertisement, a BGP speaker must decide whether or not to use this path and, if the path is chosen, whether or not to propagate the advertisement to neighboring ASes (after adding its own AS number to the AS path). A BGP speaker withdraws an advertisement when the prefix is no longer reachable with this route, which leads to a sequence of withdrawals by upstream ASes that are using this path.

The simplest distance-vector protocol would employ shortest-path routing. BGP allows a much wider range of policies based on how the routers are configured. An AS can favor one path over another by assigning a *local preference*. BGP also allows an AS to send a hint to a neighbor on the preference that should be given to a route by using the *community* attribute. An AS can control how traffic enters its network by assigning a different *multiple exit discriminator* (MED) value to the advertisements it sends on each link to a neighboring AS. Otherwise, the neighboring AS would select the link based on its own intradomain routing protocol. An AS can also discourage traffic from entering its network by performing *AS prepending*, which inflates the length of the AS path by listing an AS number multiple times. Processing an advertisement involves three steps — *import policies* that decide which routes to consider, *path selection* that decides which route to use, and *export policies* that decide whether (and what) to advertise to a neighboring AS — that are discussed in more detail in Section 3.

2.2 Protocol Dynamics

The growing importance and complexity of the Internet routing infrastructure has sparked interest in understanding BGP protocol dynamics. Previous work consists of measurement-based studies of BGP protocol traffic and theoretical analysis of BGP convergence properties. Extensive traces of BGP update messages have been used to characterize the structure (and growth) of the Internet topology, as well as the stability of routes to destination prefixes [7; 13; 14; 15]. In contrast, research on BGP convergence has focused on determining what combination of BGP policies would cause a group of ASes to continually advertise and withdraw routes to a given prefix [6; 8; 9; 10; 20]. BGP convergence problems would not arise if every AS selects shortest-path routes. However, ASes can have conflicting local policies when they use the local-preference attribute to favor a route with a non-minimal AS path. This can result in route oscillation, where an AS makes a decision and advertises a new route to its neighbors, which causes neighbors to change their decisions; then, these ASes withdraw their previous route and advertise new ones, and the process repeats.

Previous research has studied route convergence under the assumption of global knowledge of the topology and routing policies. The work in [20] analyzes route oscillation in simple ring topologies, and suggests maintaining a global routing registry of interdomain policies that can be checked for potential convergence problems [2; 6; 16; 20]. Expanding on these observations, the work in [10] presents a formal model of BGP that focuses on local-preference and AS-path-length attributes. Since the paper proves *negative* results about BGP convergence properties, it is sufficient to consider a restricted subset of the protocol. In particular, the study establishes that the problem of checking the convergence properties is NP-complete, even with full knowledge of the routing policies of each AS. In addition, the paper presents several examples of conflicting BGP policies, including scenarios when the divergence occurs only after a link failure. A follow-up paper [9] presents a dynamic model that captures the asynchronous processing of updates at each AS. The paper formalizes the notion of a *stable state* where no AS would change its routes, and a *safe BGP system* that is guaranteed to converge to a stable state. The paper presents a sufficient condition for a BGP system to be safe. However, testing adherence to the condition requires full knowledge of the AS graph and the set of routing policies for each AS.

These results suggest that it may be possible to restrict local policies in a way that guarantees BGP convergence, while still allowing greater flexibility than shortest-path routing. Our paper focuses on constructing a set of reasonable policy guidelines that guarantee a safe BGP system, even under changes in network topology and routing policies, without requiring coordination between ASes.

3. ABSTRACT MODEL OF BGP

In this section, we present an abstract model of BGP that we use in establishing the stability properties in Section 5. The model extends the work in [9; 10] to include interior BGP (iBGP) and exterior BGP (eBGP), additional BGP attributes and operations (such as MEDs, community set, and AS prepending), and the possibility that an AS has multiple BGP speakers. This more complete model of BGP is necessary for establishing *positive* results about system stability.

3.1 BGP Routing

The topology of a BGP system is modeled as a clustered graph $G = (N, V, E)$, where the set N consists of ASes, the vertex set V consists of all BGP-speaking routers, and the edge set E consists of all eBGP peering sessions. Each BGP speaker belongs to one AS and an AS can have one or more BGP speakers. Let $a(i) \in N$ denote the AS that BGP speaker i belongs to. Each eBGP peering session involves a pair of BGP speakers in different ASes. Each BGP-speaker pair in the same AS has an iBGP session and a cost metric that represents the distance between the two BGP speakers based on the intradomain routing protocol. BGP speakers i and j in different ASes (i.e., $a(i) \neq a(j)$) has a set of eBGP sessions $E(i, j) \subseteq E$, which may be empty. Figure 1 shows an example of the topology in a BGP system.

A route update r includes the destination prefix ($r.prefix$), next-hop interface address ($r.next\_hop$), AS path ($r.as\_path$), local preference ($r.local\_pref$), multiple-exit discriminator ($r.med$), and community set ($r.c\_set$). Each BGP speaker originates updates for one or more prefixes, and can send

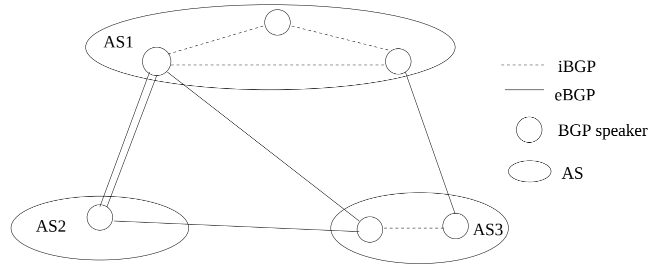

Figure 1: An example of a BGP system topology

the updates to the immediate neighbors via an iBGP or eBGP session. BGP-speaker pairs in the same AS use iBGP to exchange routes learned from BGP peers. In practice, an AS may configure its iBGP sessions to avoid a full exchange of routing updates between speaker pairs (e.g., by using route reflectors). These optimizations are intended to reduce iBGP traffic without affecting the routing decisions [11] and, hence, are not included in our model. Routing updates exchanged via eBGP sessions are transformed according to the BGP policies. We consider an eBGP session $l \in E$ between two BGP speakers, u and v. BGP speaker v receives a set of route updates R on l from u. BGP speaker v applies *import policies* to transform incoming route updates, and applies *export policies* before sending updates to the neighbor u.

An AS can apply both implicit and explicit import policies. Let $im\_import(l, v)[R]$ denote the set of updates after applying the implicit import policy of v on edge l. Every edge has an implicit import policy that discards a routing update when the receiving BGP speaker's AS already appears in the AS path; this is essential to avoid introducing a cycle in the AS path. That is, if $a(v) \in r.as\_path$, then $im\_import(l, v)[\{r\}] = \{\}$; otherwise $im\_import(l, v)[\{r\}] = \{r\}$. Let $ex\_import(l, v)[R]$ represent the set of updates after applying the explicit import policy, such as denying or permitting an update, and assigning a local-preference value. For example, an explicit import policy could assign $r.local\_pref = 100$ if AS 1 appears in $r.as\_path$ or deny any update that includes AS 2 in the path. Ultimately, the import policy transforms the set of updates R as $import(l, v)[R] = ex\_import(l, v)[im\_import(l, v)[R]]$.

After applying the import policies for a route update from an eBGP session, v exchanges the update with all other BGP speakers in the same AS, using iBGP sessions. Each BGP speaker v then follows a route selection process $Select(S)$ that picks the best route for each prefix. The BGP speaker picks the route with the highest $r.local\_pref$, breaking ties by selecting the route with the shortest $r.as\_path$. Note that local preference overrides the AS-path length. Amongst the remaining routes, v picks the one with the smallest $r.med$, breaking ties by selecting the route with the smallest cost to the BGP speaker that passes the route via an iBGP session. Note that, since the tie-breaking process draws on intradomain cost information, two BGP speakers in the same AS may select different best routes for the same prefix. If a tie still exists, v picks the route with the smallest $r.next\_hop$.

Each BGP speaker sends its best route (one best route for each prefix) via eBGP sessions. The BGP speaker u applies implicit and explicit export policies on each eBGP session l to a neighboring BGP speaker v, defined as $im\_export(l,u)$ and $ex\_export(l,u)$, respectively. Each BGP speaker u applies an implicit policy that sets $r.local\_pref$ and $r.med$ to default values, assigns $r.next\_hop$ to u's interface connecting to l, and prepends u to $r.as\_path$. Explicit export policies include permitting or denying the route, assigning $r.med$, assigning $r.c\_set$, and prepending u one or more times to $r.as\_path$. For example, AS u could decline to advertise routes to AS v that have community 10 in the community set. Also, AS u could prepend u two times to the AS path for prefix 1.2.3.0/24 and for any route that includes AS 2 in the AS path. Ultimately, the export policy transforms the set of updates R as $export(l,u)[R] = ex\_export(l,u)[im\_export(l,u)[R]]$. Then, u transmits these transformed updates to v using eBGP sessions.

3.2 Distributed Path Selection

The route-selection process proceeds in a distributed and asynchronous fashion, triggered by advertisements and withdrawals of routes. Rather than modeling the exact timing of message transmissions, we focus on the decision-making process of each BGP speaker. For the sake of simplicity, we focus on a single destination prefix d that originates from AS_d; since address aggregation does not affect the convergence properties, it is sufficient to consider the set of routes to a single destination prefix. Each speaker applies the BGP selection process to pick its best path to d, after applying import policies to the routes that have been exported by its neighbors. BGP is an incremental protocol, where each speaker remembers the routes advertised by neighbors until they are withdrawn, and selects a best path from this set. In a stable state, a BGP speaker remembers precisely those routes that have been chosen by its neighbors. Hence, for studying convergence properties, it is sufficient to define the state of the BGP system in terms of the route chosen by each BGP speaker. That is, we assume that each speaker remembers only its own best route, selected from the set of routes exported by its neighbors. As such, we define the system state as a vector $s = (s_1, s_2, ..., s_n)$, where s_i denotes the route chosen by speaker $i = 1, 2, \ldots, n$.

Changes in the system state occur when one or more BGP speakers apply the route selection process. Formally, *activating* a speaker applies the export policies of the BGP speakers in neighboring ASes, the speaker's import policies, and the BGP path-selection process [9]. In particular, if the BGP speaker resides in AS_d, the route to d is a route (denoted as r_0) that contains a null AS path. Otherwise, the selection of s_i can be affected by the route chosen by any BGP speaker j that has a BGP session with a speaker $k \in a(i)$. This includes the BGP peers of speaker i, as well as the BGP peers of the other speakers in the same AS, since i could learn about these routes via iBGP sessions. The choices available to speaker i depend on the route s_j, the export policies of j, and the import policies of k:

$$Choices(i,s) = \begin{cases} r_0, & \text{if } a(i) = AS_d \\ \cup_{l \in E(k,j) \wedge k \in a(i)} import(l,k)[export(l,j)(s_j)], & \\ \quad \text{otherwise} & \end{cases}$$

Then, i selects a route $BestRoute(i,s) = Select(Choices(i,s))$. Note that the model assumes that each external neighbor's route is immediately available and that these routes are propagated via iBGP sessions. This simplifying assumption does not affect the BGP convergence properties, as the neighbors' updates would eventually become available (e.g., after finite propagation delay).

Since each BGP speaker operates independently, we cannot assume that every BGP speaker is activated at the same time. Instead, as in [9], we consider a subset $A \subseteq V$ of speakers that are activated at a given time. The remaining BGP speakers do not apply the path-selection process and, hence, do not change their best route. Therefore, the next state $s' = (s'_1, s'_2, ..., s'_n)$ has $s'_i = BestRoute(i,s)$ for $i \in A$, and $s'_i = s_i$ for $i \notin A$. We let $s \xrightarrow{A} s'$ denote the transition from state s to s' given the activation set A. The definition of the state of a BGP system, and the notion of an activation set, allows us to precisely define the notion of stability. Formally, a state s is *stable* if and only if $s \xrightarrow{A} s$ for any activation set A. That is, when the system is in state s, no AS would change to a different route.

To study convergence, we define an *activation sequence* as a (possibly infinite) sequence of activations. Let σ denote the activation sequence and $\sigma(j) \subseteq V$ denote the jth activation in σ. In studying convergence, we need to consider sequences that activate each AS several times. In particular, a *fair* activation sequence σ is an infinite sequence that has infinitely many elements j such that $n \in \sigma(j)$, for each BGP speaker $n \in V$. A BGP system *converges* on an activation sequence and an initial state if it arrives at a stable state after the activation sequence. Formally, for an activation sequence σ and an initial state s^0, a BGP system converges if there is a finite j such that $s^0 \xrightarrow{\sigma(1)} s^1 \xrightarrow{\sigma(2)} \ldots \xrightarrow{\sigma(j)} s^j$ and s^j is a stable state.

Thus far, we have defined the notion of a stable state. But, some BGP systems have a stable state without necessarily converging. For example, Figure 2 shows an example where three ASes are connected pairwise and AS 0 originates destination prefix d [10]; with each AS, we list the set of possible routes in order of preference. Both AS 1 and AS 2 prefer the path through the neighbor over the direct route to reach d. The system has a stable state. For example, AS 2 could use the direct route (0) and AS 1 could use the route $(2,0)$. However, the system could also oscillate between two unstable states. In the first state, both ASes have selected the direct route (0). Then, if activated simultaneously, both ASes switch to their indirect routes (e.g., AS 2 switches to $(1,0)$). Then, if activated again, both ASes return to their direct routes, and the process repeats. Whether or not the system eventually reaches a stable state depends on the exact timing of the reception and processing of the route updates. Hence, we define a stronger notion of a *safe* BGP system [9]. A BGP system is *safe* if it has a stable state and converges under any fair activation sequence and any initial state.

4. HIERARCHICAL AS GRAPH

Our policy configuration guidelines capitalize on the fact that ASes are interconnected in a hierarchical fashion. In this section, we describe the relationships between ASes and the resulting hierarchical structure.

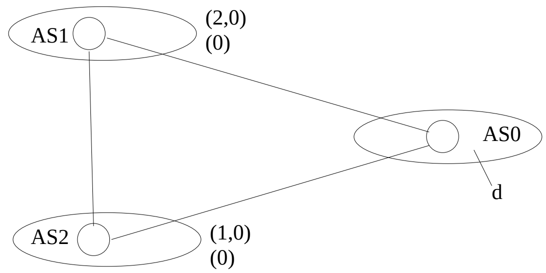

Figure 2: A BGP system has a stable state but might not converge

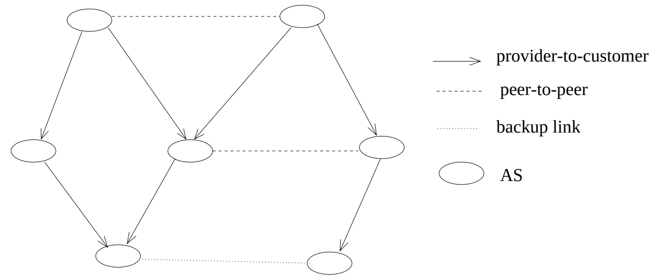

Figure 3: A hierarchical AS interconnection

4.1 Clients, Providers, and Peers

AS relationships arise from contracts that define the pricing model and the exchange of traffic. In a *customer-provider* relationship, the customer is typically a smaller AS that pays a larger AS for access to the rest of the Internet. The provider may, in turn, be a customer of an even larger AS. In a *peer-to-peer* relationship, the two peers are typically of comparable size and find it mutually advantageous to exchange traffic between their respective customers. Each eBGP session defines a relationship between the two ASes it connects. Although there might be multiple eBGP sessions between two ASes, the relationship between the two ASes should be uniquely defined. An AS a may have multiple customers, providers, and peers. We define $customer(a)$, $peer(a)$, and $provider(a)$ as the set of customers, peers, and providers of a, respectively. We let $first(r.as\_path)$ denote the first AS in $r.as\_path$. A route r is classified as a customer route of a if $first(r.as\_path) \in customer(a)$, a peer route if $first(r.as\_path) \in peer(a)$, or a provider route if $first(r.as\_path) \in provider(a)$. Two ASes may also have a bilateral *backup* agreement, as discussed in more detail in Section 5.2.

The customer-provider and peer-to-peer agreements translate into several rules governing BGP export policies [1; 12]:

- **Exporting to a provider:** In exchanging routing information with a provider, an AS can export its routes and the routes of its customers, but can not export routes learned from other providers or peers.
- **Exporting to a customer:** In exchanging routing information with a customer, an AS can export its routes, as well as routes learned from its providers and peers.
- **Exporting to a peer:** In exchanging routing information with a peer, an AS can export its routes and the routes of its customers, but can not export the routes learned from other providers or peers.

Drawing on our abstract model, consider a BGP speaker u and with a link l connecting to an AS $v \in provider(a(u)) \cup peer(a(u))$. For each r, if $first(r.as\_path) \in provider(a(u)) \cup peer(a(u))$, then $ex\_export(l, u)[\{r\}] = \{\}$.

4.2 Hierarchy

We assume that there is a hierarchical customer-provider relationship among ASes. The hierarchical structure arises because an AS typically uses an AS of larger size as a provider. An AS u serving a metropolitan area is likely to have a regional provider v, and a regional AS v is likely to have a national provider w; it is very unlikely that a nationwide AS w would be a customer of a metropolitan-area AS u. That is, if $u \in customer(v)$ and $v \in customer(w)$, then $w \notin customer(u)$. AS v is a *direct* provider of u, whereas AS w is an *indirect* provider of w. Any direct or indirect provider of u cannot be a customer of u. To simplify the discussion, we define two directed graphs formed by the customer-provider relationships. In the *provider-to-customer graph*, the edges are directed from provider to customer. The resulting subgraph formed by only provider-customer relationships should be a *directed acyclic graph* (DAG), as shown in the example in Figure 3. In the *customer-to-provider graph*, the edges are directed from customer to provider.

A route registry can be used to verify the hierarchical relationships. Each AS a supplies its set $provider(a)$, updating the registry upon adding or deleting a provider. The registry can check for a cycle whenever any AS changes its set of providers. This could happen when an AS adds or removes a provider, or when an AS changes its relationship with one of its neighbors; for example, a pair of ASes may transition from a customer-provider relationship to a peer-to-peer arrangement. The algorithm for checking whether there is a cycle in a directed graph takes $O(|N| + |E|)$ time [4], where $|E|$ is the number of edges and $|N|$ is the number of nodes of the directed graph. As of January 2000, there were 6474 ASes and 13895 AS interconnections known to globally-connected BGP systems [17]. BGP permits at most $2^{16} = 65536$ AS numbers and the number of AS interconnections tends to grow linearly in the number of ASes [5]. Therefore, it is possible to run the cycle-detection algorithm whenever an AS updates its list of providers to ensure the conformity to the hierarchical relationships at all times.

If the provider-to-customer or customer-to-provider graph has a cycle, the registry can efficiently identify the sequence of ASes involved. If more detailed information is available about the routing policies of these ASes, the registry could check for possible convergence problems. Although checking for convergence is an NP-complete problem [10], the check would be applied on the subgraph, which would involve much fewer vertices and edges than the initial AS graph. Alternatively, the registry could instruct the ASes in the cycle to coordinate amongst themselves to avoid policies that would cause convergence problems, or to force the use of a restrictive policy (such as shortest AS path) that would guarantee convergence.

5. BGP POLICY GUIDELINES

This section presents policy guidelines that ensure that the BGP system is safe. To simplify the discussion, we initially consider only customer-provider and peer-to-peer relationships. We then extend the guidelines to include backup relationships. Since the route selection process for each destination prefix is independent of other prefixes, it is sufficient to consider only one destination prefix d in describing and analyzing the guidelines.

5.1 BGP Systems with no Backup Link

In this section, we present the policy configuration guidelines for BGP systems that have only customer-provider and peer-to-peer relationships. We first consider the guideline for the case that any AS pair can have a peer-to-peer agreement. Then, we expand the set of local policies by imposing realistic restrictions on which AS pairs can have peer-to-peer relationships.

5.1.1 *Unconstrained Peer-to-Peer Agreements*

Our guideline requires an AS to prefer a route via a customer over a route via a provider or peer. Formally, we have guideline A for the explicit import policy of each BGP speaker in AS a:

Guideline A
if $((first(r_1.as\_path) \in customer(a))$ and $(first(r_2.as\_path) \in peer(a) \cup provider(a)))$ then $r_1.loc\_pref > r_2.loc\_pref$

Note that Guideline A does *not* restrict the preference among customer routes or among provider or peer routes, which leaves ISPs with significant flexibility in selecting local policies. In addition, ISPs have a financial incentive to follow the guideline since an ISP does not have to pay its customer to carry traffic. Guideline A allows a large number of possible configurations, much larger than policies based only on AS-path length. To implement the guidelines, an AS could allocate a range of local-pref values for each type of route (e.g., 86–100 for customer routes and 75–85 for peer and provider routes).

Guideline A ensures that the BGP system is safe. The proof draws on how the local-pref assignment affects how each BGP speaker picks its best route.

THEOREM 5.1. *For a BGP system that has only customer-provider and peer-to-peer relationships, if all ASes follow guideline A, then the BGP system is safe.*

We prove the theorem by two lemmas. The first lemma claims that the BGP system has a stable state. The second lemma claims that the BGP system converges to the stable state for any initial state and any fair activation sequence.

LEMMA 5.1. *The BGP system has a stable state.*

PROOF: We prove the lemma by constructing an activation sequence σ^* that leads to a stable state for any initial state. Let d denote the destination prefix and AS_d denote the AS that originates prefix d. Since the activation order among the BGP speakers within an AS does not affect the best route selection of the BGP speakers, we activate all BGP speakers of an AS simultaneously. For simplicity of explanation, we use the activation of an AS to represent the activation of all BGP speakers in the AS. We activate ASes in two phases. In the first phase, a AS selects a customer route if one is available, following Guideline A. This is accomplished by activating the ASes in an order that conforms to the partial order in the customer-to-provider DAG. In the second phase, the ASes that do not have a customer route after Phase 1 get provider or peer routes. This is accomplished by activating ASes in an order that conforms to the partial order in the provider-to-customer DAG. Formally, we have a two-phase activation sequence σ^* as follows.

Phase 1: Activate ASes in a linear order that conforms to the partial order in the customer-to-provider DAG.
Phase 2: Activate ASes in a linear order that conforms to the partial order in the provider-to-customer DAG.

For the simplicity of the discussion, we partition the ASes into two classes; the first class consists of AS_d and ASes that select a customer route in Phase 1. The second class consists of the remaining ASes. We call ASes in the first class *Phase-1 ASes* and ASes in the second class *Phase-2 ASes*. Similarly, we call BGP speakers in a Phase-1 AS *Phase-1 BGP speakers* and BGP speakers in a Phase-2 AS *Phase-2 BGP speakers*. The activation sequence results a stable state independent of the initial state. We prove that each Phase-1 BGP speaker reaches a stable state after its activation in Phase 1 and each Phase-2 BGP speaker reaches a stable state after its activation in Phase 2. In other words, we prove the following two claims.

Claim 1: A Phase-1 BGP speaker reaches a stable state after its activation in Phase 1.

Proof: We prove by induction on the order that Phase-1 BGP speakers are activated in Phase 1. Clearly, among Phase-1 BGP speakers, BGP speakers in AS_d are the first to be activated. BGP speakers in AS_d reach a stable state as soon as AS_d is activated. Let Phase-1 BGP speaker i belong to AS_n. Suppose all Phase-1 BGP speakers that belong to an AS preceding AS_n in Phase 1 reach a stable state after their activation. BGP speaker i selects the best route amongst its customer routes. All of the customers precede AS_n in the activation sequence for Phase 1. Hence, each customer has either reached a stable state (earlier in Phase 1) or does not get a customer route in Phase 1. Any customer that does not get a customer route in Phase 1 does not export its route to BGP speaker i according to export policy rule. Hence, those customers' routing decisions do not affect BGP speaker i. Therefore, BGP speaker i reaches a stable state after its activation in Phase 1.

Claim 2: A Phase-2 BGP speaker reaches a stable state after its activation in Phase 2.

Proof: Following a similar approach, we prove by induction on the order that Phase-2 BGP speakers are activated in Phase 2. Let AS_0 be the first Phase-2 AS that is activated in Phase 2. Clearly, AS_0 does not have any Phase-2 provider. Since AS_0's BGP speakers are not Phase-1 BGP speakers, these BGP speakers can only get routes from AS_0's peers and providers. AS_0's peers either (a) are stable after Phase 1 (if there is a customer route) or (b) do not export their routes AS_0 (if the best route is a provider or peer route). The peers that fall in case (a) are stable before AS_0 are activated. The peers that fall in case (b) do not affect AS_0's BGP speakers' route. Since AS_0 does not have any Phase-2 provider, its providers are stable after Phase 1. Therefore, AS_0's BGP speakers are stable after their activation in Phase 2.

Let Phase-2 BGP speaker i belong to AS_n. Suppose all BGP speakers that belong to an AS preceding AS_n in Phase 2 reach a stable state after their activation in Phase 2. Since no customer route was learned in Phase 1, BGP speaker i must select a route from one of its providers or peers. Each provider has already reached a stable state (either in Phase 1, or earlier in the activation sequence of Phase 2). Each peer is either a Phase-1 AS or a Phase-2 AS. If a peer is a Phase-1 AS, the peer's route is available to BGP speaker i when it is activated in Phase 2. If a peer is a Phase-2 AS, then this peer selects a route from one of its providers or one of its other peers. The peer would not announce such a route to BGP speaker i and, hence, the routing decision would not affect BGP speaker i. Therefore, BGP speaker i reaches a stable state after its activation in Phase 2. ∎

LEMMA 5.2. *The BGP system converges to the stable state for any initial state and any fair activation sequence.*

PROOF: Given any fair activation sequence σ, we prove by induction on the ASes in the order given by Phase-1 ASes followed by Phase-2 ASes where both Phase-1 and Phase-2 ASes are in the order of activation sequence σ^*. It is clear that each BGP speaker in AS_d reaches a stable state after a single activation. Suppose that all BGP speakers in the ASes that precede AS_n are stable after activation $\sigma(t)$. Let $\sigma(t')$ be the first activation set such that all BGP speakers in AS_n have been activated at least once between $\sigma(t)$ and $\sigma(t')$. Note that we can find t' since any fair activation sequence activates a BGP speaker infinitely many times. Using the same argument as above, we can prove that all BGP speakers in AS_n reach a stable state after $\sigma(t')$. Therefore, the system converges to the stable state after a finite number of activations in the fair activation sequence. ∎

Figure 4 presents an example of a set of policies that violates Guideline A. The directed edges in the graph indicate the provider-to-customer relationships, and the routes of each AS are listed in the order of preference. AS 3 violates the guideline since by preferring a provider route (via AS 2) over a customer route (via AS 0). This BGP system is not safe. Each AS initially selects route (0) and then decides to change to a route through its counterclockwise neighbor. This process can continue indefinitely. As another example, consider the BGP system given in Figure 2. AS 1 and AS 2 are peers and both are providers of AS 0. Both AS 1 and AS 2 prefer the peer route over the customer route, which violates Guideline A. The resulting BGP system is not safe.

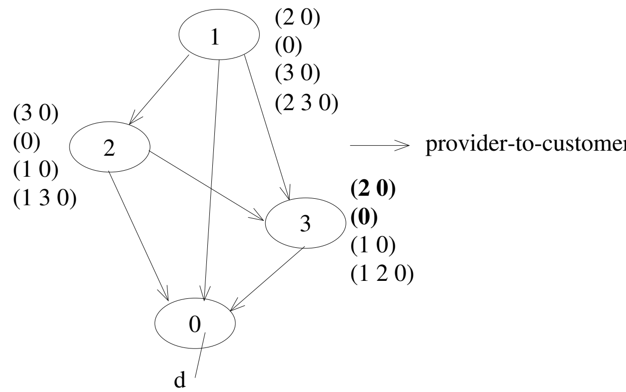

Figure 4: A BGP system that violates Guideline A

5.1.2 *Constrained Peer-to-Peer Relationships*

Guideline A assumes that any pair of ASes could have a peer-to-peer agreement. In this section, we make some realistic assumptions about peering agreements so as to relax the guideline. In particular, we allow peer routes to have the *same* local-pref as customer routes, to give ISPs greater flexibility in balancing network load. Typically, a peer-to-peer relationship is between two ASes of similar size. An AS is unlikely to have a peer-to-peer relationship with one of its (direct or indirect) providers. More generally, we say that AS u is a *peer-provider* of v if there exists an AS $w \in peer(v)$ such that u is a (direct or indirect) provider of w. That is, u is a (direct or indirect) provider of one of v's peers. We assume that peer-to-peer relationships satisfy the following condition:

Assumption P: *For any pair of ASes a_1 and a_n, there is no sequence of ASes $a_2, a_3, ..., a_{n-1}$ such that a_i is a peer-provider of a_{i+1} for any $1 \leq i < n$.*

A routing registry can check for violations of Assumption P and notify the ASes involved, or force the system to abide by Guideline A.

Assumption P allows us to relax Guideline A to allow a peer route to have the same local-pref as a customer route. Formally, we have Guideline B for the explicit import policy of each BGP speaker in AS a:

Guideline B
if $((first(r_1.as\_path) \in customer(a))$ and $(first(r_2.as\_path) \in peer(a))$ then $r_1.loc\_pref \geq r_2.loc\_pref$
if $((first(r_1.as\_path) \in customer(a))$ and $(first(r_2.as\_path) \in provider(a))$ then $r_1.loc\_pref > r_2.loc\_pref$

Assumption P is essential for the stability of BGP system.

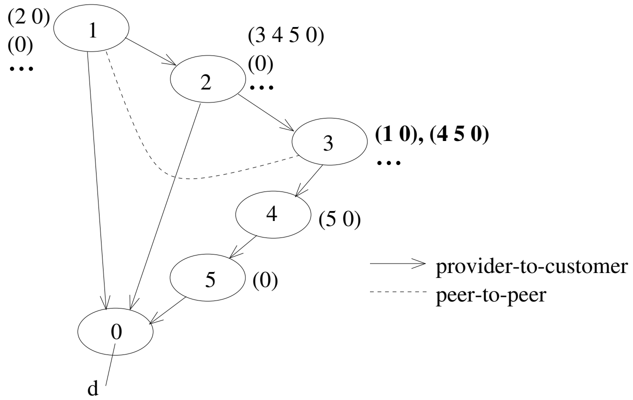

Figure 5: A BGP system that obeys Guideline B but violates Assumption P

For example, the BGP system in Figure 5 violates Assumption P since AS 1 is an indirect provider *and* a peer of AS 3. Applying Guideline B, AS 3 assigns equal preference to the route $(1,0)$ through its peer and the route $(4,5,0)$ through its customer; AS 3 ultimately favors the route $(1,0)$ with the shorter AS path. However, AS 1 prefers the customer route $(2,0)$ over its direct route (0), and AS 2 prefers the route $(3,4,5,0)$ through its customer over the route $(1,0)$ through its provider. Assume that initially none of the ASes have a route to d. After AS 5 and AS 4 have been activated, assume that ASes 1, 2, and 3 are always activated together. The first activation leads ASes 1, 2, and 3 to select routes (0), (0), and $(4,5,0)$, respectively. On the next activation, they switch to $(2,0)$, $(3,4,5,0)$, and $(1,0)$, and the process repeat indefinitely. This system would be safe if it followed Guideline A by requiring AS 3 to favor the customer route $(4,5,0)$ over the peer route $(1,0)$.

THEOREM 5.2. *For a BGP system that has only customer-provider and peer-to-peer relationships and conforms to Assumption P, if all ASes follow guideline B, then the BGP system is safe.*

PROOF: We prove the theorem by demonstrating that the BGP system has a stable state and converges to the stable state for any initial state and any fair activation sequence. Since the second part is similar to Theorem 5.1, we concentrate on proving that the BGP system has a stable state.

Similar to Lemma 5.1, we construct a two-phase activation sequence that leads to a stable state. We activate all ASes in a linear order that conforms to the partial order in the customer-to-provider DAG in Phase 1. We impose additional constraints on the order of AS activations in Phase 1 based on the peer-to-peer relationships and the AS-path length. Therefore, BGP speakers get their customer and peer routes in Phase 1. The BGP speakers that do not get a route in Phase 1 then select a route from a peer or a provider. Therefore, in Phase 2, ASes are activated in an order that conforms to the partial order given in the provider-to-customer DAG. Formally, we have a two-phase activation sequence σ^* as follows.

Phase 1: Activate ASes in a linear order that conforms to the partial order in the customer-to-provider DAG. When the partial order allows more than one AS to be activated next, activate an AS whose peers have all been activated, breaking ties arbitrarily. When all of the ASes have at least one peer that has not been activated, activate an AS only if all of the Phase-1 customers of its peers have been activated. If more than one such AS exists, activate the AS who has the shortest AS path among its customer routes (minimum length among all paths learned from customers).
Phase 2: Activate ASes in a linear order that conforms to the partial order in the provider-to-customer DAG.

Note that we have the same Phase 2 as in Theorem 5.1. Our proof of the stability of Phase-2 ASes follows the same argument. Therefore, we concentrate on Phase-1 ASes. In Phase 1, we impose additional order on ASes so that an AS is activated only if all of its peers are stable or the routes of its unstable peers would not affect the routing decision. The order conforms to the length of the shortest customer route. Since a peer route never has a *larger* local-pref than a customer route, an AS never selects a peer route over a customer route with a *shorter* AS path. Hence, this additional restriction on activation order ensures that a Phase-1 AS is stable after its activation, following a similar argument as in Lemma 5.1. Next, we prove by contradiction that all ASes are activated in Phase 1.

Assume that u is a (direct or indirect) provider of AS_d and never became eligible for activation in Phase 1. Then, either one of u's customers was not activated, or one of the customers of u's peers was not activated. Without loss of generality, assume that u has a customer u_1 that is a (direct or indirect) provider of AS_d that was not activated. Similarly, u_1 either has a customer that was not activated, or one of the customers of u_1's peers was not activated. Without loss of generality, assume that u_1 has a peer with a customer u_2 that was not activated. This process cannot repeat indefinitely without encountering some AS at steps i and j (i.e., $u_i = u_j$), since AS_d has a finite number of (direct or indirect) providers. However, this is a contradiction, since it implies that Assumption P has been violated. Therefore, Phase 1 activates all AS_d's (direct or indirect) providers. ■

5.2 BGP Systems with Backup Links

Customer-provider and peer-to-peer are the two most common relationships between two ASes. However, an AS may also have a backup relationship with a neighboring AS. Having a backup relationship with a neighbor is important when an AS has limited connectivity to the rest of the Internet. For example, ASes A and B could establish a bilateral backup agreement for providing the connection to the Internet in the case that one AS' link to providers fails. AS C is a provider of AS A and AS D is a provider of AS B. ASes C and D have a peer-to-peer agreement. Typically, A reaches others via C and B reaches others via D. If the link between A and C (or B and D) fails, the backup link between A and B is used for A (or B) to connect to the Internet. Initially, we assume that an AS pair cannot have both a backup relationship and a customer-provider or peer-to-peer arrangement; we relax this assumption in Section 6.2.

Backup links are not meant to be used unless a failure occurs. Hence, routes involving backup links should have a *lower* local-pref than other routes. Note that a route through a backup link is a route that contains one or more backup links — the backup link does *not* have to be first hop. Formally, we have Guideline C for each BGP speaker:

Guideline C

if ((r_1 does not contain a backup link) and
(r_2 does not contain a backup link))
follow Guideline A or B to assign $r_1.loc\_pref$
if ((r_1 contains a backup link) and
(r_2 does not contain a backup link))
$r_1.loc\_pref < r_2.loc\_pref$

Note that, unlike Guideline A or B, enforcing Guideline C requires cooperation between ASes. An AS can not tell which routes involve backup links between other AS pairs. Hence, the BGP advertisements must identify these routes. This is typically achieved using the community attribute (c_set). Providers and customers agree on a community number that indicates which routes includes a backup link [3]. When the customer sends the provider a backup route, it assigns the community number to the route so that the provider can assign an appropriate loc_pref. See [2] for an example of the configuration specified using Routing Policy Specification Language (RPSL). Now, we prove that Guideline C ensures that the BGP system is safe.

THEOREM 5.3. *If all ASes follow guideline C in setting up their policies, then the BGP system is safe.*

PROOF: We prove the theorem for the case that all non-backup routes follow Guideline A. The similar argument follows for the case that all backup routes follow Guideline B. Let AS_d denote the AS that originates the destination prefix d. We construct an activation sequence that leads the BGP system to a stable state. We then prove that the system always converges to the stable state. The activation sequence first propagates routes using customer-provider and peer-to-peer links, and then propagates routes using backup links. There are five phases; the first two phases are the same as in Theorem 5.1, the third and fourth phase propagates routes that use AS_d's backup link, and the fifth phase activates the ASes that have not gotten a route and have a backup link. Formally, we construct an activation sequence σ^* that leads to a stable state. The activation sequence activates the BGP speakers in each AS simultaneously.

Phase 1: activate ASes in a linear order that conforms to the partial order given in the customer-to-provider DAG.
Phase 2: activate ASes in a linear order that conforms to the partial order given in provider-to-customer DAG.
Phase 3: activate the ASes that have a backup link with AS_d and have not gotten a route in the first two phases, and these ASes' (direct and indirect) providers. The order the activation of the providers conforms to the customer-to-provider DAG.
Phase 4: activate the ASes that have not gotten a route in the first three phases in the order that conforms to the provider-to-customer DAG.
Phase 5: activate the ASes that have not gotten a route to d in the first four phases and have a backup link. Since these ASes do not provide transit service, they can be activated in an arbitrary order.

Using the same argument as in Theorem 5.1, the first two phases ensure a stable state for all ASes that have a route to d without using a backup link. Similarly, in Phase 3 and 4, all ASes that do not get a route in the first two phases and have a route via AS_d backup link reach a stable state. Finally, in Phase 5, the remaining ASes that have a route via their respective backup link get to a stable state. All of these can be proven by induction, as in Theorem 5.1.

Note that in Theorem 5.1, the activation sequence gives a linear order of ASes. Using the same argument, we can prove that the BGP system converges to the stable state for any fair activation sequence. ∎

6. PRACTICAL IMPLICATIONS

In this section, we discuss the applicability of our guidelines to diverse and changing network topologies and routing policies. Then, we demonstrate how our methodology can be applied to more complex relationships between ASes, and describe how an AS pair can transition to a new relationship without disrupting system stability.

6.1 Robustness of the Guidelines

The network topology and routing policies are very dynamic in today's rapidly growing Internet. Router and link failures, and the deployment of additional network equipment, result in frequent changes to the underlying topology. ISPs often fine-tune their policy configurations to adapt to fluctuations in traffic demands and changes in their internal topology and connections to neighboring ASes. In addition, ASes periodically change their relationships by adding or removing customers, peers, or providers. Our guidelines ensure the stability of the BGP system even in this dynamic environment. Although these changes may trigger the exchange of new routing information, and may ultimately result in new routing decisions, the partial ordering among routes to each destination ensures that the system reaches a stable state. Alternate approaches [9] that establish convergence properties by performing a check on the topology and policy configurations would have to reconfirm these properties, with no guarantee that the new BGP system would be safe.

Similar to earlier work on BGP convergence properties [9; 10; 20], our guidelines focus on the application of local-pref to prefer some routes over alternatives with a shorter AS path. Since our work aims to prove *positive* results about the stability of the resulting BGP system, it is important to consider the impact of other BGP attributes and the possibility of an AS having multiple BGP speakers. The model in Section 3, and the proofs of the theorems in Section 5, allow each AS to have one or more BGP speakers. Speakers within the same AS do not necessarily choose the same route. The ultimate routing decision may also depend on AS path length (including paths with AS prepending), multiple exit discriminators, and cost information from the intradomain routing protocol. BGP speakers consider these attributes *after* applying local-pref to the routes learned from neighboring ASes. As such, these additional attributes only impact selection of routes *within* a preference class. For example, AS path length may determine *which* customer route is chosen but would not cause a BGP speaker to pick a provider route over a customer route.

6.2 Complex AS Relationships

As presented in Section 4, the hierarchical relationships apply at the level of AS pairs. That is, the discussion implicitly assumes that an AS pair has a customer-provider or peer-to-peer relationship for *all* destination prefixes. Since the path selection process proceeds independently for each prefix, this restriction is not actually necessary. In fact, allowing an AS pair to have their relationship depend on the destination prefix is important for expressing more complex policies. For example, two ASes may have both a peer-to-peer and a backup relationship, where each AS provides backup connectivity to the rest of the Internet in the event of a failure. This arrangement does *not* violate our guidelines, since the relationship is still uniquely defined for each destination prefix. The ASes have a peer-to-peer relationship for any prefixes belonging to either AS, and a backup relationship for all other prefixes. The ASes would need to use different ranges of local-pref values based on whether the routes were learned from customers or from providers and other peers.

Similarly, an AS may act as an intermediary between two ASes that would like to establish a peer-to-peer relationship. For example, consider two ASes u and w that would like to have a peer-to-peer arrangement. Suppose that u and w do not have dedicated connections to each other, but that they each have a peer-to-peer relationship with AS v. Normally, an AS would not advertise routes learned from one peer to another peer. But, AS v can agree to export routes learned from u to w (and routes learned from w to u). That is, routes r with $first(r.as\_path) = u$ would be exported to w, and routes with $first(r.as\_path) = w$ would be exported to u. AS v would not export these routes to any of its other peers or providers. This arrangement obeys our guidelines. AS v acts as a provider for u for routes to and from w (and as a provider for w for routes to and from u), and as a peer for all other routes. Hence, guideline A ensures the stability of the resulting BGP system. We believe that a similar approach can be used to analyze other potential relationships between ASes.

6.3 Changing AS Relationships

Over time, an AS may change the nature of its relationships with its neighbors. For example, a customer may grow large enough to renegotiate its relationship with a provider, and the AS pair may transition to a peer-to-peer relationship. As part of evolving to a new relationship, the two ASes may need to change their import and export policies. Ideally, these changes would occur simultaneously. However, in practice, each AS configures its routers independently of the other. As a result, the BGP system may go through a transition period where one AS has changed its configuration and the other has not. Since these changes occur on a human time scale, it is important to carefully study the influence of the transition period on system stability. Our methodology can be used to identify potential convergence problems, and to determine which AS should change its configuration first. We focus the discussion on a BGP system that obeys guideline A. Similar arguments apply under the other guidelines.

For example, consider a customer u and a provider v that transition to a peer-to-peer relationship. Each AS may change its configuration while remaining consistent with guideline A. AS u does not need to change its export policies since v remains in $provider(u) \cup peer(u)$. Similarly, guideline A does not require u to change its import policies. AS u may in fact modify its local-pref value for routes learned from v, but differences in local-pref within a preference class do not affect system stability. AS u does not need to coordinate with v in making these changes. In contrast, AS v needs to change its import and export policies. AS v stops exporting routes learned from its providers and peers. In addition, the import policy must apply a smaller local-pref to treat u as a peer, rather than a customer. This removes an edge in the provider-to-customer graph. Since removing an edge cannot introduce a cycle, the resulting graph is still a DAG.

Next, we consider a change in the opposite direction, from a peer-to-peer to a customer-provider relationship, where u is the customer and v is the provider. We assume that the final customer-provider configuration does not violate the hierarchy in the AS graph; that is, the final customer-to-provider and provider-to-customer graphs are DAGs. As in the previous example, AS u does not need to change its import and export policies. Hence, u does not need to coordinate with v. AS v changes its export policies to advertise routes learned from other providers and its peers. In addition, v changes its import policies to apply a higher local preference to routes learned from u. Since the changes are isolated to AS v, the BGP system remains safe. Stability problems may arise if multiple ASes transition from peer-to-peer to customer-provider relationships, if the resulting AS graph does not retain its hierarchical structure. A routing registry could be consulted as each provider changes its configuration, and can flag proposed changes that would violate the hierarchical structure.

The transition is more complicated when a customer-provider relationship changes to a provider-customer relationship. This situation is extremely unlikely to happen in practice, and could be handled by performing two separate transitions from customer-provider to peer-to-peer, and from peer-to-peer to provider-customer. But, for the sake of completeness, we show how the AS pair can directly transition from customer-provider to provider-customer. Initially, u is the customer and v is the provider. Again, we assume that the final configuration does not violate our assumptions of a hierarchical relationship between ASes. We also assume that at most one AS pair changes its relationship at a time. Applying our methodology, we can show that the provider v should change its configuration first. For example, suppose that u changes its configuration first. Then, during the transition period, u sees v as a provider and v sees u as a provider. This introduces two problems. First, there is a cycle in the provider-to-customer graph. Second, both ASes export all routes to each other. The resulting BGP system may not be safe. For example, the two ASes are vulnerable to the scenario in Figure 2.

Instead, suppose that u changes its configuration first. This removes an edge from the customer-to-provider graph and adds an edge to the provider-to-customer graph. Although the resulting provider-to-customer graph has a cycle, we can show that the BGP system is still safe during this transition period. The provider-to-customer graph has exactly one cycle — the cycle between u and v, since each AS considers the other as a provider. Consider a particular destination prefix d. We consider two cases depending on whether or not one (or both) of the ASes has a customer route to d. Without loss of generality, assume that AS u has a customer route to d. Then, applying guideline A, u would prefer this route over any route via v. Hence, the decision made by v has no influence on u, and the system is safe. In the second case, assume that neither AS has a customer route to d. Then, both u and v must select from routes learned from providers and peer routes. Neither u nor v would export such a route to each other, since a customer does not tell a provider about routes learned from peers or from other providers. Hence, the decision made by each AS does not affect the other, and the BGP system is safe. As such, our methodology demonstrates that the provider v should change its configuration first.

7. CONCLUSIONS

In this paper, we present a detailed model of BGP, along with a set of guidelines for ASes to apply in configuring their BGP import policies. These guidelines capitalize on the commercial relationships between ASes, and provably guarantee route convergence for all possible initial states without requiring global coordination. As part of ongoing work, we are investigating how ASes can verify conformity with our proposed guidelines. Since router configuration files are typically managed by humans, the stability properties can be compromised by human errors. We propose to use the route registry that contains the hierarchical interconnection structure of ASes to check for consistency. For example, export policies should ensure that no AS path has a provider-to-customer link followed by either a customer-to-provider or peer-to-peer link. Using IRR database, each ISP can verify the validity of a route announcement. The verification can be done statically by periodically checking routing updates or routing table entries; upon identifying an invalid route, the offending AS can be notified. In addition, an AS' router configuration files can be checked to ensure that local-pref values are consistent with the desired relationship with the neighboring AS (and the associated export policies). The iBGP configuration can be checked to ensure that techniques for reducing protocol traffic do not affect the routing decisions.

Acknowledgments

We would like to thank Anja Feldmann and Tim Griffin for their comments on earlier drafts of this paper.

8. REFERENCES

[1] C. Alaettinoglu. Scalable router configuration for the Internet. In *Proc. IEEE IC3N*, October 1996.

[2] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra. Routing Policy Specification Language (RPSL). Request for Comments 2622, June 1999.

[3] E. Chen and T. Bates. An Application of the BGP Community Attribute in Multi-home Routing. Request for Comments 1998, August 1996.

[4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. McGrawHill, 1990.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the Internet topology. In *Proc. ACM SIGCOMM*, August 1999.

[6] R. Govindan, C. Alaettinoglu, G. Eddy, D. Kessens, S. Kumar, and W. Lee. An architecture for stable, analyzable Internet routing. *IEEE Network Magazine*, January/February 1999.

[7] R. Govindan and A. Reddy. An analysis of Internet inter-domain topology and route stability. In *Proc. IEEE INFOCOM*, April 1997.

[8] T. Griffin and G. Wilfong. A safe path vector protocol. In *Proc. IEEE INFOCOM*, March 2000.

[9] T. G. Griffin, F. B. Shepherd, and G. Wilfong. Policy disputes in path-vector protocols. In *Proc. Inter. Conf. on Network Protocols*, November 1999.

[10] T. G. Griffin and G. Wilfong. An analysis of BGP convergence properties. In *Proc. ACM SIGCOMM*, September 1999.

[11] B. Halabi. *Internet Routing Architectures*. Cisco Press, 1997.

[12] G. Huston. Interconnection, peering, and settlements. In *Proc. INET*, June 1999.

[13] 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.

[14] C. Labovitz, R. Malan, and F. Jahanian. Internet routing stability. *IEEE/ACM Trans. Networking*, 6(5):515–528, October 1998.

[15] C. Labovitz, R. Malan, and F. Jahanian. Origins of pathological Internet routing instability. In *Proc. IEEE INFOCOM*, March 1999.

[16] D. Meyer, J. Schmitz, C. Orange, M. Prior, and C. Alaettinoglu. Using RPSL in Practice. Request for Comments 2650, August 1999.

[17] National Laboratory for Applied Network Research, Routing Data (`http://moat.nlanr.net/AS/`).

[18] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). Request for Comments 1771, March 1995.

[19] J. W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1999.

[20] K. Varadhan, R. Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. Technical Report 96-631, USC/ISI, February 1996.