Route Oscillations in I-BGP with Route Reflection

Anindya Basu
Bell Laboratories
basu@research.bell-labs.com

Chih-Hao Luke Ong
Oxford University
Luke.Ong@comlab.ox.ac.uk

April Rasala
MIT
arasala@theory.lcs.mit.edu

F. Bruce Shepherd
Bell Laboratories
bshep@research.bell-labs.com

Gordon Wilfong
Bell Laboratories
gtw@research.bell-labs.com

Abstract

We study the route oscillation problem [16, 19] in the Internal Border Gateway Protocol (I-BGP) [18] when route reflection is used. We propose a formal model of I-BGP and use it to show that even deciding whether an I-BGP configuration with route reflection can converge is an NP-Complete problem. We then propose a modification to I-BGP and show that route reflection cannot cause the modified protocol to diverge. Moreover, we show that the modified protocol converges to the same stable routing configuration regardless of the order in which messages are sent or received.

Categories & Subject Descriptors: C.2.2 Routing Protocols.
General Terms: Algorithms
Keywords: I-BGP, Route Reflection, Route Oscillations, Stability.

1. Introduction

The Border Gateway Protocol (BGP) [18] has become the de-facto standard for inter-domain routing in today's Internet. External-BGP [10, 11] (or E-BGP) is the protocol used for exchanging external routing information among administrative domains (called Autonomous Systems or AS-es). In contrast, Internal-BGP [10, 11] (or I-BGP) is used for exchanging this external routing information among routers within the same AS.[1] It has been observed in practice that *persistent route oscillations* can occur when using I-BGP [16, 19] with route reflection [1] or confederations [20]. That is, some subset of the routers within an AS may exchange routing information forever without being able to settle on a stable routing configuration. This happens when no stable routing configuration exists. The other kind of route oscillation that can occur in a system is a *transient route oscillation*. In this case, some subset of routers may undergo route oscillations due to a timing co-incidence (such as message delays or a particular order in which routers send and receive messages). These route oscillations are transient in nature since they disappear when the timing co-incidences no longer exist.

The persistent route oscillation problem for I-BGP was first reported in a Field Notice from Cisco Systems [19]. This document described the persistent oscillation problem as an "Endless BGP Convergence Problem in Cisco IOS Software" as reported by certain customers in the field. We note that the persistent oscillation problem was reported for both route reflection configurations as well as confederation configurations. The positive results in the present paper address route reflection configurations exclusively.

As has been observed (e.g., by Walton et al. [23]), the key problem in persistent route oscillation (under route reflection scenarios) is the use of the Multi-Exit-Discriminator (or MED) attribute for route comparison. The MED attribute of a BGP route is a non-negative integer that is used to compare routes that pass through the same neighboring AS. The lower the MED value, the more preferred the route. The MED attribute value is used in configurations where multiple links connect the same AS pair. In such situations, the MED value of a route is used by the AS receiving traffic to indicate (to the sending AS) which links it prefers when receiving traffic. The BGP protocol requires that routers in the sending AS respect the MED values assigned to a route by the receiving AS. Since MED values are not used to compare routes that pass through different neighboring AS-es, the use of MED values may periodically hide certain routes from view and create the possibility for route oscillations — we explain this in greater detail in Section 3.

In their analysis of the route oscillation problem, McPherson et al. [16] suggest two possible approaches for solving the problem. The first approach is to constrain the use of MEDs according to some guidelines in order to avoid oscillations. For instance, MEDs could be disallowed or their use could be modified (e.g., using the Cisco "always-compare-med" command that compares the MED values of all routes even if they go through different neighboring AS-es). It is also possible to adjust link metrics in a way that eliminates some of these oscillations. The second approach is to modify the core protocol itself such that route oscillations are eliminated in the modified protocol. One such remedy was proposed by Walton et al. in [23] — we show that their solution fails to prevent persistent oscillations in certain cases (see Section 8 for a full discussion).

In this paper, we follow the second approach and suggest an alternate modification to BGP that is provably correct. The key idea in our modification is that each router, in addition to its best path, also advertises some additional paths to all its I-BGP peers. These extra paths are useful to avoid a situation where paths are periodically "hidden from view" due to MED comparisons. Obviously, such a modification

[1] Note that the RFC defining BGP [18] does not explicitly refer to the internal and the external versions of BGP as E-BGP and I-BGP, respectively. However, this terminology is in common usage when referring to the two uses of BGP.

SIGCOMM'02, August 19-23, 2002, Pittsburgh, Pennsylvania, USA.

raises some scalability issues since each router must advertise multiple paths instead of a single best path for each destination prefix. However, there are two distinct advantages. First, the modification admits a fairly simple analysis which proves that both persistent and transient oscillations are eliminated without restricting router configurations or the use of MEDs in any way. Second, we show that given the same collection of E-BGP routes injected into an AS, the modified protocol will converge to the same stable routing solution, even after the failure and restart of certain routers and independent of message ordering and delays. This may prove a substantial advantage in terms of debugging routing anomalies in an AS. It is particularly true for configurations that ordinarily may non-deterministically converge to one of multiple stable routing solutions.

We summarize the main contributions of the paper as follows. We provide a formal study of the route oscillation problem in I-BGP when route reflection is used. In particular, we describe a modification to I-BGP and give a proof that the modified protocol is guaranteed to solve the persistent *and* transient route oscillation problems. In contrast to E-BGP, I-BGP runs over a set of routers under the control of the same administrative entity. Therefore, our modifications may potentially be more easily deployed in operational networks since it does not require cooperation between different AS-es.

The rest of the paper is organized as follows. In Section 2, we give an overview of some aspects of the I-BGP protocol. Section 3 contains examples illustrating persistent route oscillations as well as transient route oscillations. In Section 4, we formally model I-BGP using a graph-theoretic formulation. In Section 5, we show that even checking whether a particular AS can converge to a stable routing solution is an NP-Complete problem. In Sections 6 and 7, we present our modifications to I-BGP and prove the convergence of the modified protocol. Section 8 describes certain failure scenarios for the solution proposed by Walton et al. We then discuss related work in Section 9 and conclude in Section 10.

2. Overview of I-BGP

We begin with a brief description of the I-BGP protocol and the route reflection mechanism. We then provide an overview of the route selection process used by I-BGP.

Description of I-BGP and Route Reflection. Internal BGP [18] is used to distribute externally-learned routes within an Autonomous System. A crucial difference between I-BGP and E-BGP is that they use separate mechanisms to prevent looping in the routing announcements. In E-BGP, routers look at the AS-PATH attribute that contains a list of AS-es that the routing announcement has passed through. If an AS occurs more than once in the list, a loop has occurred in the routing announcement. Since all participants in I-BGP belong to the same AS, this technique of using the AS-PATH attribute to detect loops cannot be used. Instead, for I-BGP, a full mesh of connections is maintained among all I-BGP speakers in the same AS, and no I-BGP speaker forwards routes that it receives from an I-BGP peer.

Maintaining a full mesh of connections has scaling problems since it requires the number of I-BGP peering sessions to be quadratic in the number of I-BGP routers. We now give an overview of a solution to alleviate this problem called route reflection [1]. The main concept in route reflection is to use a two-level hierarchy. The set of I-BGP speakers in an AS is partitioned into a collection of disjoint sets called *clusters*. Each cluster consists of one or more special routers called *route reflectors*. All other routers in a cluster are *clients* for the route reflectors in the cluster.[2] The route reflectors in an AS maintain a full mesh of I-BGP connections among themselves. These reflectors form the top level in the hierarchy. Furthermore, the clients in a cluster maintain I-BGP sessions with each route reflector in the cluster. These clients form the bottom level in the hierarchy. There are no I-BGP sessions between clients in one cluster and routers in a different cluster. In practice, this configuration can significantly reduce the number of I-BGP sessions. Of course, each cluster itself can be partitioned into subclusters and so on creating an arbitrarily deep hierarchy. We concentrate on the case of a two-level hierarchy.

When route reflection is used, I-BGP behavior is modified slightly. Client routers continue to behave as before. The behavior of a route reflector is modified as follows (see also [1]). On receiving a new route from a (Internal or External) BGP peer, the route reflector selects the best route according to the BGP route selection procedure (described later in this section). Depending on the peer it received the best route from, the route reflector does the following: (a) if the peer is an E-BGP peer, the route is forwarded to all client peers and all non-client peers in other clusters, (b) if the peer is a non-client peer in a different cluster, the route is forwarded to all client peers, or (c) if the peer is a client peer, the route is forwarded to all non-client peers in other clusters and to all client peers except the originator.

Route Selection Procedure. When an I-BGP speaker receives a route update from a BGP peer, it uses the following procedure to select the best route.

1. The route with the highest "degree of preference" is chosen.

2. If there are multiple such routes, the route with the minimum length of the AS-PATH attribute is chosen.[3]

3. If there are multiple such routes, for each neighboring AS, consider all the routes with the minimum value of the MULTI-EXIT-DISCRIMINATOR (MED) attribute going through the AS. (Note that if there are multiple neighboring AS-es, there could be routes with minimal MED values corresponding to each AS.) If there is exactly one such route, this route is chosen.

4. If there are multiple such routes, and there are one or more routes received via E-BGP (E-BGP routes), the E-BGP route with the minimum cost IGP path to the NEXT-HOP router is chosen. Otherwise, go to rule 6.

5. If there are no E-BGP routes and multiple I-BGP routes, the route with the minimum cost IGP path to the NEXT-HOP router is chosen. [4]

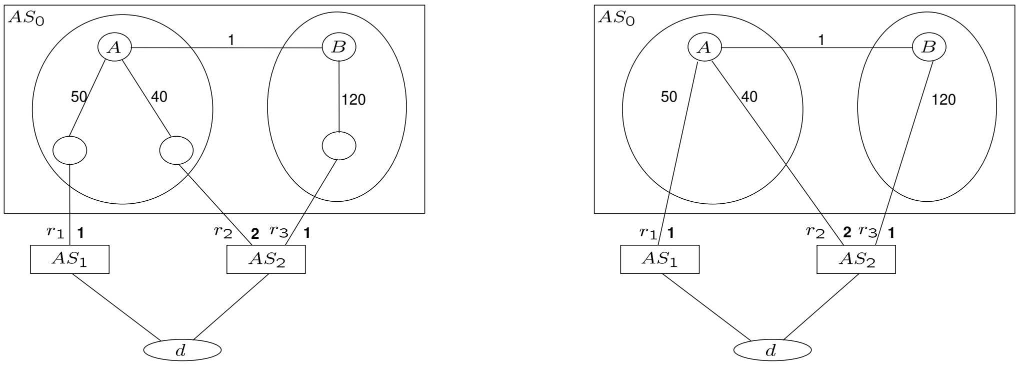
6. If there are multiple such routes, the route received from the neighbor with the minimum BGP identifier is chosen.

Note that the specification in [18] says that the degree of preference for a route is calculated by a BGP speaker on receiving the route. If

[2]Note that a cluster may consist only of route reflectors and no clients. In the extreme case, a cluster may have only one member, a route reflector — this is fully-meshed I-BGP.

[3]The BGP specification [18] does not mention use of the AS-PATH length to break ties though both [10] and [11] do. We assume that the AS-PATH length is used.

[4]The route selection process as described in [11] and [18] applies rules 4 and 5 differently. Here, the route with the minimum cost IGP path to the NEXT-HOP is chosen, irrespective of whether it is an E-BGP route or an I-BGP route (rule 4). If there are multiple minimum IGP cost routes, E-BGP routes are given preference over I-BGP routes (rule 5). However, implementations by Cisco and Juniper as well as the route selection process as described in [10] apply rules 4 and 5 as described here. In other words, external routes are preferred over internal routes, irrespective of the cost of the path to the NEXT-HOP.



(a) I-BGP with Route Reflection

(b) Fully Meshed I-BGP

Figure 1: Persistent Route Oscillations in I-BGP

the route is received via I-BGP, the recipient *may* use the value of the LOCAL-PREF attribute as the degree of preference. However, if the LOCAL-PREF attribute is *not* used as the degree of preference, then it is possible to create routing oscillations very easily by assigning a route's degree of preference in a manner similar to that in [7]. Hence, for the purpose of this paper, we assume that the value of the LOCAL-PREF attribute is used as the "degree of preference" in I-BGP.

3. Route Oscillations

In this section we provide examples of persistent route oscillations as well as transient route oscillations.

We begin by looking at the example shown in Figure 1(a) where persistent route oscillations occur. This is essentially the example presented in [16]. The configuration consists of two clusters, one with route reflector A (with two clients) and another with route reflector B (with one client). In the figure, the MED values for routes over the inter-AS links are shown in bold text, and the link costs are shown in normal text. The route oscillations are generated as follows:

- Route reflector A selects route r_2 (lower IGP metric) and route reflector B selects route r_3.
- A receives r_3 and selects r_1. This is because r_3 is better than r_2 (lower MED) and r_1 is better than r_3 (lower IGP metric).
- B receives r_1 and selects r_1 over r_3 (lower IGP metric) and withdraws r_3.
- A selects r_2 over r_1 (lower IGP metric) and withdraws r_1.
- B selects r_3 over r_2 (lower MED) and the cycle begins again.

The core problem here is the following. Since MED comparisons only take place between routes that pass through the same neighboring AS, the presence or absence of a route may change the relative ranking of a different route and thereby cause persistent oscillations. Walton et al. [23] propose a modification to I-BGP route reflection which thwarts the oscillation problem in this example. Their proposal is that each reflector advertises not only its best path, but a vector consisting of its best path through each neighboring AS. In this way, I-BGP peers can modify their own choice of best path according to the extra information.

It should be noted that McPherson et al. [16] indicate that it is a combination of route reflection and the way in which MEDs are compared that can cause persistent route oscillations to occur. They suggest that one solution is to only permit fully-meshed I-BGP. However, as mentioned earlier, fully-meshed I-BGP has scaling problems, and both solutions to the scaling problem (route reflection and confederations) exhibit routing oscillations of this nature.

Finally, we point out that if the order in which the selection rules are applied is changed to the ordering in [18] or [11], it is possible to create persistent oscillations in fully-meshed I-BGP in a manner similar to Figure 1(a). Namely, the configuration of Figure 1(b) will diverge just as in our previous example under these modified rules. It converges under our present route selection procedure since B always prefers its E-BGP route to either of the (shorter) routes through A.

We now come to transient route oscillations. Consider the network in Figure 2. The dotted lines represent additional (IGP) links between nodes in AS_0 over which no I-BGP sessions run. All routes have the same LOCAL-PREF, AS-PATH length and MED value 0 (shown in bold next to the inter-AS links). Route oscillations can be created in this network as follows:

1. Reflector RR_1 chooses r_1 and reflector RR_2 chooses r_2.
2. The two reflectors advertise their best paths to each other. Now RR_1 chooses r_2 (lower IGP cost to NEXT-HOP) and RR_2 chooses r_1 (lower IGP cost to NEXT-HOP).
3. Reflector RR_1 withdraws r_1 as best path and reflector RR_2 withdraws r_2 as the best path.
4. Once again, reflector RR_1 chooses r_1 and reflector RR_2 chooses r_2, and the cycle repeats.

Note that in this example, two stable routing configurations exist. In the first configuration, both RR_1 and RR_2 choose r_1, and in the second, both RR_1 and RR_2 choose r_2. It is easy to check that both these configurations are stable. It is also possible to reach either of these configurations, if the reflectors RR_1 and RR_2 send and receive messages in a certain order. For example, the first stable configuration can be reached if the following steps occur:

1. Reflector RR_1 chooses r_1 and advertises it to reflector RR_2.
2. Reflector RR_2 receives r_1 and r_2 and chooses r_1. Since it received r_1 from reflector RR_1, it does not advertise r_1 back to RR_1. Thus, the system achieves a stable configuration.

We note that the crucial difference in the two executions (one unstable, and the other stable) is the order in which the route reflectors send and receive messages. In other words, this is an example of a transient route oscillation where the ordering of messages may cause route oscillations. Later in this section, we give another example which shows that delays in messaging can cause transient oscillations. However, we note that the solution of Walton et al. (which does not purport to address transient oscillations) does not avoid such oscillations. Indeed, for this example, there is only one neighboring AS, so their adaptation behaves exactly the same as for classical I-BGP. Thus the routing configuration achieved by standard I-BGP and by the modified version of Walton et al. can be either of the two stable solutions, or it may continue to oscillate, depending on non-deterministic timing considerations. However, we show that our modified protocol always converges to the same routing configuration, irrespective of timing issues.

We now present an example of transient route oscillation in a system configured such that (in contrast to the previous example) the I-BGP peering sessions correspond to IGP links (see Figure 3). In this example, transient route oscillations are caused by message delays. Routers A, B, and C are I-BGP speakers in Autonomous System AS_0 and are connected to (routers in) AS_1, AS_2, and AS_3 as shown. The MED value for each inter-AS link is shown next to the link. The link cost for each of these links is 0. Each inter-AS link represents an external route to destination d — from left to right, let these routes be r_1 through r_6, respectively. We assume that all these routes have the same LOCAL-PREF value and we note that these routes all have the same AS-PATH length. We also assume that the routes represented by dotted lines have lower BGP identifiers than the solid ones. The links connecting the I-BGP speakers are not shown. One may check that this example again has two stable solutions.

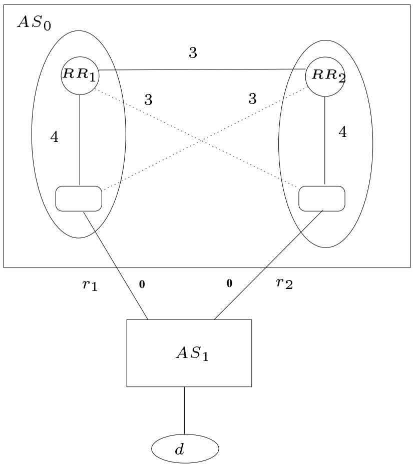
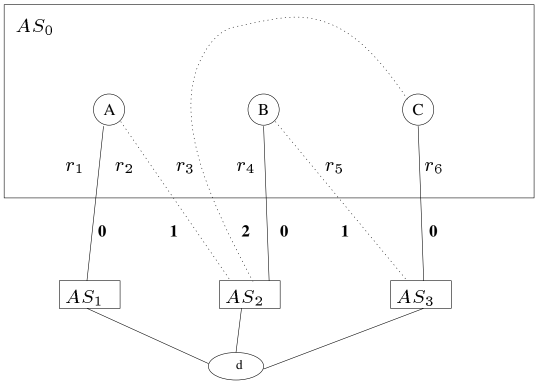


Figure 2: An Example of Transient Route Oscillations



Three routers each with two exit paths, the preferred (by speaker number) is indicated as a dotted line. MED values are shown in bold next to the routes.

Figure 3: Another Example of Transient Route Oscillations

All through this example, we assume that whenever a router selects a new route, it withdraws its previously advertised route, if any. The route oscillation behavior is now produced by the sequence of updates presented in Table 1.

We note that this example can be simplified somewhat (by deleting router A and autonomous system AS_1) and transient oscillations can still occur but it will rely on the timing of when the routes through AS_2 and AS_3 are injected into AS_0.

4. Modeling I-BGP with Route Reflection

In this section, we present a graph-theoretic model to formalize the behavior of I-BGP speakers (i.e., routers) within some given autonomous system, AS_0, that uses route reflection. For the remainder of this paper, we concentrate only on routes for a single external destination (prefix), namely, d. Of course, since fully-meshed I-BGP can be thought of as a special case of I-BGP with route reflection where each router is a route reflector without any clients, this is also a model of fully-meshed I-BGP.

It should be noted that the Safe Path Vector Protocol (SPVP) models (see [6] and [8]) can not be used to model I-BGP when MED values are used. This is because the SPVP models rely on each router having a fixed order of preference for routes but MED values can cause the relative ordering of routes to vary depending on what other routes are being considered.

Physical and Logical Graphs. We start by defining a connected graph $G_P = (V, E_P)$ called the *physical graph* that captures the physical connectivity of the autonomous system. Each node in V represents a router (i.e., an I-BGP speaker) in AS_0. We use the notation $\boldsymbol{\rho}_v$ to denote the router represented by the node v. There is an edge $uv \in E_P$ if and only if $\boldsymbol{\rho}_u$ and $\boldsymbol{\rho}_v$ have a physical link connecting them in AS_0. Each edge $uv \in E_P$ has a positive integer cost, $cost(uv)$, representing the IGP metric for uv. We define $cost(p)$ of a path p in G_P to be the sum of the costs of the edges in p. The *shortest path*, $SP(u, v)$, between two nodes in V, is chosen (deterministically) from one of the least cost paths in G_P between u and v. Finally, let $AS_1, AS_2, \ldots, AS_m$ be the autonomous systems which have routers that maintain E-BGP peering sessions with routers in AS_0.

We define a second graph $G_I = (V, E_I)$ called the *logical graph* that represents I-BGP peering relationships. Here, there is an edge $uv \in E_I$ if the routers $\boldsymbol{\rho}_u$ and $\boldsymbol{\rho}_v$ are I-BGP peers. To model route reflection, we define a partition of the nodes in V into sets $C_1, C_2, \ldots, C_k$. Each C_i represents a router cluster in AS_0. Let $R_i \subseteq C_i$ be the set of nodes representing the route reflectors in the cluster C_i. Let N_i be the set of nodes in C_i but not in R_i. A node in R_i is called a *reflector node* and a node in N_i is called a *client node*. Let $R = \bigcup_{i=1}^{k} R_i$ and $N = \bigcup_{i=1}^{k} N_i$ (see Figure 4). A client node in cluster C_i is called a *client of* all the nodes in R_i. Note that the edges in E_I satisfy some constraints imposed by the conditions described in Section 2. Namely,

1. there is an edge $uv \in E_I$ for every pair of nodes u, v in R,

2. there is an edge from every node in N_i to every node in R_i, $1 \leq i \leq k$,

router updated	routes learned via E-BGP	A	B	C	routes removed rule 3	rule 4	rule 6	best route
C	r_3, r_6						r_6	r_3
B	r_4, r_5						r_4	r_5
A	r_1, r_2						r_1	r_2
C	r_3, r_6	r_2	r_5		r_3, r_5	r_2		r_6
B	r_4, r_5	r_2		r_6	r_2, r_5	r_6		r_4
A	r_1, r_2		r_4	r_6	r_2	r_4, r_6		r_1
C	r_3, r_6	r_1	r_5 *		r_5	r_1	r_6	r_3
B	r_4, r_5	r_1		r_3	r_3	r_1	r_4	r_5
A	r_1, r_2		r_5	r_3	r_3	r_5	r_1	r_2
				⋮				

* Timing delay results in stale information.

Table 1: Transient Route Oscillation.

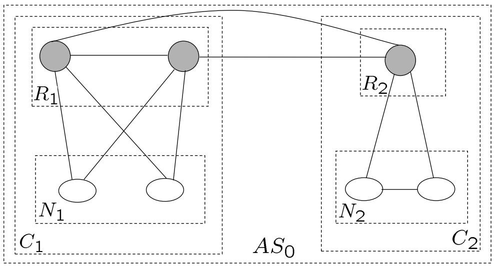


Figure 4: Route Reflectors and Clients: The nodes shaded gray are route reflectors and the other ones are clients

3. there are no edges from any node in N_i to any node in C_j where $i \neq j$ and

4. there may be edges between arbitrary pairs of nodes u and v if $u, v \in N_i$ for some i.

In practice, it is often the case that each router cluster has exactly one route reflector and client nodes in the same cluster do not maintain I-BGP adjacencies. However, we allow multiple reflectors per cluster as well as I-BGP peering sessions among clients in the same cluster to make our model more general. Observe that the specification [1] does not explicitly disallow such configurations.

Routes and Exit Paths. We now introduce the concept of an "exit path". An *exit path* p represents a BGP route $\mathbf{b_p}$ to destination d in an E-BGP message injected into AS_0. An exit path p has the following attributes:

- $\mathsf{localPref}(p)$ is a non-negative integer that represents the local preference assigned to $\mathbf{b_p}$ when it is injected into I-BGP running on AS_0.
- $\mathsf{AS\text{-}Path}(p)$ is a list of autonomous systems $AS_0, AS_{i_1}, \ldots, AS_{i_s}$ and represents the AS-PATH attribute of the BGP route $\mathbf{b_p}$.
- $\mathsf{AS\text{-}path\text{-}length}(p)$ is a positive integer representing the length of the AS-PATH attribute of $\mathbf{b_p}$.
- $\mathsf{nextAS}(p)$ is the autonomous system from which AS_0 received the BGP route $\mathbf{b_p}$ via E-BGP. Thus if $\mathsf{AS\text{-}Path}(p) = AS_0, AS_{i_1}, \ldots, AS_{i_s}$ then $\mathsf{nextAS}(p) = AS_{i_1}$.

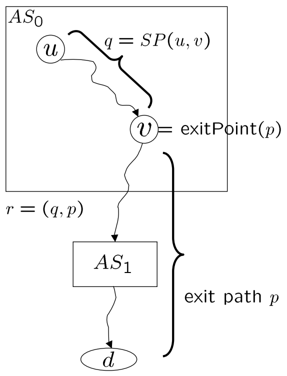


Figure 5: Exit Paths and Routes

- $\mathsf{MED}(p)$ is a non-negative integer that represents the Multi-Exit-Discriminator (MED) assigned to $\mathbf{b_p}$.
- $\mathsf{nextHop}(p)$ is an IP-address representing the usual NEXT-HOP attribute associated with an E-BGP route.[5]
- $\mathsf{exitPoint}(p)$ is the node in V that represents the router in AS_0 which learned of $\mathbf{b_p}$ via E-BGP. We say that p is an *exit path from* $v = \mathsf{exitPoint}(p)$. Note that $\mathsf{exitPoint}(p)$ is uniquely defined since there is a one-one correspondence between the NEXT-HOP attribute for $\mathbf{b_p}$ and $\mathsf{exitPoint}(p)$.[6]
- $\mathsf{exitCost}(p)$ is some non-negative integer value representing the cost associated with the link from $\mathsf{exitPoint}(p)$ to $\mathsf{nextHop}(p)$. This metric is usually 0 in practice, but can be set to a value that is > 0.

A *route* r *from a node* $u \in V$ is an ordered pair (q, p) where p is an exit path and q is a path in G_P which joins u to the node

[5] In practice, the NEXT-HOP is typically a BGP speaker in a neighboring autonomous system. This implies that the IGP running in AS_0 must know how to get to the NEXT-HOP address, even though it is outside the AS.

[6] In real networks, the NEXT-HOP refers to the IP address of the remote end of a numbered link (in other words, a port on the neighboring router). Hence, we have the one-one correspondence. However, for simplicity, we do not explicitly model ports since it does not affect the correctness of our proofs.

$v = \mathsf{exitPoint}(p)$ (see Figure 5). In addition, the path q must coincide with the selected shortest path $SP(u, v)$. We sometimes refer to q and p respectively as the *internal* and *external* parts of r; we also let $\mathsf{exit}(r)$ denote the external part p of the route. Such a route inherits all the attributes from its external part, e.g., we may refer to $\mathsf{MED}(r)$ but this refers simply to $\mathsf{MED}(p)$. In addition, we let $\mathsf{metric}(r)$ denote the length of the (shortest) path q plus $\mathsf{exitCost}(p)$. If $u = v$, then r corresponds to an E-BGP route (as opposed to an I-BGP route) and is essentially equivalent to the exit path p. Note that in such cases, $\mathsf{metric}(r)$ is simply $\mathsf{exitCost}(p)$, since the internal part is the trivial single node path, which has cost 0. The other attribute we associate with a route r is the attribute $\mathsf{learnedFrom}(r)$ which denotes the BGP identifier of the (BGP) peer from which u received the route r. In case of E-BGP, $\mathsf{learnedFrom}(r)$ is the same as the BGP identifier for $\mathsf{nextHop}(r)$. For I-BGP, $\mathsf{learnedFrom}(r)$ denotes the BGP identifier for the I-BGP neighbor that advertised r to u.

Note that a route is uniquely determined by an exit path p and a node u. We thus let $\mathsf{route}(p, u)$ denote the route $(SP(u, v), p)$ where $v = \mathsf{exitPoint}(p)$. For a set of exit paths P, we define:

$$\mathsf{route}(P, u) = \{\mathsf{route}(p, u) | p \in P\}$$

and for a set of routes S, we define:

$$\mathsf{exit}(S) = \{\mathsf{exit}(s) | s \in S\}.$$

Operational Description of I-BGP. We now provide an operational description of an I-BGP router. We consider a discrete model of time $t = 1, 2, \ldots$. Let S be the set of external routes (learned via E-BGP or I-BGP) known to a router $v \in V$. We define the best route according to v as $\mathsf{best}_v(S) = \mathrm{Choose\_best}(v, S)$ where the procedure $\mathrm{Choose\_best}(v, S)$ is as shown in Figure 6.

A *configuration at time* t, $\mathsf{config}(t)$, consists of the following for each $v \in V$:

1. $\mathsf{MyExits}(v)$, a set of exit paths from v (i.e., $\mathsf{exitPoint}(p) = v$ for $p \in \mathsf{MyExits}(v)$) that does not vary with time (we explain why later).

2. $\mathsf{PossibleExits}(v, t)$, a set of exit paths, and

3. $\mathsf{BestRoute}(v, t)$, a route from v.

These objects satisfy the following conditions:

1. $\mathsf{PossibleExits}(v, t) \supseteq \mathsf{MyExits}(v)$, and

2. $\mathsf{BestRoute}(v, t) = \mathsf{best}_v(\mathsf{route}(\mathsf{PossibleExits}(v, t), v))$.

Intuitively, $\mathsf{MyExits}(v)$ represents the E-BGP routes that the router $\boldsymbol{\rho}_v$ currently knows about. The set $\mathsf{PossibleExits}(v, t)$ represents the exit paths (learned by router $\boldsymbol{\rho}_v$ either via E-BGP or via I-BGP) that router $\boldsymbol{\rho}_v$ could choose from at time t. $\mathsf{BestRoute}(v, t)$ corresponds to the best route chosen by router $\boldsymbol{\rho}_v$ at time t. Depending on certain conditions (described below), $\boldsymbol{\rho}_v$ advertises the exit path for its best route to some of its I-BGP peers.

The configuration $\mathsf{config}(t)$ is *valid at time* t if for each $v \in V$ and $p \in \mathsf{PossibleExits}(v, t)$, then $p \in \mathsf{MyExits}(\mathsf{exitPoint}(p))$. That is, in a valid configuration, all exit paths that are in the system are ones that are currently known by their exit points (i.e., they have not been subsequently withdrawn after they were injected into AS_0).

Modeling Communication. We now model how routers communicate. For a set of exit paths P and distinct nodes $u, v \in V$, we define the subset $\mathsf{Transfer}_{v \to u}(P) \subseteq P$ such that p is in $\mathsf{Transfer}_{v \to u}(P)$ if and only if $p \in P$, vu is an edge in E_I and

1. $\mathsf{exitPoint}(p) = v$ or,

2. $v \in R_i$, $u \in R_j$, for some $i \neq j$, and $\mathsf{exitPoint}(p) = w$ for some node $w \in N_i$ or,

3. $v \in R_i$ and $u \in N_i$ for some i and $\mathsf{exitPoint}(p) \neq u$.

The subset $\mathsf{Transfer}_{v \to u}(P)$ models communication between routers $\boldsymbol{\rho}_v$ and $\boldsymbol{\rho}_u$. Suppose $p \in P$ is the path associated with BGP route $\mathbf{b_p}$. Then $\mathsf{Transfer}_{v \to u}(P)$ models the fact that $\boldsymbol{\rho}_v$ announces $\mathbf{b_p}$ to I-BGP peer $\boldsymbol{\rho}_u$ if one of three conditions hold. The first condition says that $\boldsymbol{\rho}_v$ has learned $\mathbf{b_p}$ from an E-BGP neighbor. The second condition says that $\boldsymbol{\rho}_u$ and $\boldsymbol{\rho}_v$ are route reflectors in different clusters and $\mathbf{b_p}$ is an exit path from a client of $\boldsymbol{\rho}_v$. The third condition says that $\boldsymbol{\rho}_u$ is a client of $\boldsymbol{\rho}_v$ and $\mathbf{b_p}$ is not an exit path from $\boldsymbol{\rho}_u$ (this prevents loops in routing announcements). Note that we do not model neighbor specific incoming and outgoing filters for BGP routes here since such filters are only applied for E-BGP peers, and not for I-BGP peers.

A *fair activation sequence* σ of node set V is a sequence $\sigma_1, \sigma_2, \ldots,$ of non-empty subsets of V called *activation sets*, such that every node $u \in V$ occurs in infinitely many σ_i's. Intuitively, an activation sequence represents an ordering of when the individual routers transfer messages and update their best routes to d. Since each router appears in a fairsequence infinitely many times, it implies that no router crashes. A failed router would stop executing at some finite

```
proc Choose_best(v, S) {
   maxPref := max_{r∈S} localPref(r)
   S := {r : r ∈ S and localPref(r) = maxPref}
   if (|S| = 1) return(b ∈ S)

   minASPL := min_{r∈S} AS-path-length(r)
   S := {r : r ∈ S and AS-path-length(r) = minASPL}
   if (|S| = 1) return(b ∈ S)

   for(j = 1, 2, ... m){
        S_j := {r : r ∈ S and nextAS(r) = j}
        minMed_j := min_{r∈S_j} MED(r)
        S_j := {r : r ∈ S_j and MED(r) = minMed_j}
   }
   S := ⋃_j S_j
   if (|S| = 1) return(b ∈ S)

   if( ∃ r ∈ S : exitPoint(r) = v)
        then S := {r : r ∈ S and exitPoint(r) = v}
   if (|S| = 1) return(b ∈ S)

   minMetric := min_{r∈S} metric(r)
   S := {r : r ∈ S and metric(r) = minMetric}
   if (|S| = 1) return(b ∈ S)

   minId := min_{r∈S} learnedFrom(r)
   S := {r : r ∈ S and learnedFrom(r) = minId}
   return(b ∈ S)
}
```

Figure 6: Procedure $\mathrm{Choose\_best}$ **for defining** $\mathsf{best}_v(S)$

time (when it fails) and therefore occur in the activation sequence only finitely many times. Suppose $\mathsf{config}(t_0)$ is a configuration at time t_0. Then for any $t > t_0$, if $u \notin \sigma_t$, then $\mathsf{PossibleExits}(u,t) = \mathsf{PossibleExits}(u,t-1)$, and $\mathsf{BestRoute}(u,t) = \mathsf{BestRoute}(u,t-1)$. However if $u \in \sigma_t$, then define

$$\begin{aligned}\mathsf{PossibleExits}(u,t) &= \bigcup_{v \in V} \mathsf{Transfer}_{v \to u}(\mathsf{exit}(\mathsf{BestRoute}(u,t))) \\ &\quad \bigcup \mathsf{MyExits}(u) \\ \mathsf{BestRoute}(u,t) &= \mathsf{best}_u(\mathsf{route}(\mathsf{PossibleExits}(u,t),u))\end{aligned}$$

In other words, whenever a router takes a step, it receives advertisements from each of its neighbors about their best routes. It then updates its own best route based on the new information. We note here that we do not explicitly model message delays in transit. However, this does not affect the correctness (proofs) of our algorithms since all the properties that we prove are valid *eventually*.

Convergence. We wish to be able to determine if a system consisting of a physical graph G_P, a logical graph G_I, and a starting valid configuration $\mathsf{config}(t_0)$ can eventually converge to a set of stable routes. Thus, we wish to determine for such a system, if there is a fair activation sequence σ such that starting at $\mathsf{config}(t_0)$, there is a time t_s after which the set $\mathsf{PossibleExits}(u,t)$ never changes for all $u \in V$. Note that this problem attempts to characterize the set of configurations for which there are no stable solutions, i.e., those that exhibit persistent oscillations.

5. The Complexity of the STABLE I-BGP WITH ROUTE REFLECTION Problem

We define an instance SR of the STABLE I-BGP WITH ROUTE REFLECTION problem to be a tuple

$$SR = (G_P, G_I, \mathsf{config}(0))$$

where G_P, G_I, and $\mathsf{config}(0)$ were defined in the previous section. Next, we assume that $\mathsf{BestRoute}(u,0) = \emptyset$, and $\mathsf{PossibleExits}(u,0) = \mathsf{MyExits}(u)$ for all vertices u. We then ask the question whether there is some activation sequence σ and a time t_s where t_s is bounded by some polynomial in $|SR|$ such that the sets $\mathsf{PossibleExits}(v,t) = \mathsf{PossibleExits}(v,t_s)$ for all $t \geq t_s$ and for all $v \in V$. If so, we say that SR *stabilizes* at time t_s and we say that the routes $\mathsf{BestRoute}(v,t_s)$ at each vertex v form a *stable solution*.

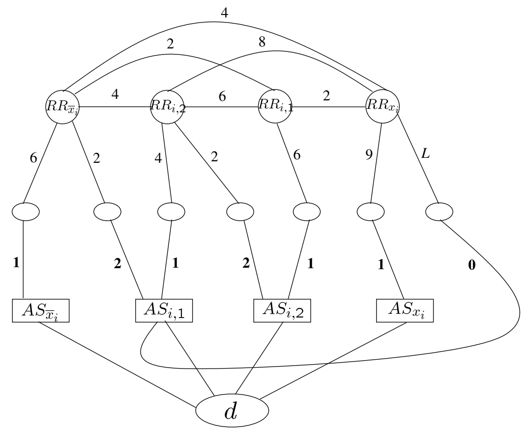
THEOREM 5.1. *The problem* STABLE I-BGP WITH ROUTE REFLECTION *is NP-complete even if* $E_I = E_P$.

Proof Sketch: It should be noted that only the essential construction is described here due to space constraints — the details are left for the full paper.

The problem is clearly in NP so we need only show that it is NP-hard. To do so, we define a reduction from the NP-complete problem 3-SAT to STABLE-IBGP WITH ROUTE REFLECTION.

An instance J of 3-SAT consists of a collection of variables $X = \{x_1, x_2, \ldots, x_n\}$ and clauses $\mathcal{K} = \{K_1, K_2, \ldots, K_m\}$ where each clause K_i is a disjunction of three literals $l_1^i \vee l_2^i \vee l_3^i$. The instance J is satisfiable if and only if there is a truth assignment of the variables that satisfies each clause simultaneously.

To show that STABLE I-BGP WITH ROUTE REFLECTION is NP-hard, we define an instance $SR_J = (G_P, G_I, \mathsf{config}(0))$ of STABLE I-BGP WITH ROUTE REFLECTION whose size is polynomial in $|J|$. We then show that J is satisfiable if and only if for SR_J, there is an activation sequence σ and a time t_s, such that SR_J stabilizes at time t_s. In our instance, we assume that $\mathsf{localPref}(p)$ is the same for all exit paths p and that $\mathsf{learnedFrom}(r)$ is some uniquely defined integer for each route r. We also point out that all exit paths p in our instance have $\mathsf{AS\text{-}path\text{-}length}(p) = 3$. Finally, we remind the reader that in SR_J, the physical and logical graphs are identical, i.e., $G_P = G_I$.



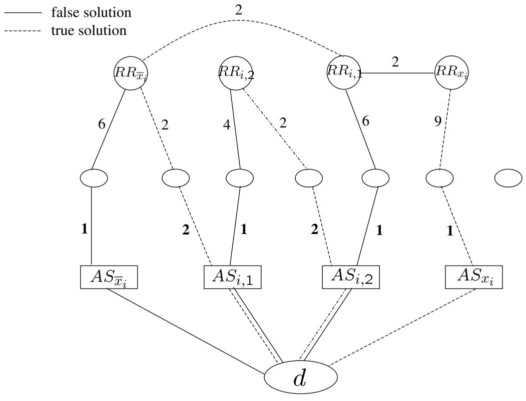
Link costs are shown in plain text next to each link.
MED values are shown in bold text beside each route.

Figure 7: The variable graph

In order to visualize the instance SR_J, we represent it by using extra nodes to represent the destination d as well as each neighboring AS. In this way, an exit path can be visualized as a sequence of two edges, one between a router node and an AS node, and the other between an AS node and the destination node. We refer to such an expanded graph as a *configuration graph*. In our illustrations of configuration graphs (as in Figure 7), router nodes are shown as large circles (for route reflectors) or small ovals (clients), and AS nodes are shown as rectangles. Note that an edge in the figures between a route reflector R (that is, a circular node) and a client router c, (that is, an oval shaped node) means also that c is in the cluster for which R is acting as a route reflector. Furthermore, link costs are shown next to each link, and MED values are shown in bold next to each exit path. Note that the link cost L denotes some large value, such as 1000. The subgraph of a configuration graph that consists of all the edges associated with all the routes in a stable solution will be called a *stable routing graph*. In the construction described, it can be seen that a stable routing graph uniquely defines a stable route at each node.

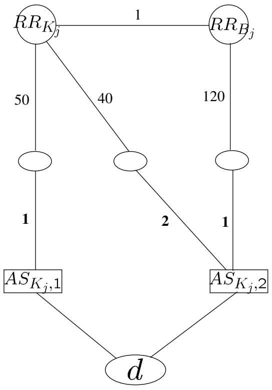
We now define two types of gadgets, one for variables in J, and the other for clauses in J. The first type of gadget (called a variable graph) is shown in Figure 7 for the variable x_i. This gadget has two stable "solutions" as shown in Figure 8. Note that the solution shown with dotted lines has a path through node AS_{x_i} but none through node $AS_{\overline{x}_i}$ — we call this the *true solution graph*, corresponding to a true setting of the variable x_i. The opposite holds for the solution shown with solid lines — we call this the *false solution graph*, corresponding to a false setting of the variable x_i. The two nodes labeled $AS_{i,1}$ and $AS_{i,2}$ are just auxiliary nodes that allow these types of gadgets to have two stable solutions.

The second type of gadget (called a clause graph) is shown in Figure 9 for the clause $K_j = l_1^j \vee l_2^j \vee l_3^j$. Without loss of generality, we can assume that no variable and its negation appears in the same clause since such a clause would always be trivially satisfied. We also point out that a clause graph does not have a stable solution when considered in isolation. However, consider the truth settings for all the variables x_i that occur in K_j such that K_j is satisfied. We next describe how



Link costs are shown in plain text next to each link.
MED values are shown in bold text beside each route.

Figure 8: Two stable routing graphs for a variable graph.



Link costs are shown in plain text next to each link.
MED values are shown in bold text beside each route.

Figure 9: The clause graph.

the clause gadget is connected to the three relevant variable gadgets and then show that the variable graphs, in conjunction with the clause graph for K_j, has a stable solution corresponding to a truth assignment in which K_j is satisfied.

Suppose x_i occurs in clause K_j. Then, we connect the node RR_{K_j} in the clause graph for K_j to the node RR_{x_i} in the variable graph for x_i, using an edge of cost 1. If, on the other hand, $\overline{x}_i$ is in K_j, then we use an edge of cost 1 to connect the node RR_{K_j} to the node $RR_{\overline{x}_i}$ in the variable graph for x_i.

Also, there are edges from node RR_{K_j} to each of the nodes

- $AS_{K_a,1}$ and $AS_{K_a,2}$, for $a \neq j$,
- $AS_{i,1}$ and $AS_{i,2}$, $1 \leq i \leq n$,
- $AS_{\overline{x}_i}$ and AS_{x_i} where $x_i \neq l_a^j$ and $\overline{x}_i \neq l_a^j$ for $a = 1, 2, 3$,
- AS_{x_i} if $l_a^j = \overline{x}_i$ for some a and
- $AS_{\overline{x}_i}$ if $l_a^j = x_i$ for some a.

Each of these edges has MED value of 0 and cost L where L is just some large value (for example, take $L = 1000$). The motivation for such MED and cost values is that having a low MED will cause other paths to the same AS to be ignored in the route selection process but the large cost means that it will not be chosen over paths to other ASes. In addition, each node $RR_{\overline{x}_i}$, $RR_{i,2}$, $RR_{i,1}$ and RR_{x_i} has edges to each of the nodes

- $AS_{\overline{x}_a}$, $AS_{a,1}$, $AS_{a,2}$ and AS_{x_a} for $a \neq i$ and
- $AS_{K_j,1}$ and $AS_{K_j,2}$ for $1 \leq j \leq m$.

All of these edges have MED 0 and cost L. Finally, we must define costs on the edges between any pair of route reflector nodes that we have not already explicitly stated. We do this arbitrarily such that their costs are no more than the shortest paths between their endpoints, thereby ensuring that the triangle inequality is satisfied.[7] This can easily be done by setting these costs one at a time to be equal to the shortest path in the graph consisting of the edges with costs so far defined. Clearly this instance of STABLE I-BGP WITH ROUTE REFLECTION can be constructed in time polynomial in the size of J.

Suppose J has a satisfying assignment A. We now describe a stable routing graph G_A associated with A. For each variable x_i, if x_i is true in A then let the true solution graph for x_i be a subgraph of G_A. Otherwise let the false solution graph be a subgraph of G_A. For each literal l_i^j, define p_i^j to be the exit path through $AS_{l_i^j}$. For each clause $K_j = l_1^j \vee l_2^j \vee l_3^j$, choose p_i^j as an edge in the routing graph such that l_i^j is true according to A and also satisfies the following condition. If l_a^j also evaluates to true in A, then $\mathsf{learnedFrom}(p_i^j) < \mathsf{learnedFrom}(p_a^j)$. It is a straightforward exercise to confirm that G_A represents a stable solution.

Similarly, it can be verified if G admits a stable solution, then the path in G from RR_{K_j} must be through $RR_{l_i^j}$ and $AS_{l_i^j}$ for some i. Verifying that no stable solution can contain both AS_{x_i} and $AS_{\overline{x}_i}$ is also straightforward and so a stable routing graph defines a satisfying assignment to the 3-SAT instance. ■

6. Modeling the New I-BGP

In this section, we extend the graph-theoretic model of I-BGP from Section 4 so that the modified protocol is guaranteed to converge. The convergence proof is given in Section 7. Broadly speaking, the change amounts to halting the best-path selection procedure early, and then advertising all of the routes which are not yet eliminated. We halt just after the point when paths are removed based on their MED value (i.e., after the application of rule 3 in the best route selection process described in Section 2). Once these paths have been exported, the router continues as before to narrow down its selection to a single best route.

For any router, we denote the set of routes advertised to its peers (in the new protocol) as the set $S^\flat$. For a set of exit paths S, this is computed as $S^\flat = \mathrm{Choose\_max}^\flat(S)$. Figure 10 shows the procedure $\mathrm{Choose\_max}^\flat(S)$. Note that $\mathrm{Choose\_max}^\flat(S)$ is essentially the first part of the procedure $\mathrm{Choose\_best}(u, S)$.

Now consider a *fair activation sequence*, σ of node set V as described in Section 4. Suppose that $\mathsf{config}(0)$ is a valid configuration at time $t = 0$. Then for any $t > 0$, if $u \notin \sigma_t$, then $\mathsf{PossibleExits}(u, t) = \mathsf{PossibleExits}(u, t-1)$, $\mathsf{BestRoute}(u, t) = \mathsf{BestRoute}(u, t-1)$ and $\mathsf{GoodExits}(u, t) = \mathsf{GoodExits}(u, t-1)$. However, if $u \in \sigma_t$, then define

[7] The I-BGP sessions (co-incident with physical edges) typically run over TCP. This implies that the I-BGP sessions are routed using shortest path routes. Therefore, the costs on the physical edges (or I-BGP sessions) have to satisfy the triangle inequality.

proc $\mathrm{Choose\_max}^{\flat}(S)$ {

$\mathrm{maxPref} := \max_{p \in S} \mathsf{localPref}(p)$

$S := \{p : p \in S \text{ and } \mathsf{localPref}(p) = \mathrm{maxPref}\}$

$\mathrm{minASPL} := \min_{p \in S} \mathsf{AS\text{-}path\text{-}length}(p)$

$S := \{p : p \in S \text{ and } \mathsf{AS\text{-}path\text{-}length}(p) = \mathrm{minASPL}\}$

for$(j = 1, 2, \ldots m)$ {

$S_j := \{p : p \in S \text{ and } \mathsf{nextAS}(p) = j\}$

$\mathrm{minMed}_j := \min_{p \in S_j} \mathsf{MED}(p)$

$S_j := \{p : p \in S_j \text{ and } \mathsf{MED}(p) = \mathrm{minMed}_j\}$

}

$S := \bigcup_j S_j$

return(S)

}

Figure 10: Procedure $\mathrm{Choose\_max}^{\flat}(S)$ **for defining** $S^{\flat}$

$$\begin{aligned}
\mathsf{PossibleExits}(u,t) &= \bigcup_{v \in V} \mathsf{Transfer}_{v \to u}(\mathsf{GoodExits}(v, t-1)) \\
&\quad \bigcup \mathsf{MyExits}(u) \\
\mathsf{BestRoute}(u,t) &= \mathsf{best}_u(\mathsf{route}(\mathsf{PossibleExits}(u,t), u)) \\
\mathsf{GoodExits}(u,t) &= \mathrm{Choose_max}^{\flat}(\mathsf{PossibleExits}(u,t)).
\end{aligned}$$

Note that it would be equivalent to define

$$\mathsf{BestRoute}(u,t) = \mathsf{best}_u(\mathsf{route}(\mathsf{GoodExits}(u,t), u)).$$

Intuitively, the changes described in this section do the following. Each I-BGP router r advertises a set of good exit paths (which have passed part of the best path selection procedure) to all its I-BGP peers instead of a single best exit path. All the exit paths in this set have the highest LOCAL-PREF and the lowest AS-PATH length among all the possible exit paths known to r. Furthermore, if p is a exit path in this set and passes through neighboring AS AS_k, then p has the lowest MED among all exit paths passing through AS_k that are known to r. Obviously, there may be multiple such exit paths corresponding to each AS_k (or none, if they do not have the appropriate values of LOCAL-PREF and AS-PATH length).

7. Convergence of Modified I-BGP

In this section, we show that the algorithm proposed in the previous section converges. The proof is in two parts — we first show that each router eventually selects a route that does not change in the absence of any E-BGP updates. Next, we show that the collection of routes chosen by all the I-BGP speakers in an AS is loop-free.

We assume throughout this section that we are given a physical graph $G_P = (V, E_P)$, a logical graph $G_I = (V, E_I)$, a starting configuration $\mathsf{config}(0)$ and a fair activation sequence σ and show that the algorithm proposed in the previous section converges.

Convergence Proof. We think of $\mathsf{MyExits}(v)$ (as defined by $\mathsf{config}(0)$) as representing all the possible exit paths that router $\boldsymbol{\rho}_v$ knows of (via E-BGP) getting to destination d at time 0. We assume that after

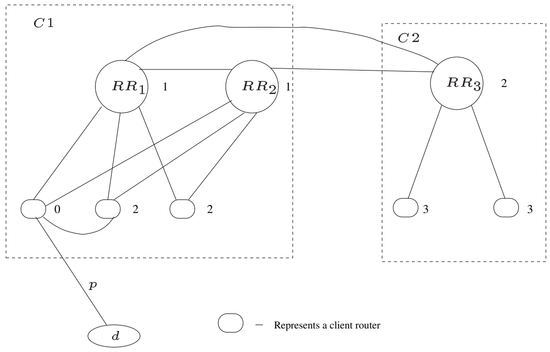


Figure 11: An example illustrating level$_p(u)$.

time 0, there are no more E-BGP updates injected into AS_0.[8] Thus, $\mathsf{MyExits}(v)$ for each node v remains fixed after time 0. Furthermore, it is possible that a path that had been injected earlier into AS_0 by some external router is no longer valid.[9] We show that all such (invalid) paths are eventually flushed out.

Let S be the set of all exit paths in $\mathsf{config}(0)$, i.e.,

$$\mathsf{S} = \bigcup_{v \in V} \mathsf{MyExits}(v).$$

For the given activation sequence σ and any time t let

$$\mathsf{Options}(t) = \bigcup_{v \in V} \mathsf{PossibleExits}(v, t).$$

We show that there is some time τ_0 such that for all times $t \geq \tau_0$, if $p \in \mathsf{Options}(t)$, then $p \in \mathsf{MyExits}(\mathsf{exitPoint}(p))$.

Consider any exit path p and suppose $v = \mathsf{exitPoint}(p)$ and $v \in C_i$. For each node $u \in V$ we define the *level* of u with respect to p, written $\mathrm{level}_p(u)$ as follows:

- $\mathrm{level}_p(u) = 0$ if $u = v$,
- $\mathrm{level}_p(u) = 1$ if $u \in R_i$ and $u \neq v$,
- $\mathrm{level}_p(u) = 2$ if $u \in N_i$ and $u \neq v$,
- $\mathrm{level}_p(u) = 2$ if $u \in R_j$ and $j \neq i$, and
- $\mathrm{level}_p(u) = 3$ if $u \in N_j$ and $j \neq i$.

Figure 11 shows how the levels are allocated for path p to destination d. The level for each node is shown in bold next to the node.

LEMMA 7.1. *Let P be a set of exit paths. If for $u, w \in V$, and $p \in P$, $\mathit{level}_p(u) \geq \mathit{level}_p(w)$, then $p \notin \mathsf{Transfer}_{u \to w}(P)$.*

Proof: This follows directly from the definition of $\mathsf{Transfer}_{u \to w}(P)$. ■

We now show that there is some time τ_0 such that for all $t > \tau_0$, $\mathsf{config}(t)$ is a valid configuration. In other words, after $t > \tau_0$, all the invalid external routes are flushed out of AS_0.

LEMMA 7.2. *For all nodes $u \in V$, and any exit path p, there is some time t_u such that for all $t \geq t_u$, if $p \notin \mathsf{MyExits}(\mathsf{exitPoint}(p))$, then $p \notin \mathsf{PossibleExits}(u, t)$.*

[8] We explain why this assumption is reasonable later.

[9] Such a situation can occur if the withdrawal messages announcing invalid paths have not reached all the I-BGP speakers by time 0.

Proof: Consider any node u. Let $\text{level}_p(u) = h$ and $v = \mathsf{exitPoint}(p)$. Throughout the proof we assume that $p \notin \mathsf{MyExits}(v)$.

Consider the case where $h = 0$. That is, suppose $u = v$. For any other node w, $\text{level}_p(w) > \text{level}_p(v)$ and so $p \notin \mathsf{Transfer}_{w \to v}(P)$ by Lemma 7.1. Moreover $p \notin \mathsf{MyExits}(v)$ which implies that $p \notin \mathsf{PossibleExits}(v, t)$ for any time $t > 0$.

Suppose $h > 0$. Then $u \neq v$ and $p \notin \mathsf{MyExits}(u)$. Consider any node $w \neq u$. There are two cases.

1. $\text{level}_p(w) < h$.
   Assume (inductively) that the claim is true for any node x with $\text{level}_p(x) < h$. Let $\tau > \max\{t_x : \text{level}_p(x) < h\}$ where t_x is such that for all $t \geq t_x, p \notin \mathsf{PossibleExits}(x, t)$. Let t be any time where $t > \tau$. Then by the induction hypothesis, $p \notin \mathsf{PossibleExits}(w, t-1)$, and hence $p \notin \mathsf{GoodExits}(w, t-1)$. Thus $p \notin \mathsf{Transfer}_{w \to u}(\mathsf{GoodExits}(w, t-1))$.

2. $\text{level}_p(w) \geq h$.
   By Lemma 7.1, we can assert that
   $p \notin \mathsf{Transfer}_{w \to u}(\mathsf{GoodExits}(w, t-1))$.

So for all nodes $w \neq u$, $p \notin \mathsf{Transfer}_{w \to u}(\mathsf{GoodExits}(w, t-1))$. Since σ is a fair activation sequence, there is some time $t_u > \tau$ where $u \in \sigma_{t_u}$. Then for all $t > t_u$, $p \notin \mathsf{PossibleExits}(u, t)$. ∎

An exit path p is *valid* if $p \in \mathsf{MyExits}(\mathsf{exitPoint}(p))$. From Lemma 7.2, for any fair activation sequence, there is some time after which all the exit paths in the system are valid. That is, from any configuration, any fair activation sequence eventually results in such a valid configuration. Clearly, once a valid configuration is reached, it remains valid. Thus we can assume without loss of generality that we start with a valid configuration. We now show that if we start with a valid configuration, then there is some time τ_1 such that for all times $t \geq \tau_1$ and for all $v \in V$, $\mathsf{PossibleExits}(v, t) = \mathsf{PossibleExits}(v, \tau_1)$. We start with the following lemma:

LEMMA 7.3. *Let $u \in V$ and let P be a set of exit paths. Suppose p is some exit path in P. Then for all $h > 0$ if $\text{level}_p(u) = h$, then there is some node w with $\text{level}_p(w) < h$ such that $p \in \mathsf{Transfer}_{w \to u}(P)$.*

Proof: Suppose $v = \mathsf{exitPoint}(p) \in C_i$. Let $h = \text{level}_p(u)$.

If $h = 1$ then $u \in R_i$ and so there is an edge $vu \in E_I$. Since $v = \mathsf{exitPoint}(p)$, then $\text{level}_p(v) = 0$ and by Case 1 in the definition of $\mathsf{Transfer}_{v \to u}(P)$, $p \in \mathsf{Transfer}_{v \to u}(P)$.

If $h = 2$ then there are two cases. Suppose $u \in N_i$. Then for any $w \in R_i$ there is an edge $uw \in E_I$. Also by Case 3 in the definition of $\mathsf{Transfer}_{w \to u}(P)$, $p \in \mathsf{Transfer}_{w \to u}(P)$. So either $w = v$ and so $\text{level}_p(w) = 0$ or $w \neq v$ and so $\text{level}_p(w) = 1$. Suppose instead that $u \in R_j$, for some $j \neq i$. Then by Case 2 in the definition of $\mathsf{Transfer}_{w \to u}(P)$ it must be that $p \in \mathsf{Transfer}_{w \to u}(P)$ for any $w \in R_i$ and so $\text{level}_p(w)$ is either 0 or 1. In any case, $\text{level}_p(w) < h = 2$.

Suppose $h = 3$. That is, $u \in N_j$ for some $j \neq i$. Then again by Case 3 in the definition of $\mathsf{Transfer}_{w \to u}(P)$, $p \in \mathsf{Transfer}_{w \to u}(P)$ for any $w \in R_j$ and so $\text{level}_p(w) = 2 < h$. ∎

Now recall the definition of

$$\mathsf{S} = \bigcup_{v \in V} \mathsf{MyExits}(v).$$

and let $\mathsf{S}^\flat = \text{Choose\_max}^\flat(\mathsf{S})$. Then the following lemma is a straightforward consequence of the definition of $\mathsf{S}^\flat$.

LEMMA 7.4. *If P is a set of exit paths such that $S^\flat \subseteq P \subseteq S$, then* $\text{Choose\_max}^\flat(P) = S^\flat$.

We now use Lemma 7.3 to show that for any fair activation sequence, eventually for all $u \in V$, $\mathsf{GoodExits}(u, t) = \mathsf{S}^\flat$.

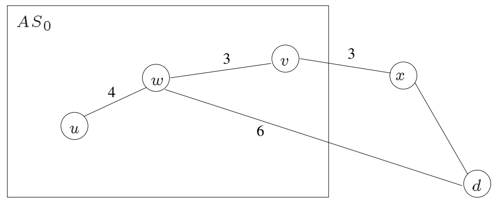


Figure 12: Real vs. Calculated Routes

LEMMA 7.5. *Let $p \in S^\flat$. Then for every node u there is some time τ_u such that $p \in \mathsf{PossibleExits}(u, t)$ for all $t \geq \tau_u$.*

Proof: The proof is by induction on the level of u. If $\text{level}_p(u) = 0$ then $\mathsf{exitPoint}(p) = u$ and so $p \in \mathsf{MyExits}(u)$. Then by definition, $p \in \mathsf{PossibleExits}(u, t) \supseteq \mathsf{MyExits}(u)$ for all t.

Suppose $\text{level}_p(u) = h + 1 > 0$ and the claim holds for all w where $\text{level}_p(w) \leq h$. That is, there is some time $t[h]$ such that for all $t \geq t[h]$, $p \in \mathsf{PossibleExits}(w, t)$ for all w with $\text{level}_p(w) \leq h$. Since $p \in \mathsf{S}^\flat$, it is clear that $p \in \mathsf{GoodExits}(w, t)$ for all $t \geq t[h]$. By Lemma 7.3, there must be some v with $\text{level}_p(v) \leq h$ where $p \in \mathsf{Transfer}_{v \to u}(\mathsf{GoodExits}(v, t))$. Since σ is a fair activation sequence there is some time $\tau_u > t[h]$ such that $u \in \sigma_{\tau_u}$ and so $p \in \mathsf{PossibleExits}(u, t)$ for all $t \geq \tau_u$. ∎

We can conclude from Lemma 7.5 that there is some time τ_1 such that $\mathsf{S}^\flat \subseteq \mathsf{PossibleExits}(u, t)$ for all $u \in V$ and for all $t \geq \tau_1$. Since we assume we are starting from a valid configuration, we know that $\mathsf{PossibleExits}(u, t) \subseteq \mathsf{S}$ for all $u \in V$. Thus by Lemma 7.4, $\mathsf{GoodExits}(u, t) = \text{Choose\_max}^\flat(\mathsf{PossibleExits}(u, t)) = \mathsf{S}^\flat$ for all $t \geq \tau_1$. As noted in Section 6,

$$\begin{aligned} \mathsf{BestRoute}(u, t) &= \mathsf{best}_u(\mathsf{route}(\mathsf{GoodExits}(u, t), u)) \\ &= \mathsf{best}_u(\mathsf{route}(\mathsf{S}^\flat, u)) \end{aligned}$$

That is, $\mathsf{BestRoute}(u, t)$ remains fixed for every node u for all $t \geq \tau_1$. Notice that this means that for any fair activation sequence, not only does $\mathsf{BestRoute}(u, t)$ eventually converge for each $u \in V$ but it converges to the *same route* for any fair activation sequence starting from the same initial valid configuration.

Loop-Free Properties. Consider the example in Figure 12. It shows that even though node u considers u–w–v–x–d to be its best route to d, the intermediate node w routes all the packets[10] to d via w–d (E-BGP route better than I-BGP route). We refer to the actual routes taken by packets to be the *real routes*. This example shows that a real route can be different from the route that the source thinks the packet will follow. Since intermediate routers may forward packets in a way that is not envisaged by the source, there is a possibility that routing loops may be created within AS_0.

We now show that this is not the case for the algorithm described in the previous section. More specifically, we show that if $r = (q, p)$ is the best route for node u to destination d, and w is an intermediate node on q, then for all packets from u to d, w either sends them along q, or it sends all such packets out of AS_0 on an external link. In either case, no packet ever goes back to the source u.

LEMMA 7.6. *Let $p = \mathsf{exit}(\mathsf{BestRoute}(u, \tau_1))$ and $v = \mathsf{exitPoint}(p)$. If w is a node along $SP(u, v)$, then either $\mathsf{exit}(\mathsf{BestRoute}(w, \tau_1)) = p$ or $w = \mathsf{exitPoint}(\mathsf{BestRoute}(w, \tau_1))$.*

[10]For simplicity, in this discussion we refer to nodes "forwarding packets" instead of referring to the "routers associated with nodes forwarding packets".

Proof: Let $r = \mathsf{BestRoute}(u, \tau_1)$ and let $r' = \mathsf{BestRoute}(w, \tau_1)$. Let $p' = \mathsf{exit}(r')$ and $v' = \mathsf{exitPoint}(p')$. Suppose that $p' \neq p$. Since $p, p' \in \mathsf{S}^\flat$, then $\mathsf{localPref}(r) = \mathsf{localPref}(r')$ and $\mathsf{AS\text{-}path\text{-}length}(r) = \mathsf{AS\text{-}path\text{-}length}(r')$. Also if $\mathsf{nextAS}(r) = \mathsf{nextAS}(r')$ then $\mathsf{MED}(r) = \mathsf{MED}(r')$. So then one of the following conditions must be true:

1. $v' = w$ and $v \neq w$ or,
2. $cost(SP(w, v')) + \mathsf{exitCost}(p') < cost(SP(w, v)) + \mathsf{exitCost}(p)$ and either $v, v' = w$ or $v, v' \neq w$, or,
3. $cost(SP(w, v')) + \mathsf{exitCost}(p') = cost(SP(w, v)) + \mathsf{exitCost}(p)$ and either $v, v' = w$ or $v, v' \neq w$ and also, $\mathsf{learnedFrom}(r') < \mathsf{learnedFrom}(r)$.

Suppose Condition 2 is true. Then

$$\begin{aligned} & cost(SP(u, v')) + \mathsf{exitCost}(p') \\ \leq \quad & cost(SP(u, w)) + cost(SP(w, v')) + \mathsf{exitCost}(p') \\ < \quad & cost(SP(u, w)) + cost(SP(w, v)) + \mathsf{exitCost}(p) \\ = \quad & cost(SP(u, v)) + \mathsf{exitCost}(p). \end{aligned}$$

But this contradicts the fact that $p' \in \mathsf{S}^\flat$ yet $p = \mathsf{exit}(\mathsf{BestRoute}(u, \tau_1))$. Similarly, if Condition 3 holds, then $r \neq \mathsf{BestRoute}(u, \tau_1)$. Thus if $p' \neq p$ then it must be that Condition 1 holds. But then $w = v' = \mathsf{exitPoint}(r')$. ■

In fact, it is possible to show that the real paths along which packets are routed form a shortest-path tree rooted at d. Now suppose that $\mathsf{exitCost}(p) = 0$ for all exit paths p and the costs of the edges in E_I are all strictly positive. Then we could make the following stronger claim about the routes chosen by the vertices in V at time τ_1.

LEMMA 7.7. *Let $p = \mathsf{exit}(\mathsf{BestRoute}(u, \tau_1))$ and $v = \mathsf{exitPoint}(p)$. Then if w is a node along $SP(u, v)$, then $\mathsf{exit}(\mathsf{BestRoute}(w, \tau_1)) = p$.*

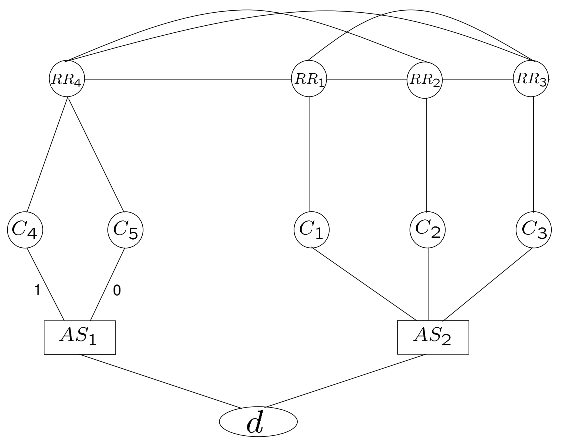
Discussion. In the course of our convergence proof, we made two assumptions. First, we assumed that after time 0, there are no more E-BGP updates that are injected into AS_0. In other words, E-BGP routes stabilize at time 0. Obviously, such is not the case in today's Internet. However, there is no algorithm that will converge if the E-BGP routes injected into AS_0 keep changing. Any route that an algorithm converges to can be withdrawn in the next E-BGP update, thereby causing route oscillations to continue indefinitely. Therefore, in order to prove convergence, it is necessary to assume that the E-BGP routes stabilize.

Second, we assumed that during an execution of our algorithm, there are no router crashes (such executions are called fair activation sequences). This is not a restrictive assumption since it is not possible (or necessary) to prove eventual convergence for a router that crashes. Indeed there is no algorithm that guarantees convergence if all routers crash at time 0.

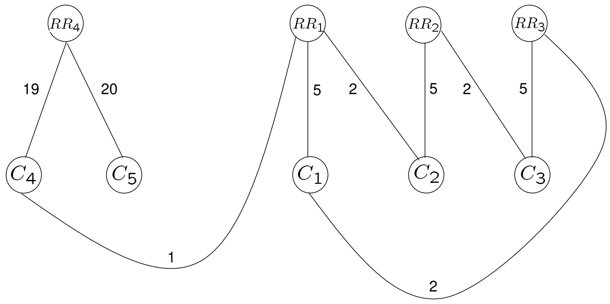
Finally, we have provided no bounds on the time taken by a router to converge to a stable route. This is because we use an asynchronous system model where we do not make any assumptions regarding message delay bounds and bounds on relative router speeds (e.g., router A may be k times slower than router B, and the value of k is not bounded). This makes the proofs more general — however, to estimate convergence times, we require a synchronous system model which is beyond the scope of this paper.

8. Comparison with Other Solutions

In this section, we provide a persistent oscillation example where the Walton et al. [23] solution fails to eliminate routing oscillations. We also describe a configuration with routing loops where our solution is able to eliminate the routing loops, whereas the the Walton et al. solution does not.



(a) BGP Sessions and MED Values



(b) Physical Links and IGP Metric Values

Figure 13: Persistent Route Oscillations for the Walton et al. Solution

Brief Overview of the Walton et al. Solution. The basic change to I-BGP is that for each neighboring AS, each route reflector computes its best route to d through that AS. If this route has the same LOCAL-PREF and the same AS-PATH length as its overall best route, the route reflector announces this route to all its I-BGP peers subject to the usual announcement rules for I-BGP with route reflection (described in Section 2). Thus, if there are m neighboring AS-es, then each route reflector sends information about at most m routes. If one of the announced routes is the single overall best route for the route reflector,[11] the route reflector indicates which route this is. All these announced routes are considered in the path selection process by other routers.

The Persistent Oscillation Counterexample. We now show an example with MED-induced (i.e., not observed if MEDs are absent) persistent oscillations that are eliminated by our algorithm but not by the algorithm proposed by Walton et al. [23]. This example is a modification of an example from [9]. Consider the configuration in Figure 13. There are four route reflection clusters, with route reflectors RR_1 through RR_4. Route reflectors RR_1 through RR_3 have clients

[11] Note that the route reflector may not always announce its own overall best route — this is subject to the usual I-BGP route announcement rules.

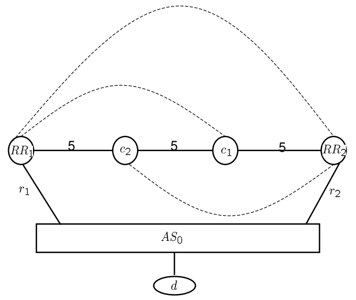


Figure 14: Example with Routing Loops

C_1 through C_3, respectively, and route reflector RR_4 has two clients — C_4 and C_5. The I-BGP sessions among the various routers are shown in Figure 13(a) and the underlying physical topology is shown in Figure 13(b). Note that the routes through C_1, C_2 and C_3 have the same MED, say 0 (which is not shown in the figure).

It is possible to see that in the absence of MEDs, RR_4 chooses and announces the route through C_4 to RR_1 (lowest IGP cost). Route reflector RR_1 selects that route (lowest IGP cost) as its best route as well. Consequently, route reflector RR_3 selects the route through C_3 (lowest IGP cost) and so does route reflector RR_2. This selection of routes forms a stable solution for the configuration. However, if MEDs are introduced, RR_4 is forced to choose the route through C_5 (lower MED). As a result, route reflectors RR_1 through RR_3 continue to oscillate between various route selections, none of which are stable. If the solution proposed by Walton et al. is used, we can see that no route reflector will announce any extra routes in this particular case. Hence, the system continues to behave in the same fashion as in the case for standard I-BGP, and the route oscillations are not eliminated.

In contrast, when our solution is used, all the route reflectors (RR_1 through RR_4) always announce to each other the routes through C_1, C_2, C_3 and C_5, respectively. Thus, RR_1 chooses the route through C_2, RR_2 chooses the route through C_3, RR_3 chooses the route through C_1, and RR_4 chooses the route through C_5. It is easy to verify that this is a stable routing solution.

Routing Loops. Consider the configuration shown in Figure 14. This configuration was first described in [2] as a scenario that causes routing loops. Here, route reflectors RR_1 and RR_2 have clients c_1, and c_2, respectively. The solid lines represent physical links, and the dotted lines show I-BGP peering relations. For example, I-BGP peering between RR_1 and c_1 goes through c_2. Both routes r_1 and r_2 have the same LOCAL-PREF, AS-PATH length, and MED value. The IGP cost for each physical link is 5. In normal I-BGP operation, RR_1 chooses r_1 (E-BGP route over I-BGP route) and RR_2 chooses r_2 (E-BGP route over I-BGP route). Thus c_1 only hears about r_1 from its route reflector and chooses r_1, similarly, c_2 chooses r_2. However, when c_2 tries to route packets to destination d, it must send it to c_1 (since the next hop to d is c_1) which sends it back to c_2, creating a routing loop.

It is easy to see that the solution proposed in [23] does not solve this problem since RR_1 only advertises r_1 to its clients and RR_2 only advertises r_2 to its clients under this scheme. In contrast, the modification that we propose solves the problem since both RR_1 and RR_2 advertise r_1 *and* r_2 to their clients (both $r_1, r_2 \in S^\flat$). Subsequently, c_1 chooses r_2 and c_1 chooses r_1 (lower IGP metric) and there are no routing loops. This example shows that our algorithm continues to work correctly even in certain "badly configured" systems.

9. Related Work

One of the first works to report on BGP convergence problems showed that there are routing policies that cause External-BGP (E-BGP) to diverge [21]. Griffin and Wilfong [7] performed an analysis of E-BGP convergence properties using graph-theoretic methods. They showed that even checking whether an E-BGP configuration can converge is an NP-Complete problem.

Various solutions have been proposed to address the E-BGP convergence problem. Govindan et al. [5] proposed a static solution where routing policies would be analyzed by programs to determine whether policy conflicts could lead to protocol divergence. A more dynamic solution uses "route flap dampening" to control the dissemination of routing updates [22]. Whenever there is any policy conflict, this mechanism prevents updates from occurring too frequently and causing update storms.

Griffin et al. used a graph theoretic formalism called the "Safe Path Vector Protocol" (SPVP) to characterize sufficient conditions for BGP (or any path vector protocol) convergence [6]. A solution that uses a new route attribute called the "route history" to guarantee the convergence of SPVP was also proposed [8]. Independently, Gao and Rexford have proposed a set of policy guidelines that guarantee convergence in E-BGP without requiring any coordination among the different AS-es [4]. The SPVP formalism, in conjunction with certain policy guidelines was later used to ensure E-BGP convergence in networks where backup routing is used [3].

I-BGP has also been an area of much investigation. Several problems with route reflection in I-BGP have been outlined by Dube and Scudder [2]. In this work, the authors show how certain route reflection configurations can lead to routing loops or incorrect routing decisions. They also provide guidelines for avoiding such problems. More recently, a different kind of routing oscillation (that we refer to as persistent route oscillations) problem for operational networks running I-BGP with route reflection or confederations was reported [19]. This problem was analyzed further in [16] and later work proposed a modification to I-BGP to address this route oscillation problem [23]. We have shown in the previous section that the solution posed in [23] fails to eliminate persistent oscillations in all cases.

The adverse effects of inter-domain route oscillations have also been studied. Empirical studies have used real routing traffic traces to describe a whole range of unexpected and anomalous behavior in inter-domain routing protocols such as BGP [14]. Other work analyzed the cause of such routing instabilities and suggested remedies [15]. More recently, Labovitz et al. studied (using empirical data) how BGP route oscillations affect the convergence times after a failure occurs in the Internet [12]. Later work examined the impact of specific Internet Service Provider policies and topologies on the speed of routing convergence [13]. Finally, Pei et al. used consistency assertions to compare similar routes and identify infeasible routes in an effort to speed up BGP convergence times [17].

10. Conclusions and Future Work

We have described a solution to the route oscillation problem in I-BGP with route reflection. The solution is a modification to I-BGP and the modified protocol provably converges. That is, it prevents persistent as well as transient route oscillations for I-BGP with route reflection. In addition, the modified protocol is guaranteed to converge to the same stable routing configuration independent of the timing and order of sent and received messages. This is helpful for analyzing and debugging scenarios where a (set of) router(s) goes down and comes back up again. Network operators prefer configurations where the routing tables before and after the crash are identical. Finally, our solution also prevents routing loops within an autonomous system.

In the future, we would like to explore three issues related to the work presented here. First, our current solution is designed to work in networks without any modifications to the current MED attribute. We would also like to explore solutions that provide the same functionality as the MED attribute, but without the associated routing oscillations. Second, the solution we propose here requires extra routing information to be propagated for each destination prefix. Such a solution may not be scalable as is — however, it is possible to treat the propagation of extra routes as a feature that is only triggered when route oscillations are detected for some destination prefix. The exact details of such a detection mechanism and how it can be integrated with our solution is another subject for future work. Third, the convergence proof for our solution works only if we assume that the external routes injected by E-BGP into an AS stop changing. However, there could be interactions between E-BGP and I-BGP that cause route changes in I-BGP to affect E-BGP routes and vice versa, resulting in route oscillations. In the future, we would like to identify such scenarios and suggest fixes for them.

Acknowledgments

We would like to thank Tim Griffin and the anonymous referees for their valuable comments.

11. References

[1] T. Bates and R. Chandra. BGP Route Reflection: An Alternative to Full Mesh I-BGP. RFC 1966, 1996.

[2] R. Dube and J. G. Scudder. Route Reflection Considered Harmful. IETF Internet Draft `draft-dube-route-reflection-harmful-00.txt`, Work in Progress. Available from http://alternic.net/drafts/drafts-d-e/draft-dube-route-reflection-harmful-00.html., November 1998.

[3] L. Gao, T. G. Griffin, and J. Rexford. Inherently Safe Backup Routing with BGP. In *Proceedings of Infocom '01*, Anchorage, Alaska, April 2001.

[4] L. Gao and J. Rexford. Stable Internet Routing without Global Coordination. In *Proceedings of SIGMETRICS '00*, Santa Clara, California, June 2000.

[5] R. Govindan, C. Alaettinoglu, G. Eddy, D. Kessens, S. Kumar, and W. Lee. An Architecture for Stable, Analyzable Internet Routing. *IEEE Network*, 13(1):29–35, 1999.

[6] T. G. Griffin, F. B. Shepherd, and G. Wilfong. Policy Disputes in Path Vector Protocols. In *Proceedings of the 7th International Conference on Network Protocols (ICNP'99)*, Toronto, Canada, November – December 1999.

[7] T. G. Griffin and G. Wilfong. An Analysis of BGP Convergence Properties. In *Proceedings of SIGCOMM '99*, Cambridge, Massachusetts, August – September 1999.

[8] T. G. Griffin and G. Wilfong. A Safe Path Vector Protocol. In *Proceedings of Infocom '00*, Tel Aviv, Israel, March 2000.

[9] T. G. Griffin and G. Wilfong. On the Correctness of IBGP Configuration. In *Proceedings of SIGCOMM '02*, Pittsburgh, Pennsylvania, August 2002.

[10] B. Halabi and D. McPherson. *Internet Routing Architectures*. Cisco Press, Indianapolis, Indiana, second edition, 2000.

[11] J. W. Stewart III. *BGP4 Inter-Domain Routing in the Internet*. Addison-Wesley, New York, New York, 1999.

[12] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian. Delayed Internet Routing Convergence. In *Proceedings of SIGCOMM '00*, Stockholm, Sweden, August–September 2000.

[13] C. Labovitz, A. Ahuja, R. Wattenhofer, and S. Venkatachary. The Impact of Internet Policy and Topology on Delayed Routing Convergence. In *Proceedings of Infocom '01*, Anchorage, Alaska, April 2001.

[14] C. Labovitz, G. R. Malan, and F. Jahanian. Internet Routing Instability. In *Proceedings of SIGCOMM '97*, Cannes, Paris, September 1997.

[15] C. Labovitz, G. R. Malan, and F. Jahanian. Origins of Internet Routing Instability. In *Proceedings of Infocom '99*, New York, New York, March 1999.

[16] D. McPherson, V. Gill, D. Walton, and A. Retana. BGP Persistent Route Oscillation Condition. IETF Internet Draft `draft-ietf-idr-route-oscillation-00.txt`, Work in Progress, March 2001.

[17] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Improving BGP Convergence Through Consistency Assertions. Submitted for Publication, 2001.

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

[19] Cisco Systems. Endless BGP Convergence Problem in Cisco IOS Software Releases. Cisco Systems Inc. Field Notice, October 10 2000.

[20] P. Traina. Autonomous System Confederations for BGP. RFC 1965, 1996.

[21] K. Varadhan, R. Govindan, and D. Estrin. Persistent Route Oscillations in Inter-domain Routing. *Computer Networks*, 32:1–16, 2000.

[22] C. Villamizar, R. Chandra, and R. Govindan. BGP Route Flap Damping. RFC 2439, 1998.

[23] D. Walton, D. Cook, A. Retana, and J. Scudder. BGP Persistent Route Oscillation Solution. IETF Internet Draft `draft-walton-bgp-route-oscillation-stop-00.txt`, Work in Progress, May 2002.