Robust Path-Vector Routing Despite Inconsistent Route Preferences

Aaron D. Jaggard
Department of Mathematics
Tulane University
New Orleans, LA USA
adj@math.tulane.edu

Vijay Ramachandran
Department of Computer Science
Stevens Institute of Technology
Hoboken, NJ USA
vijayr@cs.stevens.edu

***Abstract*— Some commonly used inter-domain-routing policies—*e.g.*, those using BGP's MED attribute for cold-potato routing—are beyond the scope of routing theory developed to date. This is because these policies cannot be expressed as a linear preference ranking of available routes at each node. Existing characterizations of well-behaved path-vector routing, however, critically depend on this linear ranking and do not naturally extend to more complex policies. In this paper, we present a framework that is able to model these more general policies. We use it to give the broadest-known sufficient condition for robust convergence of path-vector protocols, even when complex policies are used. In doing so, we present a new, unified notion of order on policies; this reduces to earlier results in the case of restricted policies, but it allows us to analyze the practically useful but inconsistent policies that could not be directly modeled before. As an application, we rigorously analyze (and improve) various robust protocol-design proposals.**

I. Introduction

Most routes on the Internet transit several independently administered network domains, called autonomous systems (ASes). Establishing connectivity between ASes, called inter-domain routing, is accomplished today using the Border Gateway Protocol (BGP) [1], a path-vector protocol. Routes are computed hop-by-hop through the network; at each step, routing decisions depend on routing policies configured locally within each AS. Convergence to a stable Internet routing thus depends on a composition of decisions involving many complex, autonomously provided inputs. Previous work [2] has shown that the interaction of these local policies can produce global anomalies in BGP, *e.g.*, nondeterministic routing and protocol divergence. To achieve greater network stability, a better understanding of the interaction of routing policies is necessary; furthermore, this must be done in a rigorous manner so that network operators can rely on provable guarantees about protocol behavior, even in worst-case scenarios.

This paper continues a line of work that explores the theoretical foundations of inter-domain routing and routing-policy interaction. Formal analysis of path-vector protocols that derived general sufficient conditions for robust convergence [3]–[7] ignore the complexities of sharing inter-domain routes within an AS; in particular, the model of the Internet assumes that every AS can be represented by one node in a graph with a single routing policy and a single link to each neighboring node. (In reality, an AS is often made up of several routers that maintain BGP sessions; these sessions often connect links to different neighboring ASes and provide multiple inter-connections between the same ASes.) As a result, these works fail to model some commonly used policies in BGP today, *e.g.*, the use of the Multi-Exit Discriminator (MED) attribute for cold-potato routing (discussed below in Example 2.8). One problem is that these policies seem to have "inconsistent preferences," because it is not possible to say that a given route is always better (or worse) than another.

On the other hand, work that addressed MEDs [8]–[10] did not give the policy-interaction analysis tools that the formal models have. This paper bridges this gap by presenting a generalized model to capture the static semantics of policy interactions for both inter-domain and intra-domain BGP sessions. We use this model to derive the first known sufficient condition—analogous to the simplified case—that guarantees robust protocol convergence *despite* inconsistent preferences.

In the rest of this section, we review the state of existing theoretical frameworks for inter-domain routing and previous attempts to analyze MED-related anomalies. In Section II, we introduce our model and show that simple uses of inconsistent preferences (which we formally define)—even at just one node—can cause routing divergence. We then derive the generalized convergence condition in Section III and discuss various applications in Section IV, including a space-efficient version of a proposal from [8] to prevent MED oscillations.

A. BGP Convergence Conditions

Gao and Rexford [5] showed that *robustness*—predictable convergence to a stable routing, even after link and node failures—is achieved when all policies follow constraints induced by a hierarchy corresponding to a simplified version of today's commercial Internet. However, minor changes to the business relationships, flexibility in the constraints, or misconfiguration could still lead to routing instability.

Griffin, Shepherd, and Wilfong [6] proposed the Stable Paths Problem (SPP) as the underlying formal problem solved by BGP. SPP captures the static semantics of routing-policy interaction as a total preference order of routes at each node. They were able to give a sufficient global condition for robustness, but showed that checking individual policies

1-4244-0593-9/06/$20.00 ©2006 IEEE

exactly for the existence of a stable routing solution is *NP*-hard. Combining the results of [5] and [6] gave a simplified version of BGP that assumed the underlying business hierarchy but allowed back-up routing while remaining robust [11].

These initial results were incorporated into theoretical frameworks [3], [4], [7] that model the behavior and design of path-vector protocols more generally, which allow rigorous analysis of convergence conditions. These works showed that consistency among the many preference orderings of routes at different nodes, together with ordering routes by path length, represents a sufficient condition for robust convergence equivalent to that of the original SPP work.

B. MED-Induced Oscillations

Unfortunately, the above convergence conditions only apply to protocols in which preferences are totally ordered at each node. We call this property *independent route ranking* (IRR) because the rank of a path does not depend on what other routes are known; the rank of two routes can be directly compared to determine which is best. (This property was also listed in [12], called *set-immune determinism*.) However, BGP's full route-selection procedure cannot be modeled in this way. In particular, use of the Multi-Exit Discriminator (MED) attribute may violate IRR (as in Example 2.8).

MED-induced oscillations are a well-known problem of BGP [13]–[15], and it has been conjectured that the violation of IRR is the major reason. These oscillations are especially difficult to analyze and debug on a real network because they are a product of not only BGP policy settings—involving attributes set in separately configured, independent ASes—but also internal distance settings within an AS (determined by an interior gateway routing protocol, or IGP).

There has been some work on the consequences of using the MED attribute, but the results have been incomplete. Basu *et al.* [8] and Musunuri and Cobb [10] proved that including in advertisements routes not chosen as best prevents MED-induced oscillations, but this change to BGP would increase the size of routing tables and the number or size of update messages. Griffin and Wilfong [9] presented examples of MED-induced oscillations and described them using an extension to their SPP model, but did not give a robustness constraint as in the original model. Other suggestions to solve the MED-oscillation problem affect the use of route reflectors and configuration of iBGP sessions within an AS [16] or require changing the interpretation of attributes [15].

This paper presents a formal model for policy routing that applies to configurations with or without IRR violations, including use of the MED attribute. We derive a constraint for policy configuration that guarantees robust convergence for the general case; it applies to instances of the original SPP model as well. Our extension to the general case is nontrivial, because previous conditions were expressed in terms of route rankings, which our model does not require. We also use the constraint to rigorously evaluate conjectured solutions to the MED-oscillation problem; in particular, we discuss a modification to the solution of Basu *et al.* [8] that requires fewer resources.

II. A Generalized Framework for Inter-Domain Routing

We begin this section by reviewing the dynamics of inter-domain routing protocols. We then define route-selection functions and independent route ranking (IRR), explaining the difference between our more general definitions and the more specific definitions used in previous theoretical work. We then present the Generalized Stable Paths Problem (GSPP) as the underlying theoretical problem being solved by routing protocols; it incorporates the generalized version of route selection. In doing so, we provide an example GSPP demonstrating a MED-induced oscillation.

A. Overview of Inter-Domain Routing

Internet traffic is *forwarded* from source to destination by routers along paths that traverse inter-domain and intra-domain links. Routers perform a basic forwarding operation, in which the destination IP address of a packet of traffic is matched to an entry in a forwarding table, and the packet is sent to the corresponding *next hop*—or neighboring router—listed in the entry. The job of routing protocols is to fill this forwarding table to form consistent, loopless paths for traffic to follow.

Intra-domain routing is well understood and is often based on simultaneous best-path calculations using some Interior Gateway Protocol (IGP)—at the intra-domain level, "best" is often defined as shortest. Inter-domain routing, however, is more complicated because the autonomy of domains and the scale of the Internet prevents both information about network topology to be distributed for such calculations and coordination or consistency among definitions of "best." Therefore, routes are computed on a hop-by-hop basis and decisions are influenced by local policy configurations.

Knowledge about destinations is learned through *advertisements* from neighboring routers; once a path to another AS is established, an AS will share that *reachability information* with its neighbors so that they gain knowledge of the destination as well. Assuming that destinations are first *originated* by the router responsible for that destination, paths are established by repeating the following three-step process:

1) Information about established routes through neighboring routers is collected, called *importing* routes. The route data stored in the local routing table depends on the route information in the update message and the *import policy*; the policy may *filter* routes entirely, *i.e.*, remove them from consideration.
2) For each destination, the protocol's best-route *selection procedure* is used to choose best routes from the local routing table. Best routes are then used to populate the forwarding table for these destinations.
3) Best routes are advertised to neighboring routers, called *exporting*. Update-message information about these routes is influenced by *export policy*, which may also filter routes.

The routers with inter-AS connections exchanging this information are *border routers*; however, any non-border routers

must learn about external destinations as well. The inter-domain protocol is thus also used to share external destinations with internal routers. As a result, path-vector protocols accomplish two inter-domain routing tasks:

1) establishing connectivity and sharing reachability information across inter-domain links; and
2) distributing knowledge of inter-domain routes to non-border routers.

Much of inter-domain-routing theory developed to date focused on task (1), *e.g.*, [4]–[7]. The Internet was modeled as a graph in which each vertex represents one AS; only inter-AS connections were considered and anomalous behavior related to task (2) was ignored. However, such anomalies have indeed been identified [13]–[15], and this paper extends routing theory to address these anomalies.

We write paths in the direction of forwarding traffic; *e.g.*, $P = v_0 v_1 \cdots v_n$ is a path from node v_0 to destination v_n. Node v_1 is the next hop on P. At the inter-domain level, most nodes v_i will represent ASes, not individual routers. However, because of task (2), it will be important to write a portion of the path from the source router to the border router such that nodes represent internal routers; *e.g.*, we may write $P = ABC(3)(6)(12)(7)$ for a path from the source AS starting at router A through internal router B to border router C, then onto ASes 3, 6, and 12 before reaching the destination AS 7. We assume that each transit and destination AS can appropriately route traffic within itself; thus inter-domain messages do not contain intra-domain information for other ASes. In general, when a router is establishing forwarding paths to a destination, we can view the Internet graph from that router's perspective as one in which all other ASes are represented by one node, neighboring ASes connect to the border routers of this router's AS, and other nodes represent the intra-domain routers and connections.

B. Route-Selection Functions and Independent Route Ranking

Step 2 in the above-described three-step process of choosing best routes from a routing table can be modeled by the following type of function.

Definition 2.1: A *route-selection function* σ_v maps a set of paths R to a set $S \subseteq R$ that is a set of "best" routes at node v; we write $\sigma_v(R) = S$. When we restrict the selection to a particular destination, we will write $\sigma_v^d(R) = S^d$ such that all paths S^d have destination d.

In most cases, including BGP, $|\sigma^d(R)| \leq 1$ for a set of paths R and some destination d (*i.e.*, for each destination, at most one best path is chosen; we refer to these as *singleton-valued selection functions*). Furthermore, we assume that choosing some permitted path is preferred to choosing no path, although some paths are filtered by local policy so that they are never considered as part of the selection process. Assuming that these filtered paths are not stored in the routing table R, then for all $R^d \subset R$ to a particular destination d, $R^d \neq \emptyset$ implies $\sigma^d(R^d) \neq \emptyset$. The process of collecting and storing routes, including what data structures are used for this purpose, and how it interacts with the selection procedure depend on the protocol implementation.

Independent route ranking (IRR) means that the preference of a path relative to other paths depends only on that path alone (and any information in that path's routing-table entry) and not knowledge of other paths.

Definition 2.2: A selection function σ obeys *Independent Route Ranking* iff, for all sets of routes R_1 and R_2 and destinations d, the following two conditions hold:

1) $\sigma^d(R_1) = S$ implies $\sigma^d(R_1 \cup R_2) \cap (R_1 \setminus S) = \emptyset$; and
2) $\sigma^d(R_1) = S$ and $\sigma^d(R_1 \cup R_2) \cap S \neq \emptyset$ implies $\sigma^d(R_1 \cup R_2) \supseteq S$.

We call violations of the first condition *type-1 IRR violations* and those of the second condition *type-2 IRR violations*. For singleton-valued selection functions, the above definition of IRR is equivalent to the following: if path P_1 is chosen over all paths in P as best, then additional knowledge of a route $P_2 \notin P$ does not permit another route $P_3 \neq P_1$ in P to be chosen as best; only P_1 or P_2 may be chosen relative to $P \cup \{P_2\}$. (Condition 2 is irrelevant for single-valued selection functions.)

Previous theoretical work [4], [6], [7] on path-vector protocols modeled only selection functions that independently assign a *rank* to each route and choose the path of minimal (or maximal) rank. Selection functions written in this way are called *linear selection functions*; at each node, the preference order on unfiltered (permitted) paths is consistent with a linear order. Because the protocol-convergence conditions described in [4], [6], [7] depended on this notion of rank, they do not apply to the more general setting involving arbitrary selection functions. (Note, however, that a preference ranking in terms of path attributes that corresponds to a given linear selection function may be quite complex.) We now show the relationship between linear selection functions and IRR; due to lack of space, these proofs may be found in the full report [17].

Definition 2.3: A selection function σ is a *linear selection function* iff there exists a map $\omega : \mathcal{P} \to \mathcal{U}$ from permitted paths $\mathcal{P}$ to a totally ordered set $\mathcal{U}$ such that

$$\forall\, R \subset \mathcal{P}, \quad \sigma(R) = \{P \mid \forall\, P' \in R,\ \omega(P) \leq \omega(P')\}\,.$$

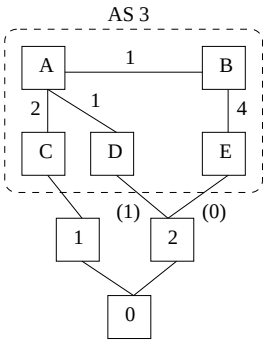
Proposition 2.4: A selection function has no IRR violations iff it can be written as a linear selection function.

It has been conjectured that IRR violations are a major cause of protocol oscillations [8], [9]. Proposition 2.5 shows that even a single IRR violation can cause divergence.

Proposition 2.5: Suppose σ_v is an IRR-violating (nonlinear) selection function. Then there exists an oscillating network instance containing node v in which all other nodes have IRR (linear) selection functions.

C. Generalized Stable Paths Problem

The Stable Paths Problem (SPP) [6] was suggested as the theoretical problem underlying inter-domain routing, but it limits nodes' route-selection functions to linear selection functions. We now present the generalized version first discussed in [9] to accommodate modeling attributes in BGP that are



Selection functions for routers A and B:

$$\sigma_A^0(AC10, AD20) = AD20$$
$$\sigma_A^0(AD20, ABE20) = ABE20$$
$$\sigma_A^0(AC10, ABE20) = AC10$$
$$\sigma_A^0(AC10, AD20, ABE20) = AC10$$

$$\sigma_B^0(BAD20, BE20) = BE20$$
$$\sigma_B^0(BAC10, BE20) = BAC10$$

Fig. 1. The GSPP MED-EVIL.

inconsistent with independent route ranking; from this point on, we assume that selection functions are singleton-valued.

Definition 2.6: An instance of the *Generalized Stable Paths Problem (GSPP)* is a network $G = (V, E)$ and a set of permitted paths $\mathcal{P}$ in G to a fixed destination node $v_0 \in V$. (The set $\mathcal{P}$ of permitted paths can be partitioned into sets $\mathcal{P}_v$, $v \in V$, which are the permitted paths at node v, *i.e.*, starting at v and ending at v_0.) All nodes $v \neq v_0$ have a route-selection function $\sigma_v^{v_0} : 2^{\mathcal{P}_v} \to \mathcal{P}_v$. A *path assignment* $\pi : V \to \mathcal{P}$ is a solution to GSPP iff $\pi(v_0) = (v_0)$ and for every $v \neq v_0 \in V$, $\pi(v) = \sigma_v^{v_0}(\{vP \in \mathcal{P} \mid P = \pi(u) \text{ and } \{u, v\} \in E\})$.

Remark 2.7: GSPP is *NP*-complete. This is because GSPP is in *NP*—given a solution, it is easy to check whether it is stable—and because SPP, an *NP*-complete problem [6], trivially reduces to GSPP by writing its path preferences as (linear) selection functions.

Example 2.8: Figure 1 shows an example GSPP, which was given in [9] and is named MED-EVIL. This instance models the route-selection procedure of BGP running on a network in which the MED attribute is used. The network is shown from the perspective of AS 3.

When a route is imported from neighbors, it is given a local-preference value that is entered into the routing table to indicate how "good" the route is; the MED attribute, on the other hand, is set by the *exporting* (or advertising) AS to indicate *its* preference among multiple inter-AS connections. The path-selection procedure for BGP is as follows:

1) Choose routes with the largest local preference.
2) In the case of a tie, routes with the shortest AS-path length are chosen.
3) In the case of a tie, if there are multiple paths to the same AS, choose the path with the lowest MED value. MED values are only compared among paths to the same AS.
4) If there remains a tie because there are paths to different ASes, choose the path with the shortest IGP distance to its egress point.

The importing AS has ultimate control by setting local-preference values, but these are often set equally for all routes through a given AS, even across different inter-AS links. In practice, this allows a neighboring AS to influence the choice between the inter-AS links by using the MED attribute.

If MEDs are not used (*i.e.*, ignoring step 3), the route-selection procedure above (via step 4) breaks ties based on minimal IGP distance; this is known as *hot-potato routing*, in which nearest egress points are used. Otherwise, the term *cold-potato routing* is used, and neighboring ASes can express alternate preferences for ingress points using the MED attribute. For example, consider a small network with high costs to carry traffic internally, and suppose it has inter-AS connections to its Internet provider in California and New York. When advertising destinations to the provider, the small customer can attach appropriate MED values so that the provider chooses egress points closest to each destination; traffic traverses as little of the customer network as possible. If the provider instead used basic hot-potato routing, traffic would exit the provider network at first opportunity (close to the source), possibly causing the customer to handle transcontinental traffic.

The instance MED-EVIL, shown in Figure 1, was first given in [9] as an example of a MED-induced oscillation. IGP distances are listed as numbers next to links; MED values are listed next to inter-AS connections in parentheses. Let the fixed destination be AS 0, and assume that all paths have the same local-preference value assigned at AS 3. The selection functions for the internal routers A and B are also shown. It is important to note that σ_A has an IRR violation because of the MED values set by AS 2; thus, the paths cannot be ranked and this configuration cannot be represented as a standard SPP.

To see why this GSPP has no solution, assume that A and B have not advertised routes to each other; then they will choose $AD20$ and $BE20$, respectively, because of minimal IGP distances. If these nodes share these choices, B will still choose $BE20$ because, even though $BAD20$ has a shorter IGP path length, its MED value is higher than $BE20$ and both paths lead to AS 2. Router A, upon learning of $ABE20$, will no longer consider $AD20$ because of its higher MED value and will choose $AC10$ instead (because of its IGP path length is shorter than $ABE20$, the other viable option). When A's new choice is broadcast to B, router B will choose $BAC10$ because of its shorter IGP distance (over $BE20$), withdrawing $BE20$. However, this withdrawal removes the path through AS 2 with lower MED value, causing A to choose $AD20$ again, withdrawing $AC10$. Thus, we have an oscillation similar to that in the proof above.

D. Convergence Properties

We are not only interested in whether policies interact to allow a stable path assignment, *i.e.*, whether or not a GSPP has

a solution, but also in how path-vector protocols, following the three-step hop-by-hop process described above, can reach that assignment. In the next section we will provide a broad sufficient condition that guarantees robust protocol convergence to a unique solution. To derive this condition, we must investigate protocol behavior in addition to the existence of solutions. The *evaluation digraph*, which is a graph constructed from a GSPP instance and defined below, allows us to do this.

To simplify our discussion of convergence properties, we assume that routes to different destinations are computed independently; therefore, we can always discuss protocol convergence with respect to one destination. This allows us to use GSPPs to describe protocol convergence in general.

Definition 2.9: The *evaluation digraph* of a GSPP instance S is a directed graph $\mathcal{T}(S) = (V_{\mathcal{T}}, E_{\mathcal{T}})$ in which the nodes represent *protocol selection states*, and the edges represent transitions between states. A selection state is a path assignment $\pi \in \left(\prod_{v \in V} \mathcal{P}_v\right)$; if $\alpha \in V_{\mathcal{T}}$, then we denote the path assigned to α by π_α. The *start state* is the node corresponding to the empty path assignment, in which $\pi(v_0) = (v_0)$ and, for $v \neq v_0$, $\pi(v) = \epsilon$, the empty path.

The directed edge (α, β) is present in $E_{\mathcal{T}}$ iff

$$\forall\, v \in (V \setminus \{v_0\})\,,\ \pi_\beta(v) = \sigma_v^{v_0}\left(\bigcup_{\{u,v\} \in E} \{v\pi_\alpha(u)\}\right);$$

i.e., given that nodes select the paths π_α and then broadcast these selections to their neighbors through asynchronous FIFO links, nodes might next select the paths π_β. Note that there may already be path data in the links that has been delayed in transit, so that $\pi_\alpha(v) = P$ and $\pi_\beta(v) = P'$ but, for a neighbor u, $\pi_\alpha(u) = Q$ and $\pi_\beta(u) = uP$. (Therefore, states may not be consistent; these states are not acceptable as solutions.)

We can follow the execution of a path-vector protocol on a GSPP instance by its *trace*, which corresponds to a directed path in the evaluation digraph beginning at the start state. Traces end at *sink states*, *i.e.*, nodes whose only outgoing edges are loop edges. Because the evaluation digraph is finite, if all traces are acyclic (ignoring loop edges), then all protocol runs will converge. It is clear that, equivalently, if the network dynamically oscillates during route selection then there is a cycle in its evaluation digraph; each of the paths among which a node oscillates will appear in at least one of the states in the corresponding cycle.

Proposition 2.10: A path assignment corresponds to a sink state iff it is a solution.

Proof: A solution is a stable routing tree. Suppose π_α is a solution; then by Definition 2.6, for all $v \neq v_0 \in V$, $\sigma_v^{v_0}\left(\bigcup_{\{u,v\} \in E}\{v\pi_\alpha(u)\}\right) = \pi_\alpha(v)$. By Definition 2.9, this is equivalent to α having no outgoing edges in the evaluation digraph other than loop edges, *i.e.*, that α is a sink state. ■

Therefore, we can define protocol-convergence properties in terms of the structure of the corresponding evaluation digraph. The following combinations of the existence of solutions and the ability of protocols to reach those solutions are of interest.

Definition 2.11: The following are convergence properties for GSPP instances.

Solvability: A GSPP is *solvable* if there exists at least one path assignment that is a solution; *i.e.*, the evaluation digraph of the GSPP has at least one sink state.

Unique Solvability (Predictability): A routing configuration is *uniquely solvable* if there exists exactly one GSPP path assignment that is a solution; *i.e.*, the evaluation digraph contains exactly one sink state.

Safety: A routing configuration is *safe* if a path-vector protocol is able to converge to a solution; *i.e.*, all traces in the GSPP's evaluation digraph are acyclic. The existence of a solution does not determine safety.

Robustness: A routing configuration is *robust* if it and all sub-instances (resulting from node or link failures) are uniquely solvable and safe; *i.e.*, all traces in the GSPP evaluation digraph are acyclic and end at the same sink state.

We are interested in robust path-vector protocols because these avoid nondeterminism and divergence, which are problems that are difficult for network operators to understand and debug when they occur at the inter-domain level.

Remark 2.12: Note that the definition of robustness, while requiring all sub-instances to be predictable and safe, requires all traces only in the original GSPP's evaluation digraph to be acyclic and end at the same sink. This is because sub-instances have evaluation digraphs that are subgraphs of the original instance's evaluation digraph (with some paths no longer possible because of failures); the property of acyclicity holds on subgraphs.

Remark 2.13: The generalization of SPP to GSPP leads to a parallel generalization of the PVPS framework of [4]. The technical report [17] discusses that generalization, which we call the GPVPS (Generalized PVPS) framework. The convergence properties that we discuss here have GPVPS analogues, and the sufficient condition for robust GSPP convergence may be used as a global constraint on GPVPSes [17].

III. Generalized Convergence Conditions

Given a set of routing-policy inputs, we can study the corresponding GSPP instance's evaluation digraph to see how they affect path-vector-protocol execution. However, an evaluation digraph is both large and complex; it is impractical to construct it as this requires simulating all possible update sequences. Griffin, Shepherd and Wilfong [6] showed that a smaller structure, called a *dispute wheel*, can be constructed from an SPP instance that is not robust. Unfortunately, the original definition of the structure is not compatible with nonlinear selection functions.

In this section we begin by introducing a new version of dispute wheels and prove that it adequately captures oscillations in GSPPs. From that discussion, we are then able to describe oscillations in terms of an underlying order on permitted paths described by local-policy configurations. This notion of *partially ordered SPPs* first appeared in [4]; however, because our generalized version of the problem does not have

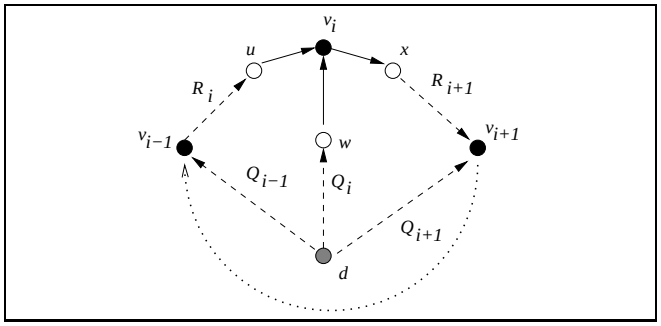


Fig. 2. Dispute wheel.

a notion of rank, we must nontrivially change the components of this order to correctly describe the robustness condition.

A. Generalized Dispute Wheels

Definition 3.1: A *generalized dispute wheel* (see Figure 2) contains *active nodes* $v_0, \ldots, v_k$ (with all subscripts interpreted modulo $k+1$) such that v_i has a *spoke path* Q_i to the destination d and v_i and v_{i+1} are connected by a *rim segment* R_{i+1} such that either:

1) $\exists\, S \supseteq \{Q_i,\ R_{i+1}Q_{i+1}\}$ s.t. $\sigma^d_{v_i}(S) = R_{i+1}Q_{i+1}$; or
2) $\exists\, S \not\ni R_{i+1}Q_{i+1}$ s.t.:
   a) $\sigma^d_{v_i}(S \cup \{Q_i\}) \neq Q_i$ and
   b) $\sigma^d_{v_i}(S \cup \{Q_i,\ R_{i+1}Q_{i+1}\}) = Q_i$; or
3) $\exists\, S \not\ni R_{i+1}Q_{i+1}$ s.t.:
   a) $\sigma^d_{v_i}(S \cup \{Q_i\}) = Q_i$ and
   b) $\sigma^d_{v_i}(S \cup \{Q_i,\ R_{i+1}Q_{i+1}\}) \notin \{Q_i,\ R_{i+1}Q_{i+1}\}$.

Remark 3.2: Note that of the three relationships between active nodes in a generalized dispute wheel, only condition (1) can occur for a linear selection function; conditions (2) and (3) imply the existence of an IRR violation. Condition (1) is analogous to the condition on rim segments found in the original definition of a dispute wheel for standard SPPs.

The generalized dispute wheel is a graph constructed from a GSPP instance, using the same nodes and edges in the instance's network graph. However, nodes may appear more than once in a dispute wheel, *e.g.*, in multiple spoke paths.

Theorem 3.3: If the evaluation digraph of a GSPP instance contains a cyclic trace, *i.e.*, if a GSPP instance is not safe, then there exists a generalized dispute wheel.

Proof: Let C be a cycle in the evaluation digraph of the instance, v_0 a node which does not select the same route throughout C, and Q_0 one of the paths that v_0 selects in C. Without loss of generality, we may assume that u is the last (and thus only) node on Q_0 that does not select the same route throughout C. Viewing Q_0 as one of the spokes of a generalized dispute wheel, we now construct another such spoke and a rim segment joining it to the spoke Q_0.

Let v_0P_1 be the next path that v_0 selects in C, and let x_1 be the first node on P_1. If x_1 oscillates its path selection in C, then let v_1 be the last cycling node on P_1, let $Q_1 = v_1 \cdots d$ be the next spoke, and $R_1 = v_0x_1 \cdots v_1$ be the rim segment connecting these two spokes. (Both Q_1 and R_1 are subpaths of P_1.) Because x_1 oscillates in C, it must broadcast and withdraw P_1 during the oscillation, and one of these actions causes the selection-state transition; thus the rim segment satisfies condition (1) in Definition 3.1.

If x_1 does not oscillate in C, let v_0P_2 be the path that v_0 selects in C after v_0P_1 and x_2 the first node on P_2. If x_2 cycles in C, we may proceed as above, otherwise we consider the path v_0P_3 that v_0 selects in C after v_0P_2, *etc.* Eventually, we either construct another spoke connected to Q_0 by a new rim segment or we progress through all of C and return to the path assignment in which v_0 selects v_0P_1. If the latter happens, then v_0 cycles through a sequence of paths in C, and each of these paths is learned from a neighbor who does not cycle in C. All of these paths are thus known to v_0 at all times, therefore all of the changes in path assignment to v_0 must be the result of IRR violations. (This is because a change in path assignment requires that v_0 know of different routes before and after the change. If the change selects a route that was already known but not chosen, by Definition 2.2, the selection function for v_0 has a type-1 IRR violation.)

In this case, assume that v_0's selection of Q_0 is the result of $\sigma^d_{v_0}(S) = Q_0$ and v_0's choice of v_0P_1 is the result of $\sigma^d_{v_0}(S_1) = v_0P_1$, with $Q_0, v_0P_1 \in (S \cap S_1)$. Because $S \,\Delta\, S_1 \neq \emptyset$, there is some route P_2 such that either learning or withdrawing v_0P_2 causes the transition from S to S_1 and Q_0 to v_0P_1. Let x be the first node on P_2 and v_1 be the last oscillating node on P_2. (There is such a node because P_2 is broadcast and withdrawn in the oscillation; otherwise we would not have this oscillation.) Then we can let $Q_1 = v_1 \cdots d$ be the next spoke, and $R_1 = v_0x \cdots v_1$ be the rim segment joining them such that either condition (2)—if P_2 is learned—or condition (3)—if P_2 is withdrawn—is satisfied.

Because the oscillation cycle is finite, we can repeat this process until we reach a selection state or path assignment that we have already visited. At this point, a subset of the spoke and rim segments will form a generalized dispute wheel. ■

Corollary 3.4: If an instance of GSPP is not solvable, then it contains a generalized dispute wheel.

Proposition 3.5: If an instance of GSPP has multiple solutions, then it contains a generalized dispute wheel.

Proof: We follow an analogous proof method in [6]. Suppose π_1, π_2 are two solutions; we can view these as trees in the network, rooted at the destination v_0: $T_i = \bigcup_{v \in V} \pi_i(v)$. Then let $H = (V,\ E(T_1) \cap E(T_2))$ be the graph induced by the intersection of the trees and let T be the component of H including v_0. $T_1 \neq T_2$ implies that $V - V(T)$ is nonempty.

In the following process, assume that all nodes u_i are assigned paths in both solutions. Choose an edge $\{u_1, v_1\} \in T_1$ where $u_1 \notin V(T)$ and $v_1 \in V(T)$. Then $\pi_1(u_1) = u_1Q_1$, where Q_1 is the path in T from v_1 to d; $\pi_1(v_1) = \pi_2(v_1) = Q_1$ so that Q_1 is in both solutions because T is the intersection of both solutions. There is some other path $P_1 = \pi_2(u_1)$ in T_2; this path is of the form R_2Q_2 where $R_2 = u_1 \cdots u_2$ contained in $T_2 \setminus H$ and $Q_2 = v_2 \cdots d$ contained in T. Note that $\pi_2(u_2) = u_2Q_2$, so we can repeat this process by examining the path $\pi_1(u_2)$. Continuing, we can alternate between both solutions until we repeat a node u_i.

The paths R_i, Q_i form a generalized dispute wheel. This is because for each i, there must exist some $S \subset \mathcal{P}_{u_i}$ such that $\sigma_{u_i}^{v_0}(S \cup \{R_{i+1}Q_{i+1}, u_iQ_i\}) = R_{i+1}Q_{i+1}$ because for either $i = 1$ or $i = 2$, $\pi_i(u_i) = R_{i+1}Q_{i+1}$ given the construction above. (If not, π_i is not a stable solution: Because Q_i is in the intersection of both solutions, the path u_iQ_i must be available.) This satisfies condition (1) in Definition 3.1. ■

The contrapositive of the above three assertions forms a sufficient condition on GSPP instances that guarantees robust protocol convergence; we summarize this as the following.

Proposition 3.6: If a GSPP instance has no generalized dispute wheel, it is robust.

B. Partially Ordered GSPPs; Generalized Dispute Digraphs

The three types of conditions described in Definition 3.1 that connect dispute-wheel spokes by rim segments can be used to define relations between permitted paths in a GSPP. Here, we use these relations to define another tool for characterizing policy disputes—a generalization of the dispute digraph [4], [6]. Intuitively, when policies are consistent with a partial order defined by these path relations, they do not induce a global routing anomaly.

Definition 3.7: Define the following four relations on permitted paths in a GSPP instance; assume that v_0 is the fixed destination node and that $u, v \in V$ are other network nodes.

Subpath: $P_1 \ominus P_2$ iff

$$P_1 = v \cdots v_0,\ P_2 = u \cdots v_0,\ \text{and } uP_1 = P_2$$

Linear Selection: $P_1 \oslash P_2$ iff

$$P_1 = v \cdots v_0,\ P_2 = u \cdots v_0,\ \text{and}$$
$$\exists\, S : \sigma_u^{v_0}(\{uP_1,\ P_2\} \cup S) = uP_1$$

Nonlinear Selection (first type): $P_1 \odot_1 P_2$ iff

$$P_1 = v \cdots v_0,\ P_2 = u \cdots v_0,\ \text{and } \exists S \not\ni uP_1 :$$
$$\sigma_u^{v_0}(\{P_2\} \cup S) \neq P_2 \text{ and } \sigma_u^{v_0}(\{uP_1,\ P_2\} \cup S) = P_2$$

Nonlinear Selection (second type): $P_1 \odot_2 P_2$ iff

$$P_1 = v \cdots v_0,\ P_2 = u \cdots v_0,\ \text{and } \exists S \not\ni uP_1 :$$
$$\sigma_u^{v_0}(S) = P_2 \text{ and } \sigma_u^{v_0}(\{uP_1\} \cup S) \notin \{uP_1,\ P_2\}$$

We now define the following graph on the set of permitted paths using the above relations.

Definition 3.8: Given a GSPP instance S, its *generalized dispute digraph* is the directed graph $\mathcal{D}(S) = (V_\mathcal{D}, E_\mathcal{D})$. The nodes $V_\mathcal{D} = \mathcal{P}$ are the permitted paths in the network. The directed edge $(P_1,\ P_2)$ is present in $E_\mathcal{D}$ iff one of $P_1 \ominus P_2$, $P_1 \oslash P_2$, $P_1 \odot_1 P_2$, or $P_1 \odot_2 P_2$ holds.

Note that the dispute digraph is smaller than the evaluation digraph as each node is labeled with a single network route rather than a set of network routes; it is also easy to build given the definition of each node's selection function.

Because the relations correspond to transitions in the evaluation digraph and connections between dispute-wheel spokes, we can prove the following.

Theorem 3.9: A GSPP instance has a generalized dispute wheel iff it has a cycle in its generalized dispute digraph.

Proof: First assume that the instance has a generalized dispute wheel. Its rim gives a cycle in the generalized dispute digraph as follows, because the pair of paths from adjacent rim nodes to the destination each belong to one of the four relations in Definition 3.7. Begin with any active node v_i on the rim; let r_1 be the next node on the rim segment R_i. From the construction of the dispute wheel, $r_1Q_i = r_1v_i \cdots d$ is an extension of Q_i, so $Q_i \ominus rQ_i$; this relation holds for further extensions along the rim, such that $(r_i \cdots r_1Q_i) \ominus (r_{i+1}r_i \cdots r_1Q_i)$. Let R_i^* be the rim segment up to, but not including, v_{i-1}; using these relations, we see there is a path from Q_i to $R_i^*Q_i$ in the dispute digraph for each active node v_i in the dispute wheel. Call these paths D_i. Then, for every R_iQ_i and Q_{i-1}, one of the three conditions in Definition 3.1 holds. In the case of condition (1), $\exists\, S : \sigma_{v_{i-1}}^d(S \cup \{R_iQ_i,\ Q_{i-1}\}) = R_iQ_i$; thus $R_i^*Q_i \oslash Q_{i-1}$, corresponding to the edge $(R_i^*Q_i,\ Q_{i-1})$ connecting D_i and D_{i-1}. In the case of condition (2), learning R_iQ_i at v_{i-1} forces another route to be selected over Q_{i-1}; thus $R_i^*Q_i \odot_2 Q_{i-1}$, also corresponding to the edge $(R_i^*Q_i,\ Q_{i-1})$ connecting D_i and D_{i-1}. Finally, in the case of condition (3), withdrawing some route at v_{i-1} forces Q_{i-1} to be chosen; thus $R_i^*Q_i \odot_1 Q_{i-1}$, corresponding to the same edge connecting D_i and D_{i-1}. Therefore the dispute-digraph edges corresponding to pairwise relations between paths starting at adjacent rim nodes form a cycle.

Conversely, assume we have a cycle in the dispute digraph. For any edge $(P_1,\ P_2)$, examine the relation between P_1 and P_2. If $P_1 \ominus P_2$, then let the first node of P_1 be a rim node and connect it to the first node of P_2 as an adjacent rim node (counterclockwise, referencing Figure 2.). If $P_1 \oslash P_2$, $P_1 \odot_1 P_2$, or $P_1 \odot_2 P_2$, then let P_2 be a spoke Q_i and connect the first node of P_2 to the first node of P_1 on the rim segment R_{i+1}; the subpath of P_1 from the first node to the last oscillating node will be the rim segment R_{i+1} and the remainder of P_1 will be the next spoke Q_{i+1}. The resulting structure will obey one of the three conditions in Definition 3.1 for rim segments connecting spokes and will have subpaths along individual rim segments (moving clockwise); thus, this structure is the dispute wheel corresponding to the dispute-digraph cycle. ■

This immediately leads to the following corollary, which provides an equivalent sufficient condition to Proposition 3.6.

Corollary 3.10: Given a GSPP instance, if there is a cycle in its evaluation digraph, then the corresponding relation $\bigcirc = (\ominus \cup \oslash \cup \odot_1 \cup \odot_2)^*$ on permitted paths is not a partial order.

Remark 3.11: The linear-selection relation defined in [4] for SPP partial ordering (nonlinear relations did not apply) was defined as follows: assuming that ω is a ranking function, $P_1 \oslash P_2$ iff $\omega(P_1) \leq \omega(P_2)$. In this version, both paths begin at the same node, and the extension of P_1 to u in Definition 3.7 was captured in the transitive closure of $\oslash$ with the subpath relation $\ominus$. If we used an analogous relation here, *i.e.*, $P_1 \oslash P_2$ iff there exists some S such that $\sigma(\{P_1,\ P_2\} \cup S) = P_1$, then any IRR violation would automatically introduce a cycle in the dispute digraph (this fact follows directly from Definition 2.2). But, not all such IRR violations cause protocol oscillations (given other nodes' policies), and subsuming one subpath

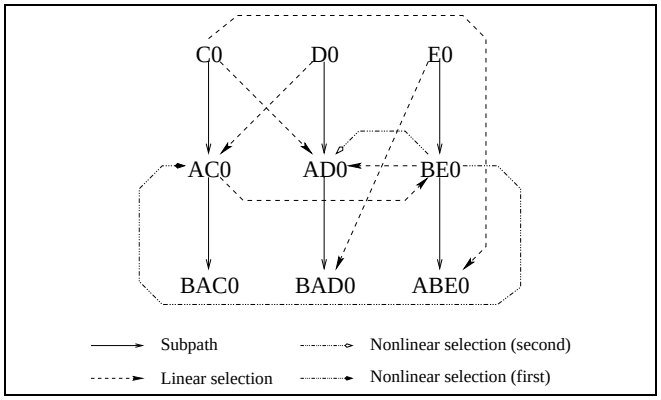


Fig. 3. Generalized dispute digraph for MED-EVIL.

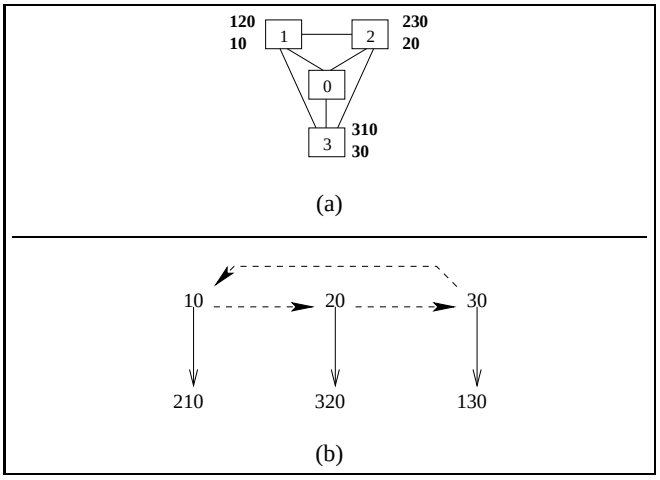


Fig. 4. (a) The SPP instance BAD GADGET and (b) its corresponding generalized dispute digraph.

relation into the selection relations eliminates these spurious cycles from dispute digraphs. Thus, the example dispute cycles in the next subsection will appear different than in [4], [6].

We may in fact generalize the relations $\ominus$ and $\oslash$ to relations $\lhd$ and $\blacktriangleleft$ that also subsume $\odot_2$ and $\odot_1$.

Definition 3.12: Given a GSPP instance, a vertex v, a (possibly empty) set of paths $S \subset \mathcal{P}_v$, and paths $vP, Q, R \in \mathcal{P}_v$, such that $\sigma_v(S) = R$ and $\sigma_v(S \cup \{vP\}) = Q$, the relations $P \lhd Q$ and $P \blacktriangleleft R$ hold.

These relations capture the effects of route export and withdrawal on neighbors' route choices. If S is empty (so that R is the empty path), then $Q = vP$ and we have $P \lhd vP$, which is the subpath relation $P \ominus Q$. If S is not empty and $vP = Q \neq R$, then $P \blacktriangleleft R$ is the linear selection relation $P \oslash R$. If S is non empty and $vP \notin \{Q, R\}$, then the relations $\lhd$ and $\blacktriangleleft$ give the nonlinear selection relations $\odot_2$ and $\odot_1$.

C. Example GSPPs and Dispute Digraphs

Example 3.13: Figure 3 shows the generalized dispute digraph for MED-EVIL, the GSPP from Example 2.8.[1] The graph's nodes are the permitted paths in the instance; edges are drawn between paths for which one of the relations in Definition 3.7 holds; the correspondence between arrow type and path relation is shown below the graph in Figure 3.

For example, the edges $(C0, AC0)$ and $(BE0, ABE0)$ are subpath edges because $AC0$ and $ABE0$ are one-hop extensions of $C0$ and $BE0$, respectively. The edge $(E0, BAD0)$ corresponds to linear selection: the selection function at node B states $\sigma^0_B(BAD0, BE0) = BE0$, which means that, at some time, $BE0$ is preferred to $BAD0$, and $E0$ is the path advertised to B to make $BE0$ available. Likewise, the edge $(BE0, AC0)$ corresponds to $BE0 \odot_1 AC0$: the selection function at node A states $\sigma^0_A(AC0, AD0) = AD0$, but after the addition of $ABE0$, we have $\sigma^0_A(AC0, AD0, ABE0) = AC0$; the broadcast of $BE0$ from B would cause A to switch to a different path it already knew, which is an IRR violation.

Note that this digraph has a cycle $AC0 - BE0$ involving nonlinear-selection edges and the paths that cause the IRR violation; the MED-induced oscillation in MED-EVIL corresponds to this cycle. However, because acyclic digraphs are sufficient—but not necessary—for robustness, the appearance of a cycle, in general, does not guarantee an oscillation.

Example 3.14: A canonical policy-induced oscillation first given by [6] is represented by the SPP BAD GADGET shown in Figure 4; it has no solution, so its dispute digraph also contains a cycle. Because it has a linear selection function, routing policy is shown as a list of permitted paths next to each node (with the most preferred path listed on top). The digraph is not acyclic; indeed, the oscillation in BAD GADGET corresponds to the cycle $10 - 20 - 30$. This cycle is equivalent to the dispute cycle in the original SPP model: The generalized model can characterize instances with or without IRR violations.

IV. Applications to Protocol Design

We now examine some strategies for constraining policies to guarantee robustness. While dispute wheels and dispute digraphs are useful tools for studying policy interactions, they can be impractical for real network configurations. The dispute digraph has size proportional to the number of loopless paths in a network, and there is no known way to directly produce a dispute wheel without an instance's dispute digraph or evaluation digraph. Furthermore, it is almost impossible to obtain Internet-wide policy information to generate these structures, and the structures may change every time nodes change policies. Ideally, we want constraints on the protocol specification or policy-configuration language that applies to a broad set of networks and routing configurations—we would like to use the sufficient condition from the previous section while allowing for as much policy expressiveness as possible.

Previous work [4], [5], [7] has given concrete local-policy constraints that guarantee robustness when MEDs are not used. However, it is difficult to generalize this work to GSPPs because these constraints use notions of order and path rank that need not be present with nonlinear selection functions; we thus consider other constraints for GSPPs. Some obvious, draconian constraints, *e.g.*, preventing the advertisement of

[1]To simplify the diagram, we have condensed ASes 1, 2, and 0 into a single AS 0 connected to routers C, D, and E; we can write analogous selection functions that maintain the oscillation in the original MED-EVIL.

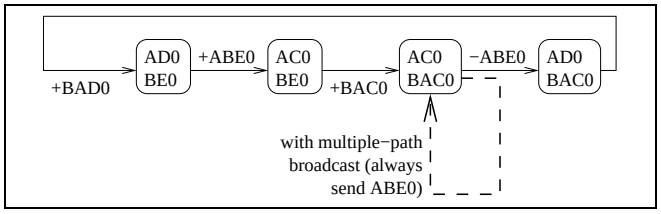


Fig. 5. Cycle in the evaluation digraph of MED-EVIL.

any route that causes an IRR violation, can be trivially shown to prevent routing anomalies, but these harshly limit expressive power. Below, we first review a specific proposal to prevent MED-induced oscillations in BGP, and we use our tools to suggest an improvement. Then, we discuss two other conjectured solutions and prove them correct using our results.

A. Multiple-Path Broadcast

Basu *et al.* [8] and Musunuri and Cobb [10] proved that a modification to BGP's update messages will prevent MED-induced oscillations. They suggested that nodes broadcast not only best routes but also any route that remains after step 3 in the BGP route-selection process (see Example 2.8); *i.e.*, all routes with minimal MED values, possibly one for each AS, are broadcast, not only the one with minimal IGP distance to the egress point. This prevents routes that cause IRR violations from being broadcast and withdrawn repeatedly. In the case of MED-EVIL in Example 2.8, node B would then always broadcast the route $BE20$, even though it would not select it if $BAC0$ were known.[2] Extensions of $BE20$ are not chosen elsewhere because they are longer than corresponding extensions of $BAC0$; so, this introduces no consistency problems. However, it (1) allows other nodes to make the correct choice of routes with respect to MED values and (2) stops the oscillation by making that choice stable. We also note that only one route is chosen for the forwarding table; thus, we need not worry about routing loops. The additional routes are used only to force the correct route choice.

We can see the effect of such a change by examining cyclic traces in the evaluation digraph. The MED-induced cycle of MED-EVIL is shown in Figure 5. The nodes show the selections of nodes A and B, and the labels on arrows show the causes of transitions (routes being advertised, denoted with a $+$, or withdrawn, denoted with a $-$). The IRR violation is clear in the transition between the first and second states; node A switches from $AD0$ to $AC0$ by learning a different route, $ABE0$. With multiple-path broadcast, the withdrawal of $ABE0$ never takes place; therefore the state $(AC0, BAC0)$ becomes a sink state and a stable assignment.

This effect generalizes to all GSPP instances involving MEDs: broadcasting additional routes will break an evaluation-digraph cycle by allowing nodes to receive MED values they otherwise would not, thus preventing one (or more) of the cycle's transitions. Because routes are always added to (not removed from) the broadcast, nodes will not choose higher-MED-valued routes when lower-MED-valued routes are available; this preserves the intended behavior of the MED attribute.

Multiple-path broadcast can increase the size of routing tables and update messages. However, we propose that IRR violations can be detected dynamically, precisely when a newly learned route causes a switch in selection without selecting the new route. Requesting that the new route always be broadcast will prevent a future oscillation due to withdrawal of that route without any route inconsistencies. Maintaining one extra route as needed is more storage-efficient than the multiple-path broadcast proposed by [8], [10]. Although this solution requires further modification to BGP, dynamic detection of IRR violations is possible in practice. Whenever a BGP update message is received, the route selection before and after the update message can be compared. If the new selection is neither the old selection nor the newly learned route, this points to an IRR violation (this is clear from Definition 2.2). Requesting this IRR-violating route to be broadcast as long as it is available prevents any induced oscillations because the route essentially becomes fixed, breaking the cycle of withdrawals and advertisements in the evaluation digraph. Formally, we have the following.

Proposition 4.1: An oscillation due to an IRR violation can be dynamically detected and stopped by requesting one additional route to be broadcast permanently.

Proof: Given a cycle in the evaluation digraph involving an IRR violation, there are transitions in this cycle involving an advertisement or withdrawal of a route that is never selected. This route can be detected by comparing path assignments in the states adjacent to these transitions. If the withdrawal transition is prevented by forcing the route to be advertised as long as it is available, even if it is not chosen, the withdrawal transition cannot take place and the cycle is broken. ■

If changes occur and routes are introduced or withdrawn for legitimate causes, the resulting GSPP instance will have a different evaluation digraph; however, the relevant IRR-violating routes can be detected for this new instance in the same way. If the IRR-violating route is no longer available, the broadcasting node can send the appropriate withdrawal—this still allows the receiving node to detect new IRR violations involving other routes. Furthermore, if any IRR-violating selections are superseded by learning new routes that are always more preferred or by other IRR-violating routes, the original routes are not needed and the broadcast can be stopped.

B. Compare All MEDs

Some routers have an option to change the route-selection procedure involving MEDs: In step 3 of the BGP procedure described in Example 2.8, instead of eliminating multiple paths to the same AS by choosing the one with lowest MED value, MED values are compared across all paths so that, regardless of AS next-hop, only paths with the lowest MED values are retained for possible selection.

[2] As in Example 3.13, we simplify the instance by condensing AS 1, AS 2, and AS 0 into a single AS 0 and modifying the selection functions accordingly.

This option changes the route-selection procedure so that it is linear: for each path, the preference of that path depends, in order, on its local preference, then path length, then MED value, and finally IGP distance. Therefore, IRR violations are no longer possible, and previous convergence constraints apply. In fact, because local-preference, AS-path length, and MED values do not change during intra-domain BGP (iBGP) sessions, and because IGP distances increase as paths are extended, the absolute rank value associated with paths increases on extension within an AS. This obeys the strict-monotonicity constraints of [4], [7], so MED-induced oscillations cannot occur. (Of course, more general policy-induced oscillations due to, *e.g.*, local-preference settings, can still occur.)

C. AS-Distinct Local-Preference Settings

McPherson *et al.* in [15] suggest a workaround for MED-induced oscillations that prevents BGP from having a conflict when it reaches the MED step. If only routes from one AS remain when MEDs are considered, then all routes have their MED values compared and, similar to above, IRR violations are not possible. One simple way to do this is to assign local-preference values such that no two routes from different ASes have the same value; then the first step of the BGP selection process will automatically eliminate all routes except those from a single AS. (One can also assign distinct local-preference values to equidistant ASes; then the first two steps eliminate all routes but those from one AS.)

This route-selection procedure is, in fact, consistent with linear selection functions because, just as above, the rank of a route independently depends on four criteria in order. Once the MED value is considered, all remaining routes have the same local preference, path length, next-hop AS, and MED value, again leaving the strictly monotonic IGP distance to be used to break ties. Therefore, this modification to BGP prevents MED-induced anomalies.

V. Conclusions and Future Work

In this paper, we investigated the use of the Generalized Stable Paths Problem (GSPP) to model route-selection functions that violate Independent Route Ranking (IRR); in particular, this model allowed us to analyze networks in which BGP's MED attribute is used and facilitated studying the interaction between inter- and intra-domain routing. Using this model, we generalized the classical dispute wheel and used our results to provide the broadest-known sufficient condition for robust routing in networks, whether or not they exhibit MED-like behavior. We also used our work to gain insight into various proposed solutions to the MED-oscillation problem.

Depending on assumptions made about filtering, *etc.*, one may construct examples in which MEDs are used but no oscillation occurs, MEDs are used but no IRR violation occurs, IRR violations occur but no oscillation occurs, or MEDs are used and an oscillation occurs even when an IRR violation does not. Whenever an oscillation occurs, regardless of the cause, the tools in this paper can characterize the corresponding policy dispute. However, it is not yet possible to use these tools alone to diagnose oscillations; this direction of research remains open, even when IRR violations are ignored.

Here we have focused on singleton-valued selection functions, but another avenue for future work is extending the theory to set-valued functions and understanding the consequences on multi-path routing; IRR is more complex in this setting (recall Definition 2.2).

Acknowledgments

This work was supported by the U.S. Department of Defense (DoD) University Research Initiative (URI) program administered by the Office of Naval Research (ONR) under grant N00014–01–1–0795. A. D. Jaggard was also supported by ONR Grant N00014–05–1–0818 and by National Science Foundation (NSF) Grant DMS–0239996. V. Ramachandran was also supported by NSF grant CNS–0524139 and by the Stevens Technogenesis program; work done in part while at Yale University. We thank the referees for their suggestions.

References

[1] Y. Rehkter and T. Li, "A border gateway protocol (BGP version 4)," RFC 1771, 1995.

[2] K. Varadhan, R. Govindan, and D. Estrin, "Persistent route oscillations in inter-domain routing," *Computer Networks*, vol. 32, pp. 1–16, 2000.

[3] C.-k. Chau, R. Gibbens, and T. G. Griffin, "Towards a unified theory of policy-based routing," in *Proceedings of IEEE INFOCOM 2006*, IEEE Communications Society. IEEE Press, April 2006.

[4] T. G. Griffin, A. D. Jaggard, and V. Ramachandran, "Design principles of policy languages for path vector protocols," in *Proceedings of ACM SIGCOMM'03*. ACM Press, August 2003, pp. 61–72.

[5] L. Gao and J. Rexford, "Stable internet routing without global coordination." *ACM/IEEE Transactions on Networking*, vol. 9, no. 6, pp. 681–692, December 2001.

[6] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *ACM/IEEE Transactions on Networking*, vol. 10, no. 2, pp. 232–243, April 2002.

[7] J. L. Sobrinho, "Network routing with path vector protocols: Theory and applications," in *Proceedings of ACM SIGCOMM'03*. ACM Press, August 2003, pp. 49–60.

[8] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route oscillations in I-BGP with route reflection," in *Proceedings of ACM SIGCOMM'02*, November 2002, pp. 235–247.

[9] T. G. Griffin and G. Wilfong, "An analysis of the MED oscillation problem in BGP," in *Proceedings of the 10th International Conference on Network Protocols (ICNP'02)*, November 2002, pp. 90–99.

[10] R. Musunuri and J. A. Cobb, "A complete solution for iBGP stability," in *Proceedings of IEEE ICC-04*. IEEE Press, June 2004.

[11] L. Gao, T. G. Griffin, and J. Rexford, "Inherently safe backup routing with bgp," in *Proceedings of IEEE INFOCOM 2001*, IEEE Communications Society. IEEE Press, 2001.

[12] N. Feamster and H. Balakrishnan, "Towards a logic for wide-area internet routing," in *Proceedings of ACM SIGCOMM Workshop on Future Directions in Network Architecture FDNA*, August 2003, pp. 289–300.

[13] Cisco Systems, "Endless BGP convergence problem in Cisco IOS software releases," Field Note, October 2001, http://www.cisco.com/warp/public/770/fn12942.html.

[14] R. Dube and J. G. Scudder, "Route reflection considered harmful," November 1998, iETF Internet Draft.

[15] D. McPherson, V. Gill, D. Walton, and A. Retana, "Border gateway protocol (BGP) persistent route oscillation condition," RFC 3345, August 2002.

[16] D. Walton, D. Cook, A. Retana, and J. Scudder, "BGP persistent route oscillation solution," May 2002, IETF Internet Draft.

[17] A. D. Jaggard and V. Ramachandran, "Robustness of path-vector protocols without independent route ranking," Yale University, Tech. Rep. YALEU/DCS/TR-1314, February 2005, ftp://ftp.cs.yale.edu/pub/TR/tr1314.pdf.