Analysis of the MED Oscillation Problem in BGP

Timothy G. Griffin — AT&T Research
Gordon Wilfong — Bell Labs

Abstract

The Multi Exit Discriminator (MED) attribute of the Border Gateway Protocol (BGP) is widely used to implement "cold potato routing" between autonomous systems. However, the use of MED in practice has led to BGP persistent oscillation. The MED oscillation problem has been described with example configurations and complicated, step-by-step evaluation of dynamic *route computations performed at multiple routers. Our work presents the first rigorous analysis of the MED oscillation problem. We employ the Stable Paths Problem (SPP) formalism that allows a* static *analysis of the interaction of routing policies. We give a formal definition of MED Induced Routing Anomalies (MIRA) and show that, in general, they can span multiple autonomous systems. However, if we assume that the BGP configurations between ASes follows a common model based on customer/provider and peer/peer relationships, then we show that the scope of any MIRA is always contained within a single autonomous system. Contrary to widely held assumptions, we show that a MIRA can occur even in a fully meshed IBGP configuration. We also show that a stable BGP routing may actually violate the stated semantics of the MED attribute.*

1 Introduction

The Border Gateway Protocol (BGP) [15, 11, 17] is used to maintain connectivity between Autonomous Systems in the Internet. Normally, a BGP speaking router within a single autonomous system (AS) uses internal (IGP) metrics in a tie breaking step when it selects its best routes. By default, BGP selects the closest egress point, a technique known as *hot potato routing*, which is often appropriate. In those cases where hot potato routing is not optimal, the Multi Exit Discriminator (MED) attribute of BGP provides a means of implementing *cold potato routing* between ASes. However, BGP divergence associated with the use of the MED attribute has been encountered in practice [14, 4].

Since BGP can diverge without the use of MED [19], it is natural to ask if there is anything special about this attribute that makes its use more likely to cause divergence problems. Or are these problems really the result of some type of feature interaction? The accounts of [14, 4] imply that MED oscillations result from an interaction between implementing cold potato routing with MEDs and features designed to scale the internal IBGP mesh within an AS (route reflection [2] or confederations [18]).

The MED attribute is treated in a unique manner by the BGP route selection algorithm. The specification of BGP in RFC 1771 [15] states that the route selection process should conform to what we will call the *rule of independent ranking*. This rule states that a route's ranking should not depend on the existence or nonexistence of other routes in the routing table. The MED attribute is unique in that its use in the BGP route selection process violates the rule of independent ranking (see examples in [14] and in Section 2). Why should the rule of independent ranking be important for BGP? Although RFC 1771 is silent about the motivation for this rule, it requires little justification for anyone having experience writing complex BGP routing policies or debugging BGP routing problems caused by unexpected policy interactions. When this rule is violated, network engineers have little hope of understanding the semantics of their routing policies. In other words, we claim that this rule is important in terms of *human factors* — it enables policy writers and network operators to develop firm intuitions about the operational consequences of their policy configurations. We argue that this is at the heart of the problem with use of the MED attribute — its semantics are so counter-intuitive and combinatorially complex that operators cannot understand the actual semantics of their router configurations.

This has implications beyond BGP and cold potatoes. BGP represents a widely deployed instance of a *vectoring* protocol that allows *local and independent policy control*. Other protocols in this family are now emerging, such as protocols for optical inter-networking [3] and Telephony Routing over IP (TRIP) [16]. The MED oscillation problem sends a clear warning to protocol designers that there are consequences associated with violating the rule of independent ranking.

Our Contributions. The descriptions of MED oscillation in [14, 4] rely on a few example configuration scenarios in which the oscillation is described using rather complicated step-by-step explanations. The goal of the current paper is to formulate a mathematical model of BGP policies that will allow us to formally define MED related routing anomalies and to examine proposed solutions rigorously and in a general way, without dealing directly with complicated dynamic scenarios.

Our approach is a formal one, based on the Stable Paths Problem (SPP) [7, 8]. The SPP is a simple framework for the analysis of interaction of BGP routing policies that abstracts away from the protocol specific details, such as the BGP attribute set, the BGP route selection procedure, and import/export policy transformations. In fact, the SPP model can be thought of as a formalism for capturing the "policy semantics" of any vectoring protocol that follows the rule of independent ranking. Since MEDs do not obey this rule, we are forced to generalize the SPP model and define the General Stable Paths Problem (GSPP), which is then used to directly model the semantics of the MED attribute. We then show that MEDs can be modeled indirectly using the SPP, via a translation from GSPP to SPP. This translation involves two steps and requires the creation of new nodes that act as "MED evaluation proxies" for other nodes. Representing BGP with MEDS in SPP, even indirectly, then allows us to apply the theory of [7, 8] to the analysis of MEDs. For example, we show that if all ASes use routing policies that conform to the customer/provider and peer/peer routing model of [12, 13, 5, 6], then the scope of each MED induced routing anomaly is confined to a single autonomous system.

The use of SPP also allows us to broaden the class of routing anomalies that we investigate. There are three main sanity conditions we would like a BGP system to satisfy. First, it should be *uniquely solvable*. That is, it should have a solution and this solution should be unique. In practice, this aids greatly in debugging routing problems, especially when they span multiple autonomous systems. Without uniqueness, the solution the protocol arrives at depends on the random ordering of routing messages. Second, we would like the system to be *safe*, meaning that the protocol always converges to a solution. Finally, the system should be *robust* in the sense that it remains uniquely solvable and safe under any combination of router or link failures. We show that, even in a fully meshed IBGP configuration, the use of MEDs can lead to multiple solutions and that these solutions are not guaranteed to conform to the stated semantics of the MED attribute. In other words, we show that the MED related routing problems go beyond the interaction of MED with IBGP mesh optimizations.

Related work. Our work is complimentary to the analysis of IBGP in [10], which presents correctness constraints for IBGP ignoring the MED attribute. As is pointed out in Section 6, a satisfactory unification of the work presented here and in [10] remains an open problem.

Our approach is to study the MED attribute as it is currently defined in RFC 1771 and to use that understanding to formulate configuration constraints that will help avoid MED related routing problems. Another approach, taken in [1], is to modify the definition of IBGP in order to eliminate the problem. The protocol modification described in [1] would augment IBGP with announcements that are not best routes, but carry enough MED related information to eliminate MED induced oscillations. In general, this comes at the price of increased memory consumption for all BGP speaking routers within an AS.

Paper outline. In Section 2 we outline the problems with MED that are discussed in [14, 4]. However, we present a *static* analysis of the problem rather than a dynamic one. In Section 3 we define the General Stable Paths Problem (GSPP) and show that it can model BGP with MEDs. Section 3.4 shows that MED can be involved in various kinds of routing anomalies besides oscillation. Section 4 shows how a GSPP representing BGP with MEDs can be encoded in an SPP. The representation involves a two step translation that requires the creation of new nodes that act as "MED evaluation proxies" for other nodes. In Section 5 we define a MED Induced Routing Anomaly (MIRA) and show that if we assume the customer/provider and peer/peer routing model, then the scope of every MIRA is confined to a single autonomous system. These results rely on the SPP theory of [5, 6].

2 MED Explained

This section reviews the route selection process of BGP [15, 11, 17], describes how MED is used, and then presents the MED oscillation problem described in [14, 4]. However, unlike [14, 4], we provide a *static* explanation as to why the system has no stable routing rather than a *dynamic* one.

2.1 BGP Routes and Route Selection

BGP route announcements are records with a varying number of attributes. One attribute represents a destination network — a collection of IP addresses represented as a CIDR block — and the associated attributes convey information about the destination or the path to the destination. These route attributes are used in two related ways. First, they are used in the implementation of *routing policy* as described below. Second, they are used in the best route selection algorithm. The attributes used in this paper are the network layer reachability information ($\mathbf{nlri}$, a CIDR block), local preference ($\mathbf{local\_pref}$), the IP address of the BGP next hop ($\mathbf{next\_hop}$), an ordered list of ASes traversed by the route announcement, called the ASPATH ($\mathbf{as\_path}$), and the Mult-Exit Discriminator attribute ($\mathbf{med}$). The local preference attribute $\mathbf{local\_pref}$ is not passed between autonomous systems, but is used internally within an autonomous system to assign a local degree of preference.

We use notation taken from [9, 7]. When router u passes a BGP record r to router w it can be modified in various ways. We will model this using *route transformations*, which are functions mapping route records to route records, $\mathcal{T}(r) = r'$. A route transformation can delete, insert, or modify the attribute values of its input r. If $\mathcal{T}(r) = \langle\rangle$ (representing the empty record), then r has been *filtered out* by $\mathcal{T}$. In order to allow composition of all transformations we insist that $\mathcal{T}(\langle\rangle) = \langle\rangle$. When a record r is sent from router w to router u it is modified by three distinct transformations. First, $r_1 = E^u_w(r)$ is w's *export policies to* u. Second, $r_2 = B^u_w(r_1)$ is a BGP-specific transformation described below. Finally, $r_3 = I^w_u(r_2)$ is u's *import policies for* w. When u and w are in different ASes, this function assigns a $\mathbf{local\_pref}$ value for r_3. We call the composition of these three transformations the *neighbor transformation*,

$$N_u^w(r) = I_u^w(B_w^u(E_w^u(r))).$$

The BGP-specific transformation $B_w^u(r_1)$ has two modes of operation. If w and u are in different autonomous systems, then it adds the AS number of w to the $\mathbf{as\_path}$ of r_1 and resets $\mathbf{next\_hop}$ to an interface on router w. It also filters out the record if its $\mathbf{as\_path}$ contains the AS number of router u. (This prevents many types of interdomain forwarding loops. However, interdomain forwarding loops still can be created by the use of aggregation and default routing [11].) If w and u are in the same autonomous systems, then the transformation will only filter out routes that are not allowed due to the rules of Internal BGP (IBGP) operation. That is, if r_1 has been learned by w from another router z inside the same autonomous system as w, then $B_w^u(r_1) \neq \langle\rangle$ only if w is a route reflector and either z or u is a client of w. This allows a route reflector to "reflect" routes from its clients to other routers and to receive IBGP routes and "reflect" them to its clients. In *fully meshed IBGP*, no route reflectors are used and a router only uses IBGP to communicate its best External BGP (EBGP) routes to other routers within an AS.

When a BGP speaking router receives two or more routes to a given destination, it must select no more than one as its best route. (Recall that it may not select any route if all are filtered out for policy reasons.) Only best routes are announced to neighboring BGP speakers. The selection of a best route is based on the attribute values contained in the route announcements. Suppose S_0 is a set of routes to the same destination learned by a BGP speaking router u. We denote the best route of S_0 at u by select(u, S_0). This selection algorithm can be described as follows. The algorithm constructs a sequence of sets $S_{i+1} \subseteq S_i$, and terminates at any point when S_i contains a single route, which is the best route of set S_0.

1. S_1 is the set of all routes in S_0 with a maximal value of $\mathbf{local\_pref}$,

2. S_2 is the set of all routes in S_1 with the minimal $\mathbf{as\_path}$ length,

3. for each neighbor AS n, let $S_2[n]$ be the set of routes in S_2 sent from AS n. Let $S_3[n] \subseteq S_2[n]$ be all routes from $S_2[n]$ that have the minimal value of $\mathbf{med}$ among routes in $S_2[n]$. Let S_3 be the union of all $S_3[n]$. That is, $\mathbf{med}$ values can only be compared between routes from the same next hop AS, and lower $\mathbf{med}$ values are preferred.

4. S_4 is obtained from S_3 by eliminating all IBGP routes only if there is at least one EBGP route in S_3.

5. S_5 is the set of routes in S_4 with minimal IGP distance to their BGP $\mathbf{next\_hop}$ attribute.

6. use some deterministic tie breaker, such as highest neighbor router ID, or lowest BGP next hop to select one route from S_5.

Note that without step (3), we could use the BGP attributes to lexically rank each route, independent of the existence or non-existence of other routes. That is, the selection process would conform to the rule of independent ranking. However, with step three this is only true if for every "MED-icated" route, the associated prefix is received in routes from a unique neighbor AS.

2.2 The Benefits of MED

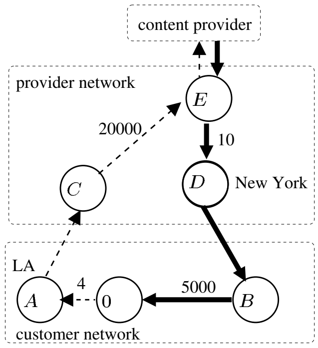

Figure 1. HOT POTATO

Without MED, BGP will use what is called *hot potato routing*. This is illustrated in Figure 1. The lower network represents a customer network, with router A in Los Angeles and Router B in the New York area. There is a client attached to router 0, which is close to the LA router. The middle network is the customer's network provider, a large ISP. The network at the top represents a content provider that serves high bandwidth data. The numbers above the links represent IGP distances between routers. Assume that router 0 learns about a destination in the content provider's network from its BGP sessions (IBGP) with routers A and B. When the client at 0 sends a request to the content provider, it will be routed to the network provider through router A, since A is closer to 0 than B (hot potato). The network provider will route the request from router C to E, and then on to the content provider. The path of the request is indicated with a dashed line. The path of the content provider's response is drawn with a thick line. When this data arrives at router E in the provider's network it will be routed to the customer through router D, using the hot potato rule. This data is then carried across country (from B to 0) on the customers lower bandwidth leased lines. From the customer's point of view this may not make economic sense. After all, it is paying the network provider to carry bits!

MED Customer Scenario: This problem can be solved with BGP using the MED attribute as illustrated in Figure 2. The annotation (4) indicates that on the LA link a MED value of 4 is sent to the provider network with the route to the client at 0. The annotation (5000) indicates that on the New York link a MED value of 5000 is sent to the provider network with the route to the client at 0. (Here the MED values are identical to the IGP distances in the customer's network. Although this is a typical configuration, MEDs do not have to be tied to IGP

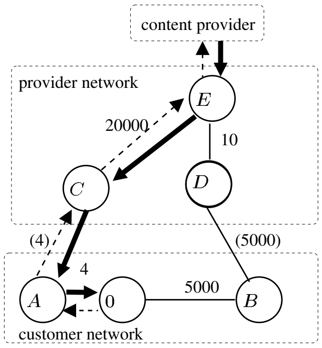

Figure 2. COLD POTATO

distances. However, the fact that they often are explains why lower MED values are preferred and why MED values from different ASes cannot be compared.) Router E will learn two routes to 0, one through router C and one through router D. The MED step in the selection process at router E will eliminate the route through D, since it has a higher MED value. Therefore, router E will route the traffic to router C as desired. Router E is said to be using *cold potato routing* since it is not throwing this traffic off of its own network as quickly as it could by sending it to router D.

MED Peer Scenario: Although this represents a common use of MED, it is not the only way it is used. Suppose that we modify the example just given so that the lower and middle networks are both large ISPs that have decided to enter into a *peering* relationship where they exchange traffic between their respective customers free of change. If the lower network thinks that it has invested more in its backbone intrastructure, it may insist as a part of the peering agreement that the middle network accept its MED values so that the middle network (a competitor!) does not get a "free ride" on its network investment.

2.3 The problem with MED

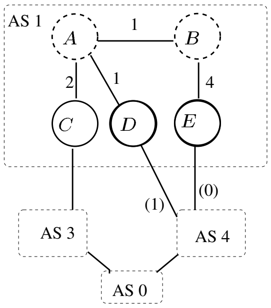

Figure 3. The system MED-EVIL**.**

If we ignore the MED step, then BGP route selection amounts to a type of lexical ranking of routes. That is, without MEDs, it is possible to select a route in two steps: first lexically rank all routes, then select the route with the highest rank. However, with MEDs present, this is no longer the case. A route's ranking can vary depending on the presence or absence of other routes. This makes reasoning about BGP with MED very difficult. For example, without MED we could process incoming routes simply by comparing each route with the current best route. If the new route has a lower rank than the current best route, we can simply store it as an inactive route. If it has a higher rank than the current best route, we can replace the best route with the new route. With MEDs, however, this simple processing no longer is valid. A newly learned route can eliminate the current best route from consideration even though the new route does not become the best route!

In addition, MED values can interact in complex ways causing the protocol to diverge. To illustrate this, consider the configuration of Figure 3, taken from [14]. We will call this example MED-EVIL. AS 1 contains five routers, A and B, which are route reflectors, C and D, which are clients of A, and E, which is a client of B. The numbers next to the links in AS 1 represent IGP weights. Routers in AS 4 send two BGP routes to AS 1: the route sent to router D has MED value 1, while the route to router E has MED value 0.

u	S	select(u, S)	due to
A	$\{r_C,\ r_D\}$	r_D	IGP
A	$\{r_D,\ r_E\}$	r_E	MED
A	$\{r_E,\ r_C\}$	r_C	IGP
A	$\{r_C,\ r_D,\ r_E\}$	r_C	MED, IGP
B	$\{r_D,\ r_E\}$	r_E	MED
B	$\{r_E,\ r_C\}$	r_C	IGP

Table 1. Best route selection at nodes A and B.

Consider route selection at routers A and B. Let r_C, r_D and r_E be the routes that A receives from routers C, D, and E, respectively. (A receives route r_E through route reflector B, while B receives routes r_C and r_D through route reflector A.) Table 1 shows the best route selections for each of the multi-route combinations that can be learned at routers A and B.

Note that we cannot rank these routes in a linear manner, since this would require that $rank(r_C) < rank(r_D) < rank(r_E) < rank(r_C)$. We now show that there can be no stable routing for these selection functions. If we assume that A always has learned routes r_C and r_D, then there are two cases to consider. Case 1: router A knows the routes $\{r_C,\ r_D,\ r_E\}$ and so selects r_C as its best route. The route set at A implies that B has chosen r_E as its best route. But this is a contradiction, since B would learn the set $\{r_E,\ r_C\}$ and select r_C. Case 2: router A learns only routes $\{r_C,\ r_D\}$ and so selects r_D as its best route. Since A does not learn a route from router B, we know that B must have selected r_C. But again this is a contradiction since B would learn r_D from A and then pick r_E as its best route. We have just demonstrated that there is no solution to this system. In practice, this configuration would cause BGP to *diverge*. That is, routers A and B would endlessly exchange update messages without arriving at a stable routing. A tedious step-by-step simulation of this type of routing exchange is presented in [14, 4].

3 A Model of BGP with MEDs

In this section we present a formal model of BGP with MEDs. This allows a formal treatment that can ignore all of the complexities associated with attributes, the route selection procedure, and import/export policy transformations. We first introduce some basic notation concerning paths in Section 3.1. Then Section 3.2 introduces the General Stable Paths Problem (GSPP), which allows a best path to be a function of the *set* of available paths. In Section 3.3, we review the Stable Paths Problem (SPP) from [7], which is a special case of the GSPP where the set selection function at each node is induced by a linear ranking of paths at that node.

3.1 Permitted Paths and Path Assignments

Let $G = (V, E)$ be a simple, undirected graph where $V = \{0, 1, 2, \ldots, n\}$ is the set of nodes and E is the set of edges. For any node u, $\text{peers}(u) = \{w \mid \{u, w\} \in E\}$ is the set of *peers* for u. We assume that node 0, called the *origin*, is special in that it is the destination to which all other nodes attempt to establish a path.

A *path* in G is either the empty path, denoted by ϵ, or a sequence of nodes, $(v_k\ v_{k-1}\ \ldots\ v_1\ v_0)$, $k \geq 0$, such that for each i, $k \geq i > 0$, $\{v_i, v_{i-1}\}$ is in E. Note that if $k = 0$, then (v_0) represents the trivial path consisting of the single node v_0. Each non-empty path $P = (v_k\ v_{k-1}\ \ldots\ v_1\ v_0)$ has a direction from its *first node* v_k to its *last node* v_0. If P and Q are non-empty paths such that the first node in Q is the same as the last node in P, then PQ denotes the path formed by the *concatenation* of these paths. We extend this with the convention that $\epsilon P = P\epsilon = P$, for any path P. For example, $(4\ 3\ 2)\ (2\ 1\ 0)$ represents the path $(4\ 3\ 2\ 1\ 0)$, whereas $\epsilon\ (2\ 1\ 0)$ represents the path $(2\ 1\ 0)$. This notation is most commonly used when P is a path starting with node v and $\{u, v\}$ is an edge in E. In this case $(u\ v)P$ denotes the path that starts at node u, traverses the edge $\{u, v\}$, and then follows path P from node v.

For each $v \in V$, $\mathcal{P}^v$ denotes the set of *permitted paths* from v to the origin (node 0). All permitted paths are assumed to be simple paths (no repeated nodes), and for each $v \in V$ we have $\epsilon \in \mathcal{P}^v$. If $P = (v\ v_k\ \ldots\ v_1\ 0)$ is in $\mathcal{P}^v$, then the node v_k is called the *next hop* of path P. Let $\mathcal{P}$ be the union of all sets $\mathcal{P}^v$. We assume that $\mathcal{P}^0 = \{(0)\}$.

A *path assignment* is a function π that maps each node $u \in V$ to a path $\pi(u) \in \mathcal{P}^u$. We assume that it is always the case that $\pi(0) = \{(0)\}$. We interpret $\pi(u) = \epsilon$ to mean that u is not assigned any path to the origin.

The set of paths candidates(u, π) is defined as follows. For $u = 0$, candidates$(u, \pi) = \{(0)\}$, and for $u \neq 0$,

$$\text{candidates}(u, \pi) = \mathcal{P}^u \cap \{(u\ v)Q \mid Q = \pi(v) \text{ and } \{u,\ v\} \in E\}.$$

This set represents all permitted paths at u that can be formed by extending the paths assigned to the neighbors of u.

3.2 The General Stable Paths Problem (GSPP)

Suppose $G = (E, V)$ is a graph with permitted paths $\mathcal{P}$. A *path selection function for node* u is a function σ_u that maps any set of permitted paths $W \subseteq \mathcal{P}^u$ to the "best path" in $W \cup \{\epsilon\}$. We insist that $\sigma_u(\phi) = \epsilon$, and for any $W \neq \phi$ that we have

$$\sigma_u(W) \in W.$$

Note that this implies that for any $P \in \mathcal{P}^u$, $\sigma_u(\{P\}) = P$. An instance of the *General Stable Paths Problem*, $S = (G,\ \mathcal{P},\ \Sigma)$, is a graph together with the permitted paths at each node and and a set of path selection functions $\Sigma = \{\sigma_u \mid u \in V\}$.

A path assignment π is a *solution* for a GSPP if for every node u we have

$$\boxed{\pi(u) = \sigma_u(\text{candidates}(u, \pi)).}$$

That is, if F is a functional that takes path assignments π to path assignments $F(\pi)$, defined as $F(\pi)(u) = \sigma_u(\text{candidates}(u, \pi))$, then the solutions of the GSPP are exactly the fixed points of F ($F(\pi) = \pi$ implies π is a solution, and for any solution π we have $F(\pi) = \pi$). A convenient abbreviation for the best path at u under π is defined to be $\text{best}(u, \pi) = \sigma_u(\text{candidates}(u, \pi))$. Then π is a solution if $\pi(u) = \text{best}(u, \pi)$ at each node u.

Given a BGP system $\mathcal{B}$ we can represent it as a GSPP system, denoted $GSPP(\mathcal{B})$, as follows. The nodes in the graph represent routers, and the links represent BGP sessions. One node, denoted 0, is the origin. If P is a path from some node u_k to 0 in this graph, then let route(P) denote the route BGP generated along path P. That is, if $P = (u_k\ u_{k-1} \cdots u_1\ u_0)$, then we define route$(P)$ is defined to be

$$N_{u_k}^{u_{k-1}}(N_{u_{k-1}}^{u_{k-2}}(\cdots N_{u_1}^{u_0}(r_o)\cdots)).$$

The permitted paths at node u are all paths P that start at the origin 0 such that route$(P) \neq \langle\rangle$.

Conversely, if r is a BGP route, let path(r) denote the path that generates r. That is, define this function so that path(route$(P)) = P$. For each subset $W \subseteq \mathcal{P}^u$ we define $\sigma_u(W)$ as follows. Let path$(W) = \{\text{path}(P) \mid P \in W\}$. Then $\sigma_u(W) = \text{path}(\text{select}(u, \text{route}(W)))$.

For example, assuming that nodes 3 and 4 only permit their direct paths to 0, the permitted paths of $GSPP(\text{MED-EVIL})$ are

$$\begin{array}{lcl}
\mathcal{P}^A & = & \{(A\ C\ 3\ 0),\ (A\ D\ 4\ 0),\ (A\ B\ E\ 4\ 0)\} \\
\mathcal{P}^B & = & \{(B\ E\ 4\ 0),\ (B\ A\ C\ 3\ 0),\ (B\ A\ D\ 4\ 0)\} \\
\mathcal{P}^C & = & \{(C\ 3\ 0),\ (C\ A\ D\ 4\ 0),\ (C\ A\ B\ E\ 4\ 0)\} \\
\mathcal{P}^D & = & \{(D\ 4\ 0),\ (D\ A\ C\ 3\ 0),\ (D\ A\ B\ E\ 4\ 0)\} \\
\mathcal{P}^E & = & \{(E\ 4\ 0),\ (E\ B\ A\ C\ 3\ 0),\ (E\ B\ A\ D\ 4\ 0)\} \\
\mathcal{P}^3 & = & \{(3\ 0)\} \\
\mathcal{P}^4 & = & \{(4\ 0)\}
\end{array}$$

In defining the selection functions $GSPP(\text{MED-EVIL})$ we need only consider the definition of σ_u on $W \subseteq \mathcal{P}^u$ where

u	W	$\sigma_u(W)$	due to
A	$\{(A\ C\ 3\ 0),\ (A\ D\ 4\ 0)\}$	$(A\ D\ 4\ 0)$	IGP
A	$\{(A\ D\ 4\ 0),\ (A\ B\ E\ 4\ 0)\}$	$(A\ B\ E\ 4\ 0)$	MED
A	$\{(A\ C\ 3\ 0),\ (A\ B\ E\ 4\ 0)\}$	$(A\ C\ 3\ 0)$	IGP
A	$\{(A\ C\ 3\ 0),\ (A\ D\ 4\ 0),\ (A\ B\ E\ 4\ 0)\}$	$(A\ C\ 3\ 0)$	MED, IGP
B	$\{(B\ E\ 4\ 0),\ (B\ A\ D\ 4\ 0)\}$	$(B\ E\ 4\ 0)$	MED
B	$\{(B\ E\ 4\ 0),\ (B\ A\ C\ 3\ 0)\}$	$(B\ A\ C\ 3\ 0)$	IGP

Table 2. Two selection functions for $GSPP$(MED-EVIL).

no two paths in W share the same neighbor (next hop node). Furthermore, since $\sigma_u(\{P\}) = P$, we need only explicitly define selection for sets of two or more paths. Table 2 presents the selection functions for nodes A and B. These correspond to the BGP route selections of Table 1.

Let erase($\mathcal{B}$) be the BGP system where all MED values have been erased. For example, the path selection functions of $GSPP$(erase(MED-EVIL)) do have a unique solution based on shortest path routing where A takes path $(A\ D\ 4\ 0)$ and B takes path $(B\ A\ D\ 4\ 0)$.

3.3 The Stable Paths Problem (SPP)

The *Stable Paths Problem* (SPP) [8] corresponds to a special class of GSPP having selection functions induced by a linear ranking of paths. Suppose that for each $v \in V$, there is a non-negative, integer-valued *ranking function* λ^v, defined over $\mathcal{P}^v$, which represents how node v ranks its permitted paths. If $P_1, P_2 \in \mathcal{P}^v$ and $\lambda^v(P_1) < \lambda^v(P_2)$, then P_2 is said to be *preferred over* P_1. Let $\Lambda = \{\lambda^v \mid v \in V - \{0\}\}$.

We then define the selection function induced by λ^u as follows. Given a node u, suppose that W is a subset of the permitted paths $\mathcal{P}^u$. The *maximal paths in* W is defined to be

$$\max(\lambda^u, W) = \{P \in W \cup \{\epsilon\} \mid\ P \text{ maximal w.r.t } \lambda^u(P)\}.$$

Then we define the natural selection function induced by this ranking as $\sigma_u(W) = P$ when $\max(\lambda^u, W) = \{P\}$. In order to ensure that this is well defined (there is always a unique maximally ranked path), we insist that the following property hold.

(strictness) If $P_1 \neq P_2$ and $\lambda^v(P_1) = \lambda^v(P_2)$, then there is a u such that $P_1 = (v\ u)P_1'$ and $P_2 = (v\ u)P_2'$ (paths P_1 and P_2 have the same next hop node).

We then restrict W so that no two paths in W ever have the same next hop nodes. This restriction follows naturally when encoding BGP, since a router can announce only one best route, and the route selection procedure has a deterministic means of breaking ties.

A *Stable Paths Problem* S is a GSPP where every selection function is induced by strict ranking of permitted paths. We will also consider "mixed" systems where some nodes have a linear ranking function while others have a general selection function.

The SPP theory of [8] provides a test, which can be checked statically, and which guarantees that an SPP will meet the conditions outlined in Section 1. That is, the system will be uniquely solvable, safe, and robust. The test is to make sure that an SPP does not have a *dispute wheel*, as described in Appendix B.

We now present several examples of SPPs that fail some or all of the conditions of unique solvability, safety, and robustness. These examples serve to highlight various types of routing anomalies similar to those that arise for uses of MED covered in later sections.

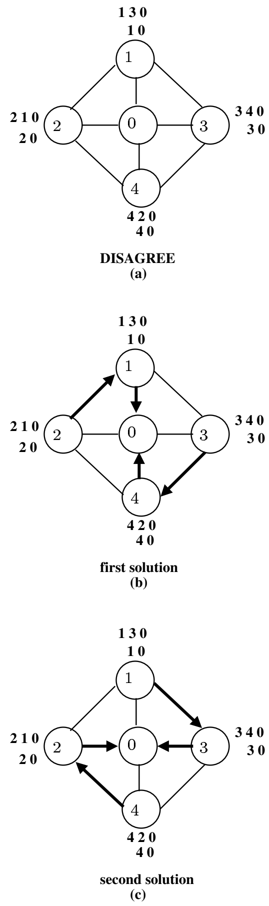

Figure 4. DISAGREE **and its two solutions.**

Figure 4 (a) presents a system called DISAGREE that does not have a unique solution. This system has two solutions, one presented in Figure 4 (b) and another in Figure 4 (c). Technically, it is also not safe. If we imagine nodes 1, 2, 3, and 4 exchanging messages in a lock-step manner, then the system could oscillate between assignments

$$((1\ 0),\ (2\ 0),\ (3\ 0),\ (4\ 0)),$$

and

$$((1\ 3\ 0),\ (2\ 1\ 0),\ (3\ 4\ 0),\ (4\ 2\ 0)).$$

However, in practice any BGP system corresponding to DISAGREE would quickly settle down to one solution or the other, due to the fact that random delays caused by message processing and delays due to rate limiting timers used by BGP make the oscillation scenario very unlikely to persist. Note that the assignment $\pi = ((1\ 0),\ (2\ 0),\ (3\ 0),\ (4\ 0))$ is not a solution to disagree, because the equation $\pi(u) = \text{best}(u, \pi)$ does not hold for any $u \in \{1,\ 2,\ 3,\ 4\}$.

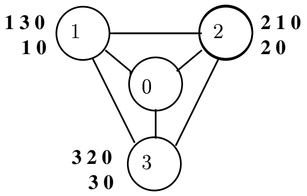

Figure 5. The system BAD GADGET**.**

In contrast, Figure 5 presents an SPP called BAD GADGET that has no solution and so a BGP-like protocol will never converge.

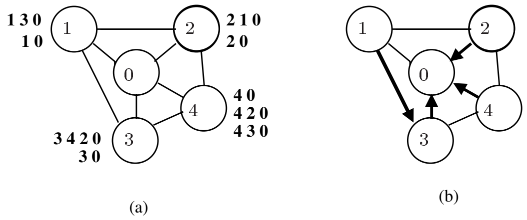

Figure 6. BAD BACKUP

Figure 6 (a) presents a system called BAD BACKUP that results from a slight modification to BAD GADGET. The node 4 is added having $(4\ 0)$ as its highest ranked path. This system has a unique solution, presented in Figure 6 (b), and it is safe. However, it is not robust. Note that if the edge $\{0, 4\}$ is deleted (modeling failure), then this system is like BAD GADGET, and has no solution.

3.4 More Problems with MED

We now present two examples that illustrate previously unknown routing anomalies associated with MEDs. The first examples shows that the stated semantics of the MED attribute can be violated, while the second shows that MED oscillation can actually span multiple ASes. In addition, neither example requires the use of route reflection or BGP confederations.

Figure 7 (a) presents the system MASHED-POTATO. There are five routers involved — one each in ASes 0, 1 and 2, and

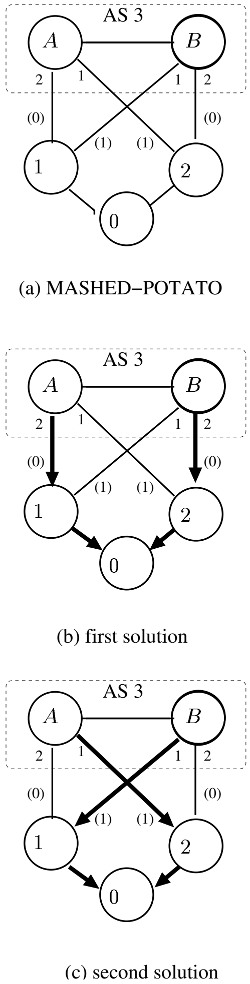

Figure 7. The system MASHED-POTATO **has two solutions**

two routers in AS 3. The routers A and B in AS 3 have an IBGP session. The weights on the EBGP links of routers A and B represent each router's preference in a BGP tie breaking, with lower values being more preferred. These need not be IGP weights, which are not always used on EBGP links, but could simply reflect other tie breaking rules. If router IDs are used for tie breaking, then the example could easily be modified so that AS 1 and 2 both had two routers, and achieving the same results. In GSPP, we can represent the path selection functions at A and B as

$$\begin{aligned}
\sigma_A(\{(A\ 1\ 0),\ (A\ 2\ 0)\}) &= (A\ 2\ 0)\\
\sigma_A(\{(A\ 1\ 0),\ (A\ 2\ 0),\ (A\ B\ 1\ 0)\}) &= (A\ 2\ 0)\\
\sigma_A(\{(A\ 1\ 0),\ (A\ 2\ 0),\ (A\ B\ 2\ 0)\}) &= (A\ 1\ 0)
\end{aligned}$$

$$\begin{aligned}
\sigma_B(\{(B\ 1\ 0),\ (B\ 2\ 0)\}) &= (B\ 1\ 0)\\
\sigma_B(\{(B\ 1\ 0),\ (B\ 2\ 0),\ (B\ A\ 1\ 0)\}) &= (B\ 2\ 0)\\
\sigma_B(\{(B\ 1\ 0),\ (B\ 2\ 0),\ (B\ A\ 2\ 0)\}) &= (B\ 1\ 0)
\end{aligned}$$

This system has two distinct solutions, presented in Figures 7 (b) and (c). Dynamically, BGP will nondeterministically settle on one or the other of these solutions. There are two things to notice. First, this example uses a "full IBGP mesh" and does not depend on route reflectors. Second, the solution of Figure 7 (c) *violates the semantics of MED*. That is, two routes picked by AS 3 both have higher MED values than the other routes sent by AS 1 and AS2. That is, solutions are not guaranteed to respect the stated semantics of the MED attribute.

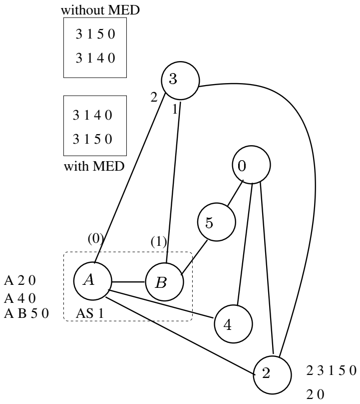

Figure 8. The system BAKED-POTATO

Figure 8 presents the system BAKED-POTATO, that has a MED oscillation problem that spans multiple ASes. That is, this system has a unique solution, is safe, and is robust if MEDs are erased, but it has no solution with the MED values shown. Furthermore, the routers involved in the oscillation are in multiple autonomous systems. Note again that this example does not depend on route reflectors.

4 Encoding BGP with MEDs in SPP

In this Section we show that BGP with MEDs can be encoded as a Stable Paths Problem. This will allow us, in the next section, to exploit the theory of [7, 8] to prove various facts about BGP with MEDs. Section 4.1 introduces a special class of GSPP, called the Two Pass Stable Paths Problem (2pSPP), and shows that BGP with MEDS can be modeled as a 2pSPP. In Section 4.2 we show that any 2pSPP can be encoded as an SPP by introducing new nodes in the graph. These nodes will act a "proxy" nodes to rank paths using MED values for paths to a single neighbor AS. Therefore, to represent a system $\mathcal{B}$ (BGP with MEDs) in SPP, we first translate to the GSPP $2pSPP(\mathcal{B})$, and then translate this to $SPP(2pSPP(\mathcal{B}))$.

4.1 The Two Pass SPP (2pSPP)

The GSPP selection function representing BGP with MED can be arrived at as a path selection function induced by a special kind of composition of two ranking functions.

A *two pass Stable Paths Problem* (2pSPP) is a GSPP in which the selection function σ_u is derived from two distinct path ranking functions. The "first pass" function sorts paths into a disjoint set of classes. It then selects a maximally ranked path from each class. Next, the "second pass" ranking function selects a best path from among those selected in the first pass.

Each node u has a partition of its permitted paths into a set of path classes. Let C_u represent class names at node u and $class(u,\ c) \subseteq \mathcal{P}^u$ denote the paths of class $c \in C_u$. Each node u has two ranking functions:

1. $\alpha^u_c(P)$: is a strict ranking defined only on permitted paths P of class $c \in C_u$.

2. $\beta^u(P)$: strict ranking of all permitted paths at u.

We define $(\beta^u \odot \alpha^u_*)(W)$ to be max(β^u, W_2), where W_2 is defined as

1. break W up into path sets $Y_c = W \cap class(u,\ c)$,

2. for each $c \in C_u$, let $P^u_c = \max(\alpha^u_c, Y_c)$,

3. $W_2 = \{P^u_{c_1}, \ldots, P^u_{c_k}\}$ where $C_u = \{c_1, \ldots, c_k\}$,

A GSPP selection function σ_u at node u is a *two pass ranking function* if can be written as $\sigma_u = \beta^u \odot \alpha^u_*$.

In order to represent BGP with MEDs as a 2pSPP, we construct, for each AS, a set of classes where each class corresponds to a neighboring AS. Then, we use $\alpha^u_c(P)$ to encode the lexical ranking of all paths from the AS associated with c to the node u. Note that now it is possible to lexically rank all such routes, since they all have comparable MED values (they are all from the same AS). Next, $\beta^u(P)$ models BGP selection without MEDs (erase all MED values and so skip the MED step in route selection). It is then fairly easy to prove that the selection function σ_u defined in Section 3.2 is equal to the resulting function $\beta^u \odot \alpha^u_*$. (The proof is omitted here for reasons of space.)

For example, let us calculate these functions for node A in the system MASHED-POTATO (Figure 7). Let A's set of classes $C_A = \{c_1,\ c_2\}$ correspond to ASes 1 and 2. Then A's permitted paths are partitioned so that paths $(A\ 1\ 0)$ and $(A\ B\ 1\ 0)$ are of class c_1 and paths $(A\ 2\ 0)$ and $(A\ B\ 2\ 0)$ are of class c_2. Then computing $\alpha^A_{c_1}$ results in a ranking where

$$\alpha^A_{c_1}((A\ B\ 1\ 0)) < \alpha^A_{c_1}((A\ 1\ 0))$$

and computing $\alpha^A_{c_2}$ results in a ranking where

$$\alpha^A_{c_1}((A\ 2\ 0)) < \alpha^A_{c_1}((A\ B\ 2\ 0)).$$

The calculation of β^A, which knows nothing about MEDs, gives

$$\begin{aligned}\beta^A((A\ B\ 2\ 0)) &< \beta^A((A\ B\ 1\ 0))\\ &< \beta^A((A\ 1\ 0))\\ &< \beta^A((A\ 2\ 0)).\end{aligned}$$

It is now easy to check that $\beta^A \odot \alpha^A_*$ is equal to σ_u defined in Section 3.4.

4.2 Encoding 2pSPP in SPP

For any 2pSPP S, we encode it as an SPP $\overline{S}$ as follows. For each node u in S, there is a node $\overline{u}$ in $\overline{S}$. For each u and each class $c_i \in C_u$, there is an auxiliary node u_{c_i} in $\overline{S}$, with an edge between $\overline{u}$ and u_{c_i}. If w is a neighbor of u then node u_{c_i} will be connected to node $\overline{w}$ as well. Node u_{c_i} will act as a "proxy" for node $\overline{u}$, and its ranking function will only encode the first pass ranking $\alpha^u_{c_i}$.

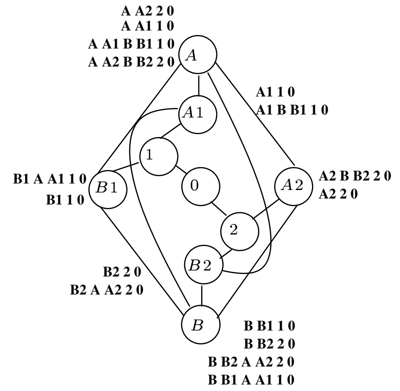

Figure 9. SPP encoding of MASHED-POTATO**.**

Before defining this formally, we present the SPP encoding of $2pSPP$(MASHED-POTATO) in Figure 9. Node A now has two proxy nodes $A1$ and $A2$ for handling paths from AS 1 and AS 2, respectively. For example, node $A1$ is connected to all nodes that A could learn of paths to AS 1. The rankings at node $A1$,

$$\lambda^{A1}((A1\ B\ B1\ 1\ 0)) < \lambda^{A1}((A1\ 1\ 0))$$

corresponds to the 2pSPP first pass ranking function

$$\alpha^A_{c_1}((A\ B\ 1\ 0)) < \alpha^A_{c_1}((A\ 1\ 0))$$

after an appropriate transformation on paths that reflects the new structure of the graph. The ranking function at node A,

$$\begin{array}{lcl} \lambda^A((A\ A2\ B\ B2\ 2\ 0)) & < & \lambda^A((A\ A1\ B\ B1\ 1\ 0)) \\ & < & \lambda^A((A\ A1\ 1\ 0)) \\ & < & \lambda^A((A\ A2\ 2\ 0)). \end{array}$$

corresponds to the 2pSPP second pass ranking function

$$\begin{array}{lcl} \beta^A((A\ B\ 2\ 0)) & < & \beta^A((A\ B\ 1\ 0)) \\ & < & \beta^A((A\ 1\ 0)) \\ & < & \beta^A((A\ 2\ 0)). \end{array}$$

after the same transformation on paths.

We formally define this translation as follows. For any path $P = (u, w)Q$ of class c at u, define $\overline{P}$ to be the path $(\overline{u}, u_c)\overline{Q}$. Let $\overline{(0)} = (\overline{0})$. Let $\overline{\overline{P}}$ be the path $u_c, \overline{Q}$ (that is, the initial node $\overline{u}$ is dropped).

If $P = (u, w)Q$ is a permitted path in S and P has class c, then add an edge $\{u_c, \overline{w}\}$ to $\overline{S}$.

The set of permitted paths at node $\overline{u}$ in $\overline{S}$ is the set of all paths $\overline{P}$, where P is permitted at u in S. The permitted paths at node u_c in $\overline{S}$ is the set of all paths $\overline{\overline{P}}$, where P is a permitted path at u and $class_u(P) = c$.

Define the ranking function λ in $\overline{S}$ as

$$\lambda^{u_c}(\overline{\overline{(u, w)Q}}) = \alpha^u_c((u, w)Q)$$

and

$$\lambda^{\overline{u}}(\overline{P}) = \beta^u(P).$$

If π is a path assignment for 2pSPP S, then $\overline{\pi}$ is the path assignment on SPP $\overline{S}$ defined as

$$\begin{array}{rcl} \overline{\pi}(\overline{u}) & = & \overline{\pi(u)} \\ \overline{\pi}(u_c) & = & \overline{\overline{\max(\alpha^u_{c_j}, \text{candidates}_1(u, c, \pi))}} \end{array}$$

Appendix A proves that there is a one-to-one correspondence between solutions of S and solutions of $\overline{S}$.

5 The Scope of MED Induced Routing Anomalies

In this section we give a formal definition of a MED induced routing anomaly (MIRA). We show that if ASes have certain types of common customer/provider and peer/peer relationships, then the scope of any MIRA is always confined to a single AS. For fully meshed ASes, we show that all MIRAs are associated with systems that have two solutions, such as the system MASHED-POTATO (Figure 7). Finally, we show that the suggested workarounds for the MED problem discussed in [14, 4] are not guaranteed to prevent MED oscillation.

A MED Induced Routing Anomaly (MIRA) of a BGP system $\mathcal{B}$ is defined to be any dispute wheel (see Appendix B) for $SPP(2pSPP(\mathcal{B}))$ that does not exist for $SPP(2pSPP(\text{erase}(\mathcal{B})))$.

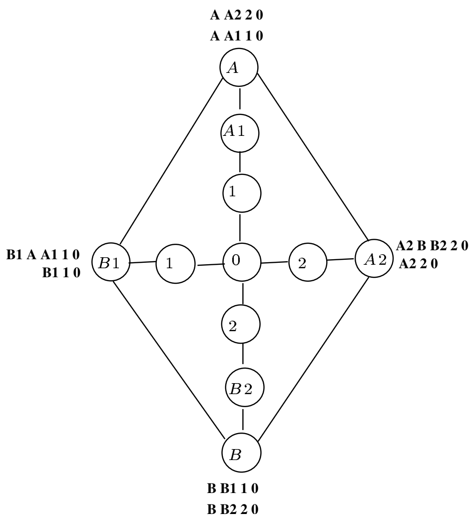

Figure 10. The dispute wheel of MASHED-POTATO**.**

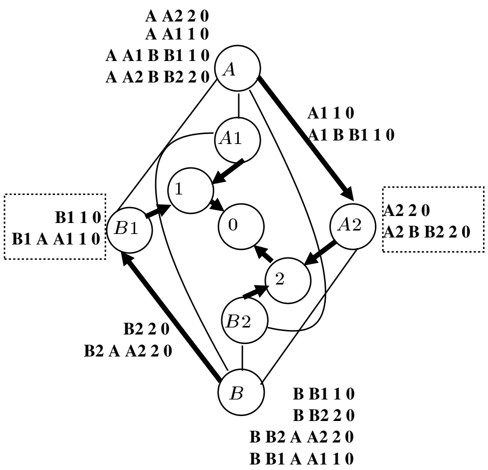

Figure 11. SPP encoding of erase(MASHED-POTATO)**.**

For example, Figure 10 presents the dispute wheel of MASHED-POTATO (Figure 9). Figure 11 presents the SPP encoding for erase(MASHED-POTATO). This system has no dispute wheel, and so it has a unique solution (indicated in the figure with thick arrows). Therefore, the dispute wheel of Figure 10 is a MIRA for MASHED-POTATO.

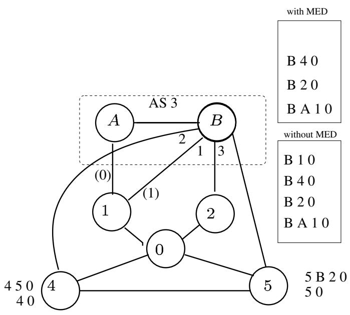

Figure 12. The system POTATO-SURPRISE

We now present an example that illustrates a subtlety with our definition of a MIRA. The system POTATO-SURPRISE is defined in Figure 12. This sytem behaves much like BAD BACKUP (Figure 6) in that it has a "hidden" divergence problem that is exposed only when link $(1, B)$ is removed. In POTATO-SURPRISE, link $(1, B)$ can be "removed" with an appropriate setting of the MED attribute. The boxes to the left of node B show that ranking of paths at B when MEDs are present and when they are erased. The effect of MED here is to eliminate the path $(B\ 1\ 0)$ from consideration at node B, and this exposes a BAD-GADGET-like oscillation between nodes B, 4, and 5. For us, this example does not represent a MIRA. The potential anomaly is present independent of the setting of the MED attribute, which can expose it just as a link failure can.

We review the common types of relationships that occur between autonomous systems [12, 13, 5, 6], which we will refer to as customer/provider and peer/peer relationships. A provider AS i receives routes from its customers and passes them on to all other customers and to all of its providers. When AS i and AS j are peers, they simply exchange routes of their respective customers. These commercial relationships tend to imply two other constraints. First, that providers will prefer customer routes over peer or provider routes (this is usually implemented with the local preference attribute). Second, the digraph representing customer/provider relationships is acyclic. So for example, a network is unlikely to buy upstream services from a customer of one if its customers! A BGP system that conforms to these rules is said to "obey AR" (for "obey the Autonomous systems Relations").

It has been shown in [6] that if a BGP system obeys AR, then it is guaranteed to be uniquely solvable, safe, and robust. However, these proofs ignored IBGP (that is, they assumed each AS was a single router), and in particular ignored the MED attribute.

Note that AR merely captures typical relationships and constraints and does not provide any hard guarantees. That is, there are no laws that force ASes to obey AR.

The system like BAKED-POTATO would be very problematic in practice since the MED oscillation problem spans several ASes. Put another way, the oscillating routers, those on the rim of the dispute wheel that is the MIRA, are in several ASes. In practice this could be very hard to debug. However, we now show that if the BGP system obeys AR, then this type of domain spanning problem cannot occur.

Theorem 5.1 *If a BGP system $\mathcal{B}$ obeys AR, then any MIRA must have a rim that is contained within a single AS.*

Proof sketch: We claim that there can be no inter-AS links along the rim of a dispute wheel. The argument proceeds much like the proof in [6], except that nodes in the rim are grouped into (contiguous) ASes. A case analysis enumerates the types of inter-AS arcs (provider/customer, peer/peer, customer/provider) that could occur on the rim and in each case finds a contradiction to AR. Therefore, all nodes on the rim must be in the same autonomous system. ■

The system MASHED-POTATO demonstrates that a fully meshed IBGP system can have a MIRA. (A fully meshed IBGP is one where all routers within the AS have IBGP sessions, and there are no route reflectors.) However, this system has multiple solutions. We now show that any fully meshed MIRA shares this property.

Theorem 5.2 *If a BGP system $\mathcal{B}$ obeys AR, and every AS uses fully meshed IBGP, then $SPP(2pSPP(\mathcal{B}))$ always has at least one solution. Furthermore, any MIRA in the system must be a multiple-solution MIRA that involves MED attributed routes associated with a single prefix received from more than one neighbor.*

Proof sketch: We will refer to steps 1-6 of the BGP selection process of Section 2.1. We know that the rim of any dispute wheel is within a single AS, so BGP routes associated with paths from the rim nodes must have the same values of $\mathbf{local\_pref}$ and $\mathbf{as\_path}$ length. To see this, suppose that the route associated with path Q_i has a lower value of $\mathbf{local\_pref}$ than the BGP route associated with path R_iQ_{i+1} (that is, the route for Q_i was eliminated at step 1). Then the route associated with Q_{i+1} must share the same $\mathbf{local\_pref}$ as R_iQ_{i+1} (both are associated with IBGP routes within the same AS). If we continue to walk around the wheel in this way we will come back to Q_i and show that it must have a higher $\mathbf{local\_pref}$ value than we assumed, which is a contradiction. This same argument works for $\mathbf{as\_path}$ length (step 2), *and* for MED values (step 3) when the source of MED-attributed routes is a unique neighbor AS. Therefore, a MIRA can only exist if MED-attributed routes (associated with a single prefix) are received from at least two different neighbors. Assume this is the case. We show that for u_i in the dispute wheel, $\lambda^{u_i}(Q_i) < \lambda^{u_i}(R_iQ_{i+1})$. (That is we can replace $\leq$ by $<$ in the IBGP case.) For each u_i, we can then ask why it is that this $\lambda^{u_i}(Q_i) < \lambda^{u_i}(R_iQ_{i+1})$ holds. The answer is **med** if the route represented by Q_i is eliminated at step 3. Otherwise, the answer is **igp**, which means that Q_i is eliminated in step 4, 5, or 6 (from the observation above concerning values of $\mathbf{local\_pref}$ and $\mathbf{as\_path}$ length, we know that steps 1 and 2 cannot eliminate any Q_i). We can then show that these reasons must *alternate* as we move around the rim of the dispute wheel. That is, it cannot be the case that both u_i and u_{i+1} give the same reason. If **igp** is followed by **igp**, then the "middle" node must have violated step 4 (prefer EBGP over IBGP). If **med** is followed by **med**, then the first node is violating step 3 and not picking the route with the best MED value. Therefore, the dispute wheel must have an even number of spokes. It is then easy to show that each MIRA has two solutions. ■

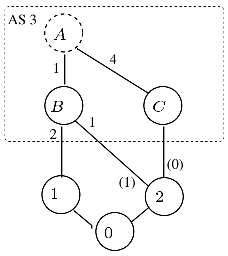

Figure 13. The system FRIED-POTATO

Route reflectors and their clients are grouped into clusters. Recommendations in [14, 4] for avoiding MED related problems when using route reflectors include a rule that the intra-cluster distances (IGP) should be less than inter-cluster distances. We would like to be able to prove a result similar to Theorem 5.2 for such systems. Unfortunately, the system FRIED-POTATO, presented in Figure 13, shows that this rule is *not guaranteed* to avoid MED oscillations. Router A is a route reflector. The weights on the EBGP links of router B represent the router's preference in a BGP tie breaking, with lower values being more preferred. (Again, these need not be IGP weights, which are not always used on EBGP links, but could simply reflect other tie breaking rules.) In GSPP, we can represent the path selection functions at A, B, and C as

$$
\begin{aligned}
\sigma_A(\{(A\;B\;1\;0),\;(A\;C\;2\;0)\}) &= (A\;B\;1\;0)\\
\sigma_A(\{(A\;B\;2\;0),\;(A\;C\;2\;0)\}) &= (A\;C\;2\;0)\\[1em]
\sigma_B(\{(B\;1\;0),\;(B\;2\;0)\}) &= (B\;2\;0)\\
\sigma_B(\{(B\;1\;0),\;(B\;2\;0),\;(B\;A\;C\;2\;0)\}) &= (B\;1\;0)\\[1em]
\sigma_C(\{(C\;2\;0),\;(A\;B\;1\;0)\}) &= (C\;2\;0)\\
\sigma_C(\{(C\;2\;0),\;(A\;B\;2\;0)\}) &= (C\;2\;0)
\end{aligned}
$$

Nodes A and B clearly have incompatible selection functions prohibiting a solution. If B selects path $(B\ 1\ 0)$, then A selects path $(A\ B\ 1\ 0)$, causing B to select path $(B\ 2\ 0)$, causing A to select path $(A\ C\ 2\ 0)$, which causes B to select path $(B\ 1\ 0)$, and so on.

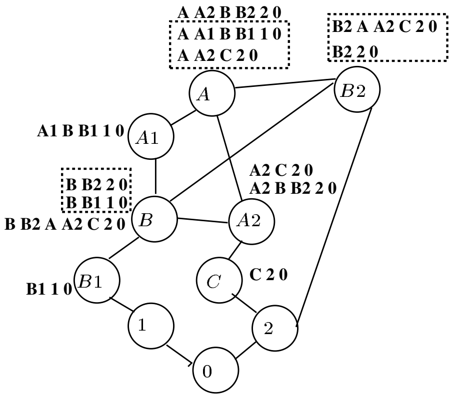

Figure 14. SPP encoding of FRIED-POTATO

Figure 14 presents the SPP encoding of FRIED-POTATO. Note that the nodes A, B, and $B2$ form a system that behaves like BAD-GADGET (Figure 5). The two path rankings at each of these nodes that correspond to the BAD-GADGET rankings are enclosed in dashed boxes to help the reader identify the embedded BAD-GADGET.

6 Conclusion and Open Problems

Changing the BGP protocol definition at this stage is an extremely difficult proposition. Therefore, we suggest that the only practical way of taming the MED attribute is for each AS to write BGP routing policies that force the MED attribute to obey the rule of independent ranking.

One approach is to force every MED-icated prefix to have $\mathbf{local\_pref}$ values unique to each neighbor. This is conjectured in [14] to solve the MED oscillation problem. Note that Theorem 5.2 actually *proves* this to be true in the IBGP full mesh case (however, we note that multiple solutions are possible, making BGP routing nondeterministic). We believe that this argument can be extended to any IBGP configuration that meets the correctness constraints given in [10]. As noted in [14], this solution is not entirely satisfactory, since it may involve complicated policies that are difficult to maintain. However, we believe that the generation of such policies could easily be automated using data from passive BGP monitors within an AS.

In addition to discussing various types of potatoes, we have presented some good news and some bad news concerning MED induced routing anomalies. The bad news is that such anomalies are more difficult to avoid than previously believed. But the good news is that they are likely to have local scope, and so debugging them does not have to span multiple autonomous systems.

There remains the problem to designing a means of doing cold potato routing with BGP that does not violate the rule of independent ranking. As mentioned, we believe that the results presented here can be combined with the IBGP constraints of [10] to prevent BGP *divergence*. However, it is not clear how MED interacts with the other problem studied in [10], that is the problem of *route deflection* in the forwarding plane.

It may be useful to generalize the notion of GSPP still further and allow $\sigma_u(W)$ to be a *subset* of W. In this way we could model path vector protocols that allow multiple best paths. This would be useful in the analysis of protocols for optical networking [3], and Telephony Routing over IP (TRIP) [16]. TRIP is a policy based interdomain protocol for advertizing the reachability of telephony destinations between telephony location servers. TRIP shares many features with BGP, but unlike BGP it does provide multiple best routes.

References

[1] A. Basu, L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong. Route oscillations in I-BGP with route reflection. In *Proceedings of ACM SIGCOMM 2002*, 2002.

[2] T. Bates, R. Chandra, and E. Chen. BGP route reflection - an alternative to full mesh IBGP. RFC 2796, 2000.

[3] Marc Blanchet, Florent Parent, and Bill St-Arnaud. Optical BGP (OBGP): InterAS lightpath provisioning. Internet Draft draft-parent-obgp-01.txt. Work in progress.

[4] Cisco. Endless BGP convergence problem in Cisco IOS software releases. Field Note, October 10 2001, `http://www.cisco.com/warp/public/770/fn12942.html`.

[5] Lixin Gao and Jennifer Rexford. Stable Internet routing without global coordination. In *Proc. ACM SIGMETRICS*, June 2000.

[6] T. Griffin, Lixin Gao, and Jennifer Rexford. Inherently safe backup routing with BGP. In *Proc. IEEE INFOCOM*, April 2001.

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

[8] T. G. Griffin, F. B. Shepherd, and G. Wilfong. The stable paths problem and interdomain routing. *IEEE/ACM Transactions on Networking*, 10(2):232–243, April 2002.

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

[10] T. G. Griffin and G. Wilfong. On the correctness of IBGP configuration. In *Proceedings of ACM SIGCOMM 2002*, 2002.

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

[12] Geoff Huston. Interconnection, peering and settlements: Part I. *Internet Protocol Journal*, 2(1), June 1999.

[13] Geoff Huston. Interconnection, peering and settlements: Part II. *Internet Protocol Journal*, 2(2), June 1999.

[14] D. McPherson, V. Gill, D. Walton, and A. Retana. BGP persistent route oscillation condition. Internet Draft `draft-ietf-idr-route-oscillation-01.txt` , Work In Progress, 2002.

[15] Y. Rekhter and T. Li. A Border Gateway Protocol. RFC 1771 (BGP version 4), March 1995.

[16] Jonathan Rosenberg, Hussein Salma, and Matt Squire. Telephony routing over IP (TRIP). RFC 3219. January 2002.

[17] John W. Stewart. *BGP4: Inter-Domain Routing in the Internet*. Addison-Wesley, 1998.

[18] P. Traina, D. McPherson, and J. Scudder. Autonomous system confederations for BGP. RFC 3065, 2001.

[19] Kanan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32:1–16, 2000.

A Proofs for Encoding 2pSPP in SPP

Given a path assignment π, the best path at u in a 2pSPP is defined as follows. Let $N(u)$ be the neighbors of node u. For any node u of S and $c \in C_u$, define

1. $\text{candidates}_1(u, c, \pi) = \mathit{class}(u, c) \cap \{(u, w)Q | w \in N(u), \pi(w) = Q\}$.
2. $P^u_c = \max(\alpha^u_c, \text{candidates}_1(u, c, \pi))$, for $c \in C_u$.
3. $\text{candidates}_2(u, \pi) = \{P^u_{c_1}, \ldots, P^u_{c_k}\}$, where $C_u = \{c_1, \ldots, c_k\}$,
4. $\text{best}(u, \pi) = \max(\beta^u, \text{candidates}_2(u, \pi))$.

Again, an assignment π is a *solution* if for each u, $\pi(u) = \text{best}(u, \pi)$.

We now show that there is a one-to-one correspondence between solutions of S and solutions of $\overline{S}$. If W is a set of paths in S, let $\overline{W} = \{\overline{P} \mid P \in W\}$ and $\overline{\overline{W}} = \{\overline{\overline{P}} \mid P \in W\}$. The first three lemmas follow easily from the definitions.

Lemma A.1 *For any W we have*

$$\overline{\max(\beta^u, W)} = \max(\lambda^{\overline{u}}, \overline{W}),$$

and

$$\overline{\overline{\max(\alpha^u_c, W)}} = \max(\lambda^{u_c}, \overline{\overline{W}}).$$

Lemma A.2 *For any path assignment π on a 2pSPP S, and any node u of S with class c, we have*

$$\overline{\overline{\text{candidates}_1(u, c, \pi)}} = \text{candidates}(u_c, \overline{\pi}).$$

Lemma A.3 *For any path assignment π on a 2pSPP S, and any node u of S, we have*

$$\overline{\text{candidates}_2(u, \pi)} = \text{candidates}(\overline{u}, \overline{\pi}).$$

Theorem A.4 *If π is stable for the 2pSPP S, then $\overline{\pi}$ is stable for the SPP $\overline{S}$.*

Proof: Assume π is stable for the 2pSPP S. Then

$$\begin{aligned}
\overline{\pi}(\overline{u}) &= \overline{\pi(u)} \\
&= \overline{\text{best}(u,\pi)} && (\pi \text{ is stable}) \\
&= \overline{\max(\beta^u, \text{candidates}_2(u,\pi))} \\
&= \max(\lambda^{\overline{u}}, \overline{\text{candidates}_2(u,\pi)}) \\
&= \max(\lambda^{\overline{u}}, \text{candidates}(\overline{u},\overline{\pi})) && \text{(Lemma A.3)} \\
&= \text{best}(\overline{u},\overline{\pi})
\end{aligned}$$

In a similar manner,

$$\begin{aligned}
\overline{\pi}(u_c) &= \overline{\overline{\max(\alpha^u_{c_j}, \text{candidates}_1(u,c,\pi))}} \\
&= \max(\lambda^{u_c}, \overline{\overline{\text{candidates}_1(u,c,\pi)}}) \\
&= \max(\lambda^{u_c}, \text{candidates}(u_c,\overline{\pi})) && \text{(Lemma A.2)} \\
&= \text{best}(u_c,\overline{\pi})
\end{aligned}$$

■

Theorem A.5 *If π' is stable for the SPP $\overline{S}$, then there exists a π stable for the 2pSPP S such that $\pi' = \overline{\pi}$.*

Proof: Suppose that π' is stable for the SPP $\overline{S}$. That is,

$$\pi'(\overline{u}) = \text{best}(\overline{u},\pi')$$

and

$$\pi'(u_c) = \text{best}(u_c,\pi')$$

Define $\pi(u) = P$ if and only if $\pi'(\overline{u}) = \overline{P}$. Observe that this means that $\overline{\pi} = \pi'$, since

$$\overline{\pi}(\overline{u}) = \overline{\pi(u)} = \overline{P} = \pi'(\overline{u}).$$

Therefore,

$$\text{candidates}(u_c,\overline{\pi}) = \text{candidates}(u_c,\pi')$$

since these depend only on $\overline{\pi}(\overline{u}) = \pi'(\overline{u})$. Therefore,

$$\begin{aligned}
\overline{\pi}(u_c) &= \overline{\overline{\max(\alpha^u_c, \text{candidates}_1(u,c,\pi))}} && (\text{def. of } \overline{\pi}) \\
&= \max(\lambda^{u_c}, \overline{\overline{\text{candidates}_1(u,c,\pi)}}) && \text{(Lemma A.1)} \\
&= \max(\lambda^{u_c}, \text{candidates}(u_c,\overline{\pi})) && \text{(Lemma A.2)} \\
&= \max(\lambda^{u_c}, \text{candidates}(u_c,\pi')) && \text{(from above)} \\
&= \text{best}(u_c,\pi') \\
&= \pi'(u_c).
\end{aligned}$$

In order to show π is stable, $\pi(u) = \text{best}(u,\pi)$, we will show that $\overline{\pi(u)} = \overline{\text{best}(u,\pi)}$.

$$\begin{aligned}
\overline{\pi(u)} &= \overline{P} && (\text{where } \pi'(\overline{u}) = \overline{P}) \\
&= \text{best}(\overline{u},\pi') && (\pi' \text{ stable for } \overline{S}) \\
&= \max(\lambda^{\overline{u}}, \text{candidates}(\overline{u},\pi')) \\
&= \max(\lambda^{\overline{u}}, \text{candidates}(\overline{u},\overline{\pi})) && \text{(observation above)} \\
&= \overline{\max(\lambda^{\overline{u}}, \overline{\text{candidates}_2(u,\pi)})} && \text{(Lemma A.3)} \\
&= \overline{\max(\beta^u, \text{candidates}_2(u,\pi))} && \text{(Lemma A.1)} \\
&= \overline{\text{best}(u,\pi)}
\end{aligned}$$

■

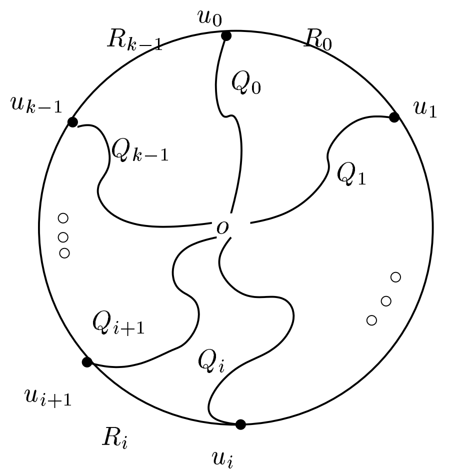

Figure 15. A dispute wheel of size k.

B Dispute Wheels

This definition is taken from [7]. A *dispute wheel W of size k* is a triple $(\vec{U}, \vec{Q}, \vec{R})$, where $\vec{U}$ is a sequence of k nodes $u_0, u_1, \cdots, u_{k-1}$, $\vec{Q}$ is a sequence of k non-empty paths $Q_0, Q_1, \cdots, Q_{k-1}$, and $\vec{R}$ is a sequence of k non-empty paths $R_0, R_1, \cdots, R_{k-1}$. This triple is such that for each $0 \leq i \leq k-1$ we have

(1) R_i is a path from u_i to u_{i+1},

(2) Q_i and R_iQ_{i+1} are permitted at u_i,

(3) $\lambda^{u_i}(Q_i) \leq \lambda^{u_i}(R_iQ_{i+1})$.

All subscripts are to be interpreted modulo k.

The set of all nodes from all paths R_i is called then *rim* of the dispute wheel. If a BGP system exhibits persistent oscillation, then that oscillation will be taking place in routers associated with the rim of a dispute wheel for an SPP encoding of the BGP system. An SPP S is *wheel-free* if it does not contain a dispute wheel. It follows from the results of [7] that if S is wheel-free, then it has a unique solution, it is safe, and it is robust.