Comprehensive Solution for Anomaly-free BGP

Ravi Musunuri, Jorge A. Cobb

Department of Computer Science, The University of Texas at Dallas,
Richardson, TX-75083-0688
{musunuri, cobb}@utdallas.edu

Abstract. The Internet consists of many self-administered and inter-connected Autonomous Systems (ASms). ASms exchange inter-AS routing information with each other via the Border Gateway Protocol (BGP). Neighboring BGP routers located in different ASms share their inter-AS routing information via external BGP (eBGP), whereas two routers in the same AS share their inter-AS routing information via internal BGP (iBGP).
From the paths received from its peers, each BGP router chooses the best path based on routing policies chosen locally at its own AS. Conflicting policies between different ASms may cause divergence problems in eBGP, i.e., permanent oscillations in the chosen path to the destination. On the other hand, divergence problems may also occur in iBGP. This is caused by the interaction of route-reflection clustering, which is a technique to improve the scalability of iBGP, and other factors, such as intra-AS link costs, among others. In this paper, we provide a comprehensive solution that avoids all the known anomalies with both eBGP and iBGP. In our solution, each AS can locally choose its routing policies, while still ensuring anomaly-free behavior.

1 Introduction

The Internet consists of many self-administered and inter-connected Autonomous Systems (ASms). Routing in the Internet is separated into intra-AS routing and inter-AS routing. Intra-AS routing (e.g., OSPF, RIP) advertises routing information that is local to the AS to all routers within the same AS. The Border Gateway Protocol (BGP) [1] advertises inter-AS routing information between BGP routers. Each pair of neighboring BGP routers in different ASms share their inter-AS routing information via external BGP (eBGP). Each pair of BGP routers within the same AS share their inter-AS routing information via internal BGP (iBGP). Contrary to eBGP, sharing routing information in iBGP is done even if the pair of routers are not neighbors, i.e., even if they are separated by multiple network hops.

BGP routers exchange inter-AS routing information via a TCP connection with each of its BGP peers. If a peer is located in a different AS, it is known as an eBGP peer, and the TCP connection to this peer is referred as an eBGP peering session. Similarly, a peer in the same AS is known as an iBGP peer, and the TCP connection to it is referred as an iBGP peering session.

From the set of paths advertised by its peers, each router chooses the best path based on the routing policies chosen locally at its AS. Conflicting routing policies [2]

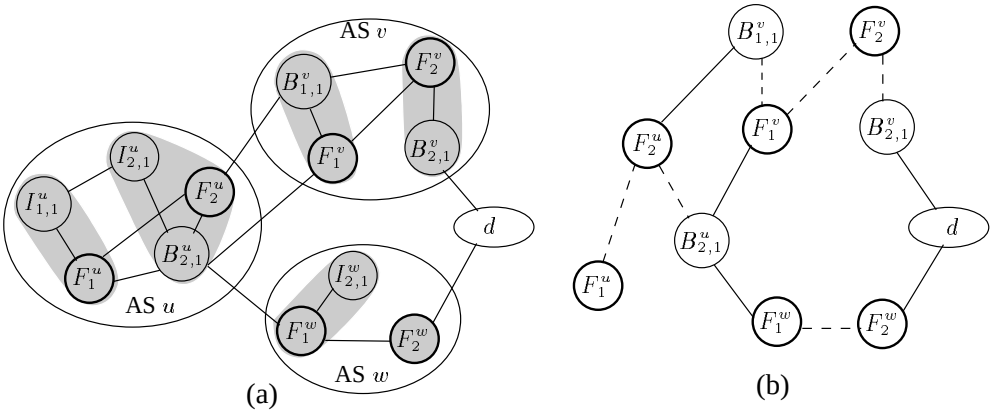

Fig. 1. a) Autonomous Systems Example b) Peering Graph

between different ASms may cause divergence in eBGP, i.e., the path to the destination continuously oscillates between several possible paths. Divergence may also occur in iBGP, even when eBGP is stable. This is caused by the interaction of route-reflection clustering [3], which is a technique to improve the scalability of iBGP, and other factors, such as intra-AS link costs, and MED values [1].

Many solutions have been proposed to avoid eBGP and iBGP divergence anomalies separately. However, we are not aware of any solution that solves divergence anomalies in both eBGP and iBGP combined. Govindan et al. [4] proposed an architecture to analyze the routing policies statically and find conflicting routing policies. On the other hand, Griffin et al. [5] have shown that checking routing policies for divergence is intractable. Gao et al. [6] proposed a set of guidelines for choosing routing policies in order to avoid eBGP divergence. However, their solution removes the freedom of each AS to choose its routing policies locally. Basu et al. [7] and Walton et al. [8] provided solutions to solve iBGP divergence anomalies. For a given destination prefix, in the original iBGP, each router only advertises a single best path to its iBGP peers. In both of these solutions, for a given destination, routers are required to advertise multiple paths to its iBGP peers, requiring higher memory and message overheads, and thus defeating the purpose of using route-reflection clustering.

In this paper, we are providing a comprehensive solution that solves all the known anomalies with iBGP and eBGP. In our solution, BGP path update message carries only two additional integer cost metric values. One cost metric is used to detect and avoid the eBGP divergence anomalies and other cost metric is used to detect and avoid iBGP divergence. For a given destination prefix, our solution does not require multiple path advertisements between iBGP peers. Also, each AS can choose routing policies locally. Our solution restricts the routing policies, only when, there exists a divergence.

[1] The Multi-Exit-Discriminator (MED) value is used to define the preference level of inter-AS links when the pair of neighboring ASms are connected by more than one inter-AS link.

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 to reduce A to a single element.
6. The best path is the single element in A.

}

Fig. 2. Best Path Selection Algorithm

2 BGP Path Selection

In this paper, we assume that each router tries to find a path to some special destination prefix d. A path P received by a router R located in AS u contains the following attributes:

- $local\_pref$: A preference value indicating the ranking of P in the local routing policy of AS u. A larger preference value indicates a greater preference for the path.
- AS_path : Sequence of ASms along the path to reach the destination prefix d from the current AS u.
- MED : For a pair of ASms 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. If the router R is an interior router then $next\_hop$ is the IP address of the border router that is the exit point from u. If the router R is a border router then $next\_hop$ is the IP address of the border router that is the entry point into the neighboring 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 [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.

node x
begin
$\pi(x) \neq best(choices(x)) \rightarrow \pi(x) := best(choices(x))$
end

Fig. 3. Greedy Protocol

3 Route-Reflection Clustering

In the original iBGP peering scheme, each border router within an AS is a iBGP peer of all other routers within the same AS. As the size of the AS increases, this scheme fails to scale due to large number of iBGP peering sessions required. A common solution is to employ route-reflection clustering [3]. In this approach, the routers within an AS are divided into disjoint sets, known as *clusters*. In Fig. 1(a), AS u is divided into two clusters depicted by the shaded regions. One distinguished router in each cluster is known as the *reflector*. The reflector within AS u and cluster i is denoted F_i^u, and to highlight this node, it is drawn in bold. Border routers within AS u and cluster i are denoted by $B_{i,j}^u$ for some j, and likewise interior routers within AS u and cluster i are denoted by $I_{i,j}^u$ 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 iBGP peering session with a reflector are known as the *clients* of the reflector. For example, in Fig. 1(a), the clients of reflector F_2^u are $I_{2,1}^u$ and $B_{2,1}^u$.

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

Reflector, F_i^u, advertises its best path to other peers as explained below:

- If F_i^u received its best path from another reflector, then F_i^u advertises its best path to all its clients and eBGP peers.
- If F_i^u received its best path from a client or from an eBGP peer, then F_i^u 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).

4 Greedy Protocol

In this section, we will reduce the BGP routing problem into an abstract and formal notation known as the Stable Paths Problem (SPP). The SPP was originally introduced

by Griffin et.al. [5] to model eBGP routing. However, in [10] and [12] it was shown that the SPP model can be extended to model iBGP routing. In this paper, we will extend the SPP model to model eBGP and iBGP combined.

An SPP instance consists of a tuple $(G, \prec)$, where G is a graph and $\prec$ is a ranking relation between the paths along the graph G.

For our purposes, we restrict G as follows. Each node in G corresponds to either a border router or a reflector router and each edge corresponds to a peering session between two routers. Notice that interior and non-reflector routers are removed from the peering graph. In general, interior and non-reflector routers do not effect the path selection; they only choose the path advertised by their reflector. Figure 1(b) presents the peering graph of the example shown in Fig. 1(a). eBGP peering sessions are shown as solid lines and iBGP peering sessions are shown as dotted lines.

Next, we define $\prec$, which is a ranking relation between the paths at a node along the peering graph G. We define $P \prec Q$ at node x, where both paths P and Q originate at node x and end at node d, as follows.

$$P \prec Q \equiv ((Q = best(\{P, Q\}) \wedge P \neq Q)$$

I.e., router x prefers Q over P when these are its only available choices.

We require relation $\prec$ to be a total-order on paths. However, a total-order is not guaranteed if the best-path selection algorithm uses MED values. Until section 7, we will ignore the MED values for path selection. In section 8, we will briefly discuss incorporating MED values into our approach. In this paper, we will use eBGP results from [11] and iBGP results from [12] to provide a comprehensive solution.

Every node x chooses a path to d among the paths offered by its neighbors in the peering graph. The path currently chosen by x is denoted by $\pi(x)$. This path is a sequence of nodes represented as $\langle x\ y \ldots d \rangle$, and its value is updated under the following constraints.

- At all times, $\pi(x)$ should be a loop-free path or the empty path.
- Node x can update $\pi(x)$ only by assigning to it the path $\langle x\ \pi(y) \rangle$ for some neighbor y in the peering graph.

Note that the path actually taken by datagrams as they traverse an AS may be different from the chosen paths above. E.g., if $\pi(x)$ is equal to $\langle x \ldots u \ldots v \ldots d \rangle$, where all the nodes in the sub-path $\langle u \ldots v \rangle$ belong to the same AS, the actual path taken by datagrams is the shortest intra-AS path between routers u and v, which may be different from the sub-path from u to v in $\pi(x)$.

Every node receives at most one path from each its peers. The set of paths advertised by all the peers of node x is denoted by $choices(x)$.

Next, we will present the greedy protocol, which simulates the working of BGP protocol with route-reflection. Specification of the greedy protocol at node x is shown in Fig. 3. The notation used in this paper is similar to the notation defined in [13], [14]. The greedy protocol consists of one action with guard $\pi(x) \neq best(choices(x))$. If the guard is true, i.e., if the current chosen path is different from the best available path, then node x greedily assigns to $\pi(x)$ the path $best(choices(x))$.

In the next two sections, we present two anomalies associated with the greedy protocol.

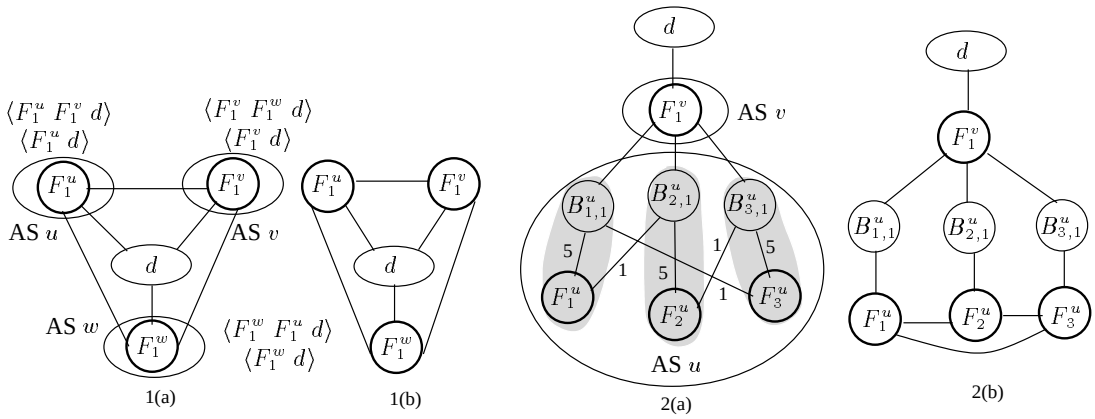

Fig. 4. 1) eBGP Divergence Example 2) iBGP Divergence Example

5 eBGP Divergence

Since the rank of each path is chosen arbitrarily at each AS, conflicting choices at neighboring ASms may prevent ASms from maintaining a stable path. That is, paths chosen by some ASms may oscillate continuously, even though neither G nor $\prec$ change.

Consider the SPP instance shown in Fig. 4.1(a).[2] Each AS contains only one BGP router. Figure 4.1(b) presents the peering graph of the SPP instance shown in Fig. 4.1(a). The paths acceptable to an AS (i.e. ranked higher than the empty path) are alongside the AS in the decreasing order of rank. Note that each AS prefers longer paths over shorter paths. E.g., F_1^u prefers the longer path $\langle F_1^u, F_1^v, d\rangle$ over the shorter path $\langle F_1^u, d\rangle$. This causes the ranking of each node to be in conflict with the ranking of its next hop to d.

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

- Initially F_1^u, F_1^v, and F_1^w choose the paths $\langle F_1^u, F_1^v, d\rangle$, $\langle F_1^v, d\rangle$, and $\langle F_1^w, d\rangle$, respectively.
- Node F_1^v notices that F_1^w chose the path $\langle F_1^w, d\rangle$. Hence, F_1^v changes its path to $\langle F_1^v, F_1^w, d\rangle$. This in turn forces F_1^u to change its path to $\langle F_1^u, d\rangle$.
- Node F_1^w notices that F_1^u chose the path $\langle F_1^u, d\rangle$. Hence, F_1^w changes its path to $\langle F_1^w, F_1^u, d\rangle$. This in turn forces F_1^v to change its path to $\langle F_1^v, d\rangle$.
- Node F_1^u notices that F_1^v chose the path $\langle F_1^v, d\rangle$. Hence, F_1^u changes its path to $\langle F_1^u, F_1^v, d\rangle$. This in turn forces node F_1^w to change its path to $\langle F_1^w, d\rangle$, and the system is back to its initial state.

Converging to a steady state is highly sensitive to the ranking of paths. For instance, in Fig. 4.1, if the ranking of paths at F_1^u is reversed, then the system is guaranteed to converge to a steady state. Due to this sensitivity to the ranking of paths, deciding if an SPP instance converges is NP-complete [5].

[2] This SPP instance is known as BAD GADGET in [2], [5].

6 iBGP Divergence

Even with stable eBGP, we consider an iBGP anomaly in which routers within an AS fail to converge to a stable assignment of paths [9]. We refer to this anomaly as clustering-induced divergence, because the interaction between route-reflection clustering and intra-AS routing link costs causes the system to diverge.

An example of clustering-induced divergence is shown in Fig. 4.2(a) [9]. Figure 4.2(b) shows the peering graph of Fig. 4.2(a). In this example, we assume that at AS u, local preference values of all the available paths to destination prefix d are equal.

Note that in the peer graph, each reflector F_i^u always prefers path $\langle F_i^u, F_{i+1}^u, B_{(i+1,1)}^u, F_1^v, d\rangle$ over path $\langle F_i^u, B_{i,1}^u, F_1^v, d\rangle$ due to following[3]:

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

Initially, assume that for each i, $\pi(F_i^u) = \langle F_i^u, B_{i,1}^u, F_1^v, d\rangle$. Consider the following sequence of events.

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

7 Comprehensive Solution

To provide a comprehensive solution that solves divergence anomalies with both eBGP and iBGP, we combine the solutions provided in [11], [12]. The general behavior of both solutions is similar. However, they have significant differences. The eBGP solution [11] models each AS as a single node. On the other hand, the iBGP solution [12] does consider the individual routers within a single AS, but it assumes the external paths from border routers are stable.

In both the solutions, each node maintains a cost value to detect divergence. Cost values grow without bound if there exists divergence in the system. If the cost value

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

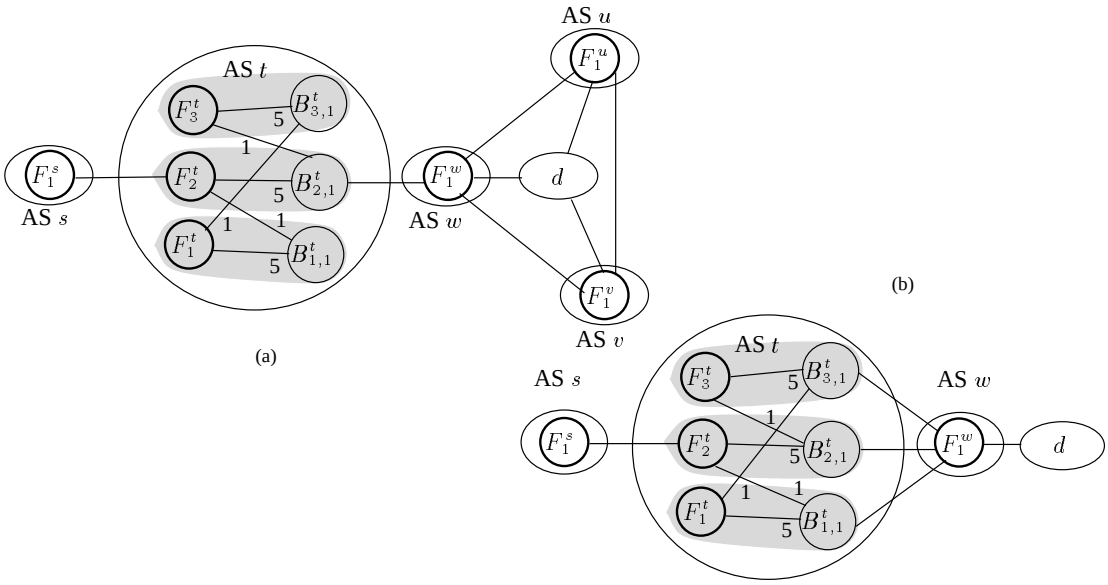

Fig. 5. a) Example with both eBGP and iBGP divergence b) Example with iBGP divergence

grows above some threshold value C, then nodes restrict their routing policies such that divergence is removed from the system. For our comprehensive solution, we have to decide whether to have two separate cost values to solve each of eBGP and iBGP divergence anomalies, or to have a single cost value to solve both eBGP and iBGP divergence anomalies.

A simple and efficient solution could be to just maintain a single cost value for solving both eBGP and iBGP divergence anomalies. Figure 5(a) shows an example with both eBGP and iBGP divergence by combining Fig. 4.1(a) and Fig. 4.2(a). In this example, eventually, the cost at a node F_1^u or F_1^v or F_1^w reaches a maximum value of C due to eBGP divergence between ASms u, v, and w. This causes the node to stop changing its chosen path. Thus, the path chosen at this node becomes stable, which stabilizes the paths at all other nodes. Hence, the single cost metric solution solves divergence anomalies with both eBGP and iBGP. However, it is not practical due to following reason.

Lets consider Fig. 5(b) obtained by removing AS u and AS v nodes from Fig. 5(a). In this example, eBGP is stable, where as iBGP in AS t diverges. Due to divergence in AS t, the cost value at router F_2^t will increase continuously. If we use a single cost metric solution, then iBGP divergence in one AS could effect many ASms. For example, divergence in AS t is effecting AS s, even though the divergence is internal to t.

To avoid these unnecessary effects, we have to use two separate cost metric values to solve each of eBGP and iBGP divergence anomalies. Thus, we denote the eBGP cost at node x by $cost_e(x)$ and the iBGP cost at node x by $cost_i(x)$. Next, we will present a Comprehensive-Divergence Avoidance Protocol, which forces convergence in both eBGP and iBGP.

7.1 Comprehensive-Divergence Avoidance Protocol

Figure 6 shows the specification of the Comprehensive-Divergence Avoidance Protocol (C-DAP). By ignoring MED values, the ranking relation $\prec$ becomes a total-order relation at each node. This protocol is motivated by the following observations. Lets consider eBGP divergence steps shown in the Fig. 4.1(a). Rank of the best path at each node decreases periodically. Divergence may not be possible, if the rank of the best path at each node increases monotonically. Eventually, every node should get the highest ranked path as the best path and the system should stabilize. We use this observation to detect divergence in eBGP. We use similar observation to detect iBGP divergence. In iBGP divergence example, rank of the best path at each reflector decreases periodically. In particular, the rank of the best path at reflector F_1^u decreases from step 1 to step 3.

In C-DAP, every node x maintains a pair cost values, $cost_e(x)$ and $cost_i(x)$. eBGP cost value, $cost_e(x)$, is increased by one, whenever the rank of the best path decreases and the best path is advertised by an eBGP peer. Similarly, iBGP cost value, $cost_i(x)$, is increased by one, whenever the rank of the best path decreases and the best path is advertised by an iBGP peer. Whenever the paths advertised between eBGP peers are oscillating continuously, value of $cost_e$ metric increases without bound. If $cost_e$ increases beyond some threshold value C_e then the node infers that the eBGP divergence is occurring. Each node takes a corrective action to remove oscillations in eBGP paths. Similarly, in an AS, value of local $cost_i$ metric increases without bound, if the paths advertised between iBGP peers are oscillating. If $cost_i$ increases beyond some threshold value C_i then the node infers that the iBGP divergence is occurring. Each node takes a corrective action to remove oscillations in iBGP paths.

Before explaining the protocol in detail, we will discuss some notation used in presenting the protocol. For every node x, $par(x)$ denotes the next hop node along the current path, $\pi(x)$, and $newpar(x)$ denotes the next hop node along the new best path, $best(choices(x))$. At node x, E represents the set of eBGP peer identifiers and I represents the set of iBGP peer identifiers. C-DAP specification consists of five actions. Each action updates $\pi(x)$, $cost_i(x)$, and $cost_e(x)$ variables depending on the guard condition, on which, they were excuted.

First action is enabled, if node x receives an update message from the current next-hop node, $par(x)$, with the current path, $\pi(x)$. This action consists of three assignment statements. If $par(x)$ is an eBGP peer then the first statement assigns $cost_i(x)$ variable to zero. Whenever a node x receives a path from an eBGP peer, node x resets the iBGP cost value to zero. If $par(x)$ is an iBGP peer then the second statement assigns $cost_i(x)$ variable to the maximum value of $cost_i$ at node x and $cost_i$ at $par(x)$. In the third statement, $cost_e(x)$ variable is always assigned to the maximum value of $cost_e$ at node x and $cost_e$ at $par(x)$.

Second action is enabled, if the rank of $best(choices(x))$ is lesser than the rank of $\pi(x)$ and $par(x)$ is not same as $newpar(x)$. This action consists of five assignments statements. First statement assigns $\pi(x)$ to the best path, $best(choices(x))$. After excution of the first statement, $par(x)$ and $newpar(x)$ are same. Next two statements update the $cost_i$ value based on whether $par(x)$ belongs to E or I. If $par(x)$ belongs to E then $cost_i$ is assigned to zero. If $par(x)$ belongs to I then $cost_i$ is increased by one. Last two statements update the $cost_e$ value depending on whether $par(x)$ belongs

node x
begin

$\pi(x) = \langle u, \pi(par(x))\rangle \rightarrow$
$\quad cost_i(x) := 0$ **if** $par(x) \in E$;
$\quad cost_i(x) := max(cost_i(x), cost_i(par(x)))$ **if** $par(x) \in I$;
$\quad cost_e(x) := max(cost_e(x), cost_e(par(x)))$;
[]
$\pi(x) \succ best(choices(x)) \wedge par(x) \neq newpar(x) \rightarrow$
$\quad \pi(x) := best(choices(x))$;
$\quad cost_i(x) := 0$ **if** $par(x) \in E$;
$\quad cost_i(x) := cost_i(x) + 1$ **if** $par(x) \in I$;
$\quad cost_e(x) := cost_e(x) + 1$ **if** $par(x) \in E$;
$\quad cost_e(x) := cost_e(par(x))$ **if** $par(x) \in I$;
[]
$\pi(x) \succ best(choices(x)) \wedge par(x) = newpar(x) \rightarrow$
$\quad \pi(x) := best(choices(x))$;
$\quad cost_i(x) := 0$ **if** $par(x) \in E$;
$\quad cost_i(x) := cost_i(par(x))$ **if** $par(x) \in I$;
$\quad cost_e(x) := cost_e(par(x))$;
[]
$\pi(x) \prec best(choices(x)) \wedge newpar(x) \in I \wedge (cost_i(newpar(x)) < C_i) \vee \pi(x) = \langle\rangle) \rightarrow$
$\quad \pi(x) := best(choices(x))$;
$\quad cost_i(x) := cost_i(par(x)$;
$\quad cost_e(x) := cost_e(par(x)$;
[]
$\pi(x) \prec best(choices(x)) \wedge newpar(x) \in E \wedge (cost_e(newpar(x)) < C_e) \vee \pi(x) = \langle\rangle) \rightarrow$
$\quad \pi(x) := best(choices(x))$;
$\quad cost_i(x) := 0$;
$\quad cost_e(x) := cost_e(par(x)$;

end

Fig. 6. Comprehensive-Divergence Avoidance Protocol

to E or I. If $par(x)$ belongs to E then node x increases the $cost_e$ value by one. But if $par(x)$ belongs to I then node x assigns its $cost_e$ value to $cost_e$ at $par(x)$.

Third action is enabled, if the rank of $best(choices(x))$ is lesser than the rank of $\pi(x)$ and $par(x)$ is equal to $newpar(x)$. This action consists of four assigment statements. First statement assigns $\pi(x)$ to the best path, $best(choices(x))$. Next two assignments update the $cost_i$ value based on whether $par(x)$ belongs to E or I. If $par(x)$ belongs to E then $cost_i$ is assigned to zero. If $par(x)$ belongs to I then $cost_i$ is assigned to $cost_i$ at $par(x)$. Value of $cost_e$ is always assigned to $cost_e$ at $par(x)$.

Fourth action is enabled, if the rank of $best(choices(x))$ is greater than the rank of $\pi(x)$, $newpar(x)$ belongs to I, and either $cost_i$ at $newpar(x)$ is less than some threshold constant C_i or $\pi(x)$ is empty. This action consits of three assignment statements. First statement assigns $\pi(x)$ to $best(choices(x))$. Next two statements assign $cost_i$, $cost_e$ values to corresponding cost values at $par(x)$.

Fifth action is enabled, if the rank of $best(choices(x))$ is greater than the rank of $\pi(x)$, $newpar(x)$ belongs to E, and either $cost_e$ at $newpar(x)$ is less than some threshold constant C_e or $\pi(x)$ is empty. First statement assigns $\pi(x)$ to $best(choices(x))$. Second statement assigns $cost_i$ value to zero and third statement assigns $cost_e$ value to $cost_e$ value at $par(x)$.

8 Other iBGP Anomalies

iBGP also suffers from two other types anomalies: MED-induced divergence, clustering-induced loops. Clustering-induced loops occur due to interaction between intra-AS link costs and route-reflection clustering. This anomaly can be avoided if the reflectors selectively advertise paths to their client routers. For complete details of this anomaly and proposed solution, readers are referred to [15]. iBGP also suffers from MED-Induced divergence anomaly. This anomaly disappears, if we ignore MED values for route selection. Reason for this anomaly is due to interaction between intra-AS routing link costs, route-reflection clustering, and MED values. MED-induced anomaly can be avoided by introducing virtual nodes [10], [12] in peering graph. Due space constaints, we are not presenting the complete details.

9 Related Work

There are several solutions proposed to solve eBGP and iBGP anomalies separately. We are not aware of any proposed comprehensive solution, that avoids anomalies with both eBGP and iBGP.

Proposed eBGP solutions can be divided into three categories. First category of solutions avoid the eBGP divergence by statically checking for conflicting routing policies in a centralized database [4]. This solution has several disadvantages. First, it requires global-coordination among all ASms. Second, Griffin et al. [5] also proved that the checking of conflicting routing policies is NP-hard. Second category of solutions avoid the divergence by presenting guidelines [6] for selecting the routing policies at each AS. This solution does not require global-coordination among ASms. But, it restricts the routing policies and removes the freedom of each AS choosing routing policies locally. Third category of solutions [16] avoid the divergence by restricting routing policies during runtime. In [16], every path update message carries the history of path update events. If a node finds a loop in the history of path update events then it removes some valid path(s) to avoid divergence. Loop in the history is only a necessary but not sufficient condition for divergence. Hence, their solution, sometimes, removes the path unnecessarily.

Proposed iBGP solutions avoid divergence by advertising multiple paths [7] [8] between each pair iBGP peers. Both solutions require high memory and message overheads. This defeats the whole purpose of using route-reflection clustering.

10 Summary and Concluding Remarks

BGP is the de-facto standard for inter-AS routing. Both external and internal forms of BGP plagued with many forms of anomalies. In this paper, we provided a comprehensive solution that solves all the known anomalies. Specification of our C-DAP protocol assumes shared memory model. But, we can easily change to more general message passing model by assuming that each path update message carries a pair of integer cost values. In our protocol, divergence increases the cost values to the maximum threshold values. We can reset these cost values by maintaining timers or by periodically using a reset protocol presented in [17].

References

1. Y. Rekhter and T. Li, "A border gateway protocol," *IETF RFC-1771*, 1995.
2. T. G. Griffi n, F. B. Shepherd, and G. Wilfong, "Policy disputes in path vector protocols," in *Proc. of IEEE ICNP conference*, 1999, pp. 21–30.
3. T. Bates and R. Chandrasekeran, "BGP route reflection - an alternative to full-mesh IBGP," *IETF RFC-1966*, 1996.
4. 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.
5. T. G. Griffi n, 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.
6. L. Gao and J. Rexford, "Stable Internet routing without global coordination," *IEEE/ACM Trans. Networking*, vol. 9, no. 6, pp. 681–692, 2001.
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. D. Walton, D. Cook, A. Retana, and J. Scudder, "BGP persistent route oscillation solution," *IETF Internet Draft*, 2002.
9. T. G. Griffi n and G. Wilfong, "On the correctness of IBGP confi guration," in *Proc. of ACM SIGCOMM conference*, 2002, pp. 17–29.
10. ——, "Analysis of the MED oscillation problem in BGP," in *Proc. of IEEE ICNP conference*, 2002, pp. 90–99.
11. J. A. Cobb and R. Musunuri, "Convergence of inter-domain routing," in *Proc. of IEEE GLOBECOM conference*, 2004, pp. 1353 – 1358.
12. R. Musunuri, , and J. A. Cobb, "Convergence of IBGP," in *Proc. of IEEE ICON Conference*, 2004.
13. M. G. Gouda, *Elements of Network Protocol Design*. John Wiley& Sons, 1998.
14. ——, "Protocol verifi cation made simple: A tutorial," *Comput. Netw. ISDN Syst.*, vol. 25, no. 9, pp. 969–980, 1993.
15. R. Musunuri, , and J. A. Cobb, "Complete solution to IBGP stability," in *Proc. of IEEE ICC conference*, vol. 2, 2004, pp. 1177 – 1181.
16. T. G. Griffi n, F. B. Shepherd, and G. Wilfong, "A safe path vector protocol," in *Proc. of INFOCOM conference*, 2000, pp. 490–499.
17. A. Arora and M. G. Gouda, "Distributed reset," *IEEE Trans. Comput.*, vol. 43, no. 9, pp. 1026–1038, 1994.