Stable Egress Route Selection for Interdomain Traffic Engineering: Model and Analysis*

Hao Wang[†]   Haiyong Xie[†]   Yang Richard Yang[†]
Li Erran Li[‡]   Yanbin Liu[§]   Avi Silberschatz[†]

[†]Computer Science Department, Yale University, New Haven, CT 06520
[‡]Networking Research Lab, Bell-labs, Murray Hill, NJ 07974
[§]Department of Computer Sciences, The University of Texas at Austin, Austin, TX 78712

Abstract

We present a general model of interdomain route selection to study interdomain traffic engineering. In this model, the routing of multiple destinations can be coordinated. Thus the model can capture general traffic engineering behaviors such as load balancing and link capacity constraints. We first identify potential routing instability and inefficiency of interdomain traffic engineering. We then derive a sufficient condition to guarantee convergence. We also show that the constraints on local policies imposed by business considerations in the Internet can guarantee stability without global coordination. Using realistic Internet topology, we evaluate the extent to which routing instability of interdomain traffic engineering can happen when the constraints are violated.

1. Introduction

The global Internet consists of a large number of interconnected autonomous systems (ASes), where each AS is administrated autonomously. Recently, ASes are increasingly adopting local route selection policies to achieve their interdomain traffic engineering objectives (*e.g.*, [37]). We have recently conducted an email survey of ISPs, and the results indicate that many ISPs choose routes to achieve their interdomain traffic engineering objectives, such as satisfying the capacity constraints of links between neighboring ASes (*e.g.*, [4]), load-balancing interdomain traffic, and/or minimizing cost (*e.g.*, [21]).

Despite this emerging trend, so far there are few systematic studies on the stability and efficiency of the global Internet with route selection for interdomain traffic engineering. As several researchers pointed out [8, 37]: "the state of the art for interdomain traffic engineering is extremely primitive." Learning anecdotal incidents causing instability in the Internet (*e.g.*, [34]) and recognizing the poten-

tial issues of using route selection for interdomain traffic engineering, researchers have proposed both configuration guidelines (*e.g.*, [8, 37]) and alternatives/extensions to the current interdomain routing protocol (*e.g.*, [1,34,41]). However, since the essential features of route selection for interdomain traffic engineering have not been pinpointed and analyzed [7], it is unclear whether these guidelines and new protocols can produce stable and efficient route selections in the global Internet.

A major breakthrough was made recently when researchers [20, 23, 24, 27, 39] proposed systematic models to study the stability of path-vector interdomain routing. In particular, Griffin *et al.* identified the existence of policy disputes as a potential reason for routing instability. By routing instability, they mean persistent route oscillations even though the network topology is stable.

Although these previous models can already capture a wide range of potential route selection behaviors for interdomain traffic engineering, since they require that the routing decisions of different destinations be separated, they cannot be applied to study a large class of common traffic engineering behaviors. In particular, a fundamental feature of route selection for interdomain traffic engineering in particular and traffic engineering in general is that route selection constraints (*e.g.*, traffic assigned to a link is within link capacity) and/or objective functions (*e.g.*, load balancing) involve the route selection of multiple destinations. Thus, in route selection for interdomain traffic engineering, whether a route will be chosen by an AS for a given destination will depend on what routes are available or chosen for other destinations. For example, if an AS selects routes for each destination independently without considering the chosen/available routes of other destinations, in the worst case it may choose the same access link for all destinations, violating link capacity constraints and/or causing load imbalance. By requiring that the routing of each destination be separated, the previous models apply only to a network where there is no AS whose routing policies require it to coordinate its route selection to multiple destinations.

In this paper, we first identify that there exist networks where the coordination of the route selection of multiple destinations due to interdomain traffic engineering considerations can cause routing instability, even though the net-

works are guaranteed to converge for each destination when each AS uses a *projected consistent ranking table* (projected from the joint ranking table for multiple destinations with lower-ranked duplicate entries removed). The identification of such routing instability shows that a general route selection model is needed to analyze the stability of route selection for interdomain traffic engineering. Motivated by this need, we propose a route selection model where each AS can partition the destinations into a set of disjoint subsets, and route selection for the destinations in each subset is coordinated, while route selection for destinations in different subsets is independent. This model is very general and is the first general model which captures the essence of route selection for interdomain traffic engineering.

Using this model, we first analyze the stability of path-vector interdomain routing when ASes choose egress routes to achieve interdomain traffic engineering objectives. We call this problem *the stable route selection for egress interdomain traffic engineering problem*. We propose the construction of *P-graph*s, and derive sufficient conditions based on P-graphs to guarantee convergence of route selections for interdomain traffic engineering.

We also investigate the efficiency of route selection for interdomain traffic engineering. We show an example with multiple stable route selections but one of them is not Pareto optimal. As a contrast, when each destination is routed independently, BGP produces Pareto optimal routing solutions [5]. These results clearly demonstrate the intrinsic challenges of route selection for interdomain traffic engineering in a generic network. It will be challenging to achieve stable and efficient outcomes for general networks even when ASes adopt explicit negotiations.

The route selection of Internet has its own special properties. Applying our general results, we investigate whether route selection for interdomain traffic engineering can lead to routing instability in Internet-like environments. We prove that, if there is no *provider-customer loop* in the network, each AS follows the *typical* export policy, and AS ranking of routes follows the *standard joint-route preference policy*, then the convergence and uniqueness of route selection for egress interdomain traffic engineering can be guaranteed. This result is particularly pleasant and somehow surprising in that the conditions of the result are highly likely to be satisfied in the current Internet due to the ISP economy of the current Internet.

We complement the preceding analysis with extensive simulations to investigate the likelihood of instability when the three conditions are violated (*e.g.*, when some ASes give non-economic considerations higher priority over economic considerations). Specifically, we use current Internet BGP routing tables to infer the AS-level topology and AS business relationships. We then conduct simulations using the inferred Internet topology. We show that even with a small number of ASes coordinating route selection for just a small number of destinations, we can observe instability.

The rest of this paper is organized as follows. In Section 2, we discuss related work. In Section 3, we study route selection for egress interdomain traffic engineering. In Section 4, we show that the constraints imposed by Internet business considerations lead to unique stable egress route selection for interdomain traffic engineering. In Section 5, we present evaluations of route selection for interdomain traffic engineering. Our conclusion and future work are in Section 6.

2. Related Work

There is a large body of literature on interdomain route selection where each destination is considered separately. In particular, researchers have conducted extensive evaluations (*e.g.*, [15,22,29,30,45]) and theoretical analysis (*e.g.*, [9, 23, 26, 27, 39]) on the stability of BGP route selection. Griffin *et al.* [24] show that "policy disputes" can cause persistent route oscillations. Griffin and Wilfong [25] then propose a protocol called SPVP3 that can detect oscillations caused by policy disputes at run time using "path history." SPVP3 is guaranteed to converge if routes whose path history contain cycles are suppressed. Feamster *et al.* [9] study routing systems with ranking independence and unrestricted filtering; they show that any routing system that has a "dispute ring", a special type of dispute wheel, is not safe under unrestricted filtering and that ASes are essentially required to rank routes based on AS-path lengths in order to guarantee convergence with ranking independence and unrestricted filtering. Gao and Rexford [19,20] observe that, if every AS considers each of its neighbors as either a customer, a provider, or a peer, and obeys certain local constraints on preferences and export policies, then BGP is guaranteed to converge. Generalizing the above commercial relationships of ISPs to a class-based system, Jaggard and Ramachandran [26] show that a global constraint that guarantees convergence can be enforced by a distributed algorithm. The major difference between our model and the previous studies is that the previous studies consider only a network where there is no AS whose routing policies require it to coordinate the route selection of multiple destinations. Thus the route for each destination can be chosen regardless of the chosen/available routes of other destinations. As a result, the routing decisions for the destinations can be separated. In this paper, we investigate the effects of the coordination of route selection among multiple destinations, which is an essential feature of interdomain traffic engineering that has been missing in previous studies.

Traffic engineering has traditionally been focused on intra-domain (for a good survey, please see [16, 17]). There is an increasing interest in tuning BGP attributes for interdomain traffic engineering [37]. However, most of the previous work focuses on the configuration of either a single AS (*e.g.*, [4, 10, 21]) or between two neighboring ASes. In particular, researchers have conducted extensive theoretical analysis (*e.g.*, [28]) and experimental evaluations (*e.g.*, [42, 43]) of hot-potato routing, which is a scheme of exit route selection between two ASes. Recognizing the potential unpredictable nature of interdomain BGP traffic engineering involving multiple ASes, Feamster *et al.* [8] propose guidelines to restrict route selection so that its impact on the traffic flow is predictable.

There is another line of research that proposes extensions/alternatives to BGP (*e.g.*, the mechanism-design ap-

proach by Feigenbaum *et al.* [11–13], the negotiation protocol by Mahajan *et al.* [33–35], the BGP pricing approach by Afergan and Wroclawski [1], the Hybrid Link-state Path-vector (HLP) approach by Subramanian *et al.* [41]). To assess the applicability and effectiveness of these new solutions to interdomain traffic engineering, we need to understand the intrinsic problems of route selection for interdomain traffic engineering. The objective of this paper is to pinpoint these problems; thus it can serve as a motivation for the initiation of these studies. It could also provide new insight to these studies. For example, we will show that the solutions may not be Pareto optimal if negotiation happens only between two neighboring ASes; this indicates that, for efficient route selection, current proposals of negotiation protocols (*e.g.*, [33]) need to be extended to handle much more general settings.

3. Route Selection for Egress Interdomain Traffic Engineering

3.1. Motivation

As we pointed out in Section 1, major ISPs are already coordinating the route selection of multiple destinations in their interdomain route selection. A very simple illustrative example is shown in Figure 1.
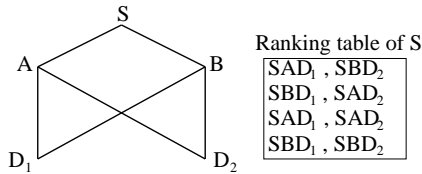
Figure 1. Egress load balancing: an example motivating the need for coordinated route selection.

In this example, the majority of the traffic of S goes to two destinations D_1 and D_2. Assume S wants to balance its outgoing traffic. Thus, it wants to choose a combination of routes for destinations D_1 and D_2 such that they use different neighbors, if possible, to have low utilization on the two links SA and SB. We refer to a combination of routes for D_1 and D_2 as a *route profile*. Since S may not know in advance the routes it will learn from its neighbors A and B, or the routes that A and B will export to S can be dynamic given network dynamics, S needs an automatic method to pick the best route profile from currently available routes.

One method that an AS can specify its preference is to use a policy language. An example policy can be: if D_1 and D_2 use different links, assign a base local preference of 100; otherwise, a base local preference of 0. If D_1 uses link SA, add 10 to local preference. If D_2 uses link SB, add 5 to local preference. A traffic engineering program then picks the available route profile with the highest preference.

Another method that an AS can specify its preference is to define an interdomain traffic engineering objective function (*e.g.*, minimize the maximum of the utilization of the

two links for this case). An advantage of using an objective function is its compact representation. Given the objective function, link capacities and traffic demands, a traffic engineering program computes the best route profile from currently available routes. Note that computing an optimal solution might be computationally expensive in some cases (*e.g.*, when the underlying problem is NP-hard). To reduce complexity, an AS might use approximation algorithms. Another method to reduce complexity is to consider the coordination of only a small set of important destination prefixes (*e.g.*, the "elephants" [14, 36, 44]). The routing of the remaining destination prefixes then will be handled individually.

To capture the essence of various methods of implementing route selection for interdomain traffic engineering, *for the purpose of analysis*, we assume that each AS partitions the destinations into arbitrary subsets. For each subset, the AS has a ranking table[1], which lists, in decreasing order, all of the potential route profiles of the destinations in the subset. Consider again the example at the beginning of this section. AS S coordinates the route selection of destinations D_1 and D_2. S may rank different route profiles according to the maximum of the utilization of the two access links. Its ranking can be represented, for the purpose of analysis, by a ranking table as shown in Figure 1, where each row is a potential route profile (*i.e.*, a combination of routes for D_1 and D_2). For example, the best route profile for S is (SAD_1, SAD_2); that is, S uses SAD_1 for destination D_1, and SAD_2 for destination D_2. The worst route profile is SBD_1 and SBD_2. Thus, if the route profile (SAD_1, SAD_2) is available, S will choose it. On the other hand, if the only available route profile is (SBD_1, SBD_2), S has no other choice but to use it.

3.2. Problem Formulation

We first state the assumptions we shall make in this section. We assume a connected network with the underlying infrastructure being stable so that we can focus on the effects of interdomain traffic engineering policies. We assume that there is only one link between two neighboring ASes; that is, we consider AS level routing only[2]. We assume that each AS has a static export policy (*e.g.*, dictated by business contracts or common practice). For scalability, each AS may coordinate the route selection of only a subset of its destinations (*e.g.*, the "elephants" [14, 36, 44]). More generally, an AS can have a set of disjoint subsets of destinations, and route selection of each subset is coordinated, while route selection of different subsets is independent. Each AS chooses the best available routes in order to achieve its own interdomain traffic engineering objectives. We assume that, the preference of an AS depends only on

---

[1] The existence of a ranking table is conceptual and for the purpose of analysis. If the route selection behavior of an AS is consistent when faced with different sets of available routes, a ranking table can be constructed accordingly.

[2] In the more general case, different border gateway routers in the same AS might choose different paths of ASes to reach the same destination. As in previous work, we do not consider this scenario in this paper and leave it as future work.

the route from the AS itself to the destinations. In other words, the ASes are conducting *egress interdomain traffic engineering*, which is one of the major tasks of ISP interdomain traffic engineering [7]. Note that we can further extend this model and study route selection for general interdomain traffic engineering, in which case the route from each source to the AS itself also matters. For initial results, please see [47]. Note also that in a more general case, the preference of an AS on a route may also depend on routes that do not pass through the AS itself. For example, these routes may share common links with the route chosen by this AS and thus cause congestion. We do not consider this problem and leave it to the study of the general congestion game [6].

Now we formally define the stable route selection for egress interdomain traffic engineering problem. Before defining stable route selection for egress interdomain traffic engineering, we first introduce some basic concepts and define our BGP system model and protocol/process model.

The network topology is represented by a simple undirected graph $G = (V, E)$, where $V = \{1, \ldots, N\}$ is the set of ASes and E the set of interdomain links.

A path P in G is either the empty path, denoted by ϵ, or a sequence of ASes $(v_k, \ldots, v_1, v_0)$, where $k \geq 0$ is the length of the path, such that $(v_i, v_{i-1}) \in E, \forall i = k, k-1, \ldots, 1$. When $k = 0$, $P = (v_0)$ represents the trivial path from AS v_0 to itself. Each nonempty path P has a direction from v_k to v_0, and $P[v_i, v_j]$ denotes the subpath of P from v_i to v_j, $\forall k \geq i > j \geq 1$. If P and Q are two nonempty paths such that the first AS in Q is the same as the last AS in P, then PQ denotes the path formed by the *concatenation* of these two paths.

We denote by $R_{i\rightarrow}$ the set of paths originating from AS i, and $R_{\rightarrow i}$ the set of paths terminating at AS i. Also, $R_{i\rightarrow j} = R_{i\rightarrow} \cap R_{\rightarrow j}$ denotes the set of paths from AS i to j.

Suppose i and j are two neighboring ASes. As a path P is exported from j and imported into i, it undergoes two transformations. First, $P_1 = \text{export}(i, j, P)$ represents the application of export policies of j to P, which includes possibly prepending j multiple times to P or filtering out P altogether. Second, $P_2 = \text{import}(i, j, P_1)$ represents the application of import policies of i to P_1. In particular, import policies at i will filter out any path that contains i itself. The collective effects of these transformations can be represented by the *peering transformation*, $\text{pt}(i, j, P)$, defined as

$$\text{pt}(i, j, P) = \text{import}(i, j, \text{export}(i, j, P)).$$

The peering transformation represents the import/export policies of all ASes in the network.

Each AS i attempts to establish a path to each destination in a given set $\mathcal{D}_i$. A *network route selection* is a function r that maps each pair of ASes $i \in V$ and $j \in \mathcal{D}_i$ to a path $r(i, j) \in R_{i\rightarrow j}$. We interpret $r(i, j) = \epsilon$ to mean that i is not assigned a path to j. We refer to an element in the product space $\Pi_{j \in \mathcal{D}_i} R_{i\rightarrow j}$ as a *route profile* of AS i, denoted by r_i, which consists of a profile of routes to all destinations in $\mathcal{D}_i$. When r_i consists of only routes to a subset $\mathcal{D} \subseteq \mathcal{D}_i$ of destinations, we call it a *partial route*

profile, denoted by $r_i^{\mathcal{D}}$. We denote by $r_i^{\mathcal{D}}(j)$ the path to destination j ($\in \mathcal{D}$) available in the partial route profile $r_i^{\mathcal{D}}$. Furthermore, we denote by

$$\mathcal{R}_i^{\mathcal{D}}(\mathcal{P}) = \{r_i^{\mathcal{D}} | r_i^{\mathcal{D}}(j) \in \mathcal{P}_{i\rightarrow j}, \forall j \in \mathcal{D}\}$$

the set of all possible partial route profiles for AS i with paths to destinations in $\mathcal{D}$ drawn from a set $\mathcal{P}$ of available paths.

For the purpose of traffic engineering, ASes would like to coordinate their route selection. A general and reasonable approach for an AS i to coordinate route selection is to partition the set of destinations, $\mathcal{D}_i$, into a family of N_i disjoint subsets $\mathcal{D}_{ik}$, where $k = 1, \ldots, N_i$. For each subset $\mathcal{D}_{ik}$, AS i chooses routes jointly for all destinations in $\mathcal{D}_{ik}$. This coordinated route selection for destinations in $\mathcal{D}_{ik}$ can be captured by a *route selection function* $\sigma_i^{\mathcal{D}_{ik}}$, which maps a set of available paths to a partial route profile for destinations in $\mathcal{D}_{ik}$. In this paper, we focus on the model of route selection which can be represented by a linear preference order. Specifically, each AS i has a ranking function $\lambda_i^{\mathcal{D}_{ik}}$ for each $\mathcal{D}_{ik}$, which maps partial route profile to a totally ordered set Λ. Given a set $\mathcal{P}$ of available paths, the route selection function $\sigma_i^{\mathcal{D}_{ik}}$ simply selects the available partial route profile with the highest rank, *i.e.*,

$$\sigma_i^{\mathcal{D}_{ik}}(\mathcal{P}) = \underset{r \in \mathcal{R}_i^{\mathcal{D}_{ik}}(\mathcal{P})}{\arg\max} \ \lambda_i^{\mathcal{D}_{ik}}(r).$$

AS i determines its route profile r_i by selecting a partial route profile for each $\mathcal{D}_{ik}$ *independently*; that is,

$$r_i = \sigma_i(\mathcal{P}), \text{ such that } r_i^{\mathcal{D}_{ik}} = \sigma_i^{\mathcal{D}_{ik}}(\mathcal{P}), \forall k = 1, \ldots, N_i.$$

We emphasize that the ranking functions $\lambda_i^{\mathcal{D}_{ik}}$ are just general representations of some more compact representations such as objective functions or policy languages.

To summarize the preceding definitions, we denote a *BGP system* by a quintuple $S = (G, \text{pt}, \sigma, \mathbb{D}, \widetilde{\mathbb{P}})$, where $G = (V, E)$ is the topology of a network, pt is a peering transformation defined on G, $\mathbb{D} = \{\mathcal{D}_i | i \in V\}$, σ_i is the route selection function of AS i, and $\widetilde{\mathbb{P}} = \{\widetilde{\mathcal{P}}_i | i \in V\}$, where $\widetilde{\mathcal{P}}_i$ is the set of *feasible* paths from i to destinations in $\mathcal{D}_i$.
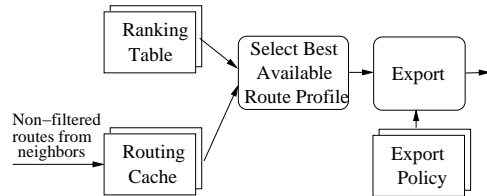
Figure 2. The protocol/process model of route selection for interdomain traffic engineering.

Given our BGP system model, we now define our protocol/process model. Figure 2 shows the standard protocol/process model of interdomain route selection [20, 23, 24,

27, 39], naturally extended to multiple destinations. Specifically, each AS maintains a routing cache A_i of currently available routes exported by its neighbors. AS i selects a route profile r_i from its routing cache A_i using its route selection function σ_i as defined above[3], which will then be used by i to route packets. Sometime we refer to this chosen route profile as the *installed* route profile. If $r_i(j)$ is different from the previously selected route to j, i then withdraws the previous route, and exports the new route to the neighbors that are allowed to receive this route according to i's export policy. We assume that BGP route update messages between neighboring ASes are delivered in FIFO order and reliably. This is reasonable as the messages are sent via TCP. We also assume that each message will be processed in a bounded time.

Given the above description of the protocol/process model of interdomain route selection, we now define the notion of a *stable* network route selection. For a given network route selection r, the set $\texttt{candidates}(i, r)$ consists of all available paths at AS i that can be formed by extending the routes chosen by neighbors of i; that is,

$$\texttt{candidates}(i, r) = \{\texttt{pt}(i, j, r_j(k)) | (i, j) \in E, k \in D_j\}.$$

The network route selection r is *stable* if no AS i can choose a higher ranked route profile from $\texttt{candidates}(i, r)$; formally, r is stable if and only if

$$r_i = \sigma_i(\texttt{candidates}(i, r)), \text{ for all } i \in V.$$

We also call a stable network route selection a *stable route solution* or *solution* for short.

Finally, a network is *robust* if BGP protocol is guaranteed to converge even with arbitrary node/link failures.

### 3.3. Multi-Destination Interactions Can Cause Instability

As we pointed out in Section 1, coordinated route selection of multiple destinations due to interdomain traffic engineering can cause routing instability. Figure 3 is one such interesting example. The export policies of A and B follow the *typical export policies* [18, 20]: 1) each AS exports to its providers only its own routes and those learned from its customers, but not the routes learned from its peers or other providers; 2) each AS exports to its customers all routes it has; 3) each AS exports to its peers its own routes and those it learned from its customers, but not those learned from its providers or other peers.

We first consider each destination separately. Specifically, for each destination, we obtain a ranking of routes from each AS to that destination by projecting the ranking table of the corresponding AS onto the destination. When a route appears more than once in a projected ranking table, we keep the one ranked highest and remove other occurrences with lower ranks so that the ranking table is consistent. We refer to a ranking table derived by applying the
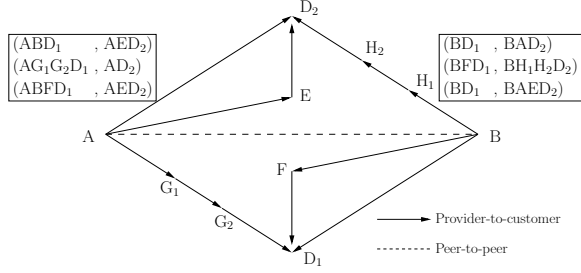
Figure 3. An example network which has no stable route selection. For clarity, only the highest three route profiles of A and B are shown.
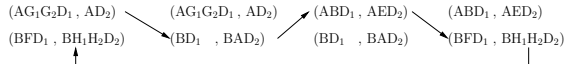
Figure 4. The BGP update process of the network in Figure 3.

preceding projection and pruning operations as a *projected consistent ranking table*.

In the example network in Figure 3, for destination D_1, A has ABD_1 and $AG_1G_2D_1$, and B has BD_1 and BFD_1, respectively, as the two highest route profiles in their projected consistent ranking tables. Consider this combination of route preference for D_1. The network has a stable route solution of ABD_1 and BD_1 for A and B, respectively. One can also verify that if we consider D_2 alone, the network has a stable route solution of AED_2 and $BH_1H_2D_2$ for A and B, respectively. Thus, if there were no interaction among destinations, A and B would settle to the stable solutions of (ABD_1, AED_2) and $(BD_1, BH_1H_2D_2)$, respectively. Note that in the projected consistent ranking tables in this example, there exists no pair of ASes that have peer-provider relationship, and peer routes can have the higher ranking than customer routes[4]. Therefore, both Assumption P and Guideline B in [20] are satisfied in the example, and the network converges on each destination alone.

Next we consider coordinated route selection for both destinations. The above solutions obtained by considering each destination alone are no longer stable. For example, B will not choose $(BD_1, BH_1H_2D_2)$ since this route profile has a lower rank. Note that for clarity, we show only the highest three route profiles of A and B in Figure 3. One can verify that the network has no stable solution at all. Specifically, we observe that the export policies of the ASes make the route profile $(AG_1G_2D_1, AD_2)$ alway available to A. Thus to see that the network has no stable solutions, we just need to verify that there is no stable route solution when A chooses $(AG_1G_2D_1, AD_2)$ or (ABD_1, AED_2). Clearly, there is no stable solution for $(AG_1G_2D_1, AD_2)$ since if A chooses $(AG_1G_2D_1, AD_2)$, B will choose (BD_1, BAD_2); this causes A to change to (ABD_1, AED_2). However,

---

[3]Due to computational complexity, for some formulations of interdomain traffic engineering, it could be the case that only approximate solutions can be obtained. We leave this consideration as future work.

[4]As in [20], a peer route is preferred over a customer route if and only if they have the same local preference and the peer route is shorter in path length.

there will be no stable route selection for (ABD_1, AED_2) neither. To make (ABD_1, AED_2) available to A, B must choose BD_1 for D_1. Since $(BFD_1, BH_1H_2D_2)$ is always available to B, it must be the case that B chooses (BD_1, BAD_2). However, this requires A to choose AD_2, which is inconsistent with (ABD_1, AED_2). Thus, the network has no stable route selections due to destination interaction! Figure 4 shows the BGP update process.

3.4. Stable, Robust Route Selection and Protocol Convergence

Given that multi-destination interactions can result in instability, we next derive a sufficient condition that can guarantee stable, robust route selection and protocol convergence.

3.4.1 Representation of protocol execution

Based on the protocol/process model described in Section 3.2, we adopt the following representation of an arbitrary protocol execution. We assume that the BGP update messages are delivered reliably and in FIFO order, and the protocol is fair [24]. We assume a total ordering of events; that is, we assign a unique index from $T = \{0, 1, 2, \ldots\}$ to each event so that the assignment is consistent with the logical "happen before" relation among events [32]. We have the following three types of events in our system: type 1) send a route update message; type 2) receive a route update message and update the route in the cache that is affected by the route update message; and type 3) select the highest-ranked route profile and install it as the current route profile. For ease of description, we refer to the ordering as *time* from now on. Specifically, when we write time t, we mean the index t assigned to an event in the total ordering. Let $r[t]$ be the network route selection at time t, then an arbitrary execution of the protocol can be represented by a sequence of network route selections, $\{r[t]\}_{t \in T}$.

3.4.2 Self-contained BGP subsystem

A stable network route selection as defined in Section 3.2 is a *network-wide* concept, where the route from any source to any destination is required to be stable. In a large network, however, it may well be the case that some routes have become stable, while others are still oscillating. It is of theoretical and practical interests, therefore, to consider *partial convergence* in a large network.

To capture this intuitive idea of partial convergence, we introduce the notion of a *self-contained BGP subsystem*. A *BGP subsystem* is a BGP system where the set D_i may not contain all of the destinations that AS i attempts to establish a route to. In a BGP subsystem, we restrict our attention to a subset of destinations for AS i, particularly those to which the routes may become stable. AS i may have routes to other destinations, but these routes are not of our interests. We do have a requirement, however, on which destinations and routes can be left out. Intuitively, we wish to omit only those routes that will not be chosen after some finite time.

Formally, the BGP subsystem S is *self-contained* if there exists $\mathcal{P}_i \subseteq \widetilde{\mathcal{P}}_i$ for all $i \in V$, such that

1. there exists t, such that for all $t' > t$ and $i \in V$, $r_i[t'] \in \mathcal{R}_i^{\mathcal{D}_i}(\mathcal{P}_i)$;

2. $\mathcal{P}_i \subseteq \{\text{pt}(i, j, Q) | (i, j) \in E, Q \in \mathcal{P}_j\}$, for all $i \in V$.
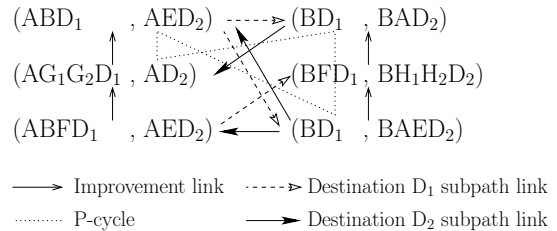
A self-contained BGP subsystem is represented by $S = (G, \text{pt}, \sigma, \mathbb{D}, \widetilde{\mathbb{P}}, \mathbb{P})$, or sometimes $S_{\mathbb{P}}$ for short when the underlying BGP system S is clear from context.

3.4.3 P-graph and P-cycle

We now introduce the notion of a P-graph to capture the interaction of interdomain traffic engineering policies of multiple ASes in a self-contained BGP subsystem $S = (G, \text{pt}, \sigma, \mathbb{D}, \widetilde{\mathbb{P}}, \mathbb{P})$. The notion of a P-graph is motivated by the partial order graph of Griffin *et al.* [23], but generalized to interdomain traffic engineering.

A P-graph is a directed graph constructed as follows. For each AS i and each $\mathcal{D}_{ik}$, there is a node which corresponds to each possible partial route profile $r_i^{\mathcal{D}_{ik}} \in \mathcal{R}_i^{\mathcal{D}_{ik}}(\mathcal{P}_i)$. Note that we do not consider partial profile formed by paths in $\widetilde{\mathcal{P}}_i \setminus \mathcal{P}_i$. There are two types of directed edges in a P-graph. The first type of edges are *improvement edges*. There is an improvement edge from node $\tilde{r}_i^{\mathcal{D}_{ik}}$ to $\hat{r}_i^{\mathcal{D}_{ik}}$ if i prefers $\hat{r}_i^{\mathcal{D}_{ik}}$ to $\tilde{r}_i^{\mathcal{D}_{ik}}$ (*i.e.*, $\lambda_i^{\mathcal{D}_{ik}}(\hat{r}_i^{\mathcal{D}_{ik}}) > \lambda_i^{\mathcal{D}_{ik}}(\tilde{r}_i^{\mathcal{D}_{ik}})$). The second type of edges are *sub-path edges*. There is a destination D sub-path edge from a node $r_i^{\mathcal{D}_{ik}}$ to another node $r_j^{\mathcal{D}_{jl}}$ if the path $r_j^{\mathcal{D}_{jl}}(D)$ from j to D is a sub path of the path $r_i^{\mathcal{D}_{ik}}(D)$ from i to D. Note that in this case $D \in \mathcal{D}_{ik} \cap \mathcal{D}_{jl}$.

A P-cycle is a loop in the P-graph of the following special format: one or more improvement edges, followed by one or more sub-path edges of the same destination, then followed by one or more improvement edges, and so on. For example, Figure 5 shows the P-graph and the P-cycle for the example of Figure 3. Note that there may be trivial loops in a P-graph which are not of the format of a P-cycle. For example, the loop consisting of (BD_1, BAD_2), $(AG_1G_2D_1, AD_2)$ and (ABD_1, AED_2) is not a P-cycle, since there are two consecutive sub-path edges of different destinations.

$(ABD_1 \quad , AED_2) \quad \dashrightarrow \quad (BD_1 \quad , BAD_2)$
$(AG_1G_2D_1 , AD_2) \quad \quad \quad \quad (BFD_1, BH_1H_2D_2)$
$(ABFD_1 \quad , AED_2) \longleftarrow (BD_1 \quad , BAED_2)$

$\longrightarrow$ Improvement link $\dashrightarrow$ Destination D_1 subpath link
$\cdots\cdots$ P-cycle $\longrightarrow$ Destination D_2 subpath link

Figure 5. The P-graph and P-cycle of the network in Figure 3. For clarity, only the highest three route profiles and their improvement links are shown.

3.4.4 BGP protocol convergence

We next apply the notion of P-graph to establish a sufficient condition for BGP protocol convergence. We first state the following lemma:

Lemma 1 *If a self-contained BGP subsystem $S_{\mathbb{P}}$ does not converge, then there is a P-cycle in the corresponding P-graph.*

Proof: As outlined in Section 3.4.1, we represent an arbitrary execution of the protocol by a sequence of network route selections, $\{r[t]\}_{t \in T}$. Let $r_u[t]$ be the route profile of AS u at time t. To simplify notation, in the following proof, we will abbreviate $r_u^{\mathcal{D}_{uk}}$ as r_u^k, and $\lambda_u^{\mathcal{D}_{uk}}$ as λ_u^k. Let $R_u^k[\infty]$ be the set of partial route profiles which u chooses infinitely often for $\mathcal{D}_{uk}$; that is, $R_u^k[\infty] = \cap_t \cup_{t' \geq t} \{r_u^k[t']\}$. There exists t_f such that for any u and any $t > t_f$, $r_u^k[t] \in R_u^k[\infty]$. In other words, after t_f, routes which are chosen only a finite number of times will no longer appear. It follows from condition 1 of a self-contained BGP subsystem that $R_u^k[\infty] \in \mathcal{R}_u^{\mathcal{D}_{uk}}(\mathcal{P}_u)$. If the BGP process does not converge, then there exists a set O of ASes such that for each AS $u \in O$, $|R_u^k[\infty]| \geq 2$ for some k. These are the ASes that have persistent oscillating partial route profiles. Since the set $R_u^k[\infty]$ is finite, we have the following observation:

Proposition 2 *For any $t > t_f$, there exists $t' > t$, such that $\lambda_u^k(r_u^k[t'-1]) > \lambda_u^k(r_u^k[t'])$; that is, u will change from a higher-ranked partial route profile for destinations in $\mathcal{D}_{uk}$ to a lower-ranked one infinitely often.*

We shall construct a P-cycle as follows. We start from an arbitrary $u_0 \in O$. By Proposition 2, there exists k_0 and $t_0 > t_f$, such that $\lambda_{u_0}^{k_0}(r_{u_0}^{k_0}[t_0]) < \lambda_{u_0}^{k_0}(r_{u_0}^{k_0}[t_0-1])$. Thus there is an improvement edge from partial route profile $r_{u_0}^{k_0}[t_0]$ to $r_{u_0}^{k_0}[t_0-1]$.

The only reason for u_0 to change from a higher-ranked partial route profile $r_{u_0}^{k_0}[t_0-1]$ to a lower-ranked partial route profile $r_{u_0}^{k_0}[t_0]$ is that, some time before t_0, a route P to some destination $d \in \mathcal{D}_{u_0 k_0}$ in $r_{u_0}^{k_0}[t_0-1]$ is withdrawn by a BGP update message from u's neighbor v. Let $P[v,d]$ denote the sub-path of P from v to d. Thus there exists some $t_f < t_1 < t_0$ and k such that v processes a type 3 event at time t_1 and changes from a partial route profile $r_v^k[t_1-1]$ containing $P[v,d]$ to a partial route profile $r_v^k[t_1]$ which does not contain $P[v,d]$.

There are two possible reasons for this change of v:

1. AS v ranks $r_v^k[t_1]$ higher than $r_v^k[t_1-1]$. In this case, let $\tilde{r}_v^k = r_v^k[t_1-1]$ and $\hat{r}_v^k = r_v^k[t_1-1]$, we have $\lambda_v^k(\tilde{r}_v^k) < \lambda_v^k(\hat{r}_v^k)$.

2. AS v ranks $r_v^k[t_1]$ lower than $r_v^k[t_1-1]$. There are two sub-cases to consider:

    (a) At time t_1, path $P[v,d]$ is still available to v. In this case, let $\hat{r}_v^k = r_v^k[t_1]$, and let $\tilde{r}_v^k$ be the partial route profile formed by replacing the route to destination d in $r_v^k[t_1]$ with $P[v,d]$. Because

$\tilde{r}_v^k$ is an available route profile to v at time t_1, but v chooses $\hat{r}_v^k$ instead, thus we have $\lambda_v^k(\tilde{r}_v^k) < \lambda_v^k(\hat{r}_v^k)$.

    (b) At time t_1, path $P[v,d]$ is no longer available to v. Let $P[v,d] = (v,w)P[w,d]$, thus v must have received a BGP update message withdrawing $P[w,d]$ from w. In this case, we take w as v, and repeat the argument. Since there are only a finite number of ASes on P, eventually we will come across v' where $P[v',d]$ is still available to v', in which case, we end up with case (2a).

Therefore, we can always find an AS v, a destination $d \in \mathcal{D}_{u_0 k_0} \cap \mathcal{D}_{vk}$, and two partial route profiles $\tilde{r}_v^k$ and $\hat{r}_v^k$, such that $\tilde{r}_v^k(d)$ is a sub-path of $r_{u_0}^{k_0}[t_0-1](d)$, and $\lambda_v^k(\tilde{r}_v^k) < \lambda_v^k(\hat{r}_v^k)$. Because the BGP subsystem is self-contained, the fact that $r_{u_0}^{k_0}[t_0-1] \in \mathcal{R}_{u_0}^{\mathcal{D}_{u_0 k_0}}(\mathcal{P}_{u_0})$ implies that both $\tilde{r}_v^k$ and $\hat{r}_v^k$ must also be in $\mathcal{R}_v^{\mathcal{D}_{vk}}(\mathcal{P}_v)$. Thus, there is a destination d sub-path edge from $r_{u_0}^{k_0}[t_0-1]$ to $\tilde{r}_v^k$, followed by an improvement edge from $\tilde{r}_v^k$ to r_v^k. After time t_1, v may go through zero or more higher-ranked partial route profiles (thus one or more improvement edges in the P-graph). By proposition 2, eventually we will have a time $t_2 > t_1$ such that, $\lambda_v^k(r_v^k[t_2-1]) > \lambda_v^k(r_v^k[t_2])$. Denote this v by u_1. Repeating the above reasoning on u_1's change at time t_2, we can construct a path with alternating improvement edges and sub-path edges in the P-graph. Since the P-graph is a finite graph, eventually we will form a P-cycle. ∎

Lemma 1 immediately leads to the following sufficient condition for convergence in a self-contained BGP subsystem.

Corollary 3 *If the P-graph of a self-contained BGP subsystem $S_{\mathbb{P}}$ has no P-cycle, then the BGP protocol converges on destinations in $\mathcal{D}_i$ for all AS $i \in V$. In addition, let r^* be the network route selection after convergence, then $r_i^* \in \mathcal{R}_i^{\mathcal{D}_i}(\mathcal{P}_i)$ for all $i \in V$. Furthermore, the BGP subsystem is guaranteed to be robust.*

The robustness result follows easily from the fact that node/link failures will not introduce any new P-cycle.

One can extend the proof in [24] to show that, the converged route selection is stable (by proving that the state are kept consistent during protocol execution in a multiple destination setting); that is, each AS's route profile is the highest ranked among all valid route profiles that can be constructed from the exported highest ranked route profile of each of its neighbors (subject to export policies).

3.4.5 Composition of self-contained BGP subsystems

In order to establish BGP protocol convergence for the whole network, we can directly apply Corollary 3 on the whole BGP system, since the whole BGP system is trivially a self-contained BGP subsystem. Sometimes, however, it may be more convenient to first establish BGP protocol convergence for two or more non-trivial self-contained BGP

subsystems, and then compose these subsystems to obtain convergence for the whole system.

There are two methods to compose two self-contained BGP subsystem $S_1 = (G, \mathtt{pt}, \sigma, \mathbb{D}^{(1)}, \widetilde{\mathbb{P}}^{(1)}, \mathbb{P}^{(1)})$ and $S_2 = (G, \mathtt{pt}, \sigma, \mathbb{D}^{(2)}, \widetilde{\mathbb{P}}^{(2)}, \mathbb{P}^{(2)})$.

The first type of composition is *parallel* composition. In this type of composition, S_1 and S_2 are disjoint in the sense that BGP protocol convergence on $\mathbb{D}^{(1)}$ and $\mathbb{D}^{(2)}$ are totally independent. Specifically, parallel composition requires that $\mathcal{D}_i^{(1)} \cap \mathcal{D}_i^{(2)} = \emptyset$, for all $i \in V$. Note that this also implies that $\widetilde{\mathcal{P}}_i^{(1)} \cap \widetilde{\mathcal{P}}_i^{(2)} = \emptyset$ and $\mathcal{P}_i^{(1)} \cap \mathcal{P}_i^{(2)} = \emptyset$. If we manage to establish convergence of S_1 and S_2, it follows immediately that BGP protocol also converges on $\mathcal{D}_i^{(1)} \cup \mathcal{D}_i^{(2)}$ for all $i \in V$.

The second type of composition is *sequential* composition; that is, BGP protocol converges on $\mathbb{D}^{(1)}$ first, and for any converged partial route profile for $\mathbb{D}^{(1)}$, routes to destinations in $\mathbb{D}^{(2)}$ will also converge. Sequential composition requires two conditions. we define some notations to formalize the conditions. For any stable route selection $\hat{r}^{(1)}$ for S_1, let $\widetilde{\mathbb{P}}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}$ be the subset of $\widetilde{\mathbb{P}}^{(2)}$ such that paths to destinations in $\mathbb{D}^{(1)}$ is given by $\hat{r}^{(1)}$; that is, $\widetilde{\mathbb{P}}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}$ is the restriction of $\widetilde{\mathbb{P}}^{(2)}$ by $\hat{r}^{(1)}$. Also define $\mathbb{P}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}$ in a similar way. Let $S_2|_{r^{(1)}=\hat{r}^{(1)}}$ be the BGP subsystem $(G, \mathtt{pt}, \sigma, \mathbb{D}^{(2)}, \widetilde{\mathbb{P}}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}}, \mathbb{P}^{(2)}|_{r^{(1)}=\hat{r}^{(1)}})$. Formally, sequential composition requires the following two conditions: 1) $\mathcal{D}_i^{(1)} \subseteq \mathcal{D}_i^{(2)}, \forall i \in V$; 2) for any $\hat{r}^{(1)}$, $S_2|_{r^{(1)}=\hat{r}^{(1)}}$ is a self-contained BGP subsystem. If we manage to show that BGP protocol converges on S_1 and $S_2|_{r^{(1)}=\hat{r}^{(1)}}$ for any stable $\hat{r}^{(1)}$, we can be sure that BGP protocol will eventually converge on $\mathcal{D}_i^{(2)}$ for all $i \in V$.

We will see an example of sequential composition of two self-contained BGP subsystems in Section 4.

3.5. Network with non-Pareto Optimal Solution

It is important that a stable route selection for interdomain traffic engineering be Pareto optimal; namely, there does not exist another stable route solution where each AS has a higher ranked route profile. In [5], Dakdouk *et al.* study interdomain routing from the social choice perspective and show that the social choice correspondence of BGP satisfies Pareto optimality property in the domain of strict route preferences. In other words, when the routing for multiple destinations is not coordinated, any stable route selection computed by BGP protocol is Pareto optimal.

However, we next show that when routing for multiple destinations is coordinated, a stable route selection for interdomain traffic engineering can be non-Pareto optimal. Figure 6 shows such an example. This example is particularly interesting in that it has two stable route solutions, as shown in Figure 7, and the solution at the second row is not even Pareto optimal. This example demonstrates again the new challenges facing route selection for interdomain traffic engineering. It also points out challenges in designing new interdomain routing protocols considering interdomain traf-

fic engineering. For example, to be effective, negotiation-based route selection [35] may involve more than two parties.
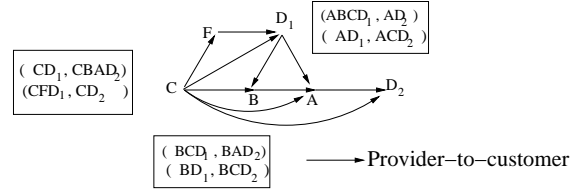

Figure 6. An example with two solutions but one of them is not Pareto optimal.

	A	B	C
Solution 1	$(ABCD_1, AD_2)$	(BCD_1, BAD_2)	$(CD_1, CBAD_2)$
Solution 2	(AD_1, ACD_2)	(BD_1, BCD_2)	(CFD_1, CD_2)

Figure 7. Two stable route selections for the network in Figure 6.

4. Stable Egress Route Selection without Global Coordination

The preceding section presents a sufficient condition to guarantee the convergence of route selection in a general network. The condition depends on checking P-cycle. In practice, it is difficult to obtain P-graph and check whether it contains a P-cycle. This is due to the fact that BGP is a distributed protocol, and generally ASes do not share their traffic engineering policies. Also, the preceding section considers general networks, while in the current Internet, the route selection policies of the ASes are not general, but are highly likely to be constrained by their business considerations. The question we will investigate in this section, therefore, is whether such constraints can lead to stability.

The constraints imposed by business considerations were first systematically studied by Gao and Rexford [19, 20]. Specifically, they observed that the business considerations of ASes in current Internet imply that ASes follow the typical export policies (please see Section 3.3 for definition). Typical export policies imply that instead of arbitrary valid routes, valid routes in the Internet have the following patterns [20]: a provider-customer link can be followed only by provider-customer links, and a peer link can be followed only by provider-customer links. Accordingly, we divide the routes from an AS i to a destination d into three categories:

- *Customer route*: each link along a customer route is a provider-customer link.

- *Peer route*: the first link along a peer route is a peer link, and the remaining links are all provider-customer links.

- *Provider route*: the first link is a customer-provider link, and the remaining route consists of zero or multiple customer-provider links, followed by zero or one peer link, and then zero or multiple provider-customer links.

Hereafter, we denote by $r_{i \to d}^C$, $r_{i \to d}^E$, and $r_{i \to d}^P$ an instance of customer, peer, and provider route, respectively. Similarly, we denote the set of customer, peer, and provider routes by $R_{i \to d}^C$, $R_{i \to d}^E$, $R_{i \to d}^P$, respectively. We can further divide the set $\mathcal{D}_i$ of destinations of an AS i into three categories, given that the above two constraints are satisfied:

- *Customer-reachable destinations*: these destinations are direct or transitive customers of AS i. Let $\mathcal{D}_i^C$ be the set of customer-reachable destinations of AS i. We have $\mathcal{D}_i^C = \{d | R_{i \to d}^C \neq \emptyset\}$.

- *Peer-provider-reachable destinations*: these destinations are direct or transitive customers of one of AS i's peers or providers, but they are not direct or transitive customers of AS i. Let $\mathcal{D}_i^E = \{d | R_{i \to d}^E \neq \emptyset\} - \mathcal{D}_i^C$ be the set of peer-reachable destinations, and $\mathcal{D}_i^P = \mathcal{D}_i - \mathcal{D}_i^C - \mathcal{D}_i^E$ the set of provider-reachable destinations. We call $\mathcal{D}_i^{EP} = \mathcal{D}_i - \mathcal{D}_i^C$ the set of peer-provider-reachable destinations of AS i.

Given the above definitions of different types of routes, Gao and Rexford [19, 20] observe that business considerations imply that an AS prefers customers routes over peer/provider routes. We call such route preference, namely, customer routes $\succ$ peer/provider routes, *the standard individual-route preference policy*. Assuming the standard export policy, the standard individual-route preference policy, together with the assumption that there is no provider-customer loop (PC-loop for short) in the business relationships formed by ASes, Gao and Rexford prove that these conditions guarantee convergence in the global Internet.

A potential issue of their analysis is that their route selection model assumes that there is no coordination among destinations. However, as we discussed in the preceding sections, in the current Internet, ISPs are increasingly adopting coordinated route selection policies to achieve their interdomain traffic engineering objectives. Given such coordination, we need to re-evaluate AS route selection behaviors and investigate whether they lead to stability. Specifically, we need to reevaluate how the standard individual-route preference policy will change if an AS coordinates its routes to multiple destinations. If economics is the first consideration, then it is still reasonable that an AS will prefer customer routes over peer/provider routes, since customer routes bring in revenue. However, in the general case, now an AS may coordinate the route selection of multiple customer-reachable destinations. As for those peer-provider-reachable destinations, now an AS can jointly select routes for multiple such destinations to load balance, and to maintain peering traffic ratios.

Specifically, the route selection behavior of each AS i can be described by ranking functions λ_i^C and λ_i^{EP}. Note that we use C and EP instead of $\mathcal{D}_i^C$ and $\mathcal{D}_i^{EP}$ as superscripts to simplify notation, we will also abbreviate $r_i^{\mathcal{D}_i^C}$ as r_i^C, and $r_i^{\mathcal{D}_i^{EP}}$ as r_i^{EP}. Suppose $\mathcal{A}_i$ is the set of paths available

to i, then i's selected route profile $\hat{r}_i$ is given by

$$\hat{r}_i^C = \arg\max_{r_i^C \in \mathcal{R}_i^{\mathcal{D}_i^C}(\mathcal{A}_i)} \lambda_i^C(r_i^C), \qquad (1)$$

$$\hat{r}_i^{EP} = \arg\max_{r_i^{EP} \in \mathcal{R}_i^{\mathcal{D}_i^{EP}}(\mathcal{A}_i)} \lambda_i^{EP}(r_i^{EP}). \qquad (2)$$

In other words, AS i's routing decision for customer-reachable destinations depend only on the routing decisions for its other customer-reachable destinations, and are independent of the routing decisions for its peer-provider-reachable destinations. Similarly, AS i's routing decisions for its peer-provider-reachable destinations are independent of that of its customer-reachable destinations. When the routing decisions of AS i are decomposed for customer- and peer-provider-reachable destinations, we say that it follows *the standard joint-route preference policy*.

We now show the pleasant but surprising result that egress route selection for interdomain traffic engineering in the current Internet is stable. In order to do so, we note that there exist two BGP subsystems in the network. The first BGP subsystem is $S_C = (G, \mathtt{pt}, \sigma, \mathbb{D}^C, \widetilde{\mathbb{P}}, \mathbb{P}^C)$, where $\mathcal{D}_i^C$ is the set of customer-reachable destinations for AS i, and $\mathcal{P}_i^C = \cup_{d \in \mathcal{D}_i^C} R_{i \to d}^C$ is the set of all customer routes of AS i. The second BGP subsystem is $S_{EP} = (G, \mathtt{pt}, \sigma, D, \widetilde{\mathbb{P}}, \widetilde{\mathbb{P}})$. It is easy to see that S_C is self-contained. Given any stable route selection $\hat{r}^C$ for S_C, $S_{EP}|_{r^C = \hat{r}^C}$ is also self-contained. Therefore, we can establish the BGP protocol convergence for the whole network through sequential composition of these two self-contained BGP subsystems:

Theorem 4 *The network has a unique stable route selection which BGP is guaranteed to converge to, and is guaranteed to be robust, if the following conditions hold:*

1. *there is no provider-customer loop in the network;*

2. *all ASes have fixed typical export policies;*

3. *the routing decisions for customer-reachable and peer-provider-reachable destinations follow the standard joint-route preference policy.*

Proof: We shall prove by sequential composition of two self-contained BGP subsystems that the network has a stable network route selection which BGP is guaranteed to converge to, and that the network is guaranteed to be stable. For a proof of uniqueness of the stable network route selection, please refer to the proof in [46].

Let $\widehat{\mathcal{P}}_i$ be the set of all possible paths for As i. The first BGP subsystem we consider is $S_C = (G, \mathtt{pt}, \sigma, \mathbb{D}^C, \widetilde{\mathbb{P}}, \mathbb{P}^C)$, where $\mathcal{D}_i^C$ is the set of customer-reachable destinations for AS i, and $\mathcal{P}_i^C = \cup_{d \in \mathcal{D}_i^C} R_{i \to d}^C$ is the set of all customer routes of AS i.

The BGP subsystem S_C is self-contained. Consider an arbitrary AS i and an arbitrary $d \in \mathcal{D}_i^C$. By definition of $\mathcal{D}_i^C$, there exists at least one customer route $P = (v_k, v_{k-1}, \ldots, v_0)$ with $v_k = i$ and $v_0 = d$, where

each link (v_i, v_{i-1}) is a provider-customer link, for $i = k, k-1, \ldots, 1$. Initially, AS d has a trivial customer route (d) to itself. Since each AS prefers customer routes strictly over peer/provider routes, it can be shown by induction that AS i eventually will get a customer route to d.

There is no P-cycle in the P-graph of S_C. Suppose for the sake of contradiction that there is a P-cycle. We will show that there is a PC-loop in this case. Since the two partial route profile connected by an improvement edge are of the same AS, it suffices to consider the sub-path edges on a P-cycle. Consider an arbitrary sub-path edge on the P-cycle from a partial route profile $\tilde{r}_u^k$ to $\hat{r}_v^l$. AS v must be a customer of u, because any link on a customer route is a provider-customer link. Thus if we follow the P-cycle and examine all the sub-path edges along our way, we will get a PC-loop, which is a contradiction.

By Corollary 3, BGP protocol will converge on S_C. Thus each AS i will have a stable partial route profile to destinations in $\mathcal{D}_i^C$.

Denote by $\hat{r}^C$ any stable route selection for S_C. The second BGP subsystem we consider is $S_{EP} = (G, \mathtt{pt}, \sigma, D, \widetilde{\mathbb{P}}, \widetilde{\mathbb{P}})$. It is easy to see that $S_{EP}|_{r^C = \hat{r}^C}$ is trivially self-contained for any stable route selection $\hat{r}^C$.

We shall prove that the P-graph of $S_{EP}|_{r^C = \hat{r}^C}$ does not contain a P-cycle. Suppose for the sake of contradiction that there is a P-cycle. We will show that there is a PC-loop in this case. Again, it suffices to consider the sub-path edges on the P-cycle. Consider an arbitrary sub-path edge on the P-cycle from a partial route profile $\tilde{r}_u^k$ to $\hat{r}_v^l$.

We first note the fact that $\hat{r}_v^l$ cannot be AS v's partial route profile to customer-reachable destinations. Otherwise, by applying similar argument as for S_C, we can show that all sub-path edges on the P-cycle are from a provider to a customer, which contradicts the assumption that there is no PC-loop. This fact also implies that v cannot be a peer u, because if $\tilde{r}_u^k(d)$ is a peer route for u, the sub-path $\hat{r}_v^l(d)$ must be a customer route for v. Thus v can only be a provider of u. If we follow the P-graph and examine all the sub-path edges along our way, we will get a PC-loop, which is a contradiction.

By Corollary 3, BGP protocol will converge on $S_{EP}|_{r^C = \hat{r}^C}$ for any stable $\hat{r}^C$. Thus each AS i will have a stable partial route profile to destinations in $\mathcal{D}_i$. But in this case, a partial route profile to $\mathcal{D}_i$ is exactly the complete route profile for i. Thus we have shown that BGP protocol converges for the whole network on all destinations.

In addition, it is not hard to see that the above proof holds even with arbitrary link/node failures, thus the network is robust. ∎

Note that when a network satisfies the conditions in the preceding theorem, there exists a unique stable route selection; therefore, this stable route selection is trivially Pareto optimal.

Note also that in the preceding theorem we require that customer routes are strictly preferred over peer routes; *i.e.*, customer routes $\succ$ peer routes. One might suspect that the above theorem still holds if customer routes $\succeq$ peer routes. However, Figure 3 gives a counter example, where all but the standard joint-route preference condition are satisfied.

As an application of Theorem 4, next we show that the recent trend of using smart routing to select egress routes does not introduce routing instability. Specifically, in [21], Goldenberg *et al.* propose algorithms to coordinate the egress route selection for multiple destinations to optimize performance under cost constraint. Using simulations, they show that their algorithms do not introduce instability. The focus of their paper is on stub ASes. Below, we show that if the conditions stated in Theorem 4 are satisfied before ASes adopting smart routing, then the conditions still hold when multihomed stub ASes adopt smart routing algorithms; thus, such algorithms do not introduce instability. First, adopting smart routing algorithms does not change the network topology; therefore, the first condition still holds. Second, adopting smart routing algorithms does not change the export policies. Third, a multihomed stub AS has only providers; therefore, its routing decisions, although coordinated, are inherently decomposed. Last, a multihomed stub AS follows the joint-route preference policy since it has only provider-routes to reach other destinations. To summarize, all of the conditions still hold when multihomed stub ASes adopt smart routing algorithms. Therefore, multihomed stub ASes adopting smart routing algorithms do not introduce routing instability. When non-stub ASes adopt smart routing, Theorem 4 suggests that a possibility for the smart routing algorithm to maintain stability is to observe the standard joint-route preference policy.

5. Simulation Studies of Route Selection for Interdomain Traffic Engineering

The preceding sections analyze the stability of route selection for interdomain traffic engineering and prove that convergence and uniqueness of route selection can be guaranteed when there is no provider-customer loop, and all ASes follow the typical export policy and standard joint-route preference policy.

In this section, we complement the preceding analysis by investigating the likelihood of instability when the policies and no-PC-loop condition are violated.

5.1. Methodology

We first present our methodology. Specifically, we derive a necessary and sufficient condition to uniquely determine provider-customer relationships. We then use this condition to infer Internet topology. Simulation setup is also described in this section.

5.1.1 Inferring AS topology

We construct an Internet AS topology from multiple vantage points by using the aggregated BGP tables of Routeviews [38] and Looking Glass servers [31]. Specifically, we remove prepended AS numbers from the AS paths in the BGP table and filter out the paths with loops. We then construct an undirected AS-level topology graph as follows. Each AS has a unique node in the graph, and there exists an edge between two AS nodes if they ever appear in pair in an

observed BGP route. The edges in this graph represent the connectivity among ASes.

We next infer business relationships among ASes to produce the *AS business-relationship graph*, denoted by G_b. Our inference of G_b consists of three steps. Firstly, we take the approach in [40] to infer peer relationships. Secondly, we infer provider and customer relationships for the remaining edges. Lastly, we remove edges with unknown relationships and label the remaining edges with the inferred relationships accordingly. In particular, in the second step, we construct a business-relationship inference graph, denoted by G_{infer}, to infer provider-customer relationships. In [3], Battista *et al.* map the inference of provider and customer relationships as a 2SAT problem. However, their method infers just one satisfiable solution. Thus, when the inferred business relationship between a pair of neighboring ASes is different from verification, it is unknown whether the error is due to ambiguity (*i.e.*, non-unique solutions) or model error. To overcome this problem, we construct a business-relationship inference graph as follows. Each pair of neighboring ASes, i and j, has two corresponding vertices in G_{infer}: v_{ij} and v_{ji}, where the vertex v_{ij} represents that i is a provider of j, while v_{ji} represents that j is a provider of i. We say that v_{ij} and v_{ji} are mirrors of each other. There exist edges between v_{ij} and v_{jk} in G_{infer} if and only if (i, j, k) or (k, j, i) appears as a segment of an observed route. In other words, from each route, we take all 3-tuple segments (i, j, k) and add two directed edges to the inference graph: one is from v_{ij} to v_{jk}, and the other from v_{kj} to v_{ji}. The directed edge from v_{ij} to v_{jk} encodes the fact that if i is a provider of j and (i, j, k) appears as a route segment, j must be a provider of k because of the no valley constraint. Given this construction and applying the result in [2], we have the following necessary and sufficient condition to check if the business relationship between a pair of neighboring ASes is uniquely determined:

Theorem 5 *If all routes are valley-free, and ASes have only provider-customer relationships, then AS i is a provider of j if and only if in G_{infer}, vertex v_{ji} has a path to its mirror vertex v_{ij} and v_{ij} has no path back to v_{ji}.*

We apply the preceding theorem on G_{infer} to infer provider and customer relationships. We find that 85% of AS relationships can be uniquely determined. In order to validate our inference results, we compare the set of inferred customers of AT&T using our approach with that using the approach in [18], where Gao verified with AT&T that 96.3% of AT&T-related relationships were correctly inferred. Our comparison shows that 98.8% of our inferred relationships are consistent with those using Gao's approach. We further validate our results by conducting email surveys with randomly selected regional transit ISPs. The results of the surveys show that all the inferred provider-customer relationships are correct.

In order to make the simulations more efficient, we iteratively remove 6157 single-homed ASes whose route selection will not affect that of others. The remaining AS graph, denoted by G'_s, has 13,048 ASes and 37,999 links and is used in our simulations.

We observe that the inferred network topology G'_s has about 1.3% of ASes involved in PC-loops. We further find that PC-loops are introduced because some customers carelessly provide transit services for their providers, and these customers are inferred as providers as a result. Note that in this section PC-loops are not defined by the real AS business relationships; instead, they are defined by the business relationships inferred from observed routes determined by the export policies. Note also that the existence of PC-loops does not invalidate Theorem 5 since the aggregated BGP table used to construct G_{infer} is not complete; therefore, the business relations of each link along a PC-loop may still be uniquely determined by applying Theorem 5.

To remove the PC-loops, we take into account the common belief that providers typically have more neighbors than their customers. Specifically, we first locate all the PC-loops in the graph. Then, for each PC-loop, we compute for each link along the loop the ratio of the provider's degree and the customer's degree, and iteratively remove the link with the lowest ratio, until there is no PC-loop. We denote by G_s the induced subgraph of G'_s after breaking all PC-loops. G'_s is only used to evaluate the impact of PC-loop on routing stability through simulation, and G_s is used in all other simulations.

5.1.2 Simulation setup

An important component of our simulation studies is route ranking tables. For AS i who does not coordinate the route selection of multiple destinations, we use the subjective routing framework to construct its route ranking table [11]. The subjective routing framework is motivated by the observation that different ASes often use different performance metrics in comparing routes. Thus, in this framework, there is a set M of performance metrics assigned to each link. Each AS computes the cost of a route using its own set of weights. Specifically, AS i has a set of weights, $W_i = \{w_{i,m} | m \in M\}$, where $w_{i,m}$ is the weight associated with the performance metric m. Note that $w_{i,m} = 0$ if i is not concerned with the metric m. Let $C_l^{(m)}$ be the value of metric m at link l. Given a route $r_{i \to d}$ from AS i to destination d, AS i computes the cost of this route as $c(r_{i \to d}) = \sum_{m \in M} w_{i,m} \sum_{l \in r_{i \to d}} C_l^{(m)}$. For each destination, AS i chooses the route with the lowest subjective cost as its best route for that destination.

For an AS i who coordinates its route selection of multiple destinations, we construct its ranking table as follows. First, for each destination d, we compute the set $R_{i \to d}$ of all feasible valley-free routes from i to d in G_s, assuming all ASes have typical export policies. Then we construct the set of all possible route profiles $R_i = \prod_{d \in \mathcal{D}} R_{i \to d}$. For efficiency, we do not explicitly store R_i; instead, we store just the set of all feasible routes to all destinations (*i.e.*, $\cup_{d \in \mathcal{D}} R_{i \to d}$), and assign a unique ID to each route in this set; therefore, we represent a route profile using a set of IDs corresponding to the routes in the route profile. Finally, we construct the ranking table of AS i by randomly permuting the entries of R_i.

We implement our own event-driven simulator to study

the stable route selection problem for interdomain traffic engineering. It simulates BGP protocol process such as route import/export, route announcement/withdrawal, and so on. Each AS selects its routes as described above. We also add random delays to route import/export events in order to simulate network asynchronousness. In each experiment, we randomly choose a set of ASes as destinations, and all other ASes exchange routes to these destinations.

To detect instability, for each AS, our simulator keeps a history of its selected route profiles. Specifically, according to its route selection history, each AS constructs a directed stability graph with each node representing a unique route profile and each directed edge representing a temporal transition between two route profiles. An AS has no stable route selection if all nodes of the stability graph are in one single strongly connected component. Hereafter, we refer to such ASes as *unstable* ASes. Since this condition is a sufficient condition, we may underestimate the extent of instability. In order to avoid taking initial route exchanges as unstable route selection, we wait for a long enough time before checking instability. Specifically, we start to keep a history of previous best route profiles for each AS after 500 seconds in simulation time when all ASes have routes to all destinations. We start to check the instability condition for each AS every 20 seconds in simulation time after the routing history starts. We run the simulation for 7,000 seconds in simulation time so that the number of ASes identified as unstable does not change any more, and take this number as the number of unstable ASes.

5.2. Routing Instability when Each Destination Is Routed Separately

We start our study of routing instability when no AS coordinates its route selection. Although the focus of this paper is on routing instability caused by coordination of route selection, since there is no previous simulation study on the single destination case, we conduct the first set of experiments as reference points. In our simulations, we randomly choose a destination AS that originates route announcements. The remaining ASes follow BGP protocol process to select the best route with the minimum subjective cost to the chosen destination. We then repeat the experiment with different random seeds and obtain the cumulative fraction of unstable ASes.

Our first experiment uses the topology with PC-loops, *i.e.*, G'_s, to study routing instability. In this experiment, all ASes have typical export policies, and strictly follow the standard individual-route preference. However, due to the existence of PC-loops, we still observe unstable ASes. Figure 8(a) shows the empirical cumulative distribution of the number of unstable ASes obtained from our experiments. We also conduct a distribution fitting and find that the extreme value distribution best fits the empirical one. Figure 8(b) also plots their density functions. To confirm that it is PC-loops that causes instability, we repeat the same experiment using G_s, where all PC-loops are removed, and we do not observe any instability in simulations.

Our second experiment uses the PC-loop-free topology, G_s, to study routing instability when ASes violate the stan-

dard individual-route preference. In this experiment, ASes have typical export policies. Each AS violates the standard individual-route preference with probability $p_v = 0.03$; for instance, with both a customer route and a peer route to a destination, an AS chooses the peer route instead of the customer route with probability 0.03. This probability is chosen because we observe that at most 3% of prefixes have routes violating the standard individual-route preference in the current Internet [46]. In order to study the impact of the violation probability on the number of unstable ASes, we also repeat the experiment with doubled violation probability $p_v = 0.06$.

Figure 9(a) shows the empirical cumulative distributions for both experiments. Similarly, we conduct a distribution fitting and find that the negative binomial distribution best fits them. We also plot in Figure 9(b) the density functions of both distributions for the case where $p_v = 0.03$. We observe that the number of unstable ASes increases when p_v is doubled. In particular, we find that on average, there are 43 unstable ASes when $p_v = 0.03$; when the violation probability is doubled, the average number of unstable ASes is more than doubled to 95. Comparing this experiment with the preceding one, we also observe that violation of the topological condition is more likely to lead to routing instability.
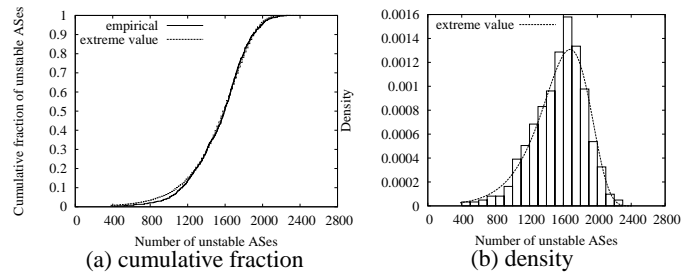

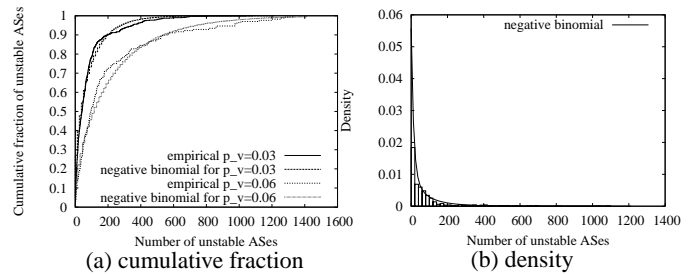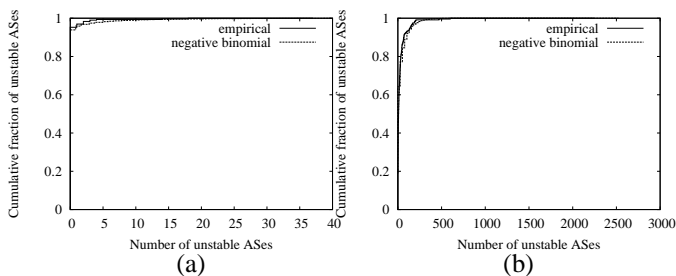**Figure 8. Distribution of total number of unstable ASes due to PC-loops.**

Figure 9. Distribution of total number of unstable ASes due to violation of the standard individual-route preference.

5.3. Routing Instability Caused by Route Coordination

Finally, we investigate routing instability caused by coordinated route selection of multiple destinations.

We start with a candidate set consisting of a randomly chosen Tier-2 AS. We then randomly choose the neighboring ASes of the candidates with probability 0.5 as the ASes that coordinate their route selections, and add them to the candidate set. This process continues until the set consists of enough number of ASes. We choose the candidate ASes in this way to model a scenario where ASes are more likely to coordinate route selections when their neighbors are doing so. We also limit our choice of candidate ASes to Tier-2 and Tier-3 ASes since Tier-1 ISPs are very cautious and less likely to actively coordinate their routes to achieve some traffic engineering objectives. To investigate the potential seriousness of the problem, we setup the experiments so that only 40 ASes coordinate route selection for only 2 destinations and violate the standard joint-route preference policy. All remaining ASes select routes for each destination separately.

We study the following two cases: (a) the remaining ASes strictly follow the standard individual-route preference; and (b) the remaining ASes violate the standard individual-route preference with probability 0.03. Figure 10 shows the empirical distribution of the number of unstable candidate ASes for both cases. We conduct a distribution fitting and find that the negative binomial distribution best fits the empirical distributions, as shown in the figures. We observe in case (a) that in worst cases, almost all 40 candidate ASes are unstable in the network. This result is surprising in that 40 ASes consist of a very small percentage (40 out of 13048) of the total number of ASes. Furthermore, 2 destinations are not many destinations. We also vary the number of ASes who coordinate route selection and the number of destinations. We observe that the number of unstable ASes further increases as the number of ASes who coordinate route selection but do not follow the joint-route preference policy increases. We also observe in case (b) that the number of unstable ASes strictly increases when the remaining ASes violate the standard individual-route preference.

Figure 10. Distributions of total number of unstable ASes due to violation of the standard joint-route preference policy, when the remaining ASes that do not coordinate route selections and either (a) strictly follow or (b) violate with probability 0.03 the standard individual-route preference.

6. Conclusion

In this paper, we conduct the first systematic study on the stability and efficiency of using route selection to achieve interdomain traffic engineering objectives. We identify that interdomain traffic engineering requires that route selection be coordinated among multiple destinations and that coordinated route selection can introduce routing instability and inefficiency. We show the surprising result that the interaction of the routing of multiple destinations can cause routing instability even when the routing of each destination individually does have a unique solution. We propose a general, simple model to capture the fundamental feature of coordinated egress route selection behaviors for interdomain traffic engineering and construct P-graphs to derive a sufficient condition to guarantee convergence and existence of stable route selection. Taking into account constraints imposed by Internet business considerations, we show the pleasant but surprising result that egress route selection for interdomain traffic engineering in the current Internet is stable if there is no provider-customer loop, and all ASes follow the typical export policy and the standard joint-route preference policy. We complement our analysis using simulations to investigate the likelihood of instability when the conditions are not satisfied. Our simulations based on realistic Internet AS topology show that if the policies are violated, even when a small number of ASes coordinate their routes for just two destinations, instability could happen.

Acknowledgments

We thank Jiang Chen, Ronny Dakdouk, Eric Friedman, Joan Feigenbaum, Tim Griffin, Arvind Krishnamurthy, Vijay Ramachandran, and Jennifer Rexford for their valuable comments and discussions. Our original proof of Theorem 4 was by induction; the one using Corollary 3 is pointed out by Aaron Jaggard. The proof also motivates us to define the notion of BGP subsystems. We are grateful to his help. We also thank the network operators who replied to our surveys, and their replies helped us to identify the current trend in interdomain traffic engineering.

References

[1] M. Afergan and J. Wroclawski. On the benefits and feasibility of incentive based routing infrastructure. In *Proceedings of ACM SIGCOMM '04 Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems*, Portland, OR, Sept. 2004.

[2] B. Aspvall, M. F. Plass, and R. E. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[3] G. D. Battista, M. Patrignani, and M. Pizzonia. Computing the types of the relationships between autonomous systems. In *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, Apr. 2003.

[4] T. Bressoud, R. Rastogi, and M. Smith. Optimal configuration for BGP route selection. In *Proceedings of IEEE INFOCOM '03*, San Francisco, CA, Apr. 2003.

[5] R. Dakdouk, H. Wang, H. Xie, and Y. R. Yang. Interdomain routing as social choice: A new perspective. Technical Report YALEU/DCS/TR-1331, Yale University, July 2005.

[6] A. Fabrikant, C. H. Papadimitriou, and K. Talwar. On the complexity of pure equilibria. In *Proceedings of the 36th Annual Symposium on Theory of Computing*, Chicago, IL, June 2004.

[7] N. Feamster, H. Balakrishnan, and J. Rexford. Some foundational problems in interdomain routing. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, CA, Nov. 2004.

[8] N. Feamster, J. Borkenhagen, and J. Rexford. Guidelines for interdomain traffic engineering. *ACM SIGCOMM Computer Communications Review*, Oct. 2003.

[9] N. Feamster, R. Johari, and H. Balakrishnan. Implications of autonomy for the expressiveness of policy routing. In *Proceedings of ACM SIGCOMM '05*, Philadelphia, PA, Aug. 2005.

[10] N. Feamster and J. Rexford. Network-wide BGP route prediction for traffic engineering. In *Proceedings of ITCOM*, Boston, MA, Aug. 2002.

[11] J. Feigenbaum, D. Karger, V. Mirrokni, and R. Sami. Subjective-cost policy routing. Technical Report YALEU/DCS/TR-1302, Yale University, Sept. 2004.

[12] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A BGP-based mechanism for lowest-cost routing. In *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC)*, pages 173–182, Monterey, CA, July 2002.

[13] J. Feigenbaum, R. Sami, and S. Shenker. Mechanism design for policy routing. In *Proceedings of the 23rd ACM Symposium on Principles of Distributed Computing (PODC)*, pages 11–20, St. John's, Newfoundland, Canada, July 2004.

[14] A. Feldmann, O. Maennel, B. Maggs, N. Kammenhuber, R. D. Prisco, and R. Sundaram. A methodology for estimating interdomain Web traffic demand. In *Proceedings of the Internet Measurement Conference*, Oct. 2004.

[15] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs. Locating Internet routing instabilities. In *Proceedings of ACM SIGCOMM '04*, Portland, OR, Aug. 2004.

[16] A. Feldmann and J. Rexford. IP network configuration for intradomain traffic engineering. *IEEE Network Magazine*, pages 46–57, Sept./Oct. 2001.

[17] B. Fortz, J. Rexford, and M. Thorup. Traffic engineering with traditional IP routing protocols. *IEEE Communication Magazine*, Oct. 2002.

[18] L. Gao. On inferring autonomous system relationships in the Internet. *IEEE/ACM Transactions on Networking*, 9(6), Dec. 2001.

[19] L. Gao, T. G. Griffin, and J. Rexford. Inherently safe backup routing with BGP. In *Proceedings of IEEE INFOCOM '01*, Anchorage, AK, Apr. 2001.

[20] L. Gao and J. Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, Dec. 2001.

[21] D. Goldenberg, L. Qiu, H. Xie, Y. R. Yang, and Y. Zhang. Optimizing cost and performance for multihoming. In *Proceedings of ACM SIGCOMM '04*, Portland, OR, Aug. 2004.

[22] R. Govindan and A. Reddy. An analysis of Internet inter-domain topology and route stability. In *Proceedings of IEEE INFOCOM '97*, Kobe, Japan, Apr. 1997.

[23] T. G. Griffin, A. D. Jaggard, and V. Ramachandran. Design principles of policy languages for path vector protocols. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[24] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(22):232–243, Apr. 2002.

[25] T. G. Griffin and G. Wilfong. A safe path vector protocol. In *Proceedings of IEEE INFOCOM '00*, Tel Aviv, Israel, Mar. 2000.

[26] A. Jaggard and V. Ramachandran. Robustness of class-based path-vector systems. In *Proceedings of the 12nd International Conference on Network Protocols (ICNP) '04*, Berlin, Germany, Oct. 2004.

[27] A. Jaggard and V. Ramachandran. Relating two formal models of path-vector routing. In *Proceedings of IEEE INFOCOM '05*, Miami, FL, Apr. 2005.

[28] R. Johari and J. N. Tsitsiklis. Routing and peering in a competitive Internet. Available at: http://web.mit.edu/jnt/www/publ.html, Jan. 2003.

[29] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet routing convergence. In *Proceedings of ACM SIGCOMM '00*, Stockholm, Sweden, Aug. 2000.

[30] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. In *Proceedings of ACM SIGCOMM '97*, Cannes, France, Sept. 1997.

[31] Looking Glass servers. http://www.traceroute.org.

[32] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, San Mateo, CA, 1996.

[33] R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan. Experiences applying game theory to system design. In *Proceedings of ACM SIGCOMM '04 Workshop on Practice and Theory of Incentives and Game Theory in Networked Systems*, Portland, OR, Sept. 2004.

[34] R. Mahajan, D. Wetherall, and T. Anderson. Towards coordinated interdomain traffic engineering. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, San Diego, CA, Nov. 2004.

[35] R. Mahajan, D. Wetherall, and T. Anderson. Negotiation-based routing between neighboring domains. In *Proceedings of USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '05)*, San Francisco, CA, May 2005.

[36] B. Quoitin, S. Uhlig, and O. Bonaventure. Using redistribution communities for interdomain traffic engineering. In *QoFIS'02 LNCS 2511*, Oct. 2002.

[37] B. Quoitin, S. Uhlig, C. Pelsser, L. Swinnen, and O. Bonaventure. Interdomain traffic engineering with BGP. *IEEE Communications Magazine*, 41(5):122–128, May 2002.

[38] RouteViews project. http://www.routeviews.org/.

[39] J. L. Sobrinho. Network routing with path vector protocols: Theory and applications. In *Proceedings of ACM SIGCOMM '03*, Karlsruhe, Germany, Aug. 2003.

[40] L. Subramanian, S. Agarwal, J. Rexford, and R. H. Katz. Characterizing the Internet hierarchy from multiple vantage points. In *Proceedings of IEEE INFOCOM '02*, New York, NY, June 2002.

[41] L. Subramanian, M. Caesar, C. T. Ee, M. Handley, Z. M. Mao, S. Shenker, and I. Stoica. HLP: A next generation inter-domain routing protocol. In *Proceedings of ACM SIGCOMM '05*, Philadelphia, PA, Aug. 2005.

[42] R. Teixeira, T. Griffin, A. Shaikh, and G. Voelker. Network sensitivity to hot-potato disruptions. In *Proceedings of ACM SIGCOMM '04*, Portland, OR, Aug. 2004.

[43] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford. Dynamics of hot-potato routing in IP networks. In *Proceedings of Joint International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, New York, NY, June 2004.

[44] S. Uhlig and O. Bonaventure. Implications of interdomain traffic characteristics on traffic engineering. In J. Crowcroft and A. Feldmann, editors, *Special issue on traffic engineering of European Transactions on Telecommunications*. 2002.

[45] K. Varadhan, R. Govindan, and D. Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32(1):1–16, 2000.

[46] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz. On stable route selection for interdomain traffic engineering: Models, analysis, and guidelines. Technical Report YALEU/DCS/TR-1316, Yale University, Feb. 2005.

[47] H. Wang, H. Xie, Y. R. Yang, L. E. Li, Y. Liu, and A. Silberschatz. On the stability of rational, heterogeneous interdomain route selection. In *Proceedings of the 13th International Conference on Network Protocols (ICNP) '05*, Boston, MA, Nov. 2005.