Enforcing iBGP Convergence

Ravi Musunuri Jorge A. Cobb
Department of Computer Science
The University of Texas at Dallas
Richardson, TX-75083-0688
Email: {musunuri,cobb}@utdallas.edu

Abstract— **BGP routers within an Autonomous System (AS) exchange their inter-AS routing information via the internal Border Gateway Protocol (iBGP). Within an AS, every BGP router needs to maintain an iBGP peering session with every border BGP router. This peering scheme fails to scale due to the large number of iBGP peering sessions required. Current solutions to this scalability limitation divide the AS into clusters, with a distinguished router, know as the reflector, acting as a representative of the cluster. Clustering, however, introduces routing anomalies, such as permanent routing loops and failure to reach a stable route to the destination. Furthermore, these anomalies are worsened by the multi-exit discriminator value used by BGP to differentiate multiple links connecting the same pair of AS'ms. In this paper, we present a simple enhancement to iBGP that prevents these routing anomalies. It requires minimal overhead, and contrary to other proposed solutions, preserves the efficiency of iBGP by having each reflector disseminate only a single path to each of its peers.**

I. Introduction

The Internet, at its highest level, is divided into administrative domains, commonly known as Autonomous Systems (AS'ms). The Border Gateway Protocol (BGP) [1] is the de-facto protocol for sharing inter-AS routing information between neighboring BGP routers. Neighboring BGP routers in different AS'ms share their inter-AS routing information via the external Border Gateway Protocol (eBGP). On the other hand, any two BGP routers within the same AS, even if they are not physically neighbors, share their inter-AS routing information via the internal Border Gateway Protocol (iBGP).

BGP routers reliably exchange the routing information with each other via peering sessions. A peering session between two routers in different AS'ms is known as an eBGP peering session, and a peering session between two routers within the same AS is known as an iBGP peering session.

Both eBGP and iBGP have been plagued with forwarding and divergence anomalies. Forwarding anomalies consist of permanent loops in the routing tables, while divergence anomalies prevent the routers from converging to a stable selection of paths. eBGP suffers mainly from divergence anomalies, and these anomalies have been studied extensively. The reader is referred to [2], [3], [4], [5] for a discussion of the problem and proposed solutions.

iBGP, on the other hand, suffers from both forwarding and divergence anomalies. Two features of iBGP are the cause for these anomalies. First, iBGP employs route-reflection clustering [6] to improve its scalability, i.e., to reduce the number of iBGP peering sessions required. In route-reflection clustering, the AS is divided into clusters, with a distinguished router, known as the reflector, acting as a representative of the cluster. Although scalability is improved, clustering has caused forwarding and divergence anomalies [7], [8], [9]. Second, a multi-exit discriminator (MED) is a integer value used to differentiate multiple links connecting the same pair of AS'ms. The MED value, in combination with clustering, may cause divergence anomalies [9], [10], [11], [12], [13].

In this paper, we present a simple, yet effective, enhancement to iBGP that prevents routing anomalies. The overhead introduced by this modification is small. Furthermore, it does not restrict the behavior of the system unless a routing anomaly is occurring. That is, the system progresses normally according to its routing policies, unless divergence occurs. Furthermore, contrary to other proposed solutions [9], [12], [13], our approach preserves the efficiency of iBGP by having each reflector disseminate only a single path to each of its peers.

II. iBGP Overview

As mentioned above, the Internet is organized as a set of inter-connected AS'ms, as shown in Fig. 1(a). Here, each node denotes an AS, and neighboring AS'ms are joined by an edge. AS'ms x and y are said to be *neighbors* iff some router in x has a communication link with some router in y.

For a given destination AS, each AS informs its neighboring AS'ms of the path it has chosen to reach this destination. For example, consider d as the destination. AS x informs v that it reaches d via the path $\langle x, y, d\rangle$, and w informs v that it reaches d via the path $\langle w, d\rangle$. It is up to v to choose one of these two paths. Its choice is influenced by several factors, such as the length of the path. Most importantly, each AS has the freedom of choosing from the available paths the one with highest preference according to a routing policy defined locally within the AS. Thus, v may choose any of these two paths according to its routing policy.

Without loss of generality, throughout the paper, we will consider a single destination AS, namely, AS d.

An AS consists of multiple routers, as shown in Fig. 1(b). This figure expands AS v, showing its routers and the communication links between them. We say that two routers are *neighbors* iff they are joined by a communications link. 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

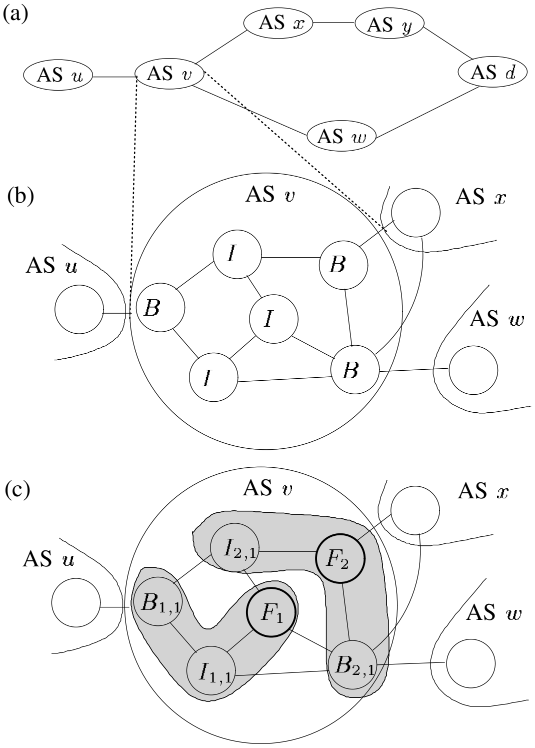


Fig. 1. Autonomous Systems and Clustering.

outside of its AS. In Fig. 1(b), internal routers are denoted by I, and border routers are denoted by B.

Two routers are said to be *peers* if they exchange routing information. In particular, each router chooses one path to the destination, and informs all its peers about the path it has chosen. Peering relationships are maintained via a reliable transport protocol, such as TCP. Note that routers need not be neighbors in order to be peers, i.e., routers may be located several network hops away from each other yet still maintain a peering relationship. This is possible because messages exchanged between peers are routed using a typical intra-AS routing protocol, such as OSPF [14] or RIP [15].

A. Route-Reflection Clustering

In a typical iBGP peering scheme, each border router within an AS 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 iBGP route reflection clustering [6]. In this approach, the routers within an AS are divided into disjoint sets, known as *clusters*. In Fig. 1(c), 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 cluster i is denoted F_i, and to highlight this node, it is drawn in bold. Border routers within cluster i are denoted by $B_{i,j}$ for some j, and likewise interior routers within cluster i are denoted by $I_{i,j}$ 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). 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(c), the clients of reflector F_2 are $I_{2,1}$, $B_{2,1}$.

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 again Fig. 1(c), in particular, border router $B_{2,1}$. Although it has a peering session with its neighbor in AS w and learns paths from it, the only router within its own AS v from whom it may learn a path is its reflector F_2. In particular, notice that even though $B_{2,1}$ is a neighbor of both F_1 and $I_{1,1}$, it does not establish a peering session with these routers.

B. Path Selection

To reach destination d, each router learns a path to d from each of its peers. If a router has no path to the destination, its path is said to be empty. The empty path is denoted by ϵ.

A path P chosen by a router in AS v consists of the following attributes:

- P_{pref} : An integer preference value indicating the ranking of P in the local routing policy of v. A higher preference value indicates a greater preference for the path.
- P_{AS} : Sequence of AS'ms traversed to reach the destination d from the current v.
- P_{MED} : In cases where there are multiple links connecting the same pair of AS'ms, each link is given a multiple-exit discriminator (MED) value. MED values indicate the preference of one link over another. A smaller MED value is preferred over a larger MED value.
- P_B : The IP address of the border router that is the exit point from v. Thus, this router has a neighbor in the first AS of the AS sequence P_{AS}.

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 [12], [16]. If a router adopts a new path, i.e., if its best path is different than before and new path is advertised by client peer or eBGP peer, the router informs each of its peers about this new path via the reliable transport protocol.

III. The Greedy Protocol

In this section, we present a formal abstraction of the behavior of the iBGP protocol with route reflection clustering. In particular, we reduce the problem of iBGP routing to an instance of the stable paths problem (SPP) [3], [4]. The SPP abstraction was developed to model eBGP routing. In [11], it was shown that iBGP may also be modeled as an instance of SPP. We take advantage of this to apply earlier SPP results [17] to iBGP routing, even though they were originally developed for eBGP routing.

First, we observe that an interior router is only able to choose the path given to it by its reflector, and does not affect

```
best($S$: path set from peers)
{
  1) $S$ is reduced to only those paths with best local preference value.
  2) If $|S| > 1$, then reduce $S$ to those paths with least $AS$ sequence length.
  3) If $|S| > 1$, then for every disjoint subset of $S$, such that all paths in the subset have the same next-hop AS, keep only those paths with least MED value.
  4) If $|S| > 1$, then:
     a) If router has a path whose border router is one of its eBGP peers, then the router reduces $S$ to those paths whose border router is an eBGP peer.
     b) The router reduces $S$ to those paths with least-cost-metric between the router and the border router.
  5) Finally if still $|S| > 1$, then use some deterministic tie breaker to reduce $S$ to a single element.
  6) The best path is the single element in $S$.
}
```

Fig. 2. Best Path Selection Algorithm

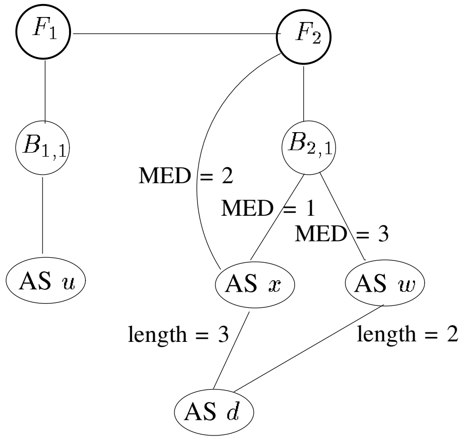


Fig. 3. Peer Graph Abstraction of AS v

```
router $R$
begin
    $\pi(R) \neq best(choices(R)) \rightarrow \pi(R) := best(choices(R))$
end
```

Fig. 4. Greedy Protocol

the selection of paths of other routers. Next, we build a *peer graph*, which is an abstraction of the peer relationship between routers. The peer graph of AS v in Fig. 1(c) is given in Fig. 3. Notice that, interior routers are removed from peer graph.

Each edge in the peer graph corresponds to a peer relationship between routers. In addition to the reflector and border routers, the graph also contains nodes that represent AS'ms. The AS'ms in the graph are the neighboring AS'ms of v and the destination AS d.

Each edge between a border router and a neighboring AS is assigned a MED value, as shown in Fig. 1. In addition, given that the scope of our paper is focused on iBGP and not on eBGP, we assume that each path from a neighboring AS to d is stable. Therefore, rather than include the entire AS path in the graph, we simply represent it by an edge from the neighboring AS to d. This edge is labeled with the length of the AS path represented by the edge. Finally, if the next hop of the neighboring AS is v itself, then the edge to d is omitted.

Each node chooses a path to d along the peer graph. But datagrams are forwarded along intra-AS shortest path between two routers in the same AS. The path chosen by node R is denoted by $\pi(R)$, and has the following properties.

- At all times, $\pi(R)$ must be either a simple path from R to d or the empty path.
- $\pi(R)$ must be *consistent* with its next node. That is, if the next node along $\pi(R)$ is B, then either $\pi(R) = R;\pi(B)$, or $\pi(R)$ is outdated and node R must update it.
- For every node u representing a neighboring AS, $\pi(u)$ is fixed, and either $\pi(u) = \langle u, d\rangle$, or $\pi(u) = \epsilon$.

As mentioned earlier, each router R will receive one path from each of its peers. Therefore, the set of paths from which R may choose its own path to d is as follows.

$$choices(R) = \{\langle R;\pi(R')\rangle \mid R' \in peers(R) \wedge R \notin \pi(R')\}$$

We may now formally define the behavior of a router R. The behavior is quite simple, and is shown in Fig. 4 using a notation similar to that in [18], [19]. Router R contains one *action*, which consists of the guard $\pi(R) \neq best(choices(R))$ and the statement $\pi(R) := best(choices(R))$. Periodically, the guard of the action is checked. If it is true, then the statement is executed. Hence, router R ensures that if $\pi(R) \neq best(choices(R))$, then eventually $\pi(R) = best(choices(R))$. We refer to this protocol as the *greedy protocol*, since it always chooses the best path.

In order for the above to be an instance of the SPP, we require a relation $\preceq$ that ranks paths at each router in order of preference. We may define $P \preceq Q$, where both paths P and Q originate at router R, to be as follows.

$$\begin{aligned} P \preceq Q &\equiv (Q = best(\{P, Q\}) \\ P \prec Q &\equiv (P \preceq Q \wedge P \neq Q) \end{aligned}$$

I.e., the router prefers Q over P when these are its only available choices. We require $\preceq$ to be a total order on paths. However, as defined above, this is not the case, due to a conflict between MED values and link-costs. This will be explored and remedied in Section VI.

The greedy nature of the protocol in Fig. 4 causes some well-known routing anomalies in iBGP. These anomalies are described in the next few sections. In addition, the greedy protocol is strengthened to detect these anomalies and to compensate for them.

In our examples below, we assume all paths from neighboring AS'ms have the same local preference value and the same AS sequence length. Hence, the chosen path at a router depends mainly on the MED value and on the link-cost to reach the border router of the path.

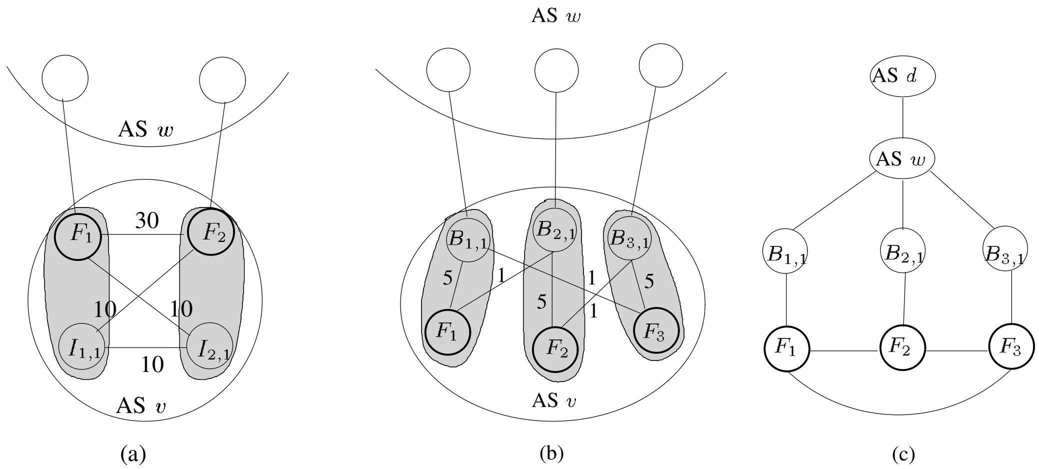


Fig. 5. Cost-Induced Anomalies.

IV. CIF Anomaly

One routing anomaly caused by clustering is a cost-induced routing loop [7], [8]. It is caused by the interaction of clustering and the intra-AS routing algorithm (such as OSPF or RIP). This anomaly is not the focus of this paper, but it is included for completeness.

Consider Fig. 5(a) [8], which shows an AS v, whose neighbor to reach the destination is AS w. AS v has two clusters, each with a reflector (which is also a border router) and an internal router. The edges correspond to network links and are labeled with their intra-AS routing link-cost.

The internal router learns its path from its reflector, and each reflector chooses the path via its external peer. Thus, $I_{1,1}$ routes its data messages via F_1, and $I_{2,1}$ routes its data messages via F_2. However, due to the costs assigned to each link, the path from $I_{1,1}$ to F_1 is $\langle I_{1,1}, I_{2,1}, F_1\rangle$, and the path from $I_{2,1}$ to F_2 is $\langle I_{2,1}, I_{1,1}, F_2\rangle$. Hence, there is a routing loop between $I_{1,1}$ and $I_{2,1}$.

Cost-induced routing loops can be avoided if each reflector selectively advertises paths, which has been presented in [9].

V. Cost-Induced Divergence

We next consider an anomaly in which routers fail to converge to a stable assignments of paths [7]. We refer to this anomaly as cost-induced divergence, because the interaction between iBGP and the link-costs of the intra-domain routing protocol causes the system to diverge.

An example of cost-induced divergence is shown in Fig. 5 [7]. Fig. 5(b) shows the routers and the links joining them. Fig. 5(c) shows the peer-graph of Fig. 5(b). Note that in the peer graph, each reflector F_i always prefers path $\langle F_i, F_{i+1}, B_{(i+1,1)}, w, d\rangle$ over path $\langle F_i, B_{i,1}, w, d\rangle$ due to following[1]:

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

Initially, assume that for each i, $\pi(F_i) = \langle F_i, B_{i,1}, w, d\rangle$. Consider the following sequence of events.

1) F_1 changes its path to $\pi(F_1) = \langle F_1, F_2, B_{2,1}, w, d\rangle$ because the path via F_2 is ranked higher than its current path via $B_{1,1}$.
2) F_2 changes its path to $\pi(F_2) = \langle F_2, F_3, B_{3,1}, w, d\rangle$ because the path via F_3 is ranked higher than its current path via $B_{2,1}$.
3) F_1 returns its path to $\pi(F_1) = \langle F_1, B_{1,1}, w, d\rangle$, because its previous path via F_2 is no longer available.
4) F_3 changes its path to $\pi(F_3) = \langle F_3, F_1, B_{1,1}, w, d\rangle$, because the path via F_1 is ranked higher than its current path via $B_{3,1}$.
5) F_2 returns its path to $\pi(F_2) = \langle F_2, B_{2,1}, w, d\rangle$, because its previous path via F_3 is no longer available.
6) F_1 changes its path to $\pi(F_1) = \langle F_1, F_2, B_{2,1}, w, d\rangle$ because the path via F_2 is ranked higher than its current path via $B_{1,1}$.
7) F_3 returns its path to $\pi(F_3) = \langle F_3, B_{3,1}, w, d\rangle$, because its previous path via F_1 is no longer available.

The state of the system after step 7 is the same as the state after step 1. The system will therefore never reach a steady assignments of paths.

A. Bounded Divergence Protocol

Throughout the remainder of this section, we assume that all paths have an equal MED value. Under this assumption, the path-ranking relation $\preceq$ becomes a total order, and thus, the peer-graph in combination with relation $\preceq$ becomes an SPP instance. This allows us to use the eBGP techniques developed in [17] to ensure the convergence of iBGP.

Given that $\preceq$ defines a total order, consider again Fig. 5(b) and its cyclic sequence of steps. Observe that the rank of $\pi(F_1)$ is periodically decreased. In particular, the rank of $\pi(F_1)$ decreases in step 3. Similarly, the rank of $\pi(F_i)$ for each i decreases periodically. Intuitively, no divergence is possible if every router monotonically increases the rank of its path, because, eventually, the node would reach and keep its highest ranking path. Therefore, during divergence, the rank of the path of diverging nodes must periodically decrease.

We use the above observation to allow nodes to infer that divergence is occurring. In particular, each node is assigned

[1] Note that mod 3 is implied on the subscript i

```
**router** $R$
**begin**
    $\pi(R) = \langle R, \pi(peer(R)) \rangle \rightarrow$
        $count(R) := max(count(R), count(peer(\pi(R))))$
[]
    $\pi(R) \succ best(choices(R)) \ \wedge$
    $peer(\pi(R)) \neq peer(best(choices(R))) \rightarrow$
        $\pi(R) := best(choices(R));$
        $count(R) := count(R) + 1$
[]
    $\pi(R) \succ best(choices(R)) \ \wedge$
    $peer(\pi(u)) = peer(best(choices(R))) \rightarrow$
        $\pi(R) := best(choices(R));$
        $count(R) := count(peer(\pi(R)))$
[]
    $\pi(R) \prec best(choices(R)) \ \wedge$
    $(count(peer(best(choices(R))) < C \vee \pi(R) = \epsilon) \rightarrow$
        $\pi(R) := best(choices(R));$
        $count(R) := count(peer(\pi(R)))$
**end**
```

Fig. 6. Bounded Divergence Protocol

an integer count. Whenever the new path of a node has a lower rank than its previous path, the count of the node is increased by one. In addition to the path of its peers, a node may read the count of its peers.[2] As the count increases beyond a threshold, a node infers that divergence is occurring, and it takes remedial action by restricting its choice of paths, and thus ensuring convergence.

The specification of the Bounded Divergence Protocol [17] is shown in Fig. 6. Each router R consists of four actions. The first action simply enforces that the cost of a node R is never smaller than the cost of its next node along its path to d.

The second action is enabled when the best path for R has a rank lower than the current path $\pi(R)$ and peer of both chosen and best paths are not equal. In this case, $\pi(R)$ is updated to the best path. However, since the rank of $\pi(R)$ decreases, the count of R is increased by one to detect divergence.

The third action is enabled when the best path for R has a rank lower than the current path $\pi(R)$ and peer of both chosen and best paths are equal. In this case, $\pi(R)$ is updated to the best path. However, the count of R assigned to count of peer along the best path.

The fourth action updates $\pi(R)$ when the best path for R has a rank higher than the current path $\pi(R)$. One way in which the R's count could be updated is simply to set it to the maximum of its current value and the count of the peer from whom the path is taken (i.e., the count of the next node along the path in the peer graph). In this way, R's count is guaranteed not to decrease, and hence, in diverging executions, such as those in Fig. 5(b), R's count is guaranteed to increase. However, we would like to keep counts as small as possible. This is because a large count indicates divergence, and when this occurs the paths available at a node are restricted (as explained below).

We would like the count of R to decrease in the event that an alternative path is found that does not lead R to diverge. E.g., consider Fig. 5(b), and assume that F_1 has an additional link to a neighboring AS x, and AS x offers a shorter path, in terms of number of AS'ms crossed, to reach destination d, and hence, AS x is more desirable than AS w. We would expect then for all routers to choose the path via x before their count increases significantly.

However, if the path information from x is slow to arrive, the count at the routers in v may grow large. Nonetheless, once the information from x arrives to F_1, we would like the counts of all nodes to decrease, and not hinder the choice of paths at each node. To ensure this, the fourth action in Fig. 6 assigns to the count of R the same count as the count of the peer router that offered the new path. In this manner, if the cost of its peer is low, then the cost of R itself will be low.

Divergence is actually prevented in the guard of the fourth action. If the new path, even if ranked higher than the current path, is from a peer whose cost has reached a threshold C, then the new path is not chosen. In this way, the chosen path stops from oscillating. The exception to this is when $\pi(R)$ is the empty path. In this case, since R is required to maintain a path to d, the best path is chosen irrespective of the count of the peer.

The above restrictions ensure that the system reaches a steady state, as indicated below.

Theorem 1: Starting from any arbitrary system state (i.e., an arbitrary value of π and node counts), and assuming all paths have equal MED values, the bounded-divergence protocol converges to a stable state within a finite number of steps.

VI. MED-Induced Divergence

The interaction between the MED value of a path and the link-costs within the AS may cause a divergence anomaly, i.e., routers fail to obtain a stable path to the destination. In this section, we present an example of this anomaly, and show how the bounded divergence protocol can be used to resolve it.

Consider the example in Fig. 7(a), which was originally presented in [10]. It consists of an AS v, and two neighboring AS'ms w and x. AS v is divided into two clusters. The network links within v are labeled with the cost of the intra-domain routing protocol. The links from a border router to a neighboring AS are labeled with the MED value of the link.

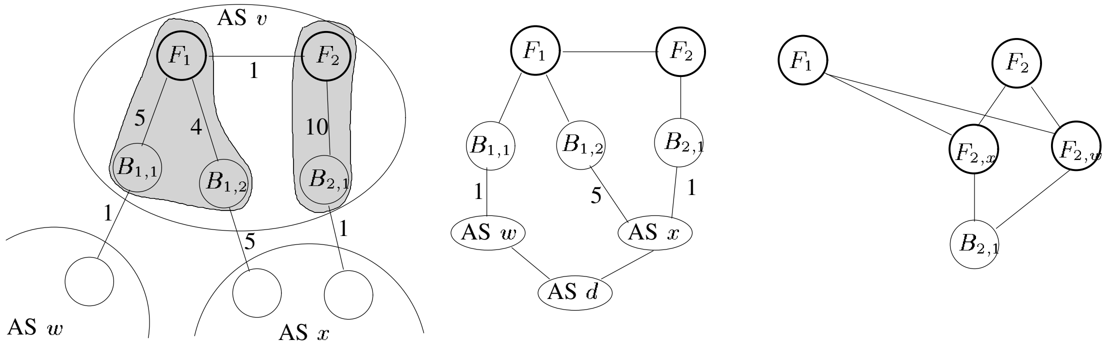
The peer-graph of AS v is shown in Fig. 7(b). We assume both x and w have an equal number of AS'ms in their paths to d. Therefore, the border routers will always choose a path via their peers in the neighboring AS. Thus, we consider only the paths taken by the reflectors. For terseness, we abbreviate each path by removing the AS nodes. For example, path $\langle F_1, F_2, B_{2,1}, x, d \rangle$ will be denoted by $\langle F_1, F_2, B_{2,1} \rangle$.

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

1) Initially,

$$\pi(F_1) = \langle F_1, B_{1,2} \rangle \wedge \pi(F_2) = \langle F_2, B_{2,1} \rangle$$

[2] This would be implemented in message passing by including the count of the node in every iBGP path-update message.



(a)

(b)

(c)

Fig. 7. MED-Induced Divergence.

That is, each reflector chooses the best path from those provided by the border routers in their clusters.

2) The choices of F_1 are $\{\langle F_1, B_{1,1}\rangle$, $\langle F_1, B_{1,2}\rangle$, $\langle F_1, F_2, B_{2,1}\rangle\}$. Applying function $best$ yields $\langle F_1, B_{1,1}\rangle^3$. Thus,

$$\pi(F_1) = \langle F_1, B_{1,1}\rangle \wedge \pi(F_2) = \langle F_2, B_{2,1}\rangle$$

3) The choices of F_2 are $\{\langle F_2, B_{2,1}\rangle, \langle F_2, F_1, B_{1,1}\rangle\}$. The path via F_1 has lower link cost than its path via $B_{2,1}$ Hence,

$$\pi(F_1) = \langle F_1, B_{1,1}\rangle \wedge \pi(F_2) = \langle F_2, F_1, B_{1,1}\rangle$$

4) The choices of F_1 are now only $\{\langle F_1, B_{1,1}\rangle, \langle F_1, B_{1,2}\rangle\}$, because the path of F_2 is via F_1. Thus, due to the link costs of $B_{1,1}$ and $B_{1,2}$,

$$\pi(F_1) = \langle F_1, B_{1,2}\rangle \wedge \pi(F_2) = \langle F_2, F_1, B_{1,1}\rangle$$

5) The choices of F_2 are $\{\langle F_2, B_{2,1}\rangle, \langle F_2, F_1, B_{1,2}\rangle\}$. The MED value of $\langle F_2, B_{2,1}\rangle$ is better than the MED of $\langle F_2, F_1, B_{1,2}\rangle$. Hence,

$$\pi(F_1) = \langle F_1, B_{1,2}\rangle \wedge \pi(F_2) = \langle F_2, B_{2,1}\rangle$$

This is the same state as the initial state.

In this scenario, the path ranking relation $\preceq$ at router F_1 is not a total order. In particular,

$$\langle F_1, F_2, B_{2,1}\rangle \preceq \langle F_1, B_{1,1}\rangle \preceq \langle F_1, B_{1,2}\rangle \preceq \langle F_1, F_2, B_{2,1}\rangle$$

To see this, consider first $\langle F_1, B_{1,2}\rangle \preceq \langle F_1, F_2, B_{2,1}\rangle$. This holds because of the MED values, since both paths exit via AS x. Next, consider $\langle F_1, B_{1,1}\rangle \preceq \langle F_1, B_{1,2}\rangle$. This holds because of link-costs, since their MED value is not used in their comparison. Finally, consider now $\langle F_1, F_2, B_{2,1}\rangle \preceq \langle F_1, B_{1,1}\rangle$. This holds again because of link-costs, since their MED value is not used in their comparison.

The correctness of the BDP [17] depends on relation $\preceq$ being a total order. Therefore, it cannot be applied directly to the peer-graph in Fig. 7(b). In the next section, we enhance the peer-graph such that $\preceq$ becomes a total order, and thus, we may apply the BDP to detect and terminate the above MED-induced divergence.

A. Applying BDP via Virtual Nodes

To ensure that relation $\preceq$ is total relation at each node, we add *virtual nodes* to the peering graph. These virtual nodes are similar to the class nodes introduced in [11]. For each router R, (R could be a reflector or a border router), and for each neighboring AS x, we introduce the virtual node R_x. This node co-exists in the peering graph along with the original node R.

The peers of R are restricted to be its virtual nodes R_x for every x. The peers of R_x are the original node R, plus any other node S that was a peer of R in the original peer graph.

The purpose of each virtual node R_x is to find the best path that exits via neighboring AS x. I.e., R_x will always offer to R a path that goes through one of its peers in the original peer graph and that exits via AS x. R then simply chooses, among the paths given to it by its virtual nodes, the one with highest rank.

For example, Fig. 7(c) shows the virtual nodes, $F_{2,x}$ and $F_{2,w}$, associated with the original node F_2. $F_{2,x}$ will choose the best path that exits via AS x. This path may occur via the original peer F_1 or via the original peer $B_{2,1}$. Similarly, $F_{2,w}$ will choose the best path that exits via AS w. This path may only occur via the original peer F_1, since the original peer $B_{2,1}$ has no path to w, and thus, the peering edge $(F_{2,w}, B_{2,1})$ could be removed. We include it in the figure simply for completeness.

Due to space limitations, the complete peering graph with virtual nodes is not shown. However, it must be noted that the remaining routers, i.e., F_1, $B_{1,1}$, $B_{1,2}$, and $B_{2,1}$, must also be expanded with virtual nodes of their own.

We next present the new relation $\preceq$ on paths. We must ensure that $\preceq$ is a total order at each node. Furthermore, we must also ensure that, for each router R, $best(choices(R))$ yields the same path in the original peering graph as in the peering graph with virtual nodes.

Consider first relation $\preceq$ on paths starting from a virtual node R_x. This relation has the following properties.

- For any path P originating at R_x, $P \prec \epsilon$ if and only if the next node after R_x is not a peer of R in the original peer graph or P does not exit via AS x. Note that if $P \prec \epsilon$ then P will never be chosen since the it is ranked below the empty path.

[3] $\langle F_1, F_2, B_{2,1}\rangle$ eliminates $\langle F_1, B_{1,2}\rangle$ because it has a lower MED, and in the next step $\langle F_1, B_{1,1}\rangle$ has lower link-cost than $\langle F_1, F_2, B_{2,1}\rangle$.

- For every pair of paths P and Q, where P and Q originate at R_x, and both are higher ranked than ϵ, we have

$$P \preceq Q \equiv (Q = best(\{P, Q\})$$

Note that $\preceq$ above is a total order, since $best$ is only used to compare paths exiting via the same AS, and hence, the MED value does not cause a cycle in relation $\prec$.

Consider now relation $\preceq$ on paths starting from an original router node R. This relation has the following properties.

- For every path P originating at R, $P \prec \epsilon$ if and only if the next node in P is not a virtual node R_x of R for some x
- For every pair of paths P and Q, where P and Q originate at R, and both are higher ranked than ϵ, we have

$$P \preceq Q \equiv (Q = best(\{P, Q\})$$

Note that $\preceq$ above is again a total order, since $best$ is only used to compare paths exiting via the different AS'ms, and hence, the MED value does not cause a cycle in relation $\prec$.

Since $\preceq$ is a total order, we have the following [17].

Theorem 2: Starting from any arbitrary system state (i.e., an arbitrary value of π and node counts), the bounded-divergence protocol converges to a stable state within a finite number of steps in the peering graph with virtual nodes.

Introducing virtual nodes does not add message overhead, because R_x is implemented in the same router as R, and hence, their communication is internal to the router. Furthermore, no new messages in practice are needed between routers. This is because, R_x needs to learn about paths from any original peers S of R, but since R_x is also located in R, in practice R receives this information anyway. Furthermore, R_x need not send its chosen path to S, since S only allows paths from its virtual nodes of the form S_y. Regarding storage overhead at router R, R needs to store $n + 1$ integers for each destination d, where n is the number of neighboring AS'ms of R. Since n is usually small, this adds little storage overhead.

VII. Related Work

Griffin et al. formally defined iBGP anomalies [7], [11]. Our solution requires only single path dissemination between every pair of iBGP peers. Furthermore, the existing routing policies need not be modified, as opposed to others that have suggested removing MED altogether.

There are two types of iBGP divergence solutions in the current literature. Both types require multiple path disseminations between iBGP peers.

The first type of solutions require multiple path disseminations between both pair of reflectors and reflector and client peers. In Walton et al. [13] solution, a reflector finds one best path through each of the neighboring AS and advertises this per AS best path if this path's local preference and AS sequence length values are equal to the reflector's overall best path's corresponding attributes. Basu et al. [12] have showed a counter-example to Walton's solution. They proposed a new solution, in which, the reflector advertises all the paths with best local preference, smallest AS sequence length, and smallest MED for each neighboring AS. They also proved the correctness of their solution. But this type of solutions, in which, multiple path advertisements are required between every pair of iBGP peers, may not be scalable. This defeats the whole purpose of using clustering.

Second type of solution only requires multiple path disseminations between pair of reflectors, but not between reflector and client peers. In [9], authors proposed a selective path dissemination between reflector and client peers.

VIII. Concluding Remarks

In this paper, we presented a simple and scalable solution to solve all the known iBGP anomalies. Our solution only requires single path disseminations between every pair of iBGP peers.

Our protocols are based on shared memory model, which is a special case of message passing model. But these protocols can easily be modified to more general message passing model. Due to space limitations, we did not provide the formal rigorous proofs to the solutions.

References

[1] Y. Rekhter and T. Li, "A border gateway protocol," *IETF RFC-1771*, 1995.

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

[3] 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.

[4] J. A. Cobb, M. G. Gouda, and R. Musunuri, "A stabilizing solution to the stable paths problem," in *Symp. on Self-Stabilizing Sys., Springer-Verlag Lecture Notes in Comp. Sci.*, vol. 2704, 2003, pp. 169–183.

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

[6] T. Bates and R. Chandrasekeran, "BGP route reflection - an alternative to full-mesh IBGP," *IETF RFC-1966*, 1996.

[7] T. G. Griffin and G. Wilfong, "On the correctness of IBGP configuration," in *Proc. of ACM SIGCOMM conference*, 2002, pp. 17–29.

[8] R. Dube, "A comparison of scaling techniques for BGP," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 3, pp. 44–46, 1999.

[9] R. Musunuri, , and J. A. Cobb, "Complete solution to IBGP stability," in *Proc. of IEEE ICC conference*, 2004.

[10] D. McPherson, V. Gill, D. Walton, and A. Retana, "Border gateway protocol persistent route oscillation condition," *IETF RFC-3345*, 2002.

[11] T. G. Griffin and G. Wilfong, "Analysis of the MED oscillation problem in BGP," in *Proc. of IEEE ICNP conference*, 2002, pp. 90–99.

[12] 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.

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

[14] J. Moy, "OSPF version 2," *IETF RFC-2328*, 1998.

[15] G. S. Malkin, "RIP version 2," *IETF RFC-2453*, 1998.

[16] S. Halabi, *Internet Routing Architectures*. Cisco Systems, 2000.

[17] J. A. Cobb and R. Musunuri, "Covergence of inter-domain routing," in *Proc. of IEEE Globecom Conference*, 2004.

[18] M. G. Gouda, *Elements of Network Protocol Design*. John Wiley& Sons, 1998.

[19] ——, "Protocol verification made simple: A tutorial," *Comput. Netw. ISDN Syst.*, vol. 25, no. 9, pp. 969–980, 1993.