# Stable and Flexible iBGP

Ashley Flavel
School of Mathematical Sciences,
University of Adelaide, SA, Australia
ashley.flavel@gmail.com

Matthew Roughan
School of Mathematical Sciences,
University of Adelaide, SA, Australia
matthew.roughan@adelaide.edu.au

## ABSTRACT

Routing oscillation is highly detrimental. It can decrease performance and lead to a high level of update churn placing unnecessary workload on routers. The general problem of stabilizing BGP is hard, given the problem is distributed between many providers. However, iBGP — the routing protocol used to distribute routes inside a single Autonomous System — has also been shown to oscillate. Despite the fact that iBGP is configured by a single provider according to apparently straight forward rules, more than eight years of research has not solved the problem of iBGP oscillation. Various solutions have been proposed but they all lack critical features: either they are complicated to implement, restrict routing flexibility, or lack guarantees of stability. In this paper we propose a very simple adaptation to the BGP decision process. Despite its simplicity and negligible cost we prove algebraically that it prevents iBGP oscillation. We extend the idea to provide routing flexibility, such as respecting the MED attribute, without sacrificing network stability.

## Categories and Subject Descriptors

C.2.2 [**Computer Communications Networks**]: Network Protocols—*routing protocols, protocol verification*; G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*

## General Terms

Algorithms, Design

## Keywords

Routing, BGP, Stability, Metarouting

## 1. INTRODUCTION

Routing oscillation is a problem. It can severely degrade network performance, and introduce many unnecessary routing updates. Some of these may propagate through the global Internet leading to redundant workload for routers worldwide. It is now well known that the Border Gateway Protocol (BGP) — the routing protocol that spreads routing information across the Internet — admits the possibility of oscillation [1–5]. Certain configurations never converge to a stable routing solution. Preventing oscillation in the global Internet is a hard problem. It would require the co-operation of many network providers, not to mention the algorithmic difficulties that would be encountered. On the other hand, large-scale oscillations have not been confirmed, and so we do not even know if this is a real problem in the Internet. But route oscillations within a single Autonomous System (AS) *have* been observed [6, 7, 16].

Inside a single AS there seems to be no excuse to allow oscillation; a single network operator has complete control of the network. Moreover, iBGP (the variant of BGP used to propagate externally learned routes inside an AS) is deceptively simple, and on the surface it appears trivial to prevent internal oscillations. However, it is far from trivial, and a significant research effort has been devoted to prevention of oscillation [2, 3, 8–13]. However, all of these approaches have limitations. It is much more appealing to modify iBGP such that oscillation is intrinsically prevented.

In this paper we present two modifications to iBGP, and we prove algebraically that these prevent oscillation. The first is very easily implementable: we simply recommend that a “minimum iBGP hop count” step be placed in the iBGP decision process. The information required for the new step is already propagated between routers, so no protocol changes are needed, and no additional state information is required at the routers. The only required change to iBGP is a minor addition to the logic used to select best routes. The change need not even be implemented network-wide, but only need be installed on route-reflectors.

This change, by itself, can be used to prevent network oscillation, but can violate the semantics of the Multi-Exit-Discriminator (MED). Our second proposed modification to iBGP is to allow propagation of more than one route from each router. This type of proposal has been made before [3]. Our contribution is to show how multiple routes can be propagated and how Sobrinho’s [10] algebraic techniques can be extended to prove stability of the resulting routing. The advantage of the mathematical approach is that we can determine the extra routes required to be propagated for a given level of routing flexibility while guaranteeing routing stability.

The problems in iBGP arise (typically) as the result of the techniques used to make it more scalable. It is the responsibility of the Internal Border Gateway Protocol (iBGP) to ensure all routers within an administrative body or AS receive the required information to route traffic to external destinations. A hierarchy of route-reflectors is one approach used to reduce the overhead associated with iBGP, but route-reflectors hide information and the result is that they can’t always choose the overall best route, but only a route from those available locally. A series of such decisions can change the information that is hidden, in turn changing other decisions, leading to oscillation. Well known cases include MED oscillation

*SIGCOMM’09,* August 17–21, 2009, Barcelona, Spain.

and pure topological oscillation resulting from differences between the logical iBGP topology and the physical Interior Gateway Protocol (IGP) topology [2, 13].

There are several current approaches to stabilization:

1. Design the network such that the hidden information has no effect [2, 10–12].

2. Test a configuration for stability before implementation [13].

3. Centralize router decisions [8, 9] to ensure all information is available for route selection.

4. Allow iBGP to propagate additional routes so that information is no longer hidden [3].

Each of the above has its limitations. Networks are rarely built from scratch. They evolve and grow over time, and the difficulty of adapting iBGP design-based methods to a changing network can be seen easily if we consider that link failures can break the carefully set up conditions of [10].

The design-based approaches also limit flexibility, either by specifying required locations for route-reflectors [11], or adding conditions on the network configuration. For instance [10] proves that if IGP weights are chosen to enforce a technical condition, then stability is guaranteed. That proof is particularly relevant here because we seek to use the same algebraic methods. However, we should note that relying on IGP distances is troublesome. IGP weights are not primarily intended to stabilize BGP. Restricting their values could interfere with other goals, e.g., those of traffic engineering. In addition, we show that it is impossible to choose weights that satisfy this condition even for some very simple networks.

In Section 3 we provide further examples of the challenges faced by design-based approaches. The natural alternative to design is to test network configurations before implementation [13]. This type of approach is desirable, but we can only test against anticipated problems such as failures. Unanticipated problems may change the network in ways that could still result in oscillation. It would be better to have an routing protocol that is inherently stable.

The third approach, centralizing router decisions [8, 9], avoids oscillation. However, questions remain about the scalability of this approach which also introduces a single critical point of failure, and may cause additional delays in a fail-over scenario. Our approach, retains the distributed nature of routing and does not suffer from these issues surrounding centralization. Further, in Section 6 we demonstrate how the separation of route propagation from route selection can allow complex policies to be implemented without sacrificing network stability.

The fourth approach — allowing BGP to propagate additional information [3] — is perhaps the best long-term solution, and we shall consider this in more detail in what follows. However, it requires changes to the iBGP protocol, and so we first consider a light-weight remedy.

Our first proposal is to change the route decision process by adding a simple "minimal iBGP hop count" step. The hop count is already implicit in the cluster-list attribute that iBGP uses to prevent looping announcements, so no new information need be propagated, or stored at the routers. Only the decision process at routers need be changed, and even that need only change at route-reflectors. However, despite its simplicity, we prove that this step can guarantee iBGP stability, even in multi-level route-reflector hierarchies and other iBGP topologies such as confederations. The new step does not require any changes to IGP weights, or impose other design requirements on networks, so it maintains the flexibility of current routing, while guaranteeing stability.

The additional decision step could result in some deviations from pure hot-potato routing, but the route-reflector hierarchy already causes such deviations. The fact that some packets may travel additional distance has rarely been seen as a concern[1]. However, to guarantee stability in all cases, this decision step must precede the MED [5] step. The result may violate MED semantics, potentially leading to an AS violating contractual agreements.

Although we may wish MEDs weren't used in practice (MEDs have non-transitive properties that make iBGP susceptible to oscillation [3–5]), their ability to implement complex contractual agreements prevents us making such an assumption. What's more, MED oscillation was the first form of routing oscillation observed "in the wild", and Wu *et al.* [16] found that in a large tier-1 ISP, MED oscillations alone accounted for 18.3% of updates from non-convergent prefixes. So we propose a second solution to the iBGP stability problem, which respects MEDs, while guaranteeing stability. As a side benefit, our second approach actually improves a route-reflector hierarchy's ability to conform to the semantics of received MEDs, where that is needed.

The approach requires propagation of additional routes. We prove that, if we choose those routes carefully, they can be used to prevent oscillation (previous work has shown that it can prevent MED oscillation but not the more general topological oscillations iBGP admits). While propagating additional routes is ideal, it does entail modifications to BGP[2], and an increase in the state information that must be retained at each router. As such, it is not as easy to implement as introducing our first suggestion. It is important, therefore, to have an understanding of the minimal information that is needed to prevent oscillation before we change the protocol, and our proofs contribute to the understanding of this question. Moreover, we show that by separation of route-propagation from route-selection, we can allow multiple flexible routing strategies to co-exist with guaranteed stability.

## 2. BACKGROUND

Routing in the Internet is undertaken on two scales: within an administrative domain or Autonomous System (AS) and between ASes. Separate routing protocols are used by an AS to spread information about internal and external destinations. The routing protocol in use within an AS is termed an Interior Gateway Protocol (IGP) and is the choice of the individual AS. The current de-facto standard routing protocol used for external destinations is the Border Gateway Protocol (BGP) [17]. However, there are two flavors of BGP. One is used to spread information externally (eBGP), and the other — the Internal Border Gateway Protocol (iBGP) — is responsible for spreading information about external destinations inside the AS. The distinction between the roles of the IGP and iBGP may seem subtle, but is critically important here.

A BGP speaking router operates by taking the information about existing routes from its BGP neighbors, the IGP and other sources. The router then makes a decision about which of these provides the "best" route to each destination. These best routes are placed in a table, and then (subject to export policies) passed to the router's BGP neighbors. The process iterates until a stable routing solution is found. When "best" equates to shortest paths, the algorithm is guaranteed to converge. However, BGP's decision process involves *policies* that can be far from shortest-paths. We outline the major steps of the BGP decision process in Figure 1.

The first three steps of the BGP decision process are AS-wide

[1] Though there is still the concern that violations of this policy may result in a forwarding-loop [2], but such problems can be cured by various other techniques [15].

[2] The BGP add-path capability is proposed in [14].

1. Highest Local Preference
2. Shortest AS Path Length
3. Lowest Origin Type
4. Lowest MED (Multi-Exit Discriminator)
5. Closest Egress (Lowest IGP Distance)
6. Tie Breaking (Lowest-Router-ID)

**Figure 1: Summarized BGP Decision Process [17], omitting vendor dependent steps.**

decision steps. That is, all routers in an AS will pick a route with equally attractive routes through these steps [18]. The fourth step involves the Multi-Exit Discriminator (MED) attribute, which allows a neighboring AS to have greater control over inbound traffic traversing their multiple interconnection links (see Section 3.1 for more details). Step 5 of the BGP decision process is a router-dependent decision step. An individual router will select a route that minimizes the IGP distance to the egress link. This is commonly referred to as the "hot-potato" step.

In its simplest form, iBGP does not pass routes more than one hop. Routers only propagate routes learned from external sources within iBGP, and so iBGP sessions are required between all pairs of routers inside an AS to ensure all routers learn the possible routes. This requires $\frac{N(N-1)}{2}$ iBGP sessions, and so does not scale well for large networks. Route-reflection [19] limits the required number of iBGP sessions by introducing a hierarchy. However, route-reflection also reduces route visibility, which can cause oscillation (see Sections 3.1 and 3.2).

The two types of routers in the route-reflector hierarchy are shown in Figure 2. Route-reflectors are shown by pentagonal nodes and their clients by circles. The lines show iBGP sessions. The clients advertise their routes from external peers to their parent. Route-reflectors 'reflect' routes across other iBGP sessions, depending on their source, according to the following rules:

| Source | Reflect to: |
|---|---|
| client | all iBGP neighbors |
| non-client | only to clients |

A valid route-reflection *signaling path* — the path of iBGP links along which a route can legitimately be propagated — is no longer a single hop. To describe a valid signaling path we use the notation of [2] (see Figure 2). An arc from a client to a route-reflector is labeled **up**. An arc from a route-reflector to a client is labeled **down**. An arc between route-reflectors is labeled **over**. A valid signaling path $S$ can be split into sub paths $S = PQR$ where $P$ contains zero or more edges $p_i \in$ **up**, $R$ contains zero or more edges $r_i \in$ **down** and $Q$ is either empty or consists of a single arc $q \in$ **over**.

An alternative to route-reflection's hierarchical approach to solving the scalability issues within iBGP is the divide-and-conquer approach of confederations [20]. Our approaches outlined in this paper are equally applicable to confederations. We focus primarily on route-reflection as it is commonly believed to be the most widely deployed solution.

The causes of oscillation often lie in the fact that the iBGP signaling network is not equivalent to the underlying network topology. Links (say between route-reflectors) can cross multiple physical links. The iBGP topology which propagates the routing information is divorced from the IGP topology, but both are still entangled in the process of determining routes. Coupled with this is the fact that each router only reveals its best route, so the diversity of routes learned in a route-reflector hierarchy is reduced. The result can be oscillation, which we discuss in more detail in the following section.

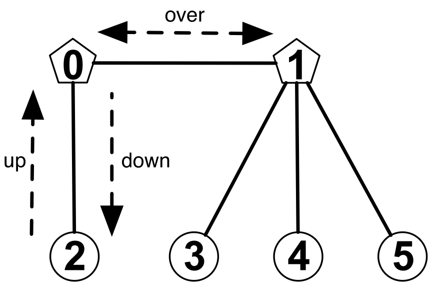


**Figure 2: iBGP route-reflection arc types**

# 3. OSCILLATION

## 3.1 MED Oscillation

ASes often interconnect in multiple locations. The MED attribute allows an AS announcing a route to define a preference over its interconnection links. The MED attribute is set locally by an AS announcing the route, so comparisons between ASes are meaningless. For example, one AS may announce routes with MED values 100 and 110 indicating a preference for the first route. However, another AS may indicate a similar preference with MED values 1000 and 1100. Hence it is only meaningful to compare MEDs with those received from the same AS.

A neighboring AS can indicate a preference for a route without setting MEDs directly, e.g., using a predetermined *community* attribute that is matched by the receiving AS for the purpose of setting the MED attribute to a *locally* comparable value. For example the receiving AS could define a community to signify a "backup route". Whenever an AS's neighbor attaches this community to a route, the AS sets the MED value to 110 instead of 100, so that backup routes among neighboring ASes have comparable MED values. However, this type of approach does not cover all uses for MEDs. The community attribute must be pre-defined, while the MED attribute is dynamic. A common implementation of the MED attribute is to set it to the (dynamically determined) IGP cost of a route to allow for "cold-potato" routing. IGP distances are administratively configured, so two neighboring ASes may have quite different "distances" defined, even for links that have the same geographical distance. Again, the resulting MEDs will not be comparable between neighbors.

Performing comparisons amongst a subset of routes, in combination with the route-reflector hierarchy's information hiding, may lead to a non-transitive ordering of routes. That is, a route $A$ may be preferred over a route $B$ and $B$ over $C$, but $C$ may is still preferred over $A$. This lack of transitivity can cause oscillation [3–5].

In Figure 3 we show an example configuration suffering from this form of oscillation. Let us start with each route-reflector learning routes from its clients. Router 0 cannot compare MEDs between routes learned from routers 2 and 3, and so will choose the route learned from 3 because of its lower IGP distance. Router 1 will choose the only route it currently knows, that learned from 4.

The route-reflectors will then inform each other of their choice. When router 0 is presented with a choice between all three egress routes, it will discard the route learned from 3 because of its higher MED value. Then it will compare the IGP distances of the other two routes, and select the closest, namely the route learned from router 2.

When router 1 learns of this egress point from router 0, it will compare the IGP distances of its client and router 2, and then choose

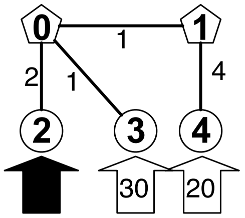


**Figure 3: MED oscillation with route-reflection. Route-reflectors are shown by pentagons and clients by circles. Links show both iBGP sessions, and IGP links with distances noted (though this is the only example where the two correspond). The large arrows show where we learn of external routes, with the color indicating the source AS. The MED values (where relevant) are shown inside the arrow.**

the closest, namely router 2. However, once it has chosen this route, it will no longer advertise its direct client to router 0, so router 0 will revert to choosing between its own clients, and we already know the decision, namely the route through router 3. This changes the information available at router 1, which once again changes its decision, leading to a persistent cycle of oscillation.

The problem arises because of the non-transitivity of router 0's route preferences, i.e., 4 is preferred to 3, 3 is preferred to 2, and 2 is preferred to 4; combined with information hiding that prevents complete information from being visible.

The potential problem of MED oscillation leads many ASes either to ignore MEDs or use the option `always-compare-med`, which leads to global MED comparisons. The latter option prevents MED-related instability, but the cost is that we make meaningless MED comparisons between routes learned from different ASes.

In Section 5 we show that the MED attribute can be compared on a per-AS basis if we introduce an additional BGP decision step prior to the comparison of the MED attribute. Further, in Section 6, we relax the restriction of propagating one route per destination and demonstrate we can satisfy the semantics of the MED attribute better than route-reflection and also guarantee network stability.

## 3.2 iBGP Topology Oscillation

Route oscillation can occur even when an AS chooses to ignore MEDs or compare the MED attribute across all ASes. It is caused by the interaction between the route-reflector iBGP topology and the IGP [2, 13]. Route-reflection determines how routes propagate, while the best route is chosen based on the IGP distance to the egress router. Consequently, route-reflectors' decisions can form a circular *reliance* which may oscillate [13].

Griffin and Wilfong [2] first demonstrated that this form of oscillation can occur. We present a simple example of such oscillation in Figure 4. The figure displays the vital IGP distances on lines connecting routers. All other distances are either irrelevant or can be considered large enough to not influence the BGP decision process. Notice that each route-reflector is closer to another route-reflector's client router. Consequently, when a route-reflector learns a route from another route-reflector's client, it will discard its own client's route. This process results in a circular reliance of router decisions and persistent oscillation ensues [2, 13].

This example led Griffin and Wilfong to prove that stability would be ensured if all route-reflectors select a client-learned route [2]. Sobrinho [10] algebraically proves the same condition and explicitly states it can be satisfied if the IGP distance from route-reflectors to their own clients is shorter than the distances to any other border router. However, as shown in Figure 5, configuring IGP distances in a way such that all distances to client routers are closer than non-clients is sometimes impossible. Further, relying on IGP distances can be troublesome as they are dynamic and can change due to link failures or additions, or when we perform traffic engineering [21].

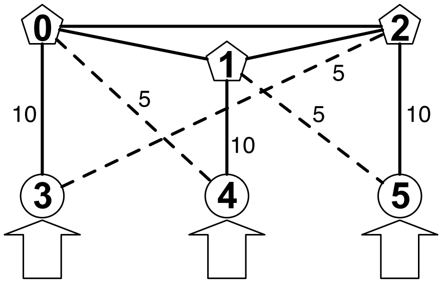


**Figure 4: iBGP topology oscillation. Vital IGP distances are shown on lines connecting routers. Solid lines are iBGP sessions while dotted lines are used only to show the IGP distance between routers. The large arrows indicate where external routes (equivalent up to step 4 of the decision process) are learned.**

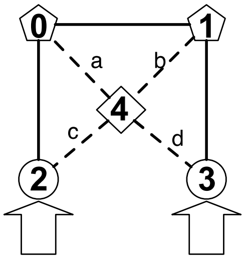


**Figure 5: The physical IGP topology is shown by dashed lines, while the iBGP topology is shown by solid lines. No distances can be configured such that $a+c < a+d$ and $b+d < b+c$, so the network cannot be configured so that clients are closer to their parents than to other route-reflectors.**

Also, Griffin and Wilfong's condition is sufficient, but not necessary, so it restricts design choices unnecessarily. Likewise the other design-based approaches [11, 12] dictate features of the network, such as which routers must be route-reflectors. Although such approaches are resilient to some failure scenarios, there is no guarantee the underlying network properties will remain identical as the network grows. Re-structuring the entire iBGP topology every time we add a link or router to the network is infeasible. Such approaches may also result in iBGP topologies that may have nice properties, but which are not logical from an operator's perspective — an important network property [22].

In addition to the standard results on oscillation that appear above, we have found that in multi-level hierarchies, configuring IGP distances such that downstream routers are closer than any others *cannot* prevent oscillation. Such hierarchies are used to improve scalability, and we know of at least two networks that have used this approach (operators are typically reluctant to reveal in internal details of their network design, so there may be many more multi-level hierarchies in operation). Figure 6 shows an example 3-level hierarchy of route-reflectors. Rather than indicate all of the IGP distances, we indicate the ordering of these distances via the list besides each route-reflector: for instance router 3 prefers (in order) the routes exiting the network from routers 6, 7 and 8. All pref-

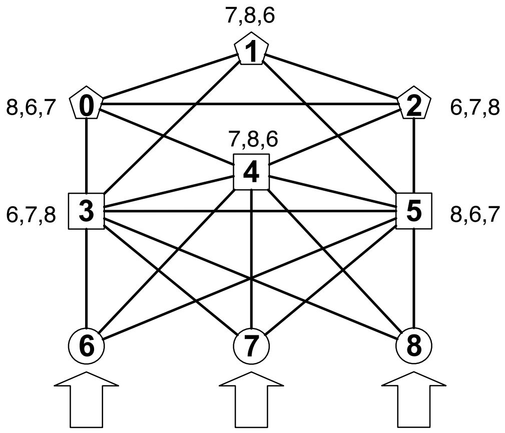


**Figure 6: Three-level iBGP topology oscillation. The preferences of each router are shown next to each node.**

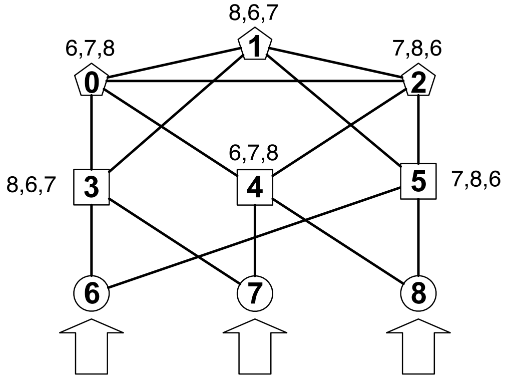


**Figure 7: Three-level iBGP topology exhibiting oscillation between levels.**

erences are configured such that a downstream egress is preferred. Now, all routers in the second level of the hierarchy will learn of their most preferred route and hence will select it. For instance, router 3 will select the route learned from 6. Now consider the routes available to the top-level route-reflectors. Notice that route-reflectors 0, 1 and 2 learn two routes. However, no route *learned* from a client router is the most preferred route. Consequently, the sufficient condition of Griffin and Wilfong that all routers select a client-learned route is *not satisfied*. The top-level route-reflectors will oscillate in a similar manner to the example in Figure 4 despite the IGP distances being configured to prefer downstream egresses over all others.

We have found that oscillation is not restricted to routers within the same level of the route-reflector hierarchy. It can also occur between routers in different levels of the route-reflector hierarchy. In the example shown in Figure 7 oscillation is caused by a middle level route-reflector selecting a route learned from a parent router over a client-learned route. For example, router 3 can learn of the route originated at router 8 from the signaling path $8-5-2-0-3$, and it will prefer this path because the IGP distance to 8 is smaller than the distance to 6. The oscillatory cycle affects the decisions of all six route-reflectors.

There are several existing methods to prevent iBGP topology based route oscillation. The first approach is to design our network in such a way that oscillation will not occur [2, 10–12]. We have already discussed the difficulties with this approach. Likewise, we have pointed out the difficulties of checking configurations before implementation [13], and of centralized router decisions schemes [8, 9, 23].

There are also proposals to increase the information propagated by route-reflectors. Bonaventure [24] propose that route-reflectors determine their clients' best route and propagate it to them. This may help route-diversity [25], however, it does not solve the underlying issue of protocol correctness and guaranteed convergence. Basu *et al.* [3] show MED oscillation can be prevented if routers propagate multiple routes. However, their approach is unable to prevent iBGP topology oscillation. In Section 6 we propose a similar concept, but we prove that our approach can satisfy all the goals of the MED attribute and also prevent oscillation, even in multi-level hierarchies.

All the examples presented in this section highlight that iBGP is not correct in the sense that it can oscillate. In the following sections, we use a routing algebra to describe iBGP and develop several approaches to prevent *all* forms of iBGP oscillation in *all* iBGP topologies — not just route-reflection. Our first approach can be implemented without altering the current information propagated between routers.

## 4. ROUTING ALGEBRAS

Sobrinho's pioneering work defining routing algebras [10] can be used for purposes such as proving properties of existing routing protocols [10,13,26] and as a language to define new protocols with provable properties [26, 27]. In this section we outline the basic building blocks of a routing algebra using a simple distance-vector routing protocol as an example. We then use the same techniques to describe the route-reflection iBGP topology.

A routing algebra consists of an ordered sextet

$$(L, \Sigma, f, W, \preceq, \oplus).$$

It comprises:

- a set of *labels* $L$;
- a set of *signatures* $\Sigma$;
- a set of *weights* $W$;
- a function $f$ that maps signatures into weights;
- a total order $\preceq$ on $W$; and
- a binary operation $\oplus$ that maps pairs of a label and a signature into a signature, i.e., $\oplus : L \times \Sigma \to \Sigma$.

The set of labels $L$ contains all feasible edge labels for a topology. In a distance-vector routing protocol (with positive integer distances), labels are simply the configured distances associated with physical links, e.g., they might be the set of natural numbers,

$$L = \mathbb{N}.$$

The set of signatures $\Sigma$ describes all feasible routes. $\Sigma$ also implicitly contains the special signature $\phi$ that represents a prohibited or invalid route. In a distance-vector routing protocol, signatures represent the distance to the destination, i.e.,

$$\Sigma = \mathbb{N}.$$

A node often has multiple signatures (or routes) it can select. Its selection is based on what it determines is the 'best'. Often the

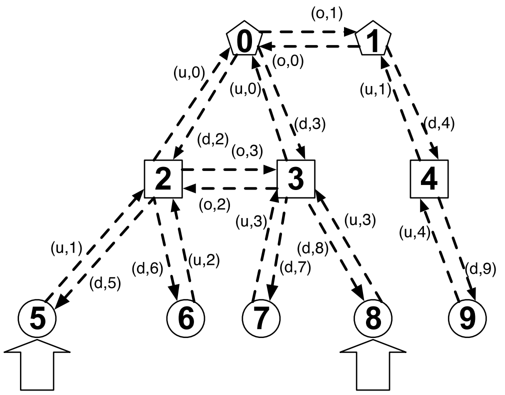


**Figure 8: Example route-reflector topology with directed edges labeled with edge types and head node identifiers.**

route signatures contain multiple attributes, and a node's selection is based on a predefined criteria. We use the function $f$ to convert signatures to a set of weights $W$ that are comparable using the operator $\preceq$. The preference of weights must be transitive. That is, if $a$ is preferred to $b$ and $b$ is preferred to $c$, then $a$ must be preferred to $c$. In the case of a distance-vector routing protocol, the function $f$ simply returns the distance of the signature, and these are compared numerically with the minimal distance route preferred.

Within a network, routers propagate their chosen route to neighboring routers. In a routing algebra, this process is undertaken by the $\oplus$ operator. The $\oplus$ operator takes a signature (or route), together with an edge label, and returns a signature. In the case of a distance-vector protocol, a node's selected signature is the distance to the destination. This signature, together with an edge label representing the distance associated with a single link are combined to create a new signature representing the new distance to the destination, i.e., the sum of the existing signature distance and the edge distance. That is for $\sigma \in \Sigma$ and $l \in L$, $\sigma \oplus l = \sigma + l$.

Sobrinho [10] showed that the important algebraic property is strict monotonicity. Strict monotonicity ensures the preference of a route strictly *decreases* when it is propagated. That is for all $\sigma \in \Sigma - \{\phi\}$, and for all $\lambda \in L, f(\sigma) \prec f(\lambda \oplus \sigma)$ where the $\prec$ operator indicates a strict preference of the former over the latter. If an algebra is strictly monotonic, then the protocol is correct and convergence is guaranteed. In the case of our simple distance-vector protocol, all distances are positive integers and hence strict monotonicity clearly holds.

## 4.1 Route-Reflection Algebra

Sobrinho describes a two-level route-reflector hierarchy as an algebra to show that IGP distances can be configured to prevent it from oscillating [10]. However, hierarchies of at least three levels are used, and as we demonstrated in Section 3, preventing oscillation in a multi-level hierarchy is more difficult with IGP distances. This is especially true when IGP distances are dynamic. We would like to have oscillation prevented under any scenario. Consequently, we now describe the multi-level route-reflector iBGP topology as an algebra (excluding MEDs for the moment).

The edge labels in a route-reflection algebra are pairs consisting of the edge type (either **down**, **up** or **over**) and the identifier of the node at the head of the directed edge. For example, in Figure 8, the **up** edge from node 6 to node 2 is labeled $(u, 2)$ while the **over** edge from node 0 to node 1 is $(o, 1)$. The set of all edge labels is defined as the lexical cross product of all edge types and node identifiers. That is,

$$L = \underset{\substack{\uparrow \\ \text{edge} \\ \text{type}}}{\{d, u, o\}} \times \underset{\substack{\uparrow \\ \text{head node} \\ \text{identifier}}}{\mathbb{Z}^+}$$

The set of signatures contains route attributes that are either used to determine a route's preference or to determine how it is propagated. We denote route signatures by a tuple containing the type of edge on which the route was learned, the identifier of the node that receives a route and the identifier of the node that originated the route (the egress node). In the example illustrated in Figure 8, the signature of a route originating at node 5 and available at node 2 would be $(u, 2, 5)$. We define an additional edge type **external** $(e)$, to represent a route learned from an external source (such as another AS). The set of all signatures is hence defined as the following cross product:

$$\Sigma = \underset{\substack{\uparrow \\ \text{edge} \\ \text{type}}}{\{d, u, o, e\}} \times \underset{\substack{\uparrow \\ \text{current} \\ \text{node}}}{\mathbb{Z}^+} \times \underset{\substack{\uparrow \\ \text{egress} \\ \text{node}}}{\mathbb{Z}^+}$$

where we also add the special signature $\phi$ to $\Sigma$ to represent an invalid or prohibited route.

A valid signaling path consists of zero or more **up** edges followed by zero or one **over** edges followed by zero or more **down** edges. The binary operator $\oplus$ incorporates these rules:

| | | Signature, $\Sigma$ | | | |
|---|---|---|---|---|---|
| | $\oplus$ | $(e, k, k)$ | $(d, i, k)$ | $(o, i, k)$ | $(u, i, k)$ |
| Link labels, L | $(d, j)$ | $(d, j, k)$ | $(d, j, k)$ | $(d, j, k)$ | $(d, j, k)$ |
| | $(o, j)$ | $(o, j, k)$ | $\phi$ | $\phi$ | $(o, j, k)$ |
| | $(u, j)$ | $(u, j, k)$ | $\phi$ | $\phi$ | $(u, j, k)$ |

Referring to our example in Figure 8, a route learned via an external source at node 5 (with signature $(e, 5, 5)$) when propagated along an **up** edge to node 2 has the signature $(u, 2, 5)$ (i.e. $(u, 2) \oplus (e, 5, 5) = (u, 2, 5)$). Also, a route learned from a **down** edge cannot be propagated via an **over** edge. Hence $(o, 2) \oplus (d, 3, 5) = \phi$.

The function $f$ converts signatures into easily comparable weights. The iBGP route decision process selects routes based on the closest IGP distance to the egress node, and if equal distances, on the lowest identifier of the egress node. Hence, the function $f$ is defined by

$$f(\sigma) = \begin{cases} (dist(i, k), k), & \text{if } \sigma = (*, i, k), \\ (\infty, \infty), & \text{if } \sigma = \phi. \end{cases}$$

where $dist(i, k)$ is the IGP distance from node $i$ to node $k$.

We compare weights lexicographically just as in the iBGP decision process. That is, we first prefer a route with the lowest IGP distance, and if equal, prefer the route with the lowest identifier of the egress node.

For iBGP to be strictly monotonic, after each iBGP hop, the preference of all signatures must decrease. The weights of signatures representing an external route will increase their IGP distance when propagated (and hence decrease their preference). The following non-trivial preferences must also be true for the algebra to be strictly monotonic:

$$\begin{aligned} f(d, i, k) &< f(d, j, k), \\ f(o, i, k) &< f(d, j, k), \\ f(u, i, k) &< f(d, j, k), \\ f(u, i, k) &< f(o, j, k), \\ f(u, i, k) &< f(u, j, k). \end{aligned}$$

Route IDs are arbitrary, so the above conditions require that at each iBGP hop the IGP distance must get progressively larger for strict monotonicity to hold. We have already shown the difficulties in ensuring that such a condition holds.

Instead we suggest changing the decision process to force the algebra to be strictly monotonic. One possible approach to this is to apply results from the eBGP context [26, 28] into iBGP. Griffin and Sobrinho [26] proved that preferring customer-learned routes over peer-learned routes over provider-learned routes in the eBGP context prevents oscillation between ASes. A similar preference in iBGP is also possible where routes learned from a client are preferred over routes learned from another route-reflector which, in-turn, are preferred over routes learned from a parent. An additional step in the BGP decision process could be implemented to ensure this condition is satisfied instead of relying on dynamic IGP distances. However, if we are required to alter the BGP decision process, why not prevent oscillation in all iBGP topologies simultaneously rather than simply route-reflection? We now define an alternative approach to ensuring strict monotonicity in route-reflection, before demonstrating it has general application to any iBGP topology.

## 4.2 Prefer Routes with Minimal iBGP Hops

We can ensure strict monotonicity in the iBGP algebra by preferring routes with minimal iBGP hops. We introduce a new parameter to the route signature — the number of iBGP hops to the router that originated the route into iBGP. The link labels remain unchanged, but the set of signatures becomes

$$\begin{array}{ccccccccc} \Sigma & = & \mathbb{Z}^+ & \times & \{d,o,u,e\} & \times & \mathbb{Z}^+ & \times & \mathbb{Z}^+ \\ & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\ & & \text{iBGP} & & \text{edge} & & \text{current} & & \text{egress} \\ & & \text{hops} & & \text{type} & & \text{node} & & \text{node} \end{array}$$

When a route is propagated, the number of iBGP hops is incremented. The $\oplus$ operator can be written

| $\oplus$ | $(0,e,k,k)$ | $(n,d,i,k)$ | $(n,o,i,k)$ | $(n,u,i,k)$ |
|---|---|---|---|---|
| $(d,j)$ | $(1,d,j,k)$ | $(n+1,d,j,k)$ | $(n+1,d,j,k)$ | $(n+1,d,j,k)$ |
| $(o,j)$ | $(1,o,j,k)$ | $\phi$ | $\phi$ | $(n+1,o,j,k)$ |
| $(u,j)$ | $(1,u,j,k)$ | $\phi$ | $\phi$ | $(n+1,u,j,k)$ |

and the function $f$ is modified such that routes with the minimal iBGP hops are preferred, i.e.,

$$f(\sigma) = \begin{cases} (n, dist(i,k), k), & \text{if } \sigma = (n, *, i, k), \\ (\infty, \infty, \infty), & \text{if } \sigma = \phi, \end{cases}$$

and once again weight comparisons occur in lexical order.

The proof that the new algebra is strictly monotonic follows directly from the demonstration in [26] that the *lexical product*, denoted by $\otimes$, of sub-algebras is strictly monotonic if they are combined as follows

$$A = \underbrace{\otimes \overbrace{A_1}^{\textbf{SM}} \overbrace{A_2 A_3 \ldots A_n}^{\textbf{irrelevant}}}_{\textbf{SM}}.$$

That is, if we compose our new algebra $A$ as a lexical product of a set of algebras $A_i$ where $A_1$ is strictly monotonic, then $A$ will also be strictly monotonic. In our case, the iBGP hop distance sub-algebra is obviously strictly monotonic, and hence so is the complete iBGP algebra including this step.

The direct result is that any part of the decision process following the iBGP hop step does not effect the monotonicity properties of the algebra, and so link types, IGP distances and node identifiers can be removed from the algebra without affecting the stability properties. Hence we can significantly simplify, and generalize our iBGP algebra to simplify the process of proving properties of more complex systems (say involving MEDs).

## 4.3 General iBGP Algebra

Our above description of iBGP is somewhat cumbersome due to the presence of route attributes that play little role. We now strip all such attributes, leaving an algebra that has the desired property of strict monotonicity and as described above can be combined with any other decision step. Our new algebra is not dependent on properties of the route-reflector topology and consequently is also applicable to other iBGP topologies such as confederations, full-mesh, or other customized iBGP topologies. The algebra essentially reduces to a simple distance-vector routing protocol described by the following algebra.

The label of an edge is irrelevant in this algebra. Consequently, we use a generic label $l$ to describe all edges. That is,

$$L = \{l\}.$$

A route has a signature based solely on the number of iBGP hops to reach the egress node. That is,

$$\Sigma = \mathbb{Z}^+.$$

The binary operator $\oplus$ simply increments the hop count

$$l \oplus n = n + 1.$$

The weight of a route is also the number of iBGP hops to the egress node so function $f$ returns the hop count

$$f(\sigma) = \begin{cases} n, & \text{if } \sigma = n, \\ \infty, & \text{if } \sigma = \phi. \end{cases}$$

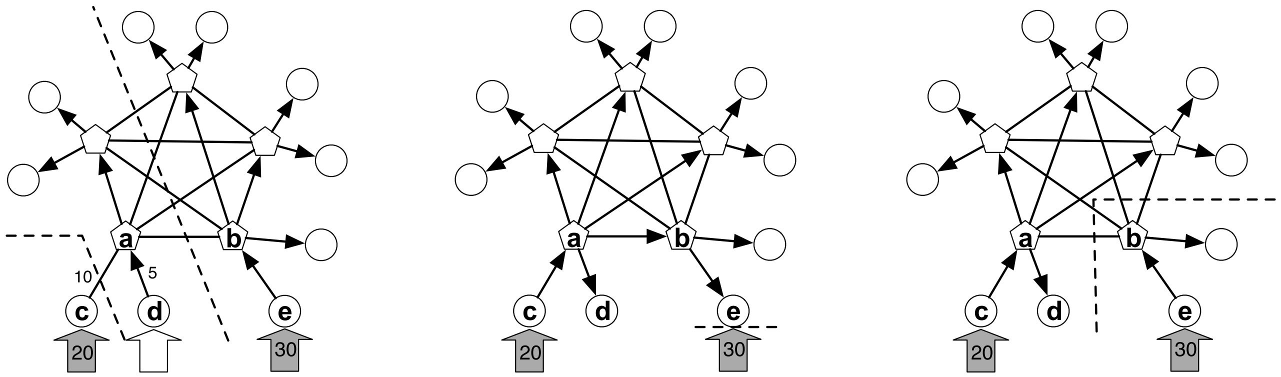
Clearly, this algebra is strictly monotonic. As mentioned earlier, when lexicographically combined (at the start) with any other routing algebras the resulting algebra will be strictly monotonic. Hence, sub-algebras such as the IGP distance step can be used without affecting the algebra's strict monotonicity (provided they are used *after* the strictly monotonic step), and we can see immediately that if MEDs are included in our routing algebra after the iBGP hop step, then the resulting algebra will be still be stable. So to simply ensure stability, the iBGP hop step should precede the MED step in the BGP decision process.

# 5. PRACTICAL CONSIDERATIONS

We have proposed an alteration to the BGP decision process that includes an extra decision step that prefers routes with the minimal iBGP hops. The *cluster-list* attribute in iBGP contains the identifiers of the clusters the route has traversed (similar to the AS-path attribute in eBGP). It is primarily used to avoid loops but it implicitly contains the number of iBGP hops, and we can use this in our new decision step. Hence, our alteration to the decision process does not require any change to the message format or information propagated between routers. Consequently, our new decision process can be incrementally deployed by router software upgrades without introducing incompatibility between routers with the existing BGP decision process.

What's more, oscillation can only occur between route-reflectors [13], so the upgrades are only necessary on route-reflectors.

The key implementation question is where to place the new step in the decision process (shown in Figure 1). The new step must be placed above the first decision that could cause oscillation, so there is no need (or desire) to place it above step 3 (lowest origin



(a) Despite the route at $c$ being the preferred egress for the dark shaded AS, a majority of routers select the route via $e$. The same route selections would be made under the current BGP decision process and our modified decision process.

(b) Under the current decision process, all routers select egress via the primary route through $c$.

(c) Under our modified decision process, three routers select egress via less preferred router $e$.

**Figure 9: The MED value is within the arrow indicating the input route. The shading of the input route indicates the neighboring AS. The relevant IGP distances are shown on links $a - c$ and $a - d$. Route-reflector $b$ is defined to be closer to $e$ than any other egress. The arrows on BGP sessions indicate the signaling path of the chosen route at each node. We use dashed lines partition the network into routers that make the same route selection.**

type) as steps 1-3 cannot cause oscillation. If we place it after step 4 (MEDs) but before step 5 (IGP distance), then we cannot prevent MED oscillation, but this may be satisfactory where MEDs are compared network wide (say through the use of community attributes). However, the obvious place to put the new decision step is between steps 3 and 4.

The additional step may have consequences for routing decisions, apart from enforcing stability, and we consider these below, but it is important to note that if this decision step were built into routers, it could be included as an option so that each network operator could decide whether to turn it on or not, or whether it should precede the MED decision.

## 5.1 Internal Optimality

We have so far concentrated on what is gained by preferring a route with the minimal iBGP hops. Here we consider what might be lost. We call a chosen route *internally optimal* if a router would select the same route (ignoring MEDs) if it learnt all possible routes, and *externally optimal* if it would make the same decision including MEDs. The first 3 steps of route selection are compared network wide, so do not change here. Hence, when our new decision process chooses an "internally suboptimal" route it means the route has a longer IGP distance to the egress point. However, note that standard route-reflector hierarchies can also cause internally suboptimal route selections by hiding some routes.

There are good reasons for a protocol to choose closest egress points. Minimizing such distances avoids transitting data an unnecessarily extra distance, and ensures consistent routing, thereby avoiding forwarding deflections [2], though this can also be prevented using a protocol such as MPLS [15] to tunnel traffic from ingress router to egress router.

Our approach prefers routes based on the iBGP hops prior to the IGP distance. Consequently, routers may select a route that is not the closest. However, when Griffin and Wilfong's condition [2] is satisfied the shortest IGP distance route will be one iBGP hop away and so when our decision step is not needed to ensure stability, it does not make route selection worse. Selecting a longer route on occasion is a small price to pay for guaranteed stability, particularly when route-reflectors already result in suboptimal decisions.

## 5.2 External Optimality

The MED attribute is used to indicate a neighboring AS's preference for its multiple links. So a route that is internally optimal, but externally *suboptimal* implies that we have ignored the semantics of the defined MEDs. Once again our new decision step (when placed above step 4) can result in such routes, but so also can standard route-reflector hierarchies. For instance, see Figure 9(a), which shows that the presence of the route from the white AS at $d$ causes a significant fraction of routers to select the route through $e$ to the dark shaded AS, despite the lower MED for the route at $c$. In that case, our additional decision step doesn't change any decisions. However, Figure 9(b) and (c) illustrate a case where our process changes the default decision made by the route-reflector hierarchy. Preferring routes with the minimal iBGP hops prior to the MED step causes three routers (see Figure 9(c)) to select the less preferred route through router $c$.

A legitimate concern is that ignoring MED semantics could violate current contractual obligations that require the MED attribute to be respected. Instead of simply considering how the MED attribute can be respected, we now consider *why* an AS may want the MED attribute to be respected and how we can implement their requirements under our preference for the route with the minimal iBGP hops. Two reasons for using MEDs are

1. Some links are intended as backup links;
2. Some links may be associated with a higher *internal* cost, perhaps because of congestion internal to the AS.

In the first case, the *link* should always be considered a backup link. In this case, the same objective can be realized by setting the community attribute as described in Section 3.1. In fact, this approach is more effective than using the MED attribute as it ensures the backup link won't be used in cases such as shown in Figure 9(a).

In the second case, the goal is to use MEDs for traffic engineering. That is, MEDs are used to control the ingress links for some traffic entering the AS, so that we can better balance internal loads on the network. An extreme case of this type of traffic engineering is cold-potato routing where we aim for traffic to enter our network as close as possible to its destination. Here the MED attribute may be more dynamic, for instance it may be selected to be the internal IGP distances (of the AS advertising the route). However, such MEDs may not even be respected on a full-mesh iBGP topology [5], so in the current network these MEDs can only be seen as indications of preference, not requirements. It is unreasonable to enforce them in preference to stability.

Despite these issues, it is possible to *completely* satisfy the objectives of the MED attribute while guaranteeing stability, and we consider this problem in the next section.

# 6. MULTIPLE ROUTES FOR MED

We demonstrated in the above section that we can ensure strict monotonicity with the cost that we may not respect MEDs that have the possibility of causing oscillation. However, we were assuming each router was only able to propagate a single route. Let us now relax this assumption and discover what is required to *completely* respect MEDs *and* guarantee iBGP correctness.

First let us consider if we have routes to a destination learned solely from a *single* AS over multiple links. Our algebraic description remains similar to that described in Section 4.2. We describe the algebra with respect to the route-reflector topology, however, it is equally applicable to our general iBGP topology from Section 4.3 (with the equivalent general iBGP topology in Appendix A).

The set of edge labels remains

$$L = \{d, o, u\} \times \mathbb{Z}^+.$$

However, the set of signatures is extended to include the MED attribute of the route, together with the number of iBGP hops, edge type, identifier of the current node and identifier of the originating node.

$$\Sigma = \underset{\substack{\uparrow \\ \text{MED}}}{\mathbb{Z}^+} \times \underbrace{\mathbb{Z}^+ \times \{d, o, u, e\} \times \mathbb{Z}^+ \times \mathbb{Z}^+}_{\text{as in Section 4.2}}$$

The binary operator $\oplus$ is equivalent to the previous definition except we also propagate the MED value associated with the route (unchanged), as shown in Table 1, and the function $f$ is defined by

$$f(\sigma) = \begin{cases} (MED, n, dist(i,k), k) & \text{if } \sigma = (MED, n, *, i, k), \\ (\infty, \infty, \infty, \infty) & \text{if } \sigma = \phi. \end{cases}$$

We compare routes lexicographically with lower numeric values preferred. That is, we first compare a route's MED attribute, and if equal, compare the number of iBGP hops, followed by the IGP distance and egress node identifier, i.e., we add our new iBGP hop decision step between steps 4 and 5 of the iBGP decision process, so that MEDs are respected. As we are only considering one neighboring AS, the MED attribute is *always* directly comparable making the weight a total order. Also, as it is a static value for a route, the ranking of routes is dependent on the number of iBGP hops and hence the algebra is strictly monotonic and the algebra is guaranteed to converge. Notice, once again, the IGP distance does not affect the strict monotonicity of the function, because it is compared after the number of iBGP hops.

Thus for a single neighboring AS, we have a strictly monotonic algebra that respects MEDs. The difficulty in ensuring the semantics of the MED attribute are fulfilled arises when multiple ASes announce the same route to a single destination and the MED attribute is compared on a per-AS basis (see Figure 9). The crux of MED oscillation is that the border router learning a route does not know that its route is less attractive than another available route. However, if we relax our restriction on each router propagating only a single route, we can ensure that the border router is aware of the availability of a better route.

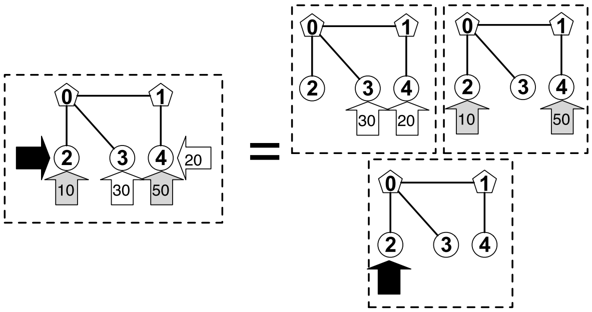


**Figure 10: Multiple ASes originating routes with MED attribute. Each AS has its own algebra for route propagation. The actual route selected by each router is irrelevant.**

Our approach is to construct one algebra *per neighboring AS* and compose these using a Cartesian product (see Appendix B for details). Strict monotonicity of the individual algebras ensures that the Cartesian product is strictly monotonic.

The idea corresponds to propagating one route per neighboring AS. We illustrate this approach in Figure 10. In this example three neighboring ASes have routes to the same destination. For each neighboring AS we allow one route to propagate through the iBGP signaling network, and this propagation is described by an algebra. The MED attribute is compared within each of these sub-algebras, thus avoiding MED oscillation (IGP related oscillation is avoided by our minimum iBGP distance step). The decision of which route to select for forwarding is made through a comparison of all available routes learned (from all algebras). This final route is not propagated, and so doesn't affect stability. In this way we decouple *route-propagation* from *route-selection*.

An obvious problem is the increase in information that need be propagated and maintained in databases at each router. We can reduce this, however, by noting that not all neighboring ASes will use MEDs. For example, MEDs are not typically accepted from provider or peer ASes, and for ASes with a single interconnection (such as many customer ASes) the MED attribute is irrelevant. We can group all such "MEDless" ASes together into one algebra.

Figure 11 shows six ASes (with routes to a particular destination), but only four algebras are required. Hence the state information at each router is reduced, as is the amount of information propagated. Notice in the third algebra only one route is announced, but this AS still has its own algebra. We have used this example to demonstrate that if an AS has the *capability* to use MEDs, a separate algebra is required, even if it is not really used. So it is useful to consider a policy of respecting the MED attribute to be a premium routing service provided only to those neighbors with special business relationships. Other ASes have their MEDs reset so that they have no effect. We could can then charge or otherwise obtain compensation for the extra resources being used in allocating a separate routing algebra to neighbors using MEDs, allowing our neighbors to make a rational (economic) decision about the costs and benefits of using MEDs.

| $\oplus$ | $(MED, 0, e, k, k)$ | $(MED, n, d, i, k)$ | $(MED, n, o, i, k)$ | $(MED, n, u, i, k)$ |
|---|---|---|---|---|
| $(d, j)$ | $(MED, 1, d, j, k)$ | $(MED, n+1, d, j, k)$ | $(MED, n+1, d, j, k)$ | $(MED, n+1, d, j, k)$ |
| $(o, j)$ | $(MED, 1, o, j, k)$ | $\phi$ | $\phi$ | $(MED, n+1, o, j, k)$ |
| $(u, j)$ | $(MED, 1, u, j, k)$ | $\phi$ | $\phi$ | $(MED, n+1, u, j, k)$ |

**Table 1: The binary operator $\oplus$ for the iBGP topology with a single neighboring AS and respecting the MED attribute.**

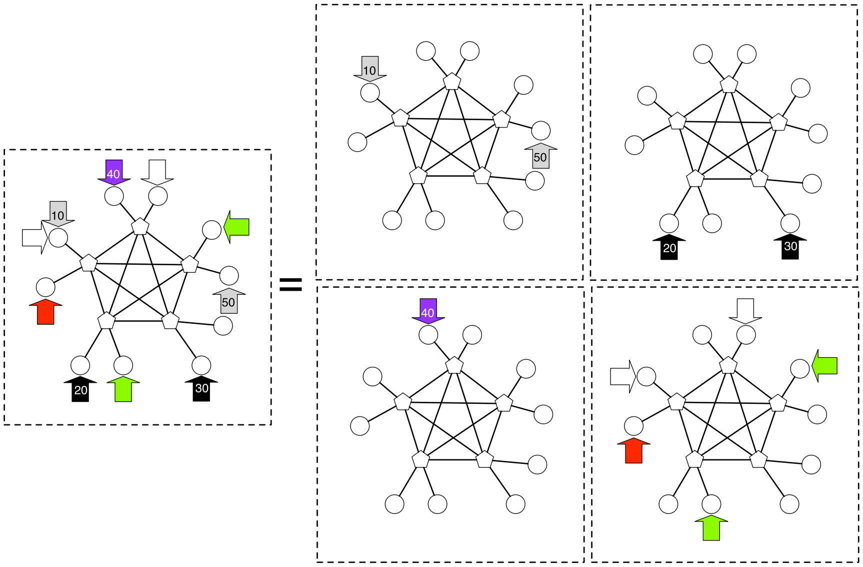


**Figure 11: Six neighboring ASes (indicated by colors) originating routes to a destination. Each AS with a business relationship stipulating MED should be respected has its own algebra for route propagation, but the other three are grouped together in one algebra. Notice that in the third algebra, only one route is announced. The AS involved (purple) has the *ability* to announce multiple routes with the MED attribute. It chooses to not to take advantage of this ability, but still requires its own algebra.**

## 7. DISCUSSION

A benefit of our approach is that we separate route propagation from route selection. That is, any route learned can be selected, including one not in the set of routes propagated[3], without affecting convergence. Hence, we can require the route *propagation* process to be strictly monotonic, but allow the route *selection* to be rather more flexible as this decision is not part of the information propagated in the protocol and so cannot cause instability.

An additional concept introduced in the previous section, was to layer multiple small, simple algebras into a larger more complex route propagation scheme without sacrificing the properties of the individual algebras. This approach lends itself to applications other than satisfying the semantics of the MED attribute, some of which we discuss in this section.

The separation of the control and forwarding plane is desirable as it allows better network management and allows ASes to offer value-added services to customers [9, 29–31]. We provide the framework that allows the design of individual solutions that can be combined without adversely affecting protocol correctness, while keeping the distributed nature of current routing protocols. Control plane virtualization has been proposed to allow multiple logical networks to run on a single physical infrastructure [32]. Similar techniques could be utilized to separate route propagation schemes. However, our approach also lends itself to forwarding tables constructed from a communal pool of routes learned across a number propagation schemes.

We show an illustration of the separation between route propagation and route selection in Figure 12. In this example, three algebras exist and each has its own propagation decision process. This is similar to the network virtualization proposed in the GENI project [33]. Each algebra can act on the entire or a subset of the topology and on all or a subset of destinations. The route selection process is orthogonal to the route propagation process, and any number of forwarding tables can be constructed — there does not need to be one forwarding table per algebra. Further, the routes selected do not need to match the routes propagated. Consequently, network applications such as differentiated quality-of-service may be implemented without affecting control-plane convergence.

Data is forwarded by the IGP. Forwarding deflections may occur when routes are selected that are not the shortest IGP path. By separating the decision process from the propagation process, the ingress router receiving data from a source outside the network selects the egress router based on its knowledge of the network and the IGP forwards the data to the egress router[4]. No intermediate routers are required to select the egress router. This approach, separates internal routing from external routing. Further, an administrator can easily alter internal traffic flows by changing the route-selection criteria without affecting the control plane. This may have applications in reducing the sensitivity of a network to internal routing dynamics [34–36], and a real-time network optimizer such as [37] could be utilized without affecting routing convergence.

An added benefit of guaranteeing the routing protocol's correct-

[3] The route propagated to routers in neighboring ASes *should* remain the route selected to ensure accurate global routing.

[4] This could be undertaken by encapsulating the packet and addressing it to the egress router or with a protocol such as MPLS [15].

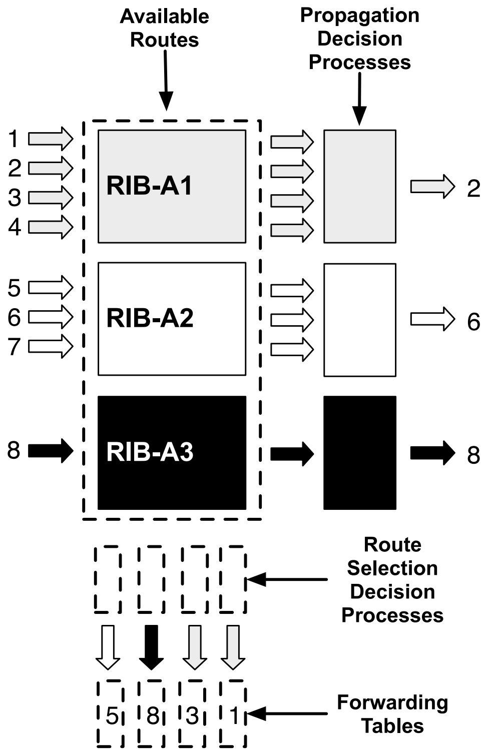


**Figure 12: Router architecture separating route propagation from route selection. Each of the three algebras shown has its own iBGP topology so a router can have different numbers of iBGP sessions in different topologies. Each algebra selects a single route to propagate to neighboring routers. Multiple decision processes are possible for differentiating traffic flows.**

ness in this way is that it implies a unique routing solution. Consequently, for each individual algebra, we could use the route selection criteria to deterministically predict the routes selected by all routers in the network. This added transparency in the iBGP decision process can aid network management through easier debugging and the ability to build automated tools predicting the impact of network changes prior to their occurrence [18, 38].

## 8. CONCLUSION

Ensuring the stability of iBGP is currently left to operators. Some guidelines for operators exist, as well as alternative proposals to stabilize iBGP, but in this paper we present methods for guaranteeing stability of the protocol itself. We use an algebraic description of routing protocols to prove convergence.

Our simplest proposal (including a minimum iBGP hop step in iBGP decisions) doesn't require additional information to be propagated between routers and only a minor modification to the BGP decision process. We have also developed a framework to allow an AS to satisfy customized contractual obligations and traffic engineering constraints by implementing highly flexible and predictable routing strategies without affecting stability. In future work we plan to further evaluation this framework, in particular its performance.

## Acknowledgments

This work was supported in part by ARC grant DP0557066. The authors would also like to acknowledge many useful discussions with Rui Zhang-Shen and Jennifer Rexford.

## 9. REFERENCES

[1] T. Griffin, F. B. Shepherd, and G. Wilfong, "The Stable Paths Problem and Interdomain Routing," *IEEE/ACM Transactions on Networking*, vol. 10, no. 2, pp. 232–243, 2002.

[2] T. Griffin and G. Wilfong, "On the Correctness of IBGP Configuration," in *ACM SIGCOMM*, 2002.

[3] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route Oscillations in I-BGP with Route Reflection," in *ACM SIGCOMM*, 2002.

[4] D. McPherson, V. Gill, D. Walton, and A. Retana, "Border Gateway Protocol (BGP) Persistent Route Oscillation Condition," 2002, RFC 3345.

[5] T. Griffin and G. Wilfong, "Analysis of the MED Oscillation Problem in BGP," in *IEEE International Conference on Network Protocols*, 2002.

[6] G. Wilfong, "Interdomain Routing," Lucent Technologies Presentation, February 2006.

[7] O. Maennel, A. Tudor, A. Feldmann, and S. Bürkle, "Observed properties of BGP convergence," 2003, RIPE 45.

[8] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and Implementation of a Routing Control Platform," in *Symposium on Networked Systems Design and Implementation*, 2005.

[9] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The Case for Separating Routing From Routers," in *ACM SIGCOMM Workshop on Future Directions in Network Architecture*, 2004.

[10] J. Sobrinho, "An Algebraic Theory of Dynamic Network Routing," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, October 2005.

[11] M. Vutukuru, P. Valiant, S. Kopparty, and H. Balakrishnan, "How to Construct a Correct and Scalable iBGP Configuration," in *IEEE INFOCOM*, Barcelona, Spain, April 2006.

[12] M. Buob, S. Uhlig, and M. Meulle, "Designing Optimal iBGP Route-Reflection Topologies," in *IFIP Networking*, 2008.

[13] A. Flavel, M. Roughan, N. Bean, and A. Shaikh, "Where's Waldo? Practical Searches for Stability in iBGP," in *IEEE International Conference on Network Protocols*, 2008.

[14] D. Walton, A. Retana, E. Chen, and J. Scudder, "Advertisement of Multiple Paths in BGP," July 2008, Internet Draft.

[15] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," January 2001, RFC 3031.

[16] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, "Finding a needle in a haystack: Pinpointing significant bgp routing changes in an ip network," in *Usenix NSDI*, 2005.

[17] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4," RFC 4271, January 2006.

[18] N. Feamster and J. Rexford, "Network-Wide Prediction of BGP Routes," *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 253–266, 2007.

[19] T. Bates, R. Chandra, and E. Chen, "BGP Route Reflection - An Alternative to Full Mesh IBGP," 2000, RFC 2796.

[20] P. Traina, D. McPherson, and J. Scudder, "Autonomous System Confederations for BGP," 2001, RFC 3065.

[21] B. Fortz and M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," in *INFOCOM*, 2000.

[22] V. van den Schrieck, P. Francois, S. Tandel, and O. Bonaventure, "Let BGP Speakers Configure Their iBGP

Sessions On Their Own," in *Workshop on Internet Routing Evolution and Design*, 2006.

[23] R. Govindan, C. Alaettinog-lu, K. Varadhan, and D. Estrin, "Route Servers for Inter-domain Routing," *Computer Networks and ISDN Systems*, vol. 30, no. 12, pp. 1157–1174, 1998.

[24] O. Bonaventure, S. Uhlig, and B. Quoitin, "The Case for More Versatile BGP Route Reflectors," 2004, work in progress, draft-bonaventure-bgp-route-reflectors-00.txt.

[25] S. Uhlig and S. Tandel, "Quantifying the BGP Routes Diversity Inside a Tier-1 Network," in *Networking*, 2006.

[26] T. Griffin and J. Sobrinho, "Metarouting," in *ACM SIGCOMM*, 2005.

[27] A. Gurney and T. Griffin, "Lexicographic Products in Metarouting," in *IEEE International Conference on Network Protocols*, 2007.

[28] L. Gao, T. Griffin, and J. Rexford, "Inherently Safe Backup Routing with BGP," in *IEEE INFOCOM*, 2001.

[29] P. Verkaik, D. Pei, T. Scholl, and A. Shaikh, "Wresting Control from BGP: Scalable Fine-grained Route Control," in *USENIX Annual Technical Conference*, 2007.

[30] H. Khosravi and T. Anderson, "Requirements for Separation of IP Control and Forwarding," November 2003, RFC 3654.

[31] Y. Wang, I. Avramopoulos, and J. Rexford, "Design for Configurability: Rethinking Interdomain Routing Policies from the Ground Up," *to appear in IEEE Journal on Selected Areas in Communications*, 2009.

[32] J. Turner, "A Proposed Architecture for the GENI Backbone Platform," in *Architecture for Network and Communications Systems*, 2006.

[33] "GENI Project," www.geni.net.

[34] R. Teixeira, A. Shaikh, T. G. Griffin, and G. M. Voelker, "Network Sensitivity to Hot-Potato Disruptions," in *ACM SIGCOMM*, 2004.

[35] R. Teixeira, A. Shaikh, T. G. Griffin, and J. Rexford, "Dynamics of Hot-Potato Routing in IP Networks," in *ACM SIGMETRICS*, 2004.

[36] R. Teixeira, N. G. Duffield, J. Rexford, and M. Roughan, "Traffic Matrix Reloaded: Impact of Routing Changes," in *Passive and Active Measurement Conference*, 2005.

[37] S. Kandula, D. Katabi, B. Davie, and A. Charny, "Walking the Tightrope: Responsive Yet Stable Traffic Engineering," in *ACM SIGCOMM*, 2005.

[38] A. Flavel, J. McMahon, A. Shaikh, M. Roughan, and N. Bean, "Humpty Dumpty: Putting iBGP Back Together Again," in *IFIP Networking*, 2009.

# APPENDIX

## A. CONVERGENCE OF IBGP WITH MEDS

In Section 6 we described the route-reflector iBGP topology and demonstrated when multiple routes are propagated, convergence is guaranteed. We now describe the general iBGP topology.

The set of generic edge labels are described as

$$L = \{l\}.$$

The set of signatures are generalized to

$$\Sigma \;=\; \underset{\text{MED}}{\underset{\uparrow}{\mathbb{Z}^+}} \;\times\; \underset{\text{iBGP hops}}{\underset{\uparrow}{\mathbb{Z}^+}}$$

The binary mapping function is equivalent to the previous definition except the MED value associated with the route is also propagated. This is described below.

| $\oplus$ | $(MED, 0)$ | $(MED, n)$ |
|---|---|---|
| $l$ | $(MED, 1)$ | $(MED, n+1)$ |

Our function $f$ converts the signature to a pair including the MED attribute and the iBGP hops.

$$\begin{aligned} f(MED, n) &= (MED, n) \\ f(\phi) &= (\infty, \infty) \end{aligned}$$

Similarly to the description of the route-reflector topology, the algebra is strictly monotonic for a single AS. Further, we can combine multiple strictly monotonic algebras as in Section 6 and guarantee convergence in a general iBGP topology.

## B. CARTESIAN PRODUCT OF ALGEBRAS

The Cartesian product of algebras is different to the *lexical* product considered previously [26, 27]. The lexical product of sub-algebras can be thought of as each algebra applying to a set of routes in series, progressively cutting the available routes at each step. The Cartesian product applies each sub-algebra's rules in parallel, allowing multiple routes to be propagated between nodes.

The Cartesian product on sets is a standard mathematical concept, most clearly seen in the extension of the real number line to $n$-dimensional Cartesian co-ordinate spaces. We typically denote the product of two sets by $A \times B = \{(a, b) | a \in A \text{ and } b \in B\}$. Also commonly defined is the Cartesian product of functions, i.e., $[f \times g](a, b) = (f(a), f(b))$.

We construct the Cartesian product of routing algebras by taking Cartesian products of each of the components of the algebra. For instance, take the algebra $A = (L, \Sigma, \oplus, f, W, \preceq) = A_1 \times A_2 \times \cdots \times A_m$, where each $A_i$ consists of $A_i = (L_i, \Sigma_i, \oplus_i, f_i, W_i, \preceq_i)$. The new edge label, signature and weight sets are composed using the standard Cartesian product for sets, i.e.,

$$\begin{aligned} L &= L_1 \times L_2 \times \ldots \times L_m, \\ \Sigma &= \Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_m, \\ W &= W_1 \times W_2 \times \ldots \times W_m, \end{aligned}$$

though note that we must extend $L_i$ so that the label $\phi$ represents all the links that don't exist in $L_i$, but are present in another sub-algebra $L_j$, i.e., so that we can define topologies for each sub-algebra, but still operate on the combined algebra. Note also that where weights in $W_i$ are already vectors, the composed weight set comprises matrices.

As with any vector space composed from a Cartesian product, the standard binary operator $\oplus$ on $A$ is simply the component-wise binary operator. That is,

$$(l_1, l_2, ..., l_m) \oplus (\sigma_1, \sigma_2, ...\sigma_m) = (l_1 \oplus_1 \sigma_1, l_2 \oplus_2 \sigma_2, ..., l_m \oplus_m \sigma_m).$$

The function that maps signatures to weights is just the Cartesian product of the component functions, i.e.,

$$f(\sigma_1, \sigma_2, ..., \sigma_m) = (f_1(\sigma_1), f_2(\sigma_2), ..., f_m(\sigma_m)).$$

Finally, the weights are compared by the operator $\preceq$ which applies each of the subcomparison operators $\preceq_i$ *component-wise*. That is, we would choose the preferred route from each sub-algebra independently based on its own comparison operator.

An algebra $A_i$ is strictly monotonic if and only if $f(\sigma_i) \prec f(\lambda_i \oplus \sigma_i)$ for all $\sigma_i \in \Sigma_i - \{\phi\}$, and $\lambda_i \in L_i$. The new algebra $A$ is strictly monotonic if and only if each $A_i$ is also strictly monotonic, as a direct result of the component-wise operation of $\oplus$ and $\prec$.