

An Overview of Solutions to Avoid Persistent BGP Divergence

Ravi Musunuri Jorge A. Cobb
Department of Computer Science
The University of Texas at Dallas
Email: {musunuri, cobb}@utdallas.edu

Abstract

The Internet is a collection of multiple interconnected and self-administered domains, known as autonomous systems. In order to find a path from one autonomous system to any other autonomous system, neighboring autonomous systems exchange routing information via the Border Gateway Protocol (BGP). However, BGP suffers from several types of divergence anomalies. A divergence anomaly occurs when BGP routers permanently fail to obtain a stable path to reach a destination autonomous system. In this article, we discuss the different types of BGP divergence anomalies, along with their proposed solutions.

I. INTRODUCTION

The Internet is a collection of multiple interconnected and self-administered domains, known as Autonomous Systems (ASes). In order for each AS to learn a path to all other ASes, neighboring ASes exchange routing information via the Border Gateway Protocol (BGP) [1]. A distinguishing feature of BGP is that each router advertises, for each destination AS, the full path of ASes that are traversed to reach the destination AS. BGP is thus referred to as a path-vector protocol. The use of path-vectors enables BGP to choose its path to the destination based on a routing policy that is defined locally at the AS. This routing policy may be based on commercial or trust relationships between the AS and its neighboring ASes. Note that this is in contrast to intra-AS routing protocols (e.g., OSPF, RIP, and EIGRP) that are based on link metrics, such as link-cost or bandwidth.

For example, consider an AS-graph shown in Fig. 1(a), where each node denotes an AS, and each link represents an inter-AS link between border routers in neighboring ASes. In BGP, the path chosen for one destination is independent of the path chosen for any other destination. Thus, for simplicity, we consider only a distinguished node d as the destination.

In Fig. 1(a), node v has two neighbors along the path to AS d , namely, x and w . Each of these neighbors informs v of its entire path to d . That is, x informs v that its path to d is (x, y, d) , and w informs v that its path to d is (w, d) . Node v has the freedom to choose any of these two paths. If v chooses the path via x , then v informs u that its path to d is (v, x, y, d) .

In practice, an AS consists of multiple routers, as shown in Fig. 1(b). This figure expands AS v , showing the routers in v and the communication links between them. A router can be either an *internal* router or a *border* router. All the neighbors of an internal router are located within its own AS, while some of the neighbors of a border router are located outside of its AS. In Fig. 1(b), internal routers are denoted by I , and border routers are denoted by B .

Two BGP routers are said to be *BGP peers* if they exchange routing information via the BGP protocol. More specifically, if a peer is located outside of the AS of a router, then the router uses the external Border Gateway

Protocol (eBGP) to communicate with the peer. On the other hand, if a peer is located within the AS of the router, then the internal Border Gateway Protocol (iBGP) is used to communicate with the peer.

The exchange of routing information between peers is performed over a reliable TCP connection. If the peers are located in different ASes (i.e., an eBGP peering session), then they must share a physical link in order to establish this TCP connection. However, peers within the same AS (i.e., an iBGP peering session) may be separated by multiple intra-AS hops. This is because routing of messages between the peers is performed by the intra-AS protocol, such as OSPF.

In the original iBGP peering scheme, each border router maintains a peering session with all other routers within its AS, which is shown in Fig. 1(c). This full-peering scheme fails to scale as the size of the AS increases. To improve scalability, route-reflection clustering [2] and confederations [3] were introduced. In this article, we focus on iBGP with route reflection, which is commonly used and has received considerable attention in the literature. We present an overview of route-reflection in Section III.

BGP has multiple forms of unstable and irregular behavior. For example, some mis-configurations and software anomalies can cause BGP to generate orders of magnitude more message overhead than necessary [4]. Also, inconsistent path advertisement from neighboring ASes may lead to very slow convergence after a route failure [5]. Our focus in this article is a problem of even greater impact: the permanent failure of BGP to converge to a stable route to the destination. We refer to this failure as a divergence anomaly.

Both external and internal BGP suffer from divergence anomalies. In eBGP, conflicting routing policies [6] between different ASes are the root cause for divergence. In iBGP, however, divergence is caused by the interaction between route-reflection clustering, multi-exit-discriminator values (defined below), and intra-AS cost values. iBGP may diverge even with an anomaly-free eBGP.

Given the continuous growth of the Internet, and the proliferation of different ASes, the occurrence of BGP divergence is bound to increase in the future. This, along with the increasing importance of the Internet in every day life, demands a solution to the BGP divergence problem that is flexible, efficient and effective. In this article, we discuss different divergence anomalies associated with BGP, and we present a survey of solutions from the literature.

II. BGP PATH SELECTION

Each router learns a path to destination d from each of its peers. A path P received by a router R located in AS v contains the following attributes

- *local_pref* : a preference value indicating the ranking of P in the local routing policy of v . A larger preference value indicates a greater preference for the path.
- *AS_path* : sequence of ASes along the path to reach the destination d from the current v .
- *MED* : for a pair of ASes connected by more than one link, the Multi-Exit Discriminator (MED) value indicates the preference of one link over another. A smaller *MED* value indicates a greater link preference.
- *next_hop* : the IP address of the “next-hop” border router along path P . If traffic from R along P traverses other routers before exiting v , then *next_hop* is the IP address of the border router that is the exit point from v . If traffic from R along P goes directly from R to a neighboring router in a different AS, then *next_hop* is the IP address of this neighboring router.

From each peer, a router receives a path (potentially empty) to reach the destination. From this set of paths, the router must choose the “best” path and adopt it as its own path. The best path is chosen according to the algorithm given in Fig. 2 [7]. If a router adopts a new path, i.e. if its best path is not its previously chosen path, then the router informs each of its peers about the newly chosen path.

III. ROUTE REFLECTION CLUSTERING

In the original iBGP peering scheme, each border router is a peer of all other routers within the same AS. As the size of the AS increases, this scheme fails to scale. A common solution is to employ route-reflection clustering [2]. In this approach, the routers within an AS are divided into disjoint sets, known as *clusters*. In Fig. 1(d), AS v is divided into two clusters depicted by the shaded regions. One distinguished router in each cluster is known as the *reflector*. The reflector of the cluster i is denoted F_i , and to highlight this node, it is drawn in bold. Border routers within cluster i are denoted $B_{i,j}$ for some j , and likewise interior routers within cluster i are denoted $I_{i,j}$ for some j .

Each reflector maintains a peering session with routers that fall in the following three categories: (a) all routers within its own cluster (via iBGP peering), (b) all reflectors of all other clusters in its AS (via iBGP peering), (c) in the case when the reflector is also a border router, all its neighboring routers outside of its AS (via eBGP peering). Figure 1(e) shows the iBGP peering sessions with route reflection. All routers, within its cluster, that establish a peering session with a reflector are known as the *clients* of the reflector. For example, in Fig. 1(d), the clients of reflector F_2 are $I_{2,1}$ and $B_{2,1}$.

Note that interior routers learn about paths to the destination only via their reflector. Furthermore, although border routers may learn paths from their neighbors outside of their AS, the only router within their own AS from whom they learn paths is their reflector. As an example, consider border router $B_{2,1}$ in Fig. 1(d). Although it has a peering session with its neighbor in AS w , it has only a single peer, reflector F_2 , within its own AS, even though it is also a neighbor of both F_1 and $I_{1,1}$.

Each reflector, F_i , advertises its best path to other peers as explained below:

- If F_i received its best path from another reflector, then F_i advertises its best path to all its clients and eBGP peers.
- If F_i received its best path from a client or from an eBGP peer, then F_i advertises its best path to all reflectors, to all its clients, and to all its eBGP peers (except the router from whom the best path was received).

IV. EBGP DIVERGENCE

Recall that path preferences are chosen locally at each AS. If path preferences at neighboring ASes conflict with each other, it may not be possible to maintain a stable path to the destination. That is, the path chosen by some ASes oscillates continuously (diverges), even though neither the AS-graph nor the path policies change.

In this section, we present an example of eBGP divergence, along with various solutions from the literature. Griffin et. al. [8] were the first to study the root causes of this problem. They presented an abstraction of the problem, known as the Stable Paths Problem (SPP), and provided sufficient conditions to ensure its convergence. They also showed that, in general, analyzing SPP, and hence also BGP, for divergence is an NP-hard problem [9].

Consider the AS-graph known as “bad-gadget” [6], shown in Fig. 3. The paths acceptable to an AS (i.e. ranked higher than the empty path) are alongside the AS in order of rank. Note that each AS prefers longer paths over shorter paths. E.g., u prefers the longer path (u, v, d) over the shorter path (u, d) . This causes the ranking of each AS to be in conflict with the ranking of its next hop to d .

The cyclic relationship between these ranking prevents any AS from obtaining a stable path to d . To see this, consider the following steps:

- 1) Initially u , v , and w choose paths (u, v, d) , (v, d) , and (w, d) , respectively, as shown in Fig. 3(a).
- 2) v notices that w chose path (w, d) . Hence, v changes from its current path (v, d) to the higher preference path (v, w, d) . This in turn forces u to change its path from (u, v, d) (which no longer exists) to (u, d) as shown in Fig. 3(b).

- 3) Similarly, w notices that u chose path (u, d) . Hence, w changes its path to (w, u, d) . This in turn forces v to change its path to (v, d) as shown in Fig. 3(c).
- 4) Finally, u notices that v chose path (v, d) . Hence, u changes its path to (u, v, d) . This in turn forces w to change its path to (w, d) , and the system is back to its initial state in Fig. 3(a).

Converging to a steady state is highly sensitive to path rankings. For instance, in Fig. 3, reversing the ranking of paths at u ensures that the system reaches a steady state.

V. EBGP SOLUTIONS

Before discussing the proposed solutions to eBGP divergence, we discussed the desired properties of a solution.

- **Scalability and Efficiency:** Given the global scale of the Internet, the proposed solution should be efficient, and scalable.
- **No global coordination:** As the number of ASes increases, any solution requiring global coordination among ASes may not be scalable.
- **No restriction on AS policies:** Every AS should independently be able to control its routing policies. Thus, the solution should avoid as much as possible restricting routing policies.
- **Minimal changes:** Due to the wide deployment of BGP, any proposed solution should change the behavior of the current BGP protocol the least possible.

The solutions to eBGP divergence can be divided into three main categories. The first category requires global coordination among ASes to avoid conflicting routing policies [10]. The second category enforces convergence by restricting the type of routing policies an AS may adopt [11]. The third category avoids divergence by detecting conflicts at run-time. There are three approaches within the run-time category. The first approach carries path histories [12] with each BGP update message, and conflicts are detected by observing cycles in the received histories. The second approach detects conflicts via diffusing computations [13], and prevent ASes from choosing a path that conflicts with other ASes. In the third approach, each AS maintains a metric value [14], which grows without bound during divergence. Divergence is avoided by restricting the policies only when the metric value grows above some threshold.

Below, we overview solutions in each of the above three categories. Each solution has its own set of advantages and disadvantages. Currently, some service providers are statically checking the conflicts in routing policies, whenever available, using global coordination. It is yet to be determined if any of the solutions presented below will gain a wide acceptance by service providers.

A. Static Checking of Routing Policies

Conflicts in routing policies may be avoided by collecting the policies of all ASes in a single location, and then analyzing these policies for conflicts before they are set into practice. An example is the Routing-Arbitrator project [10]. Here, Govindan et al. designed an inter-domain routing architecture to gather the routing policies of multiple ASes and check for conflicts. The architecture consists of describing routing policies using a common language, and storing them in a global database. A set of software tools are provided to analyze the routing policies in the global database and attempt to find conflicts among them.

One drawback of this solution is that ASes are often unwilling to share their local routing policies with others due to privacy concerns. Most importantly, however, is that Griffin et al. [6] have shown that deciding whether a set of routing policies may lead to divergent behavior is intractable, more specifically, it is NP-hard.

B. Routing Policy Restriction

Gao et al. [11] proposed a set of guidelines for choosing routing policies based on the hierarchical structure of commercial relationships between ASes. These relationships include customer-provider, and peer-peer. In general, commercial relationships are based on the size of the ASes. In a customer-provider relationship, a customer's data transits through a provider AS, which is larger than the customer AS, to reach the rest of the Internet. The provider could itself be a customer of an even larger AS. In a peer-peer relationship, two ASes of similar size use the network resources of each other to connect to the Internet.

Each AS exports its routes and the routes learned from its customer ASes to all its providers. Each AS also exports its routes, routes learned from its providers and peers to all its customers. Peer ASes export their paths and paths learned from their respective customer ASes to each other. The guidelines for choosing the paths are that each AS prefers paths via its customer AS than via a peer AS or a provider AS. These restrictions ensure the routing policies are conflict-free, and thus convergence is assured.

The advantage of this solution is that it requires no modification to the current BGP protocol. There are some disadvantages, however. Commercial relations between ASes are not always clearly defined as assumed in this work and not all ASes may desire to restrict their policies in this way. Thus, the routing policy freedom of the original BGP protocol is lost. Also, if the routing policy is mis-configured accidentally at a router, then conflicts, and hence divergence, may occur.

C. Run-Time Policy Analysis

Solutions explained in Sections V-A,V-B avoid divergence by restricting the routing policies. In this section, we present three solutions that do not restrict the routing policies. Instead, they resolve the divergence problem by discovering routing policy conflicts at runtime. These three solutions assume that each AS is composed of a single router. Extending them to support ASes with many routers is still an open issue.

1) *Run-Time Policy Analysis via Path Histories:* In [12], each AS maintains a history of the events that led it to adopt its current path to the destination. In addition to informing each of its neighbors of its newly chosen path, each AS informs its neighbors of the history associated with that path.

The history of a path is the concatenation of one local event with the history of the path provided by the neighbor. More specifically, path histories are a sequence of path-change events. If an AS changes its path to P_{new} from P_{old} upon arrival of a neighbor's update message with history $Hist$, then there are two choices. If the AS prefers P_{new} over P_{old} , then it inserts event $(+, P_{new})$ at the beginning of $Hist$. Otherwise, if the AS prefers P_{old} over P_{new} , then it inserts event $(-, P_{old})$ at the beginning of $Hist$.

For example, let's consider the bad-gadget example shown in Fig. 3. In step 3, AS w receives an update message from AS u . The message indicates that u 's current path is (u, d) and u 's history is $(-, (u, v, d))(+, (v, w, d))$. This message causes w to choose the path via u because it is ranked higher. Thus, w sets its path to (w, u, d) , and sets its history to $(+, (w, u, d))(-, (u, v, d))(+, (v, w, d))$. I.e., w 's new history is that of u with the additional term $(+, (w, u, d))$ indicating that w increased its rank. As the execution continues, path histories grow. At the end of step 4, v 's history becomes $(+, (u, v, d))(-, (v, w, d))(+, (w, u, d))(-, (u, v, d))(+, (v, w, d))$, which clearly has a cycle (event (u, v, d) is repeating). It has been shown [12] that a cycle in a path history is a necessary but not sufficient condition for divergence. Hence, even though a cycle is detected it is still possible for the system to converge.

The advantage of path histories is that the execution causing the cycle is recorded in the history, and can be used later to analyze the problem. On the other hand, one drawback is a significant increase in message and memory

overhead. I.e., for each destination, in addition to maintaining the entire routing path, each AS needs to maintain a sequence of paths, each of which may be of size proportional to the routing path of AS.

Divergence is resolved by removing a path from the set of paths allowed at an AS, and thus, breaking the cycle in the path history. However, after a sequence of topology changes, the removed path might become crucial to maintain connectivity if it becomes the only available path to the destination.

Lastly, path histories may partially reveal the routing policies of ASes, which, as mentioned above, are sometimes preferred to be kept confidential.

2) *Path Choice Restriction via Diffusing Computations:* In [13], the observation is made that, if when any AS changes its path it is guaranteed to receive a path with a local preference value at least as high as that of its current path, then a stable set of paths is guaranteed to be achieved. That is, when the local preference value of the chosen path at all ASes monotonically increases, the system cannot diverge.

Given that path preferences are chosen independently at each AS, an additional restriction is necessary to ensure the monotonicity of *local_pref* values as follows. Before q adopts a new path, q asks any other AS p whose path currently traverses q if this change of path at q will cause the preference value of the current path of p to decrease. If this is the case, q refrains from adopting the new path.

The coordination between q and p is performed via a diffusing computation [15] along the *routing tree*. The routing tree is defined as follows. For every AS p and its next-hop neighbor q along its path to d , consider the directed edge (p, q) . The union of all these directed edges over all ASes form a routing tree. If there is a path from p to q along the routing tree, then p is a *descendant* of q and q is an *ancestor* of p .

Diffusing computations are performed along the subtree of the AS desiring a new path. When AS q desires a new path, q propagates its new path along its subtree. When a descendant p in the subtree of q receives the path of q , it determines if this new path, along with the routing tree path from p to q , has a lower preference than its current path. If so, p rejects the new path, and q is prevented from adopting the new path. Otherwise, p continues the propagation of the new path of q down its subtree. If all ASes in the subtree allow the new path, a positive feedback is sent to q , and q adopts the new path.

For example, let's consider step 2 in the bad-gadget example shown in Fig. 3. AS u is the only descendant of AS v . Before changing its path to (v, w, d) , AS v asks if this change decreases the preference value of the path at AS u . Changing the path at AS v from (v, d) to (v, w, d) would force AS u to change its path to a lower ranked (u, d) path. Hence, AS u sends back a negative reply, which refrains AS v from changing its path.

The above technique has the advantage of enforcing convergence regardless of which routing policy is chosen at each AS. Although convergence is assured, the routing policy freedom of the original BGP protocol is removed. In addition, the protocol prevents some sequence of path changes that does not necessarily cause the system to diverge. Finally, diffusing computations cause additional message overhead.

3) *Run-Time Policy Analysis via Bounded Metric:* In [14], we presented a solution that uses a metric value to detect and avoid divergence. Our solution is based on the following observation: during divergence, the rank of the best path at some ASes periodically decreases. For example, from Fig. 3(a) to Fig. 3(b), the rank of the best path at AS u decreases. Stated otherwise, as observed in [13], divergence is not possible if the rank of the current path at each AS monotonically increases.

To detect divergence, each AS maintains an integer metric value. Along with path advertisements, ASes advertise their metric to their neighbors. If the rank of the path at an AS decreases, then the new metric of the AS is the maximum of its previous metric (plus one) and the metric advertised by the neighbor along the new path. If the rank of the path at an AS increases, then the AS sets its metric to the metric advertised by the neighbor along the new path. When eBGP diverges, this metric update scheme guarantees to increase the metric value without

bound [14].

Given that metric values increase when the system diverges, it is evident that the system should restrict its behavior when metric values become large. One simple option is to restrain from updating its path any AS whose metric value is greater than some threshold, even if a path with higher preference is available. The disadvantage of this is that if there is a path that offers an escape [14] of the cyclic behavior, then an AS whose metric reaches the threshold would be unable to take this escape path.

Instead, we choose to prevent an AS from choosing a new path only when the new path is advertised by a neighbor whose metric is greater than the threshold. This is because, an escape path will likely contain a low metric value, and thus, the AS is free to choose an escape path, and break the cyclic behavior.

To quickly illustrate the above protocol, consider again Fig. 3(a). Let this be the initial state of the system, and let the metric value of all ASes be zero. The transition from Fig. 3(a) to Fig. 3(b) causes v to increase the rank of its path, and hence, the metric of v is set to the metric of w , i.e., it remains zero. However, it causes the rank of u 's path to decrease, and metric of u is set to one. The transition from Fig. 3(b) to Fig. 3(c) causes w to increase the rank of its path, and hence, the metric of w is set to the metric of u , i.e., to one. Furthermore, it causes the rank of v to decrease, and the metric of v is increased to one. Finally, the transition from Fig. 3(c) to Fig. 3(a) causes u to increase the rank of its path, and hence, the metric of u is set to the metric of v , i.e., it remains one. However, it causes the rank of w 's path to decrease, and the metric of w is set to two. Therefore, the metric of all three ASes increases. A similar sequence of events occur continuously, which increases the metric values without bound.

The above technique has the advantage of restricting the routing policy only when the system diverges. Furthermore, it is scalable, and efficient, by requiring the addition of a single integer to each BGP update message. More specifically, the BGP update message [1] allows a variable length sequence of path attributes to be added. Hence, this protocol simply requires a new attribute type to accommodate for the metric value.

VI. INTERNAL BGP

iBGP suffers from two different types of divergence anomalies. These anomalies occur even with stable eBGP. We explain both anomalies using examples shown in Fig. 4. In both examples, we assume that the *local_pref* and *AS_path* length values of all the paths are equal. Hence, best path selection is based on other attributes like *MED*, intra-AS routing cost values, etc. We also assume that the path from each border router in AS v to the destination d is stable. Therefore, v 's border routers will always choose a path via their eBGP peers in the neighboring AS, and, we focus only on the paths taken by the reflectors. Network links within AS v are labeled with the cost of the intra-AS routing protocol, and inter-AS links are labeled with their *MED* values. For terseness, we abbreviate the path of the reflector by removing the interior path. For example, path $(F_1, F_2, B_{2,1}, x, d)$ at F_1 in Fig. 4(b) will be denoted as $(B_{2,1}, x, d)$. Each interior path is always a shortest path between the reflector and the border router based on intra-AS routing costs.

A. Clustering-induced Divergence

One cause for divergence is the interaction between route-reflection clustering and intra-AS routing costs [16]. We refer to this anomaly as clustering-induced divergence, because this anomaly disappears if we remove clustering. This anomaly occurs even if *MED* values are not used for route selection.

An example of clustering-induced divergence is shown in Fig. 4(a) [16]. Figure 4(b) shows the iBGP peering sessions of Fig. 4(a). Note that in this example, each reflector F_i always prefers path $(B_{(i+1,1)}, w, d)$ over path

$(B_{i,1}, w, d)$ due to following¹:

$$\text{cost}(F_i, B_{i,1}) > \text{cost}(F_i, B_{(i+1,1)}). \quad (1)$$

- 1) Lets assume F_1 's current path is $(B_{2,1}, w, d)$, F_2 's current path is $(B_{2,1}, w, d)$ and F_3 's current path is $(B_{3,1}, w, d)$.
- 2) Next, if F_2 receives the path update message from F_3 , then F_2 withdraws $(B_{2,1}, w, d)$ and changes its current path to $(B_{3,1}, w, d)$. So, F_1 changes its current path to $(B_{1,1}, w, d)$.
- 3) Next, if F_3 receives the path update message from F_1 , then F_3 withdraws $(B_{3,1}, w, d)$ and changes its current path to $(B_{1,1}, w, d)$. So, F_2 changes its current path to $(B_{2,1}, w, d)$.
- 4) Next, if F_1 receives update from F_2 , then F_1 withdraws $(B_{1,1}, w, d)$ and changes its current path to $(B_{2,1}, w, d)$. So, F_3 changes its current path to $(B_{3,1}, w, d)$.

F_i routers would continuously exchange path update messages in the cyclic manner as above. They will never agree on stable set of paths.

B. MED-induced Divergence

MED-induced divergence [17] is caused due to the interaction between *MED* values, intra-AS routing costs, and route-reflection clustering. We refer to this anomaly as the MED-induced divergence, because this anomaly disappears if we ignore *MED* values during path selection.

Consider an example in Fig. 4(c), which was originally presented in [17]. Figure 4(d) shows the iBGP peering sessions of Fig. 4(c). It consists of an AS v , and two neighboring ASes x and w . AS v is divided into two clusters.

In this scenario, F_1 and F_2 fail to achieve a stable assignment of paths, as explained below:

- Let us assume F_1 chooses path $(B_{1,1}, x, d)$ since at the moment it is the only available path, and F_2 chooses path $(B_{2,1}, x, d)$ from the available paths $\{(B_{2,1}, x, d), (B_{2,2}, w, d)\}$.
- Next, if F_2 receives an update message from F_1 with path $(B_{1,1}, x, d)$, then F_2 chooses path $(B_{2,2}, w, d)$ from the available paths $\{(B_{1,1}, x, d), (B_{2,1}, x, d), (B_{2,2}, w, d)\}$. F_2 prefers $(B_{1,1}, x, d)$ over $(B_{2,1}, x, d)$ due to smaller *MED* value, and $(B_{2,2}, w, d)$ over $(B_{1,1}, x, d)$ due to smaller intra-AS costs.
- Next, if F_1 receives an update message from F_2 with path $(B_{2,2}, w, d)$, then F_1 chooses path $(B_{2,2}, w, d)$ from the available paths $\{(B_{1,1}, x, d), (B_{2,2}, w, d)\}$. F_1 prefers $(B_{2,2}, w, d)$ over $(B_{1,1}, x, d)$ due to smaller intra-AS costs.
- Next, because $(B_{1,1}, x, d)$ has been withdrawn, F_2 chooses path $(B_{2,1}, x, d)$ from the available paths $\{(B_{2,1}, x, d), (B_{2,2}, w, d)\}$. F_2 prefers $(B_{2,1}, x, d)$ over $(B_{2,2}, w, d)$ due to smaller intra-AS costs.
- Next, because $(B_{2,2}, w, d)$ has been withdrawn, F_1 chooses path $(B_{1,1}, x, d)$ since it is the only available path.

The above cyclic exchange of path update messages may continue indefinitely, and thus, a stable set of best paths may never be achieved.

Before discussing the proposed solutions to iBGP divergence, we discuss their desired properties.

- **Scalability and Efficiency:** In the original iBGP, every router advertises only a single path to its peers. It is desirable to preserve this feature, given the large scale of the Internet.
- **Support cold-potato routing:** BGP routing that uses *MED* values in path selection is referred to as cold-potato routing. Each AS uses *MED* values to inform the neighboring AS about its preference of one inter-AS link over another. *MED* values are particularly useful if a customer AS prefers to receive traffic on a specific

¹Note that mod 3 is implied on the subscript i

inter-AS link from the provider AS. This preference might be due to a very popular node being nearer to that inter-AS link than to others. Hence, cold-potato routing gives the AS flexibility in controlling incoming traffic.

VII. IBGP DIVERGENCE SOLUTIONS

Griffin et al. [16] provided a sufficient condition to solve the clustering induced divergence anomalies. Clustering induced divergence is avoided by restricting the choice of paths at each router. Each router should prefer the paths advertised by the client nodes over the paths advertised by the non-client nodes. For example, let's consider the example shown in Fig. 4(a). Each router F_i prefers the path $(B_{i,1}, w, d)$ over $(B_{i+1,1}, w, d)$, which avoids the divergence. These conditions does not solve the MED induced anomalies.

Next, we present two categories of iBGP solutions that solve both types of iBGP anomalies. The first category solves divergence anomalies by using multiple path dissemination between iBGP peers. There are two solutions in this category, presented in [7], [18]. Their basic difference lies in how reflectors compute multiple paths. The second category detects and avoids iBGP divergence anomalies via a metric [19], by using the eBGP results presented in Section V-C.3 [14].

A. Multiple Path Dissemination

In Walton et al. [18], each reflector advertises at most l paths to each of its iBGP peers, where l is the number of neighboring ASes. The reflector selects these l paths as follows. First, it finds the overall best path among the paths advertised by all its peers. Next, the reflector divides the set of paths advertised by all its peers into subsets, where each subset consists of those paths that exit via the same neighboring AS. Each reflector finds l paths by selecting the best path from each of the l subsets. The best path in each subset is chosen by using the path-selection algorithm in Fig. 2. Finally, from the l paths obtained, the reflector only advertises those paths, whose *local_pref* and *AS_path* length values are equal to the corresponding attributes of the overall best path.

In [7], Basu et al. gave a counter-example to the solution of Walton et al.. In addition, they proposed a new solution, in which, each reflector advertises at most m paths to each of its iBGP peers, where m is the number of border routers in its AS. Each reflector finds the set of at most m paths as follows. Each reflector applies the first three steps of the path-selection algorithm in Fig. 2 using as input the entire set of paths advertised by its peers. Their solution is proven correct.

Let us see how Basu et al. [7] solution avoids the iBGP divergence anomalies in examples shown in Fig. 4. In Fig. 4(a), each reflector F_i advertises its available path $(B_{i,1}, w, d)$ to other reflectors. Hence, F_1 chooses path $(B_{2,1}, w, d)$, F_2 chooses path $(B_{3,1}, w, d)$, and F_3 chooses path $(B_{1,1}, w, d)$ respectively and the clustering induced anomaly is resolved. Similarly, in Fig. 4(c), reflector F_2 advertises two available paths, $(B_{2,1}, x, d)$, $(B_{2,2}, w, d)$, to F_1 . Hence, both F_1 and F_2 choose the path $(B_{2,2}, w, d)$, and the MED induced anomaly is resolved.

The advantage of this solution is that it supports cold-potato routing, which gives more flexibility to the AS, as compared to other suggestions, such as removing *MED* values altogether, comparing the *MED* value over all inter-AS links [20], or restricting the choice of paths [16]. However, multiple path advertisements are required between every pair of iBGP peers, which increases memory and message overheads. Hence, multiple path dissemination is limited in efficiency and scalability. Also, it is contrary to the purpose of route-reflection clustering, i.e., reducing the number of path advertisements received by each router.

B. Run-Time Policy Analysis via Metric

In [19], we presented a solution that solves both iBGP divergence anomalies by using results from Section V-C.3 [14]. The general behavior of the iBGP solution is similar to the eBGP solution [14], but with the following

important differences. The eBGP solution [14] models each AS as a single node. On the other hand, a node in the iBGP solution [19] can be either an individual router within the AS being modelled or a neighboring AS. Their similarity lies in that each router in the iBGP solution maintains a metric value to detect divergence. Metric values grow without bound if there exists divergence in iBGP. If the metric value grows above some threshold, then routers restrict their routing policies to halt iBGP divergence.

VIII. CONCLUDING REMARKS

BGP interconnects all ASes in the Internet by advertising inter-AS routing information between them. Both external and internal BGP suffer from divergence anomalies. In this article, we discussed different types of divergence anomalies along with proposed solutions. All proposed solutions solve either eBGP divergence anomalies or iBGP divergence anomalies. A solution that solves both of these concurrently is still an open problem. As the size of the Internet grows, it will be a significant challenge to find a comprehensive BGP solution that is stable, efficient, and scalable.

REFERENCES

- [1] Y. Rekhter and T. Li, "A Border Gateway Protocol," *IETF RFC-1771*, 1995.
- [2] T. Bates and R. Chandrasekeran, "BGP route reflection - an alternative to full-mesh iBGP," *IETF RFC-1966*, 1996.
- [3] P. Traina, D. McPherson, and J. G. Scudder, "Autonomous system confederations for BGP," *IETF RFC-3065*, 2001.
- [4] C. Labovitz, G. R. Malan, and F. Jahanian, "Internet routing instability," in *Proc. of ACM SIGCOMM conference*, 1997, pp. 115–126.
- [5] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "Improving BGP convergence through consistency assertions," in *Proc. of IEEE INFOCOM conference*, vol. 2, 2002, pp. 902–911.
- [6] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Trans. Networking*, vol. 10, no. 2, pp. 232–243, 2002.
- [7] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route oscillations in iBGP with route reflection," in *Proc. of ACM SIGCOMM conference*, 2002, pp. 235–247.
- [8] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "Policy disputes in path vector protocols," in *Proc. of IEEE ICNP conference*, 1999, pp. 21–30.
- [9] T. G. Griffin and G. Wilfong, "An analysis of BGP convergence properties," in *Proc. of ACM SIGCOMM conference*, 1999, pp. 277–288.
- [10] R. Govindan, C. Alaettinoglu, G. Eddy, D. Kessens, S. Kumar, and W. S. Lee, "An architecture for stable, analyzable Internet routing," *IEEE Network*, vol. 13, no. 1, pp. 29–35, 1999.
- [11] L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 681–692, 2001.
- [12] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "A safe path vector protocol," in *Proc. of INFOCOM conference*, 2000, pp. 490–499.
- [13] J. A. Cobb, M. G. Gouda, and R. Musunuri, "A stabilizing solution to the stable paths problem," in *Proc. of Symp. on self-stabilizing systems, Springer-Verlag Lecture Notes in Computer Science*, vol. 2704, 2003, pp. 169–183.
- [14] J. A. Cobb and R. Musunuri, "Convergence of inter-domain routing," in *Proc. of IEEE GLOBECOM conference*, 2004, pp. 1353 – 1358.
- [15] J. J. Garcia-Lunes-Aceves, "Loop-free routing using diffusing computations," *IEEE/ACM Trans. Networking*, vol. 1, no. 1, pp. 130–141, 1993.
- [16] T. G. Griffin and G. Wilfong, "On the correctness of iBGP configuration," in *Proc. of ACM SIGCOMM conference*, 2002, pp. 17–29.
- [17] D. McPherson, V. Gill, D. Walton, and A. Retana, "Border Gateway Protocol (BGP) persistent route oscillation condition," *IETF RFC-3345*, 2002.
- [18] 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*, 2002.
- [19] R. Musunuri, , and J. A. Cobb, "Convergence of iBGP," in *Proc. of IEEE ICON Conference*, 2004.
- [20] Cisco Systems Inc., "Endless BGP convergence problem in Cisco IOS software releases," *Cisco Field Notice*, October 10 2000.

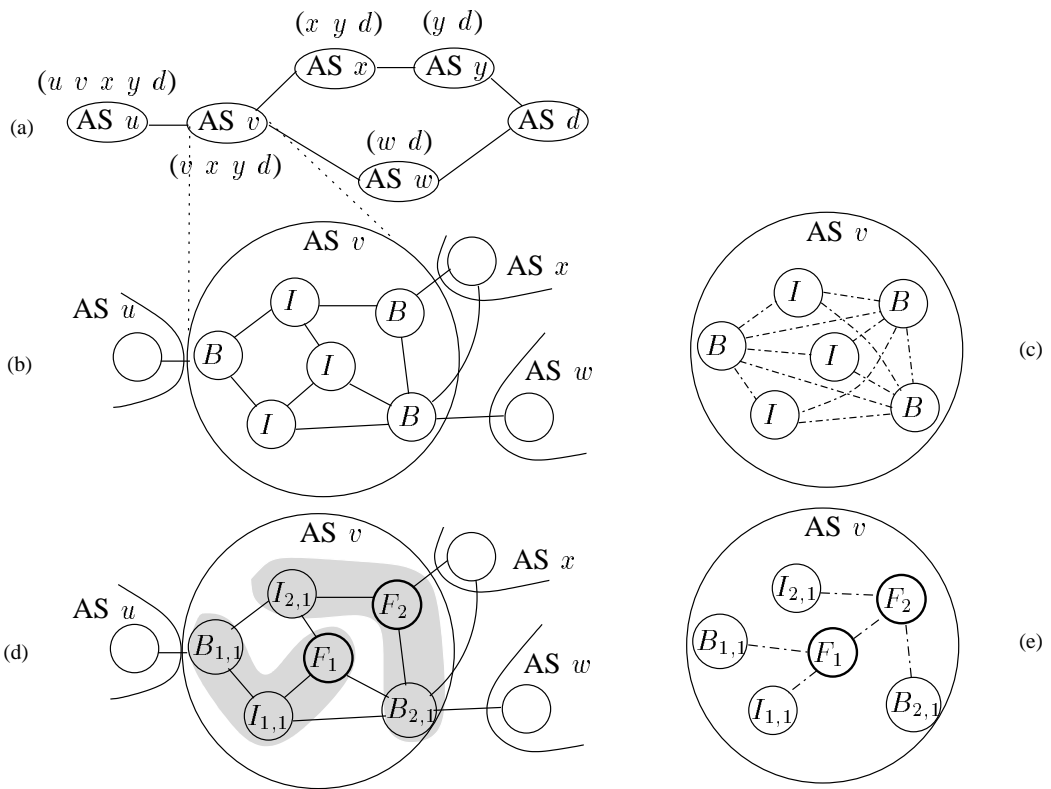


Fig. 1. a) Autonomous Systems Graph. b) AS-graph with Expanded $AS\ v$. c) Full-peering of $AS\ v$ d) $AS\ v$ with Clusters e) Peering of $AS\ v$ with Route-reflection.

```

best(input A: set of paths advertised by peers)
{
  1) A is reduced to only those paths with largest local_pref value.
  2) If  $|A| > 1$ , then reduce A to those paths with least AS_path sequence length.
  3) If  $|A| > 1$ , then separate A into disjoint subsets, where all paths in a subset exit via the same neighboring AS. Reduce each subset to those paths with smallest MED value. Set A to the union of the reduced subsets.
  4) If  $|A| > 1$ , then:
      a) If A has at least one path whose next_hop is an eBGP peer, then the router reduces A to those paths whose next_hop is an eBGP peer.
      b) If A has no paths whose next_hop is an eBGP peer, then the router reduces A to those paths whose intra-AS cost from itself to the path's border router is the least.
  5) Finally, if  $|A| > 1$ , then use some deterministic tie breaker (such as node identifier) to reduce A to a single element.
  6) The best path is the single element in A.
}

```

Fig. 2. Best Path Selection Algorithm

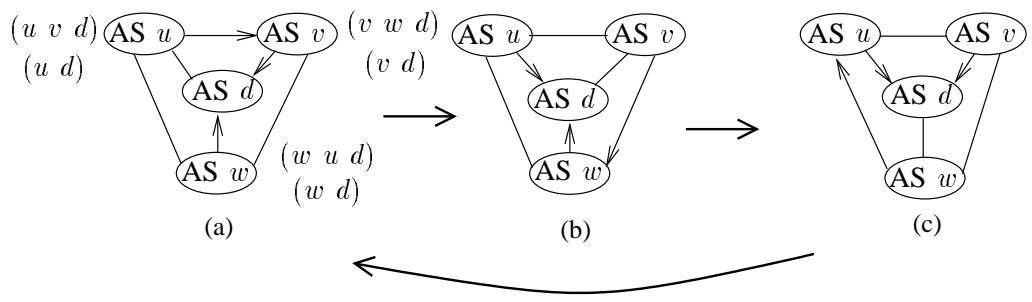


Fig. 3. eBGP Divergence Anomaly

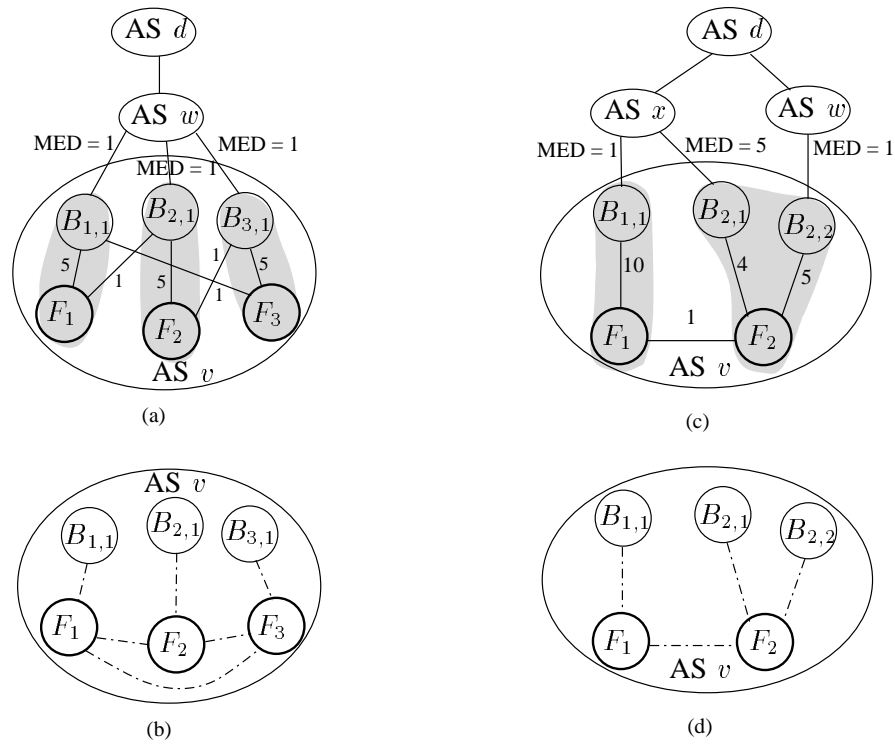


Fig. 4. iBGP Anomalies a) Clustering-induced Divergence Example b) iBGP Peering Sessions of Example (a) c) MED-induced Divergence Example d) iBGP Peering Sessions of Example (c)