Preventing Persistent Oscillations and Loops in IBGP Configuration with Route Reflection

Anuj Rawat and Mark A. Shayman
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Email: [anuj, shayman]@isr.umd.edu

***Abstract*— Internal Border Gateway Protocol (IBGP) is responsible for distributing external reachability information, obtained via External-BGP (EBGP) sessions, within an autonomous system (AS). To avoid a full mesh of IBGP sessions between all the BGP speakers of an AS, scaling schemes such as route reflection and AS confederations have been proposed. But it has been observed that employing these schemes may result in problems such as routing oscillations and forwarding loops due to Multi-Exit Discriminator (MED) attribute and path asymmetry in IBGP. In this paper we study the pathologies observed in IBGP when route reflection is used. We model the AS using the Interior Gateway Protocol (IGP) connectivity graph G_I and IBGP peering graph G_L. Then we state some simple conditions on G_I and G_L and prove that these conditions guarantee the absence of any persistent routing oscillations and forwarding loops due to MED attribute and IBGP path asymmetry. We consider the problem of constructing an IBGP configuration given the IGP connectivity such that there are no persistent oscillations and loops, and apply the conditions developed in the paper on this problem. We prove that solving the problem while minimizing some appropriate cost function is NP hard. We then give an Integer Linear Program (ILP) to construct a forwarding loop and persistent routing oscillation free IBGP configuration, for an AS with given IGP connectivity graph, which minimizes some appropriate cost while satisfying the resource constraints on all the BGP speaking nodes.**

I. Introduction

At the topmost level, the Internet can be seen as a collection of a number of large and small Autonomous Systems (AS). An AS is nothing but a collection of routers managed by a single organization. The routing of IP datagrams within an AS is independent of the inter-AS routing and different organizations are free to deploy different intra-AS routing protocols based on their needs. But, unlike the intra-AS routing, inter-AS routing protocol has to be the same throughout the Internet. Border Gateway Protocol (BGP) [1] is the de-facto standard inter-domain routing protocol currently used in the Internet.

BGP works in two distinct modes of operation: External-BGP (EBGP) and Internal-BGP (IBGP) based on whether the BGP *peers*[1] belong to different ASes or the same AS respectively. EBGP is responsible for exchanging reachability information between different ASes whereas IBGP is responsible for distributing the information gained from EBGP among all the BGP speakers within the AS.

[1]BGP peers are BGP speakers (in the same or neighboring ASes) having direct BGP connection between them.

BGP, or more specifically EBGP, is a path vector protocol in which loops are detected and avoided by checking for multiple occurrences of an AS in the AS_PATH list[2] at each BGP node. This scheme cannot be used to detect loops in IBGP since all the speakers belong to the same AS. So to avoid loops in IBGP, every BGP speaker is required to maintain an IBGP session with every other BGP speaker in its AS. Clearly maintaining a full mesh of IBGP connections is not very scalable. To overcome this scalability issue, the two widely used IBGP configuration schemes are *AS confederations* [2] and *route reflections* [3]. But in recent years it has been observed that there can be persistent route oscillations [4][5][6][7][8] when these schemes are used in conjunction with Multi-Exit Discriminator (MULTI_EXIT_DISC or MED)[3] path attribute. Later Griffin et al. [9] showed that even without taking MED into account there may be route oscillations and loops due to the path asymmetry in IBGP.

There have been several attempts to study these routing anomalies. One of the approaches to eliminate MED oscillations, taken in [10] and [11], has been to change the protocol such that the problem vanishes. While Basu et al. [10] present a counterexample for the solution provided by Walton et al. [11], their own method is plagued by scaling issues (as discussed in [9]). In [10], they also prove that checking for MED oscillations is NP complete. In [12], Griffin et al. study MED oscillations using the technique employed in [13] for analyzing oscillations in EBGP due to the path selection policies employed by various ASes. But the MED oscillations turn out to be much harder to model. In another paper [9] they do the static analysis of the oscillations and loops due to path asymmetry using a graph theoretic approach and prove that checking for such anomalies is NP hard. They also give some sufficient conditions for preventing such anomalies. In [14], Musunuri et al. propose modifications to the IBGP protocol which, when supplemented with some restrictions on the IBGP configuration, succeed in suppressing the anomalies. They assume a full mesh of IBGP sessions among all the border speakers. But since almost all the BGP speakers are border speakers, this is essentially the same as assuming a full mesh of IBGP sessions between all the BGP speakers. So the scheme

[2]List of all the ASes that a route goes through to reach its destination, kept at each BGP speaker.

[3]MED value of a BGP route is a non-negative integer used to compare two routes passing through the same next-hop AS. The route having lower MED value has higher preference.

is not very useful in practice. Gobjuka [15] finds conditions on graphs to suppress loops due to path asymmetries in IBGP with route reflection. In [16] Musunuri et al. propose changes to IBGP which solve the problems due to both MED and path asymmetry. But, until now there has been no attempt at the static analysis of anomalies due to MED attribute, IBGP path asymmetry and their interactions. In this paper we model the AS using graphs and then we state and prove conditions on these graphs which guarantee the absence of all these anomalies in IBGP configurations with route reflections, without requiring any changes to the protocol.

The rest of the paper is organized as follows. Section II provides a brief overview of the route reflection mechanism and the route selection procedure employed by IBGP. In section III, we present a simple model for AS. Section IV formally defines the problem and explains why routing oscillations and loops occur in IBGP. In section V, we state our main theorem which gives conditions on the IBGP configuration guaranteeing the absence of persistent oscillation and looping problems. In this section we also give some intuition for why these conditions should work and discuss how our conditions are tighter than the conditions specified by Griffin et al. in [9] so that they take into account both path asymmetries and MED at the same time. Sections VI and VII contain the proof of the theorem. Section VIII looks at the time complexity of the problem of constructing an IBGP configuration based on the theorem from section V (while satisfying some other constraints and minimizing some appropriate cost function), when the IGP connectivity is given. In section IX we give an algorithm based on Integer Linear Programming to solve the problem set up in section VIII. Finally section X concludes the paper.

II. IBGP Overview

We start with a brief overview of the route reflection mechanism and the IBGP route selection criteria.

A. Route Reflection

As stated earlier, route reflection is a scheme devised to avoid maintaining a full mesh of IBGP sessions between the BGP speakers of an AS. The basic idea is to use a hierarchical tree like structure. The AS is partitioned into sets of nodes called *clusters*. Each cluster must have one (or more) special node(s) called *route reflector(s)*. All the other nodes in the cluster are called *clients* of the route reflectors of that cluster. The reflectors of an AS maintain a full mesh of IBGP sessions among themselves and IBGP connections with every client in their own cluster. A client cannot have an IBGP session with any node not in its own cluster. IBGP sessions between clients of the same cluster are permitted but not required. Now each cluster may have its own sub-clusters and so on, i.e., clustering can be as deep as required.

The rules of route reflection are that whenever a reflector receives a route from an IBGP peer, it selects the best path based on its path selection rule. After the best path is selected, it must do the following depending on the type of peer it is receiving the best path from.

(*i*) from another reflector: reflect the path to all its clients.

(*ii*) from a client: reflect the path to all its IBGP peers, except the originator.

The rest of the operating rules remain the same, i.e., whenever a node receives a route from an EBGP peer and selects it as its best path, then it must announce this to all its IBGP peers. Also, the clients do not re-advertise IBGP learned routes to other IBGP peers.

B. Route Selection in IBGP

On receiving a route update, a BGP speaker employs the following procedure to ascertain the best route.

(*i*) The route having the highest *degree of preference* is selected.

(*ii*) If there are multiple routes having highest degree of preference, then the route having the minimum AS_PATH length is selected.[4]

(*iii*) If there are multiple such paths, then for each neighboring AS, the path having the least MED value among all the paths going through that AS is considered. If there is only one such route, then that route is selected.

(*iv*) If there are multiple routes after step (*iii*) then among these, all the routes learned through EBGP peers only are considered. And if there are no routes learned via EBGP sessions, then all the routes learned via IBGP sessions (i.e., all the routes obtained after step (*iii*)) are considered. If there is only one route left then that route is selected.[5]

(*v*) If there are still multiple routes in contention, then the route having minimum IGP cost to the NEXT_HOP[6] node is selected.

(*vi*) If there are multiple such routes, then some deterministic tie-breaking criteria is used.

Since the IBGP path selection process is independent for two distinct external nodes, it is sufficient to consider only one destination node for analyzing the IBGP routing issues. In this work we will assume this external destination to be node d. Also, for ease of discussion, unless otherwise stated, we will assume that all the paths to destination d are ranked equally according to the rules (*i*) and (*ii*) of the path selection procedure stated above, i.e., they have equal degree of preference and AS_PATH length.

III. Model

We define a simple, undirected graph $G_P = \{N, E\}$ which captures the physical connectivity between the routers of an AS. Here N is the set of all the routers in the AS and E is the set of physical links between the routers. There is an edge

[4]The use of AS_PATH length to break the path selection ties is not mentioned in the BGP specifications [1], but both Cisco [17] and Juniper routers [18] use it. We also assume that it is practical to use AS_PATH length.

[5]The path selection rules given in the BGP specifications [1] do not differentiate between paths learned via EBGP and IBGP peers while searching for paths with minimum IGP cost to the NEXT_HOP node. But if there are multiple such paths with minimal IGP costs to the NEXT_HOP node, then EBGP learned routes are given preference over IBGP learned routes. The selection criteria we follow in this paper is the criteria used in the Cisco [17] and Juniper routers [18].

[6]NEXT_HOP path attribute defines the IP address of the border router that should be used as the point of exit (from the AS) for reaching the destinations listed in the BGP update message.

$n_i n_j \in E$ if and only if there is a physical link between the routers represented by nodes n_i and n_j.

A path P from node $n_1 \in N$ to node $n_k \in N$ is defined as an ordered set of nodes $n_1 n_2 n_3 \ldots n_{k-1} n_k$ such that $n_i n_{i+1} \in E$ for $i = 1, 2, \ldots, k-1$. We define function $cost()$ that takes a path as its argument and returns the IGP cost associated with that path. We assume that the IGP costs are additive, i.e., the cost of a path is the sum of IGP weights of its constituent physical links (edges).

Path S is the *shortest path* between two nodes $n_i, n_j \in N$ if it is a valid path between n_i and n_j and there is no valid path S' between n_i and n_j such that $cost(S') < cost(S)$. We define function $sp(n_i, n_j)$ that gives the shortest path available between the nodes n_i and n_j.

In this paper we consider that the IGP has converged. This is a valid assumption since we want to study the problems caused by the path asymmetry between the IGP routing and the BGP signaling.[7] So we can define graph $G_I = \{V, I\}$ which captures the IGP connectivity of the BGP speakers in the AS. Here $V \subseteq N$ is the set of all the BGP speakers in the AS and there is an edge $uv \in I$ if and only if $u, v \in V$ and there is no $w \in V$ such that $w \in sp(u, v)$. So a link in IGP connectivity graph G_I actually refers to the shortest path in the physical graph G_P between two BGP nodes (if it does not contain any other BGP node).

In this work we also assume that the EBGP learned paths, in the AS under study, are stable. This is a standard assumption in all the literature studying the IBGP convergence. The reason, as stated in [10], is that if the paths learned via EBGP in the AS under study are not stable, then we can always come up with new EBGP paths and withdraw existing EBGP paths such that the IBGP never converges. So it does not make sense to study the IBGP convergence when the EBGP learned paths are not stable.

We define another graph $G_L = \{V, L\}$ to represent the IBGP peering relationships. Here L is the set of IBGP sessions between the BGP speakers. A link $uv \in L$ if and only if node u and node v are IBGP peers.

For studying the IBGP routing issues due to MED and path asymmetries we do not really need the physical graph G_P. It is enough to look at the IGP connectivity graph G_I and the IBGP graph G_L. So in this paper we shall model an AS as $\{G_I, G_L\}$ and from now onwards we shall use *link* for referring to the links in the IGP connectivity graph G_I and not in the physical graph G_P.

Unless otherwise stated, we assume that all the nodes of the AS are divided into non-overlapping clusters, i.e., $V = \cup_{i=1}^{m} V_i$ such that $V_i \cap V_j = \emptyset$, for $i \neq j$. Every node in cluster V_i is classified as either reflector r_i or client c_i. If we want to show more than one reflector in cluster V_i then we use the notation r_{ij}. Similarly for multiple clients in cluster V_i, we use the notation c_{ij}.

Also for each external path P to destination node d, we define the following functions:

- $nextAS()$: gives the next AS which the packet has to enter, after exiting from the current AS, while following route P.
- $exitN()$: gives the NEXT_HOP node for route P (this is the node $u \in V$ that learns about P via an EBGP peer).
- $med()$: gives the MED value for route P.

We also define function $bestP(u)$[8] that gives the path selected by node u to reach destination d.

In all the examples and figures in this paper, the IGP cost of the link/shortest path (as the case may be) between two nodes is indicated besides the line joining the nodes. And wherever required, the MED values of the routes is indicated in parentheses.

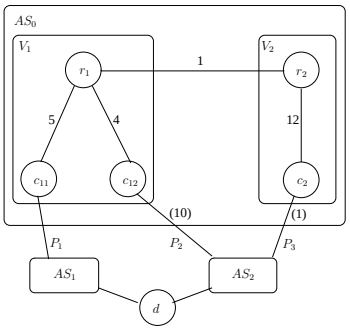

Fig. 1. Route oscillation due to MED

IV. Problem

In this section first we shall define the problem in terms of the model given in section III and then we shall study the BGP behavior responsible for the anomalies.

A. Problem Statement

Given the IGP connectivity graph G_I we want to find the conditions on the logical graph G_L which guarantee that the AS configuration with route reflections is free of persistent route oscillations and loops.

We define the two pathologies as:

- An AS is said to experience *persistent route oscillations* if, even in absence of any EBGP updates, some subset of BGP speakers of the AS keep on exchanging IBGP updates and are unable to settle down to any stable routing configuration.
- If a packet goes in a cyclic manner from one node to another without ever reaching the destination, the path is said to contain a *forwarding loop*.

B. MED

The basic problem with MED attribute is that it violates what Griffin et al. [9] call the *rule of independent ranking*. According to this rule, the relative ranking of paths at a node should be independent of the existence or non-existence of any path.

When used in conjunction with route reflection this behavior may lead to persistent route oscillation. Fig. 1 illustrates this

[7] We shall discuss this asymmetry in detail in Section IV-C.

[8] Path selected by a node depends on all the paths that a node knows about, so strictly speaking, $bestP()$ should be a function of time (or system state) also, but here we are assuming that the BGP has already converged to a stable state.

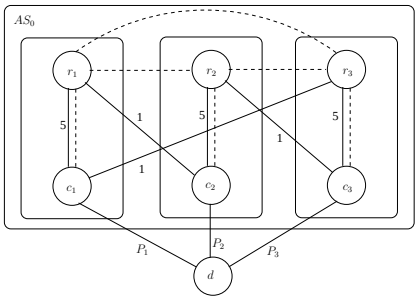

Fig. 2. Route oscillation due to IBGP path asymmetry

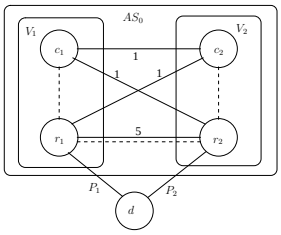

Fig. 3. Forwarding loop due to IBGP path asymmetry

by an example[9]. This is essentially the example presented in [6]. We assume that there is no IBGP session between c_{11} and c_{12}. Now the oscillations are generated due to the following steps:

(*i*) Node r_1 selects path P_2 over path P_1 (lower IGP metric) and node r_2 selects path P_3 (only path known).

(*ii*) On receiving update from node r_2, node r_1 learns about path P_3 and it selects path P_1 as its best path (path P_3 is ranked over path P_2 based on lower MED value, and then path P_1 is selected over path P_3 based on the lower IGP metric).

(*iii*) Now on receiving the update from node r_1, node r_2 learns about path P_1 and it selects path P_1 as its best path over path P_3 (lower IGP metric) and withdraws its previous best path P_3.

(*iv*) When path P_3 is withdrawn by node r_2, node r_1 selects path P_2 over path P_1 (lower IGP metric) and withdraws its previous best path P_1.

(*v*) When path P_1 is withdrawn by node r_1, node r_2 selects path P_3 over path P_2 (lower MED value) and the cycle begins again.

The underlying problem here is that since we are not using the full mesh of IBGP sessions between all the nodes, at a BGP speaker some of the available paths are invisible. When these paths become visible, the BGP speaker updates its best path and this new update may lead to path updates at other BGP speakers forcing the newly made visible path to become invisible once again. This results in route oscillation.

C. Path Asymmetries

In EBGP it is normally assumed that the peers share a physical network, so the underlying TCP link is a one-hop link. This means that usually EBGP messages are not routed. In this case the path followed by the EBGP signaling messages is same as the path followed by the data traffic, albeit in opposite direction. This is termed in [9] as *path symmetry*. On the other hand IBGP sessions are usually set up over multi-hop TCP links, so they are generally routed within the AS using the connectivity provided by the local IGP. Due to the internal routing of IBGP messages, there is an inherent *path asymmetry* in IBGP. More specifically, consider the AS modeled by its IGP connectivity graph and the IBGP peering graph $\{G_I, G_L\}$. Let BGP node $v \in V$ learn about route R to the external destination node d via IBGP peers. Now the IBGP *signaling path* corresponding to R at v is the logical path in graph G_L that the BGP updates announcing route R take to reach node v. So the signaling path $v_1 v_2 \dots v_k v$ corresponding to route R implies that BGP node v_{i+1} learns about R through IBGP peer v_i, where $i = 1, 2 \dots, k-1$, v learns about R through v_k and v_1 learns about R through some EBGP peer, i.e., $exitN(R) = v_1$. So if v selects route R as the best available route for destination d, the data packets destined for d at node v are routed to $exitN(R) = v_1$ via the IGP. Now the *data path* corresponding to R at v is the path in IGP peering graph G_I that the data packets take to reach $exitN(R) = v_1$ from v, i.e., the data path corresponding to R at v is $sp(v, exitN(R)) = sp(v, v_1)$. Usually the data path and the signaling path are not symmetric, i.e., usually data path is not equal to the signaling path in the reverse order. Griffin et al. [9] showed that these path asymmetries can cause both routing oscillations and loops.

Fig. 2 gives an example of the routing oscillations caused by the path asymmetry. This example was first presented in [9]. In the figure, the solid lines represent the IGP links, whereas the dotted lines represent the IBGP sessions. Based on the path selection criteria mentioned in section II-B we can see that the client nodes always select the paths learned via EBGP peers, i.e, $bestP(c_i) = P_i$ for $i = 1, 2, 3$. Now the paths P_1, P_2, P_3 are always visible to the reflectors r_1, r_2, r_3 respectively. But due to the lower IGP metric, when visible, reflector r_1 prefers path P_2 over path P_1, reflector r_2 prefers path P_3 over path P_2 and reflector r_3 prefers path P_1 over path P_3. Now we can verify that the reflectors will never be able to settle on a stable choice of paths. More specifically, for destination d, the path selection at r_1 will oscillate between paths P_1 and P_2, at r_2 between P_2, P_3 and at r_3 between paths P_3, P_1. The problem here is similar to the route oscillations described in section IV-B. The difference is that the here the oscillations are induced due to IBGP path asymmetry whereas in section IV-B, the problem was due to the MED path attribute. Here the signaling paths are along the dotted lines (IBGP sessions) but the actual physical paths followed are along the solid lines (IGP links). This path asymmetry leads to oscillation.

Griffin et al. [9] showed that path asymmetry can also lead to forwarding loops such as shown in Fig. 3. In the figure, the solid lines show the IGP links and the dotted lines represent the IBGP sessions. We can see that $bestP(r_1) = P_1$ and $bestP(r_2) = P_2$. Since node c_1 has IBGP connection only with node r_1, $bestP(c_1) = P_1$ (only path known). Similarly, $bestP(c_2) = P_2$. Now consider a packet at node c_1 marked for node d. Node c_1 tries to send this packet to $exitN(P_1) = r_1$

[9]Analogous example can easily be constructed where AS confederations causes route oscillations when used in conjunction with MED path attributes.

($\because$ $bestP(c_1) = P_1$). Note that $c_2 \in sp(c_1, r_1)$ therefore packet is routed through c_2. Arguing similarly, a packet destined for node d at node c_2 will be routed through node c_1. So we see that there is a loop between nodes c_1 and c_2.

In this example we see that the packet changes its intended path at node c_1 and again at node c_2. These changes in the forwarding path are called *path deflections*. Suppose for node u_1, $bestP(u_1) = P$ such that $exitN(P) = u_k$. Let the path from u_1 to u_k, according to IGP routing, be $u_1 u_2 \dots u_k$. A deflection is said to occur at node u_i if, starting from u_1, u_i is the first node $\in u_1 u_2 \dots u_k$ such that $exitN(bestP(u_i)) \neq u_k$.

As shown in the previous example, multiple deflections in the forwarding path may combine to form cycles called *forwarding loops*.

V. Theorem

In this section we state our main theorem followed by a brief intuitive explanation for the conditions stated in the theorem and compare them to the conditions given in [9].

A. Theorem Statement

Theorem 5.1: If an AS configuration with route reflection satisfies each one of the following conditions then it is free of persistent route oscillations as well as forwarding loops.

(*i*) If nodes u, v learn about paths P, Q respectively, having $nextAS(P) = nextAS(Q)$ through EBGP sessions, then u, v are IBGP peers.
(*ii*) Clients of same cluster are not IBGP peers.
(*iii*) $cost(sp(u, v)) < cost(sp(u, w))$ $\forall$ nodes u, v, w such that $u, v \in$ cluster V_i, $w \in$ cluster V_j and $i \neq j$.
(*iv*) If $u_i \in V_i$ and $u_j \in V_j$ are client nodes and $i \neq j$, then $\exists$ a reflector $u_k \in sp(u_i, u_j)$.

B. Intuitive Explanation

Condition (*i*) of Theorem 5.1 states that all the nodes, which learn about the paths with comparable MED values through EBGP sessions, should themselves be IBGP peers. The intuition is that if all the nodes, which learn about the paths having the same $nextAS$ via EBGP peers, form an IBGP mesh, then they can resolve amongst themselves which of these is the best path depending on the MED values. Now the other nodes in the AS should simply use this chosen path in their IGP metric based path ranking. This avoids the persistent routing oscillations in the system. If we look at this condition more closely we can see that if a BGP speaker learns about more than one path through its EBGP peers then it will only advertise at most one of these paths (it may not advertise any of these paths if it selects some path learned via IBGP peer). It may seem that this behavior can result in oscillations, but we shall see in section VI-B that this can only cause transient oscillations and no persistent oscillations. An important consequence of this condition is that if a node u has an EBGP learned path P through $nextAS(P) = AX_x$, and it learns about another path Q through $nextAS(Q) = AX_x$ via an IBGP peer v, then v must have learned about Q through an EBGP peer. This is because if v learns about Q through some IBGP peer w (which learns about Q via an EBGP session) then w should have an IBGP session with node u (according to condition (*i*)), and so u should have learned about Q via w and not v, which is a contradiction.

Condition (*ii*) of Theorem 5.1 takes care of forwarding loops that may form due to the IBGP path asymmetry when there are extra IBGP sessions between clients of a cluster. An example of such a forwarding loop is given in [9]. Note that conditions (*i*) and (*ii*) of Theorem 5.1 require that if nodes u, v learn about paths P, Q respectively, having $nextAS(P) = nextAS(Q)$ through EBGP sessions, then either both nodes are reflectors or they form a reflector-client pair in the same cluster. This is because by condition (*i*), we need u, v to be IBGP peers. But, by condition (*ii*), clients in the same cluster cannot be IBGP peers and according to the route reflection rules stated in section II-A, clients in different clusters cannot be IBGP peers.

According to condition (*iii*) of Theorem 5.1, if we ignore the MED values or if the MED values are same for all the paths, then for any node $u \in$ cluster C, if $\exists$ path P such that $exitN(P) \in C$ then this path should be ranked over all paths Q having $exitN(Q) \notin C$. In other words if we ignore MED values then any node prefers paths learned via clients and reflectors in its own cluster over paths learned via other reflectors. This is very similar to one of the condition given in [9] to guarantee that there are no forwarding loops due to IBGP path asymmetries. The condition in [9] states that any node should rank paths learned via clients over all the other paths. We see that the two conditions are similar but not exactly the same. Note that a consequence of condition (*iii*) is that if nodes $u, v \in$ cluster C, then $\exists$ no node $w \notin C$ such that $w \in sp(u, v)$.

Condition (*iv*) of Theorem 5.1 deals with the forwarding loops that may form due to the IBGP path asymmetries. This condition is not very intuitive but we can easily construct examples of IBGP configurations where if this condition is violated then there may be forwarding loops. In section VII we shall prove that when this condition is met in addition to the other conditions of Theorem 5.1, then there cannot be any forwarding loops in the BGP configuration.

Overall, while condition (*iii*) of Theorem 5.1, as stated earlier, is somewhat similar to one of the conditions given in [9] for removing routing oscillations, the conditions for removing forwarding loops in [9] are very different from what we present (conditions (*ii*) and (*iv*) of Theorem 5.1). We believe that our conditions are simpler to understand and less restrictive than those given in [9]. Specifically, conditions in [9] allow clients in the same clusters to be IBGP peers while we do not allow this. This is not very restrictive since it is taken as a rule of thumb in designing IBGP configurations anyway. But [9] needs the shortest path between any two nodes to be some valid signaling path, which is very restrictive in nature; we do not require this but we need a much simpler condition (condition (*iv*)) to hold. A major difference is that our conditions allow *simple deflections* in the IBGP configuration, whereas conditions in [9] remove simple deflections also. Here we define a simple deflection as the path deflection

that takes packets out of the AS. Note that as long as the path selection procedure checks for AS_PATH length, simple deflections cannot form loops on their own and therefore can be ignored to simplify the required conditions.[10] We will talk more about these in section VII. Another major difference is that we take into account the problems due to MED (mainly using condition (*i*) of Theorem 5.1) whereas [9] ignores MED.

In sections VI and VII we give proof for Theorem 5.1. Note that although the level of clustering in IBGP route reflection configuration may be more than one level deep, the proof only looks at the configurations where there are no sub-clusters within some cluster, i.e., in the proof we assume that the clustering is one level deep only.

Examples to show that if any condition is violated then the system may have persistent oscillations or forwarding loops, are not presented here due to the lack of space. But these examples are presented in [19].

VI. Routing Oscillations

To prove that the conditions stated in Theorem 5.1 guarantee the absence of persistent routing oscillations in the IBGP configuration, we use the *Stable Paths Problem* (SPP) model defined in [12][13]. We first present a brief introduction of the SPP model.

A. Modeling IBGP with MEDs

We consider the IBGP peering graph $G_L = (V, L)$ of the given AS. Now we construct an augmented logical graph $\widetilde{G}_L = (\widetilde{V}, \widetilde{L})$ where $\widetilde{V} = V \bigcup d \bigcup \bar{V}$ and $\widetilde{L} = L \bigcup \bar{L}$. Here d denotes the external destination and each node in $\bar{V}$ represents one of the various ASes which advertise d to the AS under study. $\bar{L}$ includes links between d and each node $\in \bar{V}$, and links corresponding to the EBGP sessions between BGP speakers in the AS under study and the ASes represented by node set $\bar{V}$.

For any node $v \in \widetilde{V}$, let $\mathcal{P}^v$ denote the set of *permitted paths* from v to destination d. Each permitted path at node v, is a *valid* BGP signaling path from d to v in the logical graph $\widetilde{G}_L$, taken in reverse order. Note that this is just an extension to the idea of IBGP signaling paths defined in section IV-C. The difference is that instead of using the IBGP peering graph G_L we use the augmented logical graph $\widetilde{G}_L$ for defining BGP signaling paths. Since each permitted path at node v corresponds to a BGP route to destination d that v can learn through its EBGP or IBGP peers, we use the two terms interchangeably. By selecting a route/path at IBGP node v, we mean that v has selected some $exitN$ for sending packets to the destination d. We define $\mathcal{P}$ to be the union of all sets $\mathcal{P}^v$.

Path assignment function π maps each node $u \in \widetilde{V}$ to a path $\pi(u) \in \mathcal{P}^u$, i.e., at each node u the path assignment function selects one of the permitted paths. We represent all the permitted paths at u that can be formed by extending the paths assigned to the neighbors of u by $candidates(u, \pi) = \mathcal{P}^u \bigcap \{(uv)Q \mid Q = \pi(v), (u, v) \in \widetilde{L}\}$[11]. So although all the permitted paths at node u are valid paths, the only paths visible are the paths that are advertised by its BGP peers. In other words, if the BGP peers of u select paths according to π, then the available paths at u are given by $candidates(u, \pi)$. Now we define *path selection* at node u as a function σ_u that maps any set of permitted paths $W \subseteq \mathcal{P}^u$ to the *best path* $\in W$. Path selection function describes the BGP rules that govern the selection of best path from all the permitted paths at each node. Let $\Sigma = \{\sigma_u \mid u \in \widetilde{V}\}$ be the set of all path selection functions. Note that the path ranking function σ_u at node u ascertains which $exitN$ is preferred at u. This is because each permitted path represents one of the possible $exitN$.

An instance of *General Stable Paths Problem* (GSPP) is a triple, $S = (\widetilde{G}_L, \mathcal{P}, \Sigma)$. And path assignment π is said to be a *solution* for the GSPP if $\forall u \in \widetilde{V}$ we have $\pi(u) = \sigma_u(candidates(u, \pi))$.

SPP is a special class of GSPP where the selection function is based on *linear*[12] ranking of paths. For node $u \in \widetilde{V}$, we define a non-negative, integer valued *ranking function* λ^u over $\mathcal{P}^u$, which specifies how the permitted paths are ranked at u. We assume that for $P_1, P_2 \in \mathcal{P}^u$ if $\lambda(P_1) < \lambda(P_2)$, then P_2 is *preferred over* P_1. We define $\Lambda = \{\lambda^u \mid u \in \widetilde{V} - \{d\}\}$ as the set of the ranking functions. Now the selection function induced by ranking λ^u is given by $\sigma_u(W) = P$ where P is the maximal path with respect to λ^u among all the paths $\in W \subseteq \mathcal{P}^u$. Since BGP speakers only announce their best paths, therefore clearly at each BGP node, the set of visible paths is such that, no two paths have the same next hop. Now to ensure that this is well defined, we need the ranking to be *strict*[13].

Griffin et al. [13] proved that a given SPP, and hence the BGP configuration, will converge to a unique solution if it does not have any *Dispute Wheel* (DW). A dispute wheel, $\Pi = (\bar{U}, \bar{Q}, \bar{P})$, of size k, is a sequence of nodes $\bar{U} = u_1, u_2, \ldots, u_k$ and sequences of nonempty paths $\bar{Q} = Q_1, Q_2, \ldots, Q_k$ and $\bar{P} = P_1, P_2, \ldots, P_k$, such that $\forall\ i \in \{1, \ldots, k\}$ we have:

(*i*) Q_i is a path from u_i to u_{i+1}
(*ii*) $P_i \in \mathcal{P}^{u_i}$
(*iii*) $Q_i P_{i+1} \in \mathcal{P}^{u_i}$
(*iv*) $\lambda^{u_i}(P_i) \leq \lambda^{u_i}(Q_i P_{i+1})$

Here all subscripts are interpreted modulo k. Fig. 4 gives an illustration of DW. Later in section VI-B we shall see that if all the DWs in a SPP are of even size, then the SPP has at least one stable solution. This implies an absence of persistent

[10] Consider the example given in Fig. 17 in [9]. Let AS_PATH length of P_1 be l_1 and of P_2 be l_2. Let the BGP path selection criteria look the AS_PATH lengths. Now since a node $\in AS_1$ selects P_1 as its best path and another node $\in AS_1$ selects $AS_2 - P_2$ as its best path, therefore $l_1 = l_2 + 1$. Similarly if we consider path selection at nodes in AS_2, then we get $l_1 + 1 = l_2$. So we have a contradiction and the loop should not exist.

[11] Here $(uv)Q$ represents the path formed by concatenation of edge uv and path Q. In general we define path PQ to be the concatenation of paths P, Q when the first node in Q is same as the last of node in P.

[12] By linear ranking of paths we mean that if path P_1 is ranked over path P_2 and path P_2 is ranked over path P_3, then path P_1 is ranked over path P_3.

[13] By strict ranking, we mean that if at node u, two permitted paths P_1 and P_2 are ranked equally, then u should learn only one of these paths at any time. This is clearly true when both the paths are announced to u by the same peer. In mathematical terms, we mean that if $P_1 \neq P_2$ and $\lambda^u(P_1) = \lambda^u(P_2)$, then $\exists\ v$ such that $P_1 = (uv)P_1'$ and $P_2 = (uv)P_2'$, i.e., paths P_1, P_2 have same next hop node.

route oscillations. So, in a sense, these kinds of DWs are manageable. We refer to these DWs of even size as *even DWs*.

The problem with SPP model is that if we take into account the MED attributes then the path rankings are no longer linear. So while considering MEDs, it is not straightforward to model a BGP configuration as a SPP. [12] models BGP with MEDs as a *Two Pass Stable Paths Problem* (2pSPP). A 2pSPP is a GSPP where the selection function σ_u is derived from two linear path ranking functions. In the *first pass*, the paths are sorted into disjoint classes and linear ranking is done within each class. In the *second pass* the best available paths from each class are ranked linearly. So at each node u, the permitted paths are partitioned into disjoint classes. Let C_u be the set of classes at node u and $\mathcal{P}_c^u \subseteq \mathcal{P}^u$ be the paths of class $c \in C_u$. Each node u has two ranking functions:

(*i*) α_c^u is a strict linear ranking defined only on permitted paths $\mathcal{P}_c^u$ of class $c \in C_u$
(*ii*) β^u is a strict linear ranking of all the permitted paths at u

Now for a set of given paths W, $(\beta^u \oplus \alpha^u)(W)$ is defined as the maximally ranked paths according to β^u among all the paths W_α, where W_α is obtained as:

(*i*) Divide W into sets $X_c = W \bigcap \mathcal{P}_c^u$.
(*ii*) $\forall$ classes $c \in C_u$, let γ_c^u be the maximally ranked path according to α_c^u among all the paths in X_c, i.e. γ_c^u is the path $P \in X_c$ such that $\forall$ paths $Q \in X_c$ and $Q \neq P$, $\alpha_c^u(P) > \alpha_c^u(Q)$.
(*iii*) $W_\alpha = \{\gamma_c^u \mid \forall c\}$

And a GSPP selection function σ_u is a two pass ranking function if it can be written as $\sigma_u = (\beta^u \oplus \alpha^u)$.

It is easy to see that BGP with MED can be represented as a 2pSPP. At any node u, classes represent the classification of paths based on $nextAS$. So the paths for each class can be strictly ranked since they have comparable MED values. This is the first pass ranking (α_c^u). The second pass (β^u) is just ranking all the paths ignoring the MED values. Now the overall ranking function $\sigma_u = (\beta^u \oplus \alpha^u)$ should give us the desired nonlinear path rankings.

Moreover any 2pSPP S can be reduced to SPP S' using the following simple transformation:

(*i*) $\forall$ node $u \in S$, place node u in S'. We call these *simple nodes*.
(*ii*) $\forall$ nodes $u \in S$ and $\forall$ classes $c \in C_u$, place node u^c in S'. We call u^c the *auxiliary node* of the simple node u for class c.
(*iii*) If w is a neighbor of u in S, then w is connected to each u^c in S'.

Now in the SPP S', auxiliary nodes u^c rank paths according to linear ranking function α_c^u and simple node u ranks paths according to linear ranking function β^u.

So we can model BGP configurations while considering MED attributes by first modeling it as a 2pSPP and then reducing it to an equivalent SPP as described above. Now the problem of oscillations in the BGP configuration can be studied by studying the DWs in the SPP.

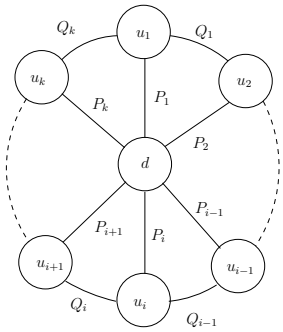

Fig. 4. Dispute Wheel

B. Proof

In this section we shall prove that a BGP configuration is free from persistent routing oscillation as long as the conditions of Theorem 5.1 are satisfied. For the purpose of the proof we assume that the given BGP configuration satisfies all the conditions of Theorem 5.1 and we have reduced it to its equivalent SPP model as defined in section VI-A. The main idea in the proof is that any DW in the equivalent SPP of a BGP configuration satisfying all the conditions of Theorem 5.1, will have at least one stable solution. And if this is the case then the SPP and hence the BGP configuration has at least one stable solution which can be reached from any starting state in finite time.

Now we present a brief outline of the proof. First we show that any even DW has at least one stable solution (Lemma 6.4). Then we prove that any DW in the given SPP is either an even DW or can be reduced to an even DW, and hence has at least one stable solution (Lemma 6.5). We then show that since all the DWs in the given SPP have stable solutions, therefore the SPP also has at least one stable solution which is reachable from any starting state in finite time (Lemma 6.6). This implies absence of persistent route oscillations in the BGP configuration (Lemma 6.7), which completes the proof.

The tricky part of the proof is to show that in the given SPP, if a DW is not even, then it can be reduced to an even DW. To prove this we use some properties of the structure of DWs in the given SPP. More specifically, we show that every DW in the given SPP satisfies the following:

(*i*) Every DW in the given SPP has at least one auxiliary node (Lemma 6.1)
(*ii*) Every DW in the given SPP has at least one simple node (Lemma 6.2)
(*iii*) Every simple node is followed by one or more auxiliary nodes of the same class and then another simple node. Here the first node following the simple node is always one of its own auxiliary nodes (Lemma 6.3).

Properties (*i*) and (*ii*) are used to prove the important property (*iii*). Now using condition (*i*) of the Theorem 5.1 along with property (*iii*), we prove that any DW in the given SPP can be reduced to a DW which has the same simple nodes as the original DW, but in which every simple node is followed by just one auxiliary node. Clearly this reduced DW is of even size.

We now present the formal proofs of all the claims stated above.

Lemma 6.1: Any DW in the SPP should have at least one auxiliary node.

Proof: Let all the nodes of a DW be simple nodes as shown in Fig 4. Let us assume that $exitN(P_i) = p_i \in$ cluster $V_i \ \forall \ i \in \{1, \ldots, k\}$, were we interpret all subscripts to be modulo k. Note that since P_i and $Q_{i-1}P_i$ are permitted paths representing the same external route at different nodes, $exitN(Q_{i-1}P_i) = exitN(P_i) = p_i$. Also, since the selection criterion at simple nodes is independent of the MED value, therefore node u_i selecting path Q_iP_{i+1} over path P_i means that $cost(sp(u_i, p_{i+1})) \leq cost(sp(u_i, p_i))$.

Now we must have one of the following two possible cases.

(*i*) $u_1 \in V_1$: In this case, since $u_1, p_1 \in V_1$ and $cost(sp(u_1, p_2)) \leq cost(sp(u_1, p_1))$, therefore from condition (*iii*) of Theorem 5.1, we can infer that $p_2 \in V_1$, i.e., $V_1 = V_2$. But if Q_1P_2 is a permitted path at u_1 having $exitN(Q_1P_2) = p_2$, with both $u_1, p_2 \in V_1$, then every node $\in Q_1P_2$ should $\in V_1$, therefore $u_2 \in V_1$. Similarly we can show that $u_i \in V_1 = V_2 \cdots = V_k \ \forall i$ in the DW, i.e., the DW is within one single cluster. Now we note that within a single cluster any valid permitted path is one of the following types[14]: (*a*) $Rf \leftarrow Ep$, (*b*) $Cl \leftarrow Ep$, (*c*) $Cl \leftarrow Rf \leftarrow Ep$, (*d*) $Rf_x \leftarrow Rf_y \leftarrow Ep$, (*e*) $Rf \leftarrow Cl \leftarrow Ep$ or (*f*) $Cl_x \leftarrow Rf \leftarrow Cl_y \leftarrow Ep$. Clearly P_1 cannot be of type (*a*) or (*b*), since then u_1 would have selected EBGP learned path P_1 over IBGP learned path Q_1P_2. Now consider path Q_kP_1. This also cannot be of the form (*a*) or (*b*), since $u_1 \in Q_kP_1$. Looking at P_1 and Q_kP_1 we can see that the only possible case when permitted paths are valid is when P_1 is of type (*e*) and Q_kP_1 is of type (*f*). This requires u_1 to be a reflector (and u_k to be a client node). Applying similar reasoning on paths P_2 and Q_1P_2 we get that u_1 should be a client node (and u_2 should be a reflector). This is a contradiction, therefore this case is impossible.

(*ii*) $u_1 \notin V_1$: We assert that this means $u_k \notin V_1$. We show this assertion by contradiction. Let $u_k \in V_1$. Now since Q_kP_1 is a permitted path at u_k having $exitN(Q_kP_1) = p_1$ with both $u_k, p_1 \in V_1$, every node $\in Q_kP_1$ should $\in V_1$. But this means that $u_1 \in V_1$, which is a contradiction, therefore $u_k \notin V_1$. Also $u_k \notin V_k$, since otherwise u_k should rank path P_k over path Q_kP_1. Similarly we can show that $u_i \notin V_i, V_{i+1} \ \forall i$ in the DW. Now if we have a permitted path spanning multiple clusters then it has to be of one of the following types[15]: (*a*) $Rf_x \leftarrow Rf_y \leftarrow Ep$, (*b*) $Cl_x \leftarrow Rf_x \leftarrow Rf_y \leftarrow Cl_y \leftarrow Ep$, (*c*) $Cl_x \leftarrow Rf_x \leftarrow Rf_y \leftarrow Ep$ or (*d*) $Rf_x \leftarrow Rf_y \leftarrow Cl_y \leftarrow Ep$. Since $u_{i+1}, u_i \notin V_{i+1}$ therefore paths P_{i+1}, Q_iP_{i+1} should be from one of the above mentioned types. Considering both at once we can see that the only possible cases when permitted paths are valid are when either P_{i+1}, Q_iP_{i+1} are of types (*a*) and (*c*) respectively or P_{i+1}, Q_iP_{i+1} are of types (*d*) and (*b*) respectively. Note that in any case we need u_i to be a client node (and u_{i+1} to be a reflector). Applying similar reasoning on paths P_i and $Q_{i-1}P_i$ we get that u_i should be a reflector (and u_{i-1} should be a client node). This is a contradiction, therefore this case is also impossible.

This shows that we cannot have a DW with only simple nodes. Hence in any DW there should be at least one auxiliary node. ■

[14]Here Cl means some client node, Rf means some reflector and Ep means some external peer. All the nodes belong to the same cluster. And $a \leftarrow b$ signifies that a node of type a learns about the path from a node of type b.

[15]Here the subscripts signify the cluster in which the node belongs.

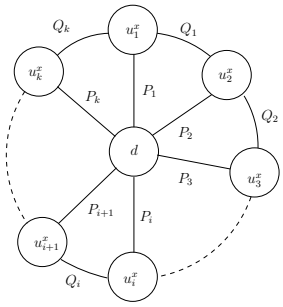

Fig. 5. Dispute Wheel with only auxiliary nodes

Lemma 6.2: All the nodes of a DW cannot be auxiliary nodes.

Proof: Let us consider a DW consisting only of auxiliary nodes. A part of such a DW is shown in Fig. 5. The notation u_1^x represents an auxiliary node of node u_1 for class x (AS_x). Note that any permitted path at node u_1^x has to pass through AS_x, therefore any auxiliary node on any of the valid signaling paths at u_1^x should also be for class x. Hence, as shown in the figure, node u_2^x, which lies on a valid path at u_1^x, is also for class x. Similarly we can show that all the auxiliary nodes on the DW are for class x. By condition (*i*) of Theorem 5.1, as noted in section V-B, u_2^x (i.e., node u_2) must have learned about path P_2 via an EBGP peer. Now for u_2^x to rank IBGP learned path Q_2P_3 over EBGP learned path P_2, we require $med(Q_2P_3) < med(P_2)$, i.e., $med(P_3) < med(P_2)$. Applying similar reasoning over the DW we see that we need $med(P_2) < med(P_3) < \cdots < med(P_k) < med(P_1) < med(P_2)$, which is a contradiction. Therefore we cannot have a DW with all its nodes as auxiliary nodes. ■

Lemma 6.3: In a DW, any simple node v is followed by one or more auxiliary nodes of the same class and then another simple node. Also the first auxiliary node following the simple node v, is an auxiliary node of v.

Proof: By Lemma 6.1 and Lemma 6.2, we have at least one simple node and one auxiliary node in DW. Without loss of generality we can consider u^xv to be the ordered nodes in DW. Here u^x is an auxiliary node of node u for class x (AS_x) and v is a simple node. Now since path Q_1P_2 is permitted at u^x, this path should be through AS_x. So by condition (*i*) of Theorem 5.1, as noted in section V-B, v must have learned about path P_2 via an EBGP peer. Now the only way v ranks another path Q_2P_3 over EBGP learned path P_2 is that Q_2P_3 is also learned via some EBGP peer and has a higher local preference. So the next node in the DW should be an auxiliary node of node v for some class y, i.e., the next node is v^y. This

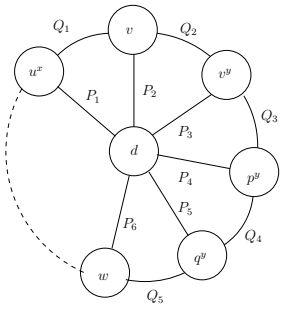

Fig. 6. Dispute Wheel in SPP

situation is shown in Fig. 6. Now we know that any permitted signaling path at v_y should be for class y, so if the next node on the DW is an auxiliary node then it must also be for class y and similarly any following consecutive auxiliary nodes on the DW are for class y until we reach the next simple node w[16]. Also we can see that whenever we put a simple node in the DW, it would be preceded by some auxiliary node and therefore, as argued, any simple node in the DW should be followed by one of its auxiliary nodes, which in turn may be followed by some auxiliary nodes of the same class and then we have another simple node. ■

Lemma 6.4: If a general DW has even number of nodes, then it has two stable solutions.

Proof: It is easy to see that for a DW of even size, if node u_i selects $P_i \forall$ odd value of i and selects $Q_i P_{i+1} \forall$ even value of i, then the system is stable. Similarly if u_i selects P_i $\forall$ even value of i and selects $Q_i P_{i+1}$ $\forall$ odd value of i, then also the system is stable. So these are the two stable solutions for any DW of even size. ■

Lemma 6.5: Any DW in the SPP should have at least one stable solution.

Proof: If the number of nodes in the DW is even then by Lemma 6.4 the DW has stable solutions. So we only need to look at the case when the number of nodes in the DW is odd. By Lemma 6.3, the nodes in a DW come in groups, where each group consists of a simple node followed by one or more auxiliary nodes. If the number of nodes in the DW is odd then at least one of these groups must have an odd number of nodes, i.e., at least one of the groups must have a simple node followed by more than one auxiliary nodes. Fig. 6 depicts a case when we have multiple auxiliary nodes (v^y, p^y, q^y) following a simple node (v). Now by virtue of condition (*i*) of Theorem 5.1, we can see that the node p^y learns about paths P_4 via some EBGP peer. Similarly node q^y learns about path P_5 via some EBGP peer. Also we saw in the proof of Lemma 6.3 that v^y learns about P_3 via an EBGP peer. Again by condition (*i*) of Theorem 5.1, all the permitted paths at any of the auxiliary nodes v^y, p^y, q^y are also permitted at the other two nodes. Now since at auxiliary nodes the ranking criteria is based on MED values, v^y, p^y, q^y should rank the path advertised by $exitN(P_5) = q$ over the other paths (seen in the DW) through AS_y. So as far as nodes v^y, p^y, q^y are concerned, there is no dispute and therefore we can ignore nodes v^y, p^y from our DW. Similarly we can remove the *extra* auxiliary nodes from all the groups just keeping a pair of nodes (a simple node and an auxiliary node). Now this *reduced* DW has even number of nodes and therefore, by Lemma 6.4, has stable solutions. Note that although two stable solutions exist for the reduced DW, if we look carefully we can see that there is only one of the two solutions is possible. This can be seen in the example in Fig. 6. Since nodes v^y, p^y, q^y rank the path advertised by $exitN(P_5) = q$ over the other paths (seen in the DW) through AS_y, therefore to get a stable solution for the original DW we need that q^y selects path P_5, p^y selects path Q_4P_5, v^y selects path $Q_3Q_4P_5$ and v selects path $Q_2Q_3Q_4P_5$ (and w does not select path P_6 which therefore remains invisible at all the other nodes). This is because the *other* possible solution for the DW requires node v to select path P_2, over EBGP learned (and therefore always visible) path Q_2P_3, which is not possible. Similar is the case for the nodes in all the groups having *extra* auxiliary nodes, and therefore only one of the two solutions is possible. So we see that in any case a DW in the SPP should has at least one stable solution. ■

Lemma 6.6: If a BGP configuration satisfies the conditions of Theorem 5.1, then its equivalent SPP has at least one stable solution which is reachable from any initial path assignment in finite time.

Proof: If the equivalent SPP has no DW then as proved in [13], the solution to SPP is unique and is always reached. If we have DWs in the SPP then according to Lemma 6.5 all of the DWs have stable solutions. Now if the DWs are non-overlapping then clearly the SPP has at least one stable solution. It is easy to see that even if DWs overlap we have stable solutions. This is because if two DWs overlap at some node u, then we consider the path assignment such that u selects the best possible path (among the four choices). We can now get to a stable solution for one of the DWs. And the other DW breaks down, since irrespective of the path selections of the other nodes of the DW, node u no longer selects any of the two paths available at u in that DW. Now in [13], Griffin et al. prove that in any SPP the nodes can be classified into two disjoint classes: *stable* and *oscillating*. The stable nodes provably reach their stable state in finite time regardless of the initial path assignment, and the oscillating nodes form the DWs. We have already shown that a stable solution exists for any DW in our SPP, and we have also discussed what the stable solution should be. We now note that this solution is indeed reachable irrespective of the starting system state. This is because at least one of the paths on each of the nodes of any DW is learned by the node via an EBGP session[17] and therefore irrespective of the starting path assignment, this is visible to the node. So we can easily construct an IBGP

[16]Note that path P_6 is also through AS_y but the path Q_6P_7 (not shown in the figure) can be through some other $nextAS$.

[17]Consider node u_i of the DW from Fig. 4 (u_i can be a simple node or an auxiliary node). Path P_i is learned by u_i via some EBGP peer.

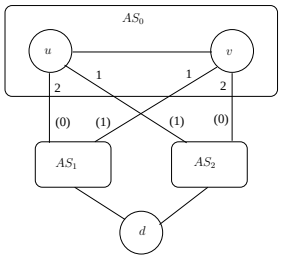

Fig. 7. Example for transient oscillations

message exchange sequence where the solution is reached[18]. ■

Lemma 6.7: If a BGP configuration satisfies the conditions of Theorem 5.1, then it is free from persistent oscillations.

Proof: From Lemma 6.6 we can see that since we always reach some stable solution, there are no persistent oscillations. ■

Note that although a stable state exists and can be reached in finite steps, there might still be transient oscillations, i.e., in certain cases we can construct IBGP message exchange sequences such that the system oscillates. This is due to the presence of DWs in the system. An example is given in Fig. 7. The local preference values for paths are indicated besides the lines representing the paths (lower value signifies higher preference) and, as always, the MED values are indicated in parentheses. If initially both nodes u, v select the path that they learn through their EBGP peers in AS_1 and they always update and advertise their best paths simultaneously then, we can see that there will be oscillations (after the first BGP message exchange both nodes u, v will select the path that they learn from their EBGP peers in AS_2, after the second message exchange they will revert back to the initial path assignment and the cycle will continue as long as they keep on updating and advertising their updated paths simultaneously). Actually the example has a DW with four nodes and two stable solutions as described in [12].

These kinds of oscillations are not persistent and they are highly dependent on the timing and delay between the BGP updates. Due to the random delays in the system these oscillations (if they occur) break down very soon and the system converges. So these are not fatal in nature. The main problem with such cases is that since there are multiple stable solutions, we are not sure to which state will the configuration converge. This may pose difficulty in debugging. In this paper we ignore the transient oscillations mentioned above.

VII. Forwarding Loops

As mentioned earlier, path asymmetry can lead to deflections, which may then combine to form loops. We will show that if an IBGP configuration satisfies the conditions of Theorem 5.1, then there may be deflections but they will never form loops. As there is no point in talking about loops if there are routing oscillations in the system, so in this section we assume that the IBGP configuration is in stable state. We also assume that the given configuration satisfies all the conditions stated in Theorem 5.1.

We start our analysis by proving certain properties about the structure of the forwarding loops when the BGP configuration satisfies all the conditions of Theorem 5.1. In particular, we prove the following.

(*i*) In a forwarding loop, if a packet gets deflected at a reflector then the next deflection must occur at some client node (Lemma 7.3).

(*ii*) If some reflector in the AS selects path P having $exitN(P) \in$ cluster C, then no node $\in C$ should select any path Q having $exitN(Q) \notin C$ (Lemma 7.4).

(*iii*) If a packet encounters a forwarding loop, then the only path deflections forming the loop occur at the nodes at which the packet enters a cluster (Lemma 7.5). We refer to such nodes as the *point of entry* to the cluster.

Now using the first property, we show that there can be only two types of forwarding loops. The other two properties place further restrictions on the structure of the possible loops. We study these two classes of forwarding loops individually in sections VII-A and VII-B and show that if all the conditions of Theorem 5.1 hold, then even these are not possible.

We first prove a couple of observations that we, in turn, use to prove the above listed properties of the forwarding loops. The first observation is on the visibility of paths to the IBGP peers in an AS. It states that the best path selected at a node is always visible to its IBGP peers.

Lemma 7.1: If two nodes are IBGP peers then they know about each other's best path.

Proof: Let nodes u, v be IBGP peers and let $bestP(u) = P$. Clearly if P is an EBGP learned path at u, then u should advertise P to all its IBGP peers including v. Hence P should be visible at v. Now let us assume the P is an IBGP learned path at u. Since u, v are IBGP peers, we have one of the following three possible cases.

- u is a reflector and v is its client: According to the rules of route reflection, u always reflects path P to its clients. Hence P should be visible at v.
- both u, v are reflectors: Here we have one of the following two subcases.
  - u learns about P via some client: In this case u reflects path P to all its IBGP peers including v.
  - u learns about P via some reflector w: Now since reflectors form a full IBGP mesh, w, v are also IBGP peers. And so if w announces P to reflector u then it should announce P to all its reflector peers including v.

  So in both the cases, path P should be visible at v.
- v is a reflector and u is its client: Due to condition (*iii*) of Theorem 5.1, the only IBGP peers of a client are its reflectors. So if u is a client and its learns about P via an IBGP session, then it must have learned it form

[18]Suppose we assume that the DW has even number of nodes and node u_i has initially selected path P_i. Now we can see that if u_i advertises path P_i to node u_{i-1}, and then u_{i-1} calculates its best path, u_{i-1} will select path $Q_{i-1}P_i$. Proceeding in similar fashion to the nodes $u_{i-2}, \ldots, u_1, u_k, \ldots, u_{i+1}$ we reach one of the stable solutions in finite number of message exchanges. Stable state can be reached similarly for the case when the DW has odd number of nodes. Now since stable solution is reachable for all the DWs in the SPP, stable solution for the SPP is also reachable in finite time irrespective of the starting path assignments.

some reflector w. Now since reflectors form a full IBGP mesh, w, v are also IBGP peers. The only time when w announces P to its client u but not to its reflector peer v is when w learns about P through another reflector peer x. Now with the similar argument as used in the second subcase of the above case, we see that P should be visible at v.

We have proved that if u, v are IBGP peers then $bestP(u)$ is always visible at v. We can similarly show that $bestP(v)$ is always visible at u. So if u, v are IBGP peers then they know about each others best path. ■

The next observation lists two cases when we have simple deflections. It states that if two nodes u, v are either IBGP peers or clients in the same cluster, then the only possible path deflection at v, on the packets coming from u, is a simple deflection.

Lemma 7.2: If $v \in sp(u, exitN(bestP(u))$ and u, v are either IBGP peers or clients in the same cluster, then either $bestP(u) = bestP(v)$ or $exitN(bestP(v)) = v$.

Proof: Let $bestP(u) = P$ with $exitN(P) = p$ and let $bestP(v) = Q$ with $exitN(Q) = q$ and $P \neq Q$.

First we observe that if u, v know about each other's best path, then the following are true.

- At u, both paths P and Q are IBGP learned paths. To see that P is not an EBGP learned path at u, we note that $v \in sp(u, p) \Rightarrow p \neq u$. For path Q, we note that since $bestP(u) \neq Q$ but $bestP(v) = Q$, therefore $\exists$ node $w \neq u$ which advertises path Q.
- At v, path P is an IBGP learned path.[19] This is because $bestP(u) = P$ but $bestP(v) \neq P$, therefore $\exists$ node $x \neq v$ which advertises path P.

Now if u, v know about each other's best path, such that $bestP(u) = P$ with $exitN(P) = p$, $bestP(v) = Q$ with $exitN(Q) = q$ and $P \neq Q$ then we have one of the following two possible cases.

1) $cost(sp(u, p)) \leq cost(sp(u, q))$
   and if equality, then P is ranked over Q based on BGP tie-breaking criteria.
   Using this inequality along with the following facts
   - $v \in sp(u, p) \Rightarrow$ $cost(sp(u, v)) + cost(sp(v, p)) = cost(sp(u, p)$
   - $cost(sp(u, q)) \leq cost(sp(u, v)) + cost(sp(v, q))$

   we can show that
   $cost(sp(v, p)) \leq cost(sp(v, q))$
   with equality only if P is ranked over Q based on BGP tie-breaking criteria.
   Now since v selects Q over P, therefore $\exists$ path P' visible at v but not at u such that:
   - $nextAS(P') = nextAS(P)$
   - $med(P') < med(P)$
   - $cost(sp(v, q)) \leq cost(sp(v, exitN(P')))$

   Note that in this case we have, $cost(sp(v, p)) \leq cost(sp(v, exitN(P')))$. Using this inequality along with the fact that v learns about path P via some IBGP peer, we can see that in this case P' is also an IBGP learned path at v.
2) $cost(sp(u, p)) \geq cost(sp(u, q)$
   and if equality, then P is ranked over Q based on BGP tie-breaking criteria.
   Now since u selects P over Q, therefore $\exists$ path Q' visible to u but not to v such that:
   - $nextAS(Q') = nextAS(Q)$
   - $med(Q') < med(Q)$
   - $cost(sp(u, p)) \leq cost(sp(u, exitN(Q')))$

   Note that in this case we have, $cost(sp(u, q)) \leq cost(sp(u, exitN(Q')))$. Using this inequality along with the fact that u learns about path Q via some IBGP peer, we can see that in this case Q' is also an IBGP learned path at u.

Now we start the actual proof. We first study the case when u, v are IBGP peers. In this case by Lemma 7.1, u, v know about each other's best path. So the only deflections possible are due to the cases described above. When u, v are IBGP peers, then we have one of the following cases.

- u is reflector and v is client, both $\in$ cluster C. First we consider case 1. In this case, since P' is an IBGP learned path at v, it should be visible to all the reflectors $\in C$ including u. This is a contradiction, hence 1 is impossible. Now consider case 2. In this case, since both Q and Q' are IBGP learned paths at u, they should be visible to all the other reflectors $\in C$ as well. But then no reflector should advertise path Q, so the only way v can learn about Q is through an EBGP session, i.e., we can only have a simple deflection at v.
- u is client and v is reflector, both $\in$ cluster C. In case 1, since both P and P' are IBGP learned paths at v, they should be visible to all the other reflectors $\in C$ as well. But then no reflector should advertise path P and so u cannot learn about P through IBGP. This is a contradiction, hence case 1 is impossible. In case 2, since Q' is an IBGP learned path at u, it should be visible to all the reflectors $\in C$ including v. This is a contradiction, hence 2 is also impossible.
- Both u and v are reflectors. We can split this into the following two subcases:
  - Both $u, v \in$ came cluster C. In case 1, since P' is an IBGP learned path at v, it should be visible to all the reflectors $\in C$ including u. This is a contradiction, hence 1 is impossible. Similarly, in case 2, since Q' is an IBGP learned path at u, it should be visible to all the reflectors $\in C$ including v. This is a contradiction, hence 2 is also impossible.
  - u, v are reflectors in different clusters. Let $u \in C_u$ and $v \in C_v$. According to case 1, P' is an IBGP learned path at v. Let v learn about P' via some IBGP peer w. Now w can either be a reflector of the AS or it can be a client $\in C_v$. If w is a reflector which advertises P', then P' should be visible to all the reflectors in the AS, including u. This is a contradiction, hence w cannot be a reflector. If w is a client $\in$

[19] Note that there is no such restriction for path Q, i.e., node v may learn about path Q via some EBGP or IBGP peer. But if Q is an EBGP learned path at v then the deflection at v would take the packet out of the AS, i.e., at v there would be a simple deflection only.

C_v which advertises P', then it should have learned about P' via some EBGP peer, i.e., $exitN(P') = w$. Now since $cost(sp(v, exitN(P'))) \geq cost(sp(v, p))$ and $exitN(P') = w \in C_v$ therefore by condition (*iii*) of Theorem 5.1, $p \in C_v$. So u must have learned about path P via some reflector $\in C_v$. But since client node $w \in C_v$ announces path P', it should be visible to all the reflectors $\in C_v$ and in that case no reflector $\in C_v$ should select and advertise P. This is a contradiction, so 1 is impossible. Now we consider case 2. In this case, we have $cost(sp(u, exitN(Q'))) \geq cost(sp(u, v))$.[20] Now applying condition (*iii*) of Theorem 5.1 on this inequality and using the fact that $v \notin C_u$, we infer that $exitN(Q') \notin C_u$. So the only way u can learn about Q' is via some reflector $w \notin C_u$. But in that case w should announce Q' to all the reflectors in the AS, including v. This is a contradiction, hence case 2 is also impossible.

So there can be no deflection at v when both u, v are reflectors.

Now we consider the case when both u and v are clients $\in$ cluster C. Since clients $\in$ the same cluster are not IBGP peers, therefore, in this case, u, v need not know about each other's best paths. But note that if u, v are clients $\in C$, then the following are true.

- Since $bestP(u) = P$, u always knows about P.
- Since $bestP(v) = Q$, v always knows about Q.
- Note that P is an IBGP learned path at u.[21] So u must learn about P via some reflector $w \in C$. But in that case w also announces P to v, therefore v knows about P.

So when u, v are clients $\in C$, then we have one of the following two possible cases.

- u, v know about each other's best path. In this case we only need to look at cases 1 and 2. In case 1, P' is an IBGP learned paths at v. This means that $\exists$ a reflector $w \in C$ which announces P'. But in that case w should also announce P' to u. This is a contradiction, hence case 1 is impossible. Similarly in case 2, Q' is an IBGP learned path at u. This means that $\exists$ a reflector $x \in C$ which announces Q'. But in that case x should also announce Q' to u. This is a contradiction, hence case 2 is also impossible.
- u does not know about Q. This is only possible when v learns about Q through an EBGP session. But in that case we can only have a simple deflection at v.

■

Now we prove the three properties about the structure of the forwarding loops that we listed at the start of this section.

Lemma 7.3: In a forwarding loop, if a packet gets deflected at a reflector u, and the next deflection occurs at node v, then v should be a client node.

Proof: Let v be a reflector. Since u is also a reflector in the same AS, u, v should be IBGP peers. Now applying Lemma 7.2, we can infer that the for a packet going through u, if the next deflection occurs at v, then it should be a simple deflection. But in that case we cannot have a forwarding loop in the AS. This is a contradiction, hence v has to be a client node. ■

By Lemma 7.3, we can see that there cannot be any forwarding loop with deflections at reflectors only. So there can only be the following two kinds of forwarding loops in the system.

- Forwarding loop consisting of deflections at client nodes only.
- Forwarding loop consisting of deflections at both client nodes and reflectors. We note that in this case, by Lemma 7.3, any deflection at reflector must be preceded and succeeded by deflections at client nodes.

We shall analyze these two possible types of forwarding loops individually and show that they cannot exist. But before doing this, we prove some more properties about the forwarding loops. These properties put further restrictions on the structure of the loops.

Lemma 7.4: If $\exists$ a reflector u having $exitN(bestP(u)) \in C$, then $\forall$ node $v \in C$ $exitN(bestP(v)) \in C$.

Proof: Let $bestP(u) = P$. First note that if $u \notin C$ but still selects path P having $exitN(P) = p \in C$, then it must have learned about P via an IBGP session with some reflector $\in C$. So without loss of generality, we can assume that u is a reflector $\in C$. Now let $v \in C$ have $bestP(v) = Q$ such that $exitN(Q) = q \notin C$. Since u, v are IBGP peers therefore by Lemma 7.1, they know about each other's best path. Now using condition (*iii*) of Theorem 5.1 and the facts that $q \notin C$ and $v, p \in C$, we get $cost(sp(v, p)) < cost(sp(v, q))$. But if v still selects Q over P then we have one of the following two cases.

- $nextAS(Q) = nextAS(P)$ and $med(Q) < med(P)$. But then u should also select path Q over path P, so this cannot be the case.
- $\exists$ route P', known to v but not to u such that $nextAS(P') = nextAS(P)$, $med(P') < med(P)$ and $cost(sp(v, q)) \leq cost(sp(v, exitN(P')))$. But the only route known to v and unknown to u should have $exitN(P') = v$. Also since $bestP(v) \neq Q$ but it is visible at u, it is an IBGP learned path at v, i.e., $v \neq q$. From these observations we can infer that $cost(sp(v, q)) > cost(sp(v, exitN(P')))$. So this is also not possible.

Hence proved. ■

Lemma 7.5: Deflections that may cause loops can only occur at the point of entry to the clusters.

Proof: Let node $u \in$ cluster C_u having $bestP(u) = P$ and let the first path deflection on P occur at node v. If $v \in C_v \neq C_u$ and is not the point of entry to cluster C_v, then let the point of entry be w. Now since v, w are either IBGP peers or clients $\in C_v$ therefore by Lemma 7.2 if the first deflection after node u occurs at v, then it has to be a simple deflection. On the other hand if node $v \in C_u$, then either it has an IBGP

[20] This is because we have the following.
$cost(sp(u, exitN(Q'))) \geq cost(sp(u, p))$
And $v \in sp(u, p) \Rightarrow cost(sp(u, p)) > cost(sp(u, v))$

[21] This is because we have the following.
P is always known at u.
$v \in sp(u, exitN(P)) \Rightarrow P$ cannot be an EBGP learned path at u.

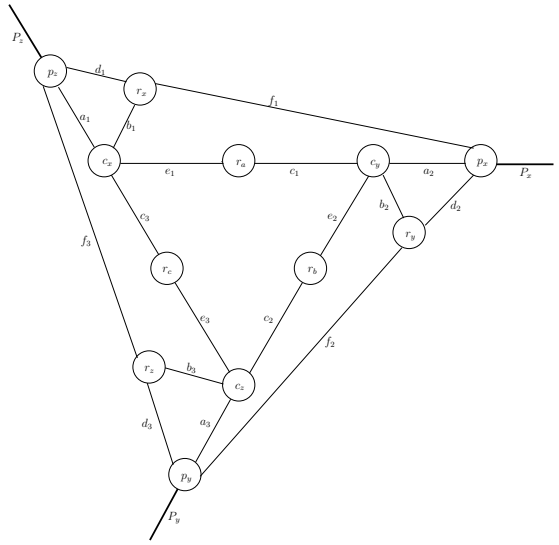

Fig. 8. Loop due to deflection at clients only - case I

session with u or u, v are clients $\in C_u$. So again by Lemma 7.2 the only deflection possible at node v should be a simple deflection.

So we have proved that if a deflection occurs at some node $v \in$ some cluster C_v such that v is not the point of entry for the packet to cluster C_v, then we can only have a simple deflection at v which takes the packets out of the AS and cannot form any loop in the AS. ■

Now we shall separately consider the two types of forwarding loops possible in the system and prove that if the IBGP configuration satisfies the conditions of Theorem 5.1, then neither of these types of loops can exist. The strategy is to establish the IGP metric based inequality relation, given as equation (11), between the consecutive client nodes at which the packet gets deflected. We then use this inequality sequentially over all the client nodes in the loop, at which the packet gets deflected, and achieve the contradiction as given in equation (13).

A. Deflections at Clients only

We first look at the loops consisting of path deflections at client nodes only. Let two consecutive deflections occur at client nodes c_x and c_y. Let $c_x \in$ cluster C_x and $c_y \in$ cluster C_y. Lemma 7.5 states that forwarding loops can only occur due to the deflections at the points of entry to the clusters. So the packets must enter C_x at c_x and C_y at c_y. Since the packets enter cluster C_y at client node c_y, therefore by condition (*iv*) of Theorem 5.1, the last link traversed by the packets before reaching c_y has to be an inter-cluster reflector-client link. Let this link be $r_a c_y$ where r_a is a reflector $\notin C_y$. Now we can further split the analysis into two cases depending on whether $r_a \in C_x$ or not.

1) $r_a \in C_a \neq C_x$*:* We study this case using Fig. 8. In the figure, the line joining any two nodes represents the shortest path between the nodes. The figure shows the structure of a forwarding loop consisting of path deflections at three client nodes c_x, c_y, c_z. Note that the proof does not rely on the number of deflections and holds for general case.

First we note the following about the structure of this loop.

- Let $c_x \in C_x$, $c_y \in C_y$ and $c_z \in C_z$.
- Let $bestP(c_i) = P_i$ with $exitN(P_i) = p_i$ for $i = x, y, z$.
- $p_x, p_y, p_z \notin C_x, C_y, C_z$. Since an inter-cluster IGP link $r_a c_y$ lies in $sp(c_x, p_x)$, therefore by condition (*iii*) of Theorem 5.1, p_x and c_x are in different clusters, i.e., $p_x \notin C_x$. Similarly $p_y \notin C_y$ and $p_z \notin C_z$. This means that client c_x learns about P_x through some reflector $r_x \in C_x$. Similarly c_y learns about P_y through reflector $r_y \in C_y$ and c_z learns about P_z through reflector $r_z \in C_z$. Now note that since reflector $r_x \in C_x$ selects path P_x having $exitN(P_x) = p_x \notin C_x$, therefore by Lemma 7.4 $\exists$ no reflector r in the AS such that $exitN(bestP(r)) \in C_x$. And therefore no node $\notin C_x$ learns about any path P' having $exitN(P') \in C_x$. So $p_y, p_z \notin C_x$. Arguing similarly we get $p_x, p_y, p_z \notin V_x, V_y, V_z$.
- As mentioned in the previous point, c_x learns about P_x through reflector $r_x \in C_x$, c_y learns about P_y through reflector $r_y \in C_y$ and c_z learns about P_z through reflector $r_z \in C_z$.
- The presence of loop requires that $r_a \in sp(c_x, p_x), c_y \in sp(r_a, p_x)$; $r_b \in sp(c_y, p_y), c_z \in sp(r_b, p_y)$; $r_c \in sp(c_z, p_z), c_x \in sp(r_c, p_z)$, as shown in the figure.

Note that according to the figure $r_x \notin sp(c_x, r_a)$, but the proof does not require this and considers the most general case.

We can now easily get the constraints on IGP costs presented in equations (1)-(5). The reasoning for each equation is provided after the equation.

$$d_1 \leq a_1 + b_1 \tag{1}$$

Equation (1) states that
$cost(sp(r_x, p_z)) \leq cost(sp(r_x, c_x)) + cost(sp(c_x, p_z))$.

$$f_1 \leq d_1 \tag{2}$$

Let $f_1 > d_1$. Now since both P_x, P_z are visible to r_x and it chooses P_x over P_z therefore $\exists$ a path P'_z visible to r_x with $exitN(P'_z) = p'_z$ having $nextAS(P'_z) = nextAS(P_z)$, $med(P'_z) < med(P_z)$ and $cost(sp(r_x, p'_z)) \geq f_1$. But since r_x, p_x are not in same cluster, therefore condition (*iii*) of Theorem 5.1 ensures that r_x, p'_z are also in different clusters. Now r_x knows about path P'_z means that P'_z is announced by some reflector, therefore r_z should also know about it. And if r_z knows about P'_z then it will never choose path P_z, so there will not be any loop. Hence $f_1 \leq d_1$, i.e., equation (2) holds.

$$e_1 + c_1 + a_2 \leq b_1 + f_1 \tag{3}$$

Equation (3) states that
$cost(sp(c_x, p_x)) \leq cost(sp(c_x, r_x)) + cost(sp(r_x, p_x))$
where we use that fact that
$r_a \in sp(c_x, p_x)$ and $c_y \in sp(r_a, p_x)$.

$$b_1 < e_1 \tag{4}$$

This is true because $r_x, c_x \in C_x$ and $r_a \notin C_x$.

$$b_2 < c_1 \tag{5}$$

This is true because $r_y, c_y \in C_y$ and $r_a \notin C_y$.

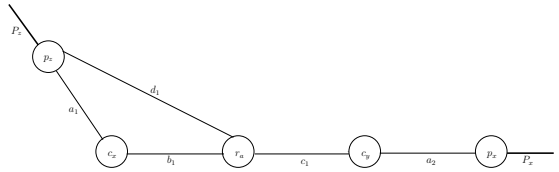

Fig. 9. Loop due to deflection at clients only - case II

Using equations (1)-(5), we get

$$\begin{aligned} & f_1 \leq a_1 + b_1 \\ \therefore \quad & e_1 + c_1 + a_2 \leq b_1 + a_1 + b_1 \\ \therefore \quad & c_1 + a_2 < a_1 + b_1 \\ \therefore \quad & a_2 + b_2 < a_1 + b_1 \end{aligned} \tag{6}$$

2) $r_a \in C_x$*:* Now we look at the case when $r_a \in C_x$. The links of the loop in such case is shown in the Fig. 9. All the observations about the structure of the loop, stated in section VII-A.1 still hold. The only difference is that since there is no deflection at r_a, therefore r_a selects P_x and we can consider it to be the reflector from which c_x learns about path P_x. So without loss of generality, we assume r_a to be the reflector r_x of section VII-A.1.

We can now easily get the constraints on IGP costs presented in equations (7)-(9). The reasoning for each equation is provided after the equation.

$$d_1 \leq a_1 + b_1 \tag{7}$$

Equation (1) states that
$cost(sp(r_a, p_z)) \leq cost(sp(r_a, c_x)) + cost(sp(c_x, p_z))$.

$$c_1 + a_2 \leq d_1 \tag{8}$$

Using the fact that $c_y \in sp(r_a, p_x)$, we get the shortest path IGP distance between r_a and p_x as $cost(sp(r_a, p_x)) = c_1 + a_2$. Now let $c_1 + a_2 > d_1$, i.e., let $cost(sp(r_a, p_x)) > cost(sp(r_a, p_z))$. Since both P_x, P_z are visible to r_a and it chooses P_x over P_z therefore $\exists$ a path P'_z visible to r_a with $exitN(P'_z) = p'_z$ having $nextAS(P'_z) = nextAS(P_z)$, $med(P'_z) < med(P_z)$ and $cost(sp(r_a, p'_z)) \geq cost(sp(r_a, p_x))$. But since r_a, p_x are not in the same cluster, therefore condition (*iii*) of Theorem 5.1 ensures that r_a, p'_z are also in different clusters. Now r_a knows about path P'_z means that P'_z is announced by some reflector, therefore r_z should also know about it. And if r_z knows about P'_z then it will never choose path P_z, so there will not be any loop. Hence $c_1 + a_2 \leq d_1$, i.e., equation (8) holds.

$$b_2 < c_1 \tag{9}$$

This is true because $r_y, c_y \in C_y$ and $r_a \notin C_y$.

Using equations (7)-(9), we get

$$\begin{aligned} & c_1 + a_2 \leq a_1 + b_1 \\ \therefore \quad & a_2 + b_2 < a_1 + b_1 \end{aligned} \tag{10}$$

So for both the cases in sections VII-A.1 and VII-A.2, from equations (6) and (10), for *ith link* in the loop, we have

$$a_{i+1} + b_{i+1} < a_i + b_i \tag{11}$$

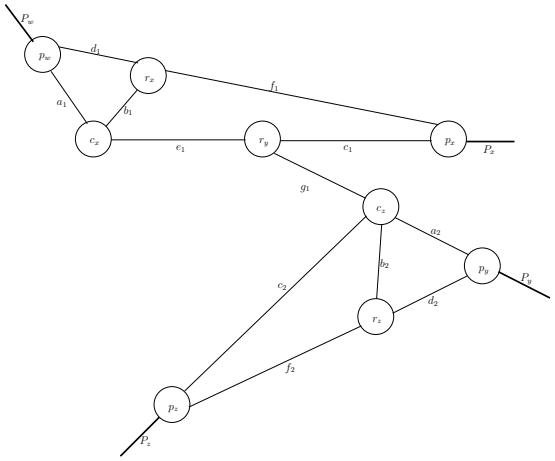

Fig. 10. Loop due to deflection at clients and reflectors

Now if the loop contains n deflections then using equation (11), we get

$$a_1 + b_1 > a_2 + b_2 > \cdots > a_n + b_n > a_{n+1} + b_{n+1} \tag{12}$$

But since this is a loop with n links, the $(n+1)$th link is same as the first link. Using this along with equation (12), we get

$$a_1 + b_1 > a_{n+1} + b_{n+1} = a_1 + b_1 \tag{13}$$

This is a contradiction, hence we see that, if the IBGP configuration satisfies the conditions stated in Theorem 5.1, then loops with deflections at client nodes only are not possible.

B. Deflections at both Clients and Reflectors

Now we look at the case when the forwarding loop has deflections at both clients as well as reflectors. By Lemma 7.3 we know that if a forwarding loop has a deflection at some reflector then it should be preceded and succeeded by deflections at client nodes. We study the structure of such deflections. Consider deflection at client c_x followed by deflection at reflector r_y followed by deflection at client c_z. We note the following about the structure of the part of the loop under study.

- Let $c_x \in C_x$, $r_y \in C_y$ and $c_z \in C_z$. By Lemma 7.5, c_x is the point of entry to C_x, r_y is the point of entry to C_y and c_z is the point of entry to C_z.
- Let $bestP(c_x) = P_x$ with $exitN(P_x) = p_x$, $bestP(r_y) = P_y$ with $exitN(P_y) = p_y$ and $bestP(c_z) = P_z$ with $exitN(P_z) = p_z$.
- $p_x, p_y, p_z \notin C_x, C_y, C_z$. Using the fact that c_x and r_y are points of entry to their respective clusters and the fact that $r_y \in sp(c_x, p_x)$, we see that $sp(c_x, p_x)$ goes through more than one cluster. So by condition (*iii*) of Theorem 5.1, p_x and c_x are in different clusters. This means that c_x learns about P_x through some reflector $r_x \in C_x$. Similarly c_z learns about P_z through some reflector $r_z \in C_z$. Now note that since for reflector r_x selects path P_x having $exitN(bestP(r_x)) = p_x \notin C_x$, therefore by Lemma 7.4 $\exists$ no reflector r in the AS such that $exitN(bestP(r)) \in C_x$. And therefore no node $\notin C_x$ learns about any path P' with $exitN(P') \in V_x$.

So $p_y, p_z \notin V_x$. Arguing similarly we get $p_x, p_y, p_z \notin V_x, V_y, V_z$.

- As mentioned in the previous point, c_x learns about P_x through reflector $r_x \in C_x$ and c_z learns about P_z through reflector $r_z \in C_z$.
- The presence of loop requires that $\exists$ node u_w having $bestP(u_w) = P_w$ with $exitN(P_w) = p_w \notin C_x, C_y, C_z$ and $c_x \in sp(u_w, p_w)$.
- The presence of loop also requires that $r_y \in sp(c_x, p_x)$ and $c_z \in sp(r_y, p_y)$.
- Note that node $r_x \notin sp(c_x, r_y)$. This is because $r_y \in sp(c_x, p_x)$ and if $r_x \in sp(c_x, p_x)$ that means $r_y \in sp(r_x, p_x)$. But since r_x and r_y are IBGP peers, by Lemma 7.2 deflection cannot occur at r_y.

The structure of such deflections is shown in the Fig. 10.[22] The line joining two nodes represents the shortest path between the nodes.

We can now easily get the constraints on IGP costs presented in equations (14)-(19).

$$f_1 \leq d_1 \tag{14}$$

The reasoning for equation (14) is similar to that for equation (2).

$$d_1 \leq a_1 + b_1 \tag{15}$$

Equation states that
$cost(sp(r_x, p_w)) \leq cost(sp(r_x, c_x)) + cost(sp(c_x, p_w))$.

$$e_1 + c_1 \leq b_1 + f_1 \tag{16}$$

This is because $r_y \in sp(c_x, p_x)$ and $r_x \notin sp(c_x, p_x)$.

$$b_1 < e_1 \tag{17}$$

This is because $r_x, c_x \in C_x$ and $r_y \notin C_x$.

$$g_1 + a_2 < c_1 \tag{18}$$

Here we use the fact that $c_z \in sp(r_y, p_y)$. The rest of the explanation is similar to that for equation (2).

$$b_2 < g_1 \tag{19}$$

This is because $r_z, c_z \in C_z$ and $r_y \notin C_z$.

Using equations (14)-(17), we get

$$\begin{aligned} & f_1 \leq a_1 + b_1 \\ \therefore \quad & e_1 + c_1 \leq b_1 + a_1 + b_1 \\ \therefore \quad & c_1 < a_1 + b_1 \end{aligned} \tag{20}$$

Using equations (18) and (19), we get

$$a_2 + b_2 < c_1 \tag{21}$$

Now equations (20) and (21) give

$$a_2 + b_2 < a_1 + b_1 \tag{22}$$

So we see that even if there is a deflection at a reflector between deflections at two client nodes (say ith and $i+1$th client nodes), equation (11) still holds. And therefore for any loop containing deflections at both clients and reflectors, the contradiction shown in equation (13) can still be achieved.

Hence we see that any kind of forwarding loops occurring due to path deflections are not possible if the IBGP configuration satisfies the conditions stated in Theorem 5.1.

[22]Node u_w in not shown in the figure.

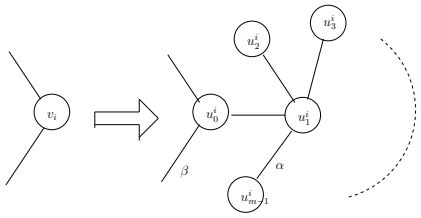

Fig. 11. Replace each node in V by a group of m nodes in V'

VIII. Complexity Analysis

Now the problem that we face is that given an IGP routing graph for an AS, we want to construct the IBGP configuration which guarantees the absence of any persistent routing oscillations and loops and is optimal in some sense and meets the given resource constraints[23]. We extend the idea used in [20][21] for evaluating the cost (and thereby determining the optimality) of a logical (IBGP) configuration. We define the *size* of the IBGP graph as the sum of all the shortest path IGP costs which constitute the IBGP links. Now the problem is to design an IBGP configuration satisfying all the conditions of Theorem 5.1 (this guarantees the absence of persistent oscillations and forwarding loops), for a given IGP routing graph, such that the size of the logical graph is minimum.

Lemma 8.1: Constructing IBGP configuration having minimum size while satisfying the conditions of Theorem 5.1 is NP hard.

Proof: Given a simple, undirected graph $G = (V, E)$, finding its *minimum vertex cover* is NP hard. We shall prove that finding the minimum size IBGP configuration while making sure that the conditions of Theorem 5.1 are satisfied is at least as hard as finding the minimum vertex cover. Consider any graph $G = (V, E)$ with k nodes $V = \{v_0, \ldots, v_{k-1}\}$. We construct a weighted, undirected graph $G' = (V', E')$ from G by replacing each node $v_i \in V$ by a *group* of m nodes $V_i = \{u_0^i, \ldots, u_{m-1}^i\}$ as shown in Fig. 11. We assume that $m > k+1$. Each group V_i has a *star* topology with node u_1^i at the center of the star and nodes $u_0^i, u_2^i, u_3^i, \ldots, u_{m-1}^i$ as the edge nodes. Now $V' = V_0 \bigcup V_1 \cdots \bigcup V_{k-1}$ and $\forall$ edges $v_i v_j \in E$ we have edges $u_0^i u_0^j \in E'$ (these are all the *inter-group edges* between $V_i, V_j \ \forall \ i, j$). E' also includes *intra-group edges* between the center node and the edge nodes (i.e., for group V_i there are edges between node u_1^i and the nodes $u_0^i, u_2^i, u_3^i, \ldots, u_{m-1}^i$, as shown in Fig. 11). We assume that all the inter-group edges have weights β and all the intra-group edges have weights α, with $\beta > 2\alpha$. We assume G' to be the IGP routing graph for some AS. We also assume that there are no resource constraints, i.e., at any node, we can have as

[23]As discussed in [20] [21], in real ASes there is a limit on the number of IBGP sessions that a node can support at a time (due to the resource constraints on the nodes). This limit may be different on different nodes and any valid IBGP configuration should respect this constraint at all the nodes.

many IBGP sessions as we like. Now we assert that even if all the paths are through different $nextAS$es, finding the optimal IBGP configuration based on conditions of Theorem 5.1 is equivalent to finding the vertex cover of original graph G. We can see that due to condition (*iii*) of Theorem 5.1 and the fact that $\beta > 2\alpha$ the only valid cases of IBGP configuration for G' can be:

(*i*) All the nodes $\in V'$ are in a single cluster.
(*ii*) All the nodes are in separate clusters, i.e., each cluster has only one node.
(*iii*) Each group of nodes is a separate cluster, i.e., we have k clusters $V_0, \ldots, V_{k-1}$, each having m nodes.

Clearly for any graph G', the size of IBGP graph having each node as a separate cluster (case (*ii*)) is greater than size of IBGP having all the nodes in one single cluster (case (*i*)). Now we will show that case (*i*) (only one cluster) is also not optimal. Let the IBGP configuration be of the form described in case (*i*), i.e., having only one cluster. Clearly the configuration with minimum size should have only one reflector. Let the reflector be in group V_i. We can see that the IBGP configuration formed by selecting u_0^i as the reflector should be smaller in size than the IBGP configuration formed by selecting reflector from nodes $u_2^i, \ldots, u_{m-1}^i$. Let $d_{ij} = cost(sp(u_0^i, u_0^j))$. Now if we select u_0^i as the reflector then we can calculate the size of the IBGP graph to be:

$$\begin{aligned} S(G') &= \sum_{j=0, j\neq i}^{k-1} m d_{ij} + k(\alpha + (m-2)2\alpha) \\ &= \sum_{j=0, j\neq i}^{k-1} m d_{ij} + k(2m\alpha - 3\alpha) \end{aligned} \quad (23)$$

And if we select u_1^i as the reflector then the size of the IBGP graph is:

$$\begin{aligned} S(G') &= \sum_{j=0, j\neq i}^{k-1} m d_{ij} + (k-1)(\alpha + (m-2)2\alpha + m\alpha) \\ &\quad +(m-1)\alpha \\ &= \sum_{j=0, j\neq i}^{k-1} m d_{ij} + (k-1)(2m\alpha - 3\alpha) \\ &\quad +(km-1)\alpha \end{aligned} \quad (24)$$

Now note that $km\alpha - \alpha > (2m\alpha - 3\alpha)$ as long as $k \geq 2$, i.e., when we have more than one group of nodes in graph G' (i.e., more than one node in G, which is the non-trivial case), the size of IBGP graph with only one cluster is smaller if u_0^i is selected as the reflector rather than u_1^i. Hence if the IBGP configuration is to have only one cluster then we should select some u_0^i as the reflector, where i is such that $\sum_{j=0, j\neq i}^{k-1} m d_{ij}$ is minimum.

If we assume that the IBGP configuration is constructed according to case (*iii*), i.e., each group of nodes is a separate cluster, then we have k clusters $V_0, \ldots, V_{k-1}$. Let u_0^i be reflectors $\forall i$. Now we can see that the size of this IBGP graph is:

$$\begin{aligned} S(G') &= \sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1} d_{ij} + k(\alpha + (m-2)2\alpha) \\ &= \sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1} d_{ij} + k(2m\alpha - 3\alpha) \end{aligned} \quad (25)$$

Now we show that when $m > k$, the RHS of (25) is less than the RHS of (23) by proving the following.

$$\min_{\forall i}\left\{ \sum_{j=0, j\neq i}^{k-1} m d_{ij} \right\} > \sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1} d_{ij} \quad (26)$$

Without loss of generality, we can assume that $i = 0$ minimizes the LHS of (26). Now we can see that:

$$\begin{aligned} \sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1} d_{ij} &\leq \sum_{i=0}^{k-1}\sum_{j=i+1}^{k-1} (d_{0i} + d_{0j}) \\ &= \sum_{i=0}^{k-1}\left((k-i-1)d_{0i} + \sum_{j=i+1}^{k-1} d_{0j} \right) \\ &= (k-1)(d_{01} + \cdots + d_{0k-1}) \\ &= (k-1)\sum_{j=1}^{k-1} d_{0j} \\ &< k\sum_{j=1}^{k-1} d_{0j} \end{aligned} \quad (27)$$

The first step in (27) follows from the following two properties of the graph G':

(*i*) Triangle equality holds, i.e., $d_{ik} \leq d_{ij} + d_{jk}\ \forall\ i, j, k$.
(*ii*) Graph weights are symmetric, i.e., $d_{ij} = d_{ji}$.

Using (27) it is easy to show that if $m > k$, then (26) holds. But since $m > k$ in graph G' due to our construction, the IBGP configuration with each group of nodes as a separate cluster and nodes u_0^i as reflectors has smaller size than any IBGP configuration with only one cluster. So IBGP configuration with one cluster is not the minimum size configuration, and we should construct IBGP graph such that each group of nodes is a separate cluster (case (*iii*)). Now the question is which of the nodes should be reflectors. As observed earlier, the IBGP configuration formed by selecting u_0^i as the reflector in all the clusters V_i should be smaller in size than the IBGP configuration formed by selecting any other node from $u_2^i, \ldots, u_{m-1}^i$ as the reflector in cluster V_i. If in some cluster V_j we select node u_1^j instead of node u_0^j as the reflector, then the change in the size of the resulting IBGP configuration is $((k-1)\alpha + (m-1)\alpha) - (\alpha + 2(m-2)\alpha) = (k+1-m)\alpha$. So if $m > k+1$, which is the case in graph G' due to our construction, then it is better to select u_1^i as reflector rather than any other node in cluster V_i. Now it is clear that the optimal IBGP configuration for graph G' should have clusters $V_0, \ldots, V_{k-1}$ and we should try to select nodes u_1^i as the reflectors. But according to condition (*iv*) of Theorem 5.1, in each inter-cluster link, we need at least one of the nodes to be a reflector. So we want to pick minimum number of reflectors of form u_0^i such that we cover all the inter-cluster links and

the reflectors in the other clusters should be of the form u_1^i. Note that this is the vertex cover for the original graph G. Hence the problem is at least as hard as the minimum vertex cover for general graphs. ■

IX. Algorithm

In section VIII, we proved that the problem of obtaining a minimum size IBGP configuration satisfying the conditions of Theorem 5.1 is NP hard. Clearly this is not desirable, but still it is not as bad as the case in [9][12], where the authors prove that even detecting anomalies due to MED and IBGP path asymmetries is NP hard.

In this section we formulate our problem as an Integer Linear Program (ILP). Although this may not be an attractive solution since solving the ILP might take exponential time, the ILP itself can be used as a starting point for obtaining other intelligent heuristics. For example, the ILP can be relaxed to a Linear Program (LP) and then some intelligent rounding can be used. The ILP formulation helps in understanding the structure of the problem, and the insights can then be applied to design other approaches such as tabu search, simulated annealing etc. The study of these alternate approaches is outside the scope of this work.

Now we give the ILP formulation of the problem. Let $\mathcal{I} = \{0, 1, \ldots, N-1\}$, where N is the number of BGP speakers in the AS. We define the following binary variables $\forall\ i, j \in \mathcal{I}$:

$$x_i = \begin{cases} 1 & \text{if node } i \text{ is a reflector} \\ 0 & \text{if node } i \text{ is a client} \end{cases}$$

$$c_{ij} = \begin{cases} 1 & \text{if nodes } i, j \text{ belong to same cluster} \\ 0 & \text{otherwise} \end{cases}$$

$$s_{ij} = \begin{cases} 1 & \text{if nodes } i, j \text{ are IBGP peers} \\ 0 & \text{otherwise} \end{cases}$$

We assume the following quantities as given:

δ_{ij}: IGP weight for shortest path between nodes i, j

α_i: max. IBGP connections permissible at node i

$$\gamma_{ij} = \begin{cases} 1 & \text{if } ij \text{ is a link in the IGP connectivity graph} \\ 0 & \text{otherwise} \end{cases}$$

$$\sigma_{ij} = \begin{cases} 0 & \text{if node } i, j \text{ learn about paths through} \\ & \text{same } nextAS \text{ via EBGP peers} \\ 1 & \text{otherwise} \end{cases}$$

We also define a function $\phi : \Re \times \Re \longrightarrow \{-1, 1\}$ as:

$$\phi(x, y) = \begin{cases} 1 & \text{if } x \geq y \\ -1 & \text{otherwise} \end{cases}$$

We define the cost function W as the size of the logical graph:

$$W = \sum_i \sum_{j \neq i} s_{ij} \delta ij$$

Now our objective is to minimize cost W.

$$\min_{s_{ij}} W \tag{28}$$

And the optimization is subject to the following constraints.

$$x_i + x_j + c_{ij} \geq \gamma_{ij} \quad \forall i, j \tag{29}$$

$$x_i + x_j + c_{ij} \geq 2s_{ij} \quad \forall i, j \tag{30}$$

$$x_i + x_j + c_{ij} \leq 2s_{ij} + 1 \quad \forall i, j \tag{31}$$

$$(c_{ij} - c_{ik})\phi(\delta_{ij}, \delta_{ik}) \leq 0 \quad \forall i, j, k \tag{32}$$

$$s_{ij} \geq \sigma_{ij} \quad \forall i, j \tag{33}$$

$$\sum_{j \neq i} s_{ij} < \alpha_i \quad \forall i \tag{34}$$

$$\sum_{j \neq i} s_{ij} \geq 1 \quad \forall i \tag{35}$$

$$c_{ik} + c_{kj} - c_{ij} \leq 1 \quad \forall i, j, k \tag{36}$$

$$s_{ij} = s_{ji} \quad \forall i, j \tag{37}$$

$$c_{ij} = c_{ji} \quad \forall i, j \tag{38}$$

The set of equations (29) state that if nodes i, j are clients in different clusters then they cannot be neighbors in IGP connectivity graph (this is the condition (*iv*) of the Theorem 5.1). Together, the sets of equations (30) and (31) relate the variables x_i, x_j, c_{ij}, s_{ij} (basically the equations state that there is an IBGP session between nodes i, j only when either i, j form a client-reflector (or reflector-client) pair in the same cluster or both are reflectors). The set of equations (32) state that if nodes i, j are in same cluster but nodes i, k are not, then $\delta_{ij} < \delta_{ik}$ (this is the condition (*iii*) of the Theorem 5.1). Equations (33) state that if nodes i, j learn about paths through the same $nextAS$ via EBGP peers, then they should be IBGP peers (this is the condition (*i*) of the Theorem 5.1). Equations (34) take care of the resource constraint (maximum number of IBGP sessions permissible) at each node. The rest of the equations (35)-(38) makes sure that the solution is consistent with the IBGP constraints (like each node should have at least one IBGP session (35), if nodes i, j and nodes i, k are in same cluster then nodes j, k are also in the same cluster (36), and IBGP peering (37) and clustering (38) should be symmetric).

Note that this ILP is flexible in the sense that if we need some node (say node k) as a reflector (or client) then it can be easily incorporated in the ILP by adding $x_k = 1$ (or $x_k = 0$) in the list of constraints.

X. Conclusion

The two straightforward approaches to tackle the routing oscillations and loops due to MED and IBGP path asymmetries are either to modify the protocol or to configure the AS in an intelligent manner such that the anomalies are absent. In [16] Musunuri et al. takes the first approach and proposes changes in BGP. But we believe that due to the large-scale deployment of BGP, it will be difficult to incorporate any major changes in the protocol at this point of time. In this paper we followed the second approach and proved conditions on IBGP configuration which are easy to check and guarantee the absence of the anomalies due to MED attribute and path asymmetry. We also look into the time complexity of the problem of constructing an IBGP configuration with minimum size, while satisfying the conditions developed in the paper and some other resource constraints, for given IGP connectivity graph. We then give an

algorithm based on integer linear programming to solve the problem.

REFERENCES

[1] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP-4)," RFC 1771, March 1995.

[2] T. Bates, R. Chandra, and E. Chen, "BGP route reflection - an alternative to full mesh IBGP," RFC 2796, April 2000.

[3] P. Traina, D. McPherson, and J. Scudder, "Autonomous system confederations for BGP," RFC 3065, February 2001.

[4] R. Dube and J. G. Scudder, "Route reflection considered harmful," IETF," Internet Draft, May 1999.

[5] "Endless BGP convergence problem in cisco ios software release," Cisco Systems, Field Notice 15641, October 2000.

[6] D. McPherson, V. Gill, D. Walton, and A. Retana, "Border gateway protocol (BGP) persistent route oscillation condition," RFC 3345, August 2002.

[7] R. Dube, "A comparison of scaling techniques for BGP," in *Proceedings of ACM SIGCOMM*, July 1999.

[8] J. G. Scudder and R. Dube, "BGP scaling techniques revisited," in *Proceedings of ACM SIGCOMM*, October 1999.

[9] T. G. Griffin and G. Wilfong, "On the correctness of IBGP configuration," in *Proceedings of ACM SIGCOMM*, August 2002.

[10] A. Basu, C.-H. L. Ong, A. Rasala, F. B. Shepherd, and G. Wilfong, "Route oscillations in IBGP with route reflection," in *Proceedings of ACM SIGCOMM*, August 2002.

[11] D. Walton, D. Cook, A. Retana, and J. Scudder, "BGP persistent route oscillation solution," IETF," Internet Draft, July 2000.

[12] T. G. Griffin and G. Wilfong, "Analysis of the MED oscillation problem in BGP," in *Proceedings of IEEE International Conference on Network Protocols (ICNP)*, 2002.

[13] T. G. Griffin, F. B. Shepherd, and G. Wilfong, "The stable paths problem and interdomain routing," *IEEE/ACM Transactions on Networking*, April 2002.

[14] R. Musunuri and J. A. Cobb, "Scalable IBGP through selective path dissemination," in *Proceedings of IASTED International Conference on Parallel and Distributed Computing and Systems (PDCS)*, November 2003.

[15] H. Gobjuka, "Forwarding-loop-free configuration for ibgp networks," in *Proceedings of IEEE International Conference on Networks (ICON)*, September 2003.

[16] R. Musunuri and J. A. Cobb, "Complete solution to stable IBGP," in *Proceedings of IEEE International Conference on Communications*, June 2004.

[17] "BGP best path selection algorithm," Cisco Systems, Document 13753, May 2004.

[18] Examine BGP routes and route selection. Juniper Networks. [Online]. Available: http://www.juniper.net/techpubs/software/nog/nog-baseline/html/verify-bgp9.html

[19] A. Rawat and M. A. Shayman. Interesting examples of IBGP configuration. University of Maryland. [Online]. Available: http://www.enee.umd.edu/∼anuj/IBGP/examples.pdf

[20] L. Xiao, J. Wang, and K. Nahrstedt, "Optimizing IBGP route reflection network," in *Proceedings of IEEE International Conference on Communications*, 2003.

[21] ——, "Reliability-aware IBGP route reflection topology design," in *Proceedings of IEEE International Conference on Network Protocols*, 2003.