### Title

Eliminating Routing Loops and Oscillations in BGP Using Total Ordering




### Author

Garcia-Luna-Aceves, J.J.



### Publication Date

2022

Peer reviewed

# Eliminating Routing Loops and Oscillations in BGP Using Total Ordering

J.J. Garcia-Luna-Aceves
*Computer Science and Engineering Department*
*University of California, Santa Cruz*
Santa Cruz, CA, USA
jj@soe.ucsc.edu

***Abstract*—OPERA is a framework recently introduced that formalizes routing etiquettes based on path information. New rules derived from OPERA to provide total ordering among paths are added to the policy mechanisms used in IBGP and EBGP, which results in OPERA-based BGP (OBGP). OBGP is a complete loop-free inter-domain multi-path routing solution based on IBGP and EBGP. OBGP is proven to be stable and loop-free at every instant. Well-known examples of systems in which IBGP and EBGP do not converge are used to illustrate the benefits of OBGP.**

## I. Introduction

The Internet routing infrastructure has been able to scale over more than 40 years since its initial implementations in the early 1980s because of a number of visionary design choices made by its architects and developers over the years. One of these choices was to split the Internet into multiple autonomous systems (AS). Each AS is a collection of routing prefixes under the control of network operators working on behalf of a single administrative authority or domain, such that the same well-defined set of policies for routing are used throughout the AS. The topology of an AS consists of one or multiple computer networks connected with each other.

Given the Internet structure, routing in the Internet is hierarchical and supported by routing protocols that operate within an autonomous system (AS), and routing protocols that operate across ASes.

There are three key reasons why this hierarchical model [19] scales far better than treating the Internet as single computer network. First, signaling and topology changes affecting the computer networks in an AS do not percolate to the entire Internet. Second, the size of the routing table at each router does not grow linearly with the number of networks in the Internet. Last but not least, computer networks in different ASes can use different routing protocols better suited to the performance characteristics of the networks.

There are many protocols for routing within ASes, but only a few protocols have been proposed and implemented for routing among ASes. Today, BGP-4 [37] is the only protocol used for routing among ASes in the Internet, and consists of two components: Internal BGP (IBGP) and External BGP (EBGP). We will refer to BGP-4 simply as BGP.

Routing across ASes using BGP is meant to: allow the use of routing policies that take into account local preferences within each AS that need not be known by other ASes; and use signaling among ASes that results in all routers computing stable paths to destinations in different ASes.

We call this type of routing a **routing etiquette**, which we define as *"the code of polite behavior adopted and followed by all routers of a group."*

The polite behavior observed by routers aims to allow all routers to attain valid routes to destinations that need not be optimum in some system-wide sense. Just like an etiquette for spectrum utilization does not have to enforce specific rules of behavior at all radios or divulge the rules adopted by each radio to use the shared spectrum, a routing etiquette should not require routers to state their local preferences publicly or require all ASes to have the same routing preferences. This design intent has been part of the Internet since its inception.

Section II provides a summary of the large body of work related to routing across ASes over the years. The original goal of BGP was to solve the limitations of the Exterior Gateway (EGP) [38] by introducing path information in routing updates stating the ASes along paths to address ranges.

In the past, the use of path information has been limited to loop detection, and BGP is well-known to have non-termination and route oscillation problems. To date, however, only partial remedies have been proposed and some require changing the signaling of BGP.

Section III presents *OPERA-based BGP* (**OBGP**). OBGP consists of the systematic embedding of total-ordering rules in the policy mechanisms for routing used in BGP. OPERA (Ordered Path Etiquette for Routing Algebra) is a framework introduced in [12] to formalize the treatment of routing etiquettes that use path information. It describes the type of information, valid operations, and ordering relations that a routing etiquette should use to be stable and loop-free.

OBGP eliminates route oscillations and looping in Exterior BGP (EBGP) by enforcing a total ordering of routes across ASes while still allowing their selection to be done based on local preferences rather than global optimality criteria. OBGP eliminates oscillations and looping in Interior BGP (IBGP) by ordering the BGP speakers in the same AS so that route selection proceeds as if all BGP speakers were fully meshed.

No BGP signaling changes are needed to implement the proposed modifications. OBGP speakers can report a single route internally and externally even when they use multiple

routes locally, and no complex AS configurations are needed. A *designated reflector* is configured or elected among the route reflectors [4] in an AS to establish the total ordering of routes known in an AS and the rest of the routers in the same AS adopt its decisions.

Section IV proves that OBGP is loop-free at every instant and converges deterministically to valid paths.

Section V discusses well-known cases of route oscillation and non-deterministic convergence in EBGP and IBGP to illustrate the major advantages of OBGP.

Section VI summarizes our results.

## II. Related Work

### A. Evolution from EGP to BGP

The rationale for the way in which routing among ASes is accomplished today dates back to the origins of routing in the Internet and the introduction of the Exterior Gateway Protocol (EGP) in 1982 [38].

EGP was proposed to provide reachability information among ASes, such that routers running EGP (EGP routers) in an AS can determine whether destination network prefixes in a remote AS can be reached through a neighbor EGP router.

EGP consisted of three procedures: neighbor acquisition, neighbor reachability, and network reachability [30], [38], [39]. The first two were intended to allow EGP routers to find other EGP routers and determine whether the link to a neighbor EGP router is operational. The third procedure allowed EGP routers to exchange lists of address ranges that could be reached through neighbor EGP routers. These lists included a routing metric; however, it did not serve the same purpose that distance values serve in protocols for routing within ASes. EGP routers in different ASes could interpret routing-metric values differently, the maximum metric value of 255 indicated an unreachable destination, and other values were used to indicate policies and preferences.

EGP required the Internet topology to be acyclic and ASes to be connected through a backbone (e.g., the ARPANET in 1983). As the Internet grew in size and complexity, this engineered-topology approach could not be sustained. This resulted in the development of the Border Gateway Protocol (BGP), which was first specified in 1989 [23]. After several revisions [24], [25], [35], the current version of BGP was developed [37]. Today, BGP-4 is the only protocol used for routing among ASes in the Internet, and we will refer to it simply as BGP.

Like EGP, BGP provides a single path from an AS to a destination and allows BGP routers to use routing policies involving local preferences in the selection of paths, rather than a system-wide optimality criteria for the selection of paths. However, there are two key differences between EGP and BGP. First, BGP relies on TCP for reliable communication between routers running BGP, which we call BGP routers; this design choice simplifies the signaling involved in discovering and maintaining neighbor BGP routers. Second, BGP uses path vectors instead of distance vectors as the method to report reachability information. A routing update in BGP carries information about the AS path traversed by a routing update from an originating AS to a destination address range. This approach can *detect* routing-table loops when an AS is listed more than once in an AS path, and allows BGP to operate in topologies that need not be a tree. Unfortunately, simply using path vectors *does not avoid routing-table loops*, and as we prove subsequently, BGP suffers from route flapping and convergence problems precisely because it is not loop-free.

### B. BGP Analysis and Extensions

Several formal frameworks called *routing algebras* have been proposed to study the properties of routing protocols and to help develop new routing approaches in a more systematic way. There has been extensive work in this area (e.g., [2], [6], [13], [18], [40], [41]). Many studies have taken advantage of these frameworks to examine the dynamic behavior of inter-AS routing based on BGP and path-vector routing protocols in general (e.g., [14], [15], [21], [22], [43]). These works helped identify slow convergence, non-convergence, and route-oscillation problems in BGP and paved the way to the current understanding of the dynamics of path-vector protocols.

The type of solutions that have been proposed in the past to solve the non-convergence problems of BGP by means of extensions to or modifications of BGP can be characterized as static and dynamic approaches. A static approach relies on programs to verify ahead of time that routing policies do not contain policy conflicts that would prevent BGP from converging to stable routes. A routing policy is used only if oscillations are not observed in the analysis. Dynamic approaches add mechanisms to the signaling of BGP in order to reduce or eliminate route oscillations.

Griffin and Wilfong [14] provide a comprehensive analysis convergence-related static analysis of BGP routing policies. This work shows that the static analysis approach to the BGP convergence problem is not practical, because the complexity of statically checking routing policies is either NP-complete or NP-hard. This leads to the conclusion that only dynamic approaches to BGP convergence are practical. However, very limited work has been reported in this area.

Dynamic schemes include the use of such features as sender side loop detection (SSLD) [21], withdrawal rate limiting (WRATE) [21], consistency assertions [32], notifying the cause and origin of route changes [26], [33] , expediting the propagation of updates regarding deleted routes [5], attempting to limit route flapping [27], and propagating more than one route [8]. However, while these techniques can help improve the speed with which BGP converges to valid routes in some cases, none can guarantee convergence, avoid the occurrence of temporary routing-table loops, or ensure faster convergence. More recently, van Beijnum et al. [42] presented an approach to support multi-path routing in BGP by requiring BGP routers to communicate the routes with the longest AS-paths among the routes locally available for each destination.

Many studies have addressed the oscillations and looping problems of IBGP (e.g., [3], [16], [17], [31], [34], [44]). Over the years, the BGP specification has been augmented

to account for the use of route reflectors and has added more path attributes in an attempt to avoid routing loops due to route reflectors [4]. However, the proposed operation of BGP with route reflectors in [4] is still prone to route oscillations and loops.

The IBGP problem is due to the use of an incorrect hierarchical routing model. More specifically, BGP assumes a hierarchical routing model in which an AS should behave as a single virtual BGP speaker, at least within a finite time, and yet the IBGP specification allows route reflectors [4] to select routes according to their own local views of what routes should be preferred, rather than an AS-wide view of the same. It is well known that this leads to permanent oscillations and looping.

Previous approaches attempting to solve the IBGP routing problems in large ASes that are not fully meshed have focused on either properly configuring ASes (e.g., [34]), or requiring BGP speakers to communicate much more path information that may induce excessive overhead [3], [31].

### C. BGP Alternatives

The use of path vectors describing path information in routing updates is not unique to BGP. Many routing protocols have been proposed for intra-AS routing in wireless and wireline networks based on path information conveyed incrementally as distances to relays along the paths to destinations or link-state information for links in paths used to reach destinations. However, BGP was the first path-vector protocol proposed for inter-AS routing, and very few alternatives to BGP have been proposed for inter-AS routing, and these are either based on re-using BGP or maintaining AS topologies and controlling the data plane for packet forwarding.

The Inter-Domain Policy Routing (IDPR) [7] architecture adopted a "link state" approach for the support of inter-AS routing taking into account multiple types of service. Its topology model is based on domains that correspond to ASes and virtual gateways that correspond to groups of border routers. It combines source routing with policies advertised in routing updates. This approach to inter-AS routing did not receive much support, which we argue is due to its complexity and the need to modify the data plane.

The Inter-Domain Routing Protocol (IDRP) [1] was a protocol for inter-AS routing proposed as an international standard that includes BGP as a proper subset. Accordingly, it is subject to the same problems found in BGP.

MIRO [45] is a multi-path approach to inter-AS routing in which routers learn default routes through the existing BGP protocol, and arbitrary pairs of ASes negotiate the use of additional paths that are bound to tunnels in the data plane. Because of its dependency on BGP, MIRO is subject to the same non-convergence problems of BGP.

## III. OPERA-Based BGP (OBGP)

### A. Overview

The type of information, operations, and ordering relations used in OBGP are defined in the context of the Ordered Path Etiquette for Routing Algebra (OPERA) we introduced in [12] to formalize routing etiquettes that use path information.

We describe OBGP by stating the changes in the BGP policy mechanisms required to implement OBGP. We assume that the reader is familiar with the neighbor acquisition, neighbor reachability, and network reachability procedures of BGP, as well as the way in which IBGP and EBGP routers operate [4], [28], [37].

For brevity, we describe the policy mechanisms for routing used in BGP as consisting of: (a) An import transformation with which routes are accepted for consideration, (b) a preference function with which valid routes are compared and preferred routes are selected, and (c) an export transformation with which preferred routes are announced.

Like BGP, OBGP consists of Exterior OBGP (E-OBGP) and Interior OBGP (I-OBGP). OBGP speakers in different ASes share routing information using E-OBGP, and OBGP speakers in the same AS share routing information using I-OBGP. The design rationale for OBGP consists of using total ordering along loop-free paths announced across ASes as part of the policy mechanisms for routing, rather than simply using loop detection as BGP does today.

E-OBGP allows routers in the same AS to select a neighbor AS as a next hop to a destination located in other AS only if the path reported by the routers in the neighbor AS satisfies a label-based ordering condition.

Specifically, independently of local route preferences that routers in an AS $a$ may have, those router are allowed to consider a neighbor AS $n$ as a next hop to a destination $d$ in another AS only if AS $n$ reports a path to destination $d$ that: (a) Does not contain AS $a$ itself; and (b) either the number of AS hops of the path reported by AS $n$ is strictly smaller than the current path reported by AS $a$, or the two paths have the same AS-hop length but the identifier of AS $n$ is lexicographically smaller than the identifier of AS $a$.

The above is similar to the way in which several loop-free routing protocols for intra-domain routing (e.g., [11]) use sufficient conditions to avoid routing loops. Such conditions do not allow routers to select neighbors as next hops to destinations if the conditions are not satisfied, even if the neighbors offer what appears to be the shortest distances.

I-OBGP induces a total ordering of loop-free routes *and* at the same time makes all OBGP speakers in an AS converge to the same choice of loop-free routes to destinations in other ASes by making all OBGP speakers select routes as a single virtual OBGP speaker, even if an AS is organized into clusters and requires route reflectors [4].

Our approach in I-OBGP consists of treating the problem as a hierarchical loop-free routing problem. Specifically, one of the route reflectors in each AS is configured or elected to serve as the *designated reflector* for the AS, and routes are totally ordered in the AS based on the route selections made by its designated reflector.

There are many ways to establish hierarchical loop-free routing, especially using path vectors. However, our approach requires the least amount of changes to the current IBGP spec-

ification [4] by taking into account the following BGP design parameters: (a) IGP performance is of secondary importance to path selection across ASes; (b) the links between BGP speakers (route reflectors, border routers, and clients of route reflectors) are TCP connections that may involve long paths and may induce forwarding loops); (c) reducing signaling and storage overhead was a key reason for introducing route reflectors; (d) each cluster in an AS has a unique route reflector; and (e) a route update in IBGP carries a cluster list stating the clusters traversed by the update [4].

### *B. Terminology and Preliminaries*

We introduce some definitions and terminology to describe OBGP sucsintly.

$N$ is a set of nodes with a node corresponding to a router executing OBGP or the group of routers in an AS executing OBGP, and $E$ is the set of edges with each edge connecting two nodes. A node in $N$ is denoted by a lower-case letter, and a link between nodes $p$ and $q$ in $N$ is denoted by $(p, q)$. Nodes $p$ and $q$ are said to be immediate neighbors of each other if link $(p, q)$ exists. The set of nodes that are immediate neighbors of node $k$ is denoted by $N^k$.

Given that each node may have multiple paths to a destination $d$, the $j$th path from node $k$ to destination $d$ listed in no particular order is denoted by $P_d^k(j)$. If node $k$ has a single path to destination $d$, then that path is denoted by $P_d^k(1)$.

Path $P_d^k(j)$ can be viewed as the sequence of links along the path or the sequence of nodes along the path. Such a path can be denoted as the augmentation of a path $P_d^q(i)$ with link $(k, q)$ to node $q$; therefore, $P_d^k(j) = (k, q)P_d^q(i) = kP_d^q(i)$.

The next hop along path $P_d^k(n)$ from router $k$ to destination $d$ is denoted by $s_d^k(n)$. Hence, path $P_d^k(n)$ consists of the concatenation of the link $(k, s_d^k(n))$ with a path $P_d^{s_d^k(n)}(m)$ offered by $s_d^k(n)$ to $k$. Accordingly,

$$P_d^k(n) = (k, s_d^k(n))P_d^{s_d^k(n)}(m) = kP_d^{s_d^k(n)}(m)$$

$W$ is a set of link weights in which each link weight describes performance-based or policy-based characteristics of the link. The weight of the link from node $p$ to node $q$ is denoted by $w(p, p)$.

To simplify our description of OBGP, we make the restriction that $w(p, q) \in \mathbb{R}^+$ for any link $(p, p)$.

BGP uses path attributes in sequence to select preferred paths as part of the Decision Process (Section 9.1 of RFC 4271). Accordingly, we define the weight of a path in terms of a sequence of attributes as stated below.

*Definition 1:* **Path Weight:** The weight $\omega_d^k(n)$ of path $P_d^k(n)$ is defined to be a tuple with a finite number of attribute values associated with the path.

The ordered sequence of the $n$ attributes of a path weight is $A = \{a_1, a_2, ..., a_{|A|}\}$. The order followed in this sequence is given by the order in which the attributes are used to determine that a path has a smaller weight than another path, i.e., that a path is preferred over another path. The value of the $j$th attribute of path $P_d^a(n)$ is denoted by $a_j[P_d^a(n)]$.

We observe that the order relation $<$ defined for real numbers is valid for the values of any path attribute, because we can assume that attribute values can be expressed as integers or real numbers.

The following definition of path -weight preference simply reflects the way in which ties of available paths are broken during Phase 2 of the BGP Decision Process (Section 9.1.2.2 of RFC 4271).

*Definition 2:* **Path-Weight Preference:** A path $P_d^b(m)$ is preferred over path $P_d^a(n)$ if the following path-preference condition is satisfied:

$$\omega_d^b(m) < \omega_d^a(n) \equiv \exists j \leq |A| \ \Big[ \ \Big( \ a_j[P_d^b(m)] < a_j[P_d^a(n)] \ \Big) \ \wedge \ \Big( \forall i < j \ \Big[ \ a_i[P_d^b(m)] = a_i[P_d^a(n)] \ \Big] \ \Big) \ \Big]$$

*Definition 3:* **Path Label:** The path label of path $P_d^k(n)$ is denoted by $\ell_d^k(n)$, it is assigned by node $k$, and is the ordered sequence of node identifiers corresponding to the nodes along the path starting with node $k$ and ending with destination $d$.

$\mathcal{M}$ is the set of routing-metric values, where the routing-metric value of path $P_d^k(n)$ is denoted by $\mu_d^k(n)$ and is defined by the tuple $\mu_d^k(n) = [\ \omega_d^k(n),\ \ell_d^k(n)\ ]$.

$\mu_o$ is the initial path metric assigned to a known destination for which a path can be found. By definition, $\mu_o = [\ \omega_o,\ \ell_o\ ]$, where $\omega_o$ and $\ell_o$ are the initial path weight and path label associated with a known reachable destination, respectively.

$\mu_\infty$ is the routing-metric value assumed for an unreachable or unknown destination. By definition, $\mu_\infty = [\ \omega_\infty,\ \ell_\infty\ ]$, where $\omega_\infty$ and $\ell_\infty$ are the path weight and path label associated with an unknown or unreachable destination, respectively.

*Definition 4:* **Label-based Ordering:** Node $a$ is ordered along path $P_d^a(n) = aP_d^b(m)$ with respect to its next-hop node $b$ if

$$\boldsymbol{L}: \ \ell_d^b(m) \prec_\ell \ell_d^a(n) \ \equiv \ \Big( \ a \notin \ell_d^b(m) \ \Big) \ \wedge \ \Big( [|\ell_d^b(m)| < |\ell_d^a(n)|] \vee [(|\ell_d^b(m)| = |\ell_d^a(n)|) \wedge (b < a)] \Big) \tag{1}$$

For any three values $\ell_d^a(i)$, $\ell_d^b(j)$, and $\ell_d^c(k)$ with $a$, $b$, and $c$ being three different nodes, the following three properties follow from Definition 4:

**(1)** *Irreflexivity:* $\ell_d^a(i) \not\prec_\ell \ell_d^a(i)$

**(2)** *Transitivity:*

$[(\ell_d^a(i) \prec_\ell \ell_d^b(j)) \wedge (\ell_d^b(j) \prec_\ell \ell_d^c(k))] \rightarrow (\ell_d^a(i) \prec_\ell \ell_d^c(k))$

**(3)** *Totality:* $(\ell_d^a(i) \prec_\ell \ell_d^b(j)) \vee (\ell_d^b(j) \prec_\ell \ell_d^a(i))$

The irreflexivity, transitivity, and totality properties of $\prec_\ell$ are satisfied by the properties of the order relation $\leq$ defined over the set of positive integers, plus the facts that node identifiers are assigned uniquely to nodes and $(\ell_d^b(i) \prec_\ell \ell_d^a(j))$ implies that either $[|\ell_d^b(i)| < |\ell_d^a(j)|]$ or $[|\ell_d^b(i)| = |\ell_d^a(j)|] \wedge [b < a]$, with both the size of path labels and node identifiers being positive integers.

The importance of the above three properties is that the path labels defined in OBGP induce a total ordering among paths reported by nodes, which allows OBGP to be loop-free.

### C. External OBGP (E-OBGP)

We describe E-OBGP based on the changes needed to the import transformation, export transformation, and local preference function of BGP [37]. We assume that all routers in an AS $k$ advertise the same route to destinations in other ASes. The way in which this is attained within a finite time is described subsequently in the context of I-OBGP.

As it is the case of BGP, each router advertises one route to any given destination $d$ if it has at least one loop-free path to the destination, and sends the same routes to all or a subset of neighbor routers in other ASes.

Because each router in an AS can advertise at most one route to any destination, a router in AS $k$ cannot have more than one route to destination $d$ through a neighbor router in another AS $q$.

The route advertised by a router in AS $k$ to destination $d$ is denoted by $P_d^k[r]$ and its label is denoted by $\ell_d^k[r]$.

We denote by $P_{dq}^k$ the route to destination $d$ stored at a router in AS $k$ and reported by a router in AS $q$, and the corresponding path label is denoted by $\ell_{dq}^k$.

The set of path labels corresponding to loop-free routes for destination $d$ that are locally available at a router in AS $k$ is denoted by $\mathcal{L}_d^k$, and the set of ASes directly connected to AS $k$ is denoted by $A^k$. It follows that $\mathcal{L}_d^k = \{\ell_{dq}^k \mid q \in A^k\}$.

The maximum path label in $\mathcal{L}_d^k$ is denoted by $\ell_{dmax}^k$ and is such that

$$\forall\ \ell_{dq}^k \in \mathcal{L}_d^k - \{\ell_{dmax}^k\}\ \left(\ \ell_{dq}^k \prec_\ell \ell_{dmax}^k\ \right) \tag{2}$$

The path label of a non-existent path is $\ell_\infty$ and its size is defined to be $|\ell_\infty| = \infty$.

Given that path labels state the AS routes advertised by routers, it is possible to determine whether a path label is a subset of another label. We denote the case in which a label value $\ell_d^q[r]$ is contained in a label $\ell_{dy}^k$ stored locally at a router in AS $k$ by $\ell_d^q[r] \in \ell_{dy}^k$.

*1) Ordered Import Transformation:* Routers in an AS are allowed to accept routes for destinations in another AS only if they are ordered according to $\boldsymbol{L}$, and order those accepted routes stored locally according to $\boldsymbol{L}$.

When a router in AS $k$ receives a route update from a neighbor router in AS $q$ for destination $d$ with a path label $\ell_d^q[r]$, the ordered import transformation of OBGP consists of accepting $\ell_d^q[r]$ only if the reported label is totally ordered with respect to the current value of $\ell_d^k[r]$, which can be stated in terms of $\boldsymbol{L}$ as follows:

$$\boldsymbol{OB}_i : \ell_d^q[r] \prec_\ell \ell_d^k[r]. \tag{3}$$

If $\boldsymbol{OB}_i$ is satisfied, then the reported route from AS $q$ is accepted and $\ell_{dq}^k \leftarrow \ell_d^q[r]$. On the other hand, if $\boldsymbol{OB}_i$ is not satisfied, the reported route is not accepted. In this case, routers in AS $k$ set $\ell_{dq}^k \leftarrow \ell_\infty$.

In addition, once a route must be reset to $\ell_\infty$ locally or as a result of an update stating that value, a router in AS $k$ must reset the labels of those routes locally stored that contained the invalidated route.

Let $\ell_{dy}^k[old]$ and $\ell_{dy}^k[new]$ denote the previous and updated value of the label for the path $P_{dy}^k$ from AS $k$ to destination $d$ through AS $y$. A router in AS $k$ sets $\ell_{dv}^k[new] \leftarrow \ell_\infty$ if $(\ell_d^q[r] = \ell_\infty) \wedge \Big(\ell_{dq}^k[old] \in \ell_{dv}^k[old]\Big)$. This is done to cope with failures of sessions between ASes more efficiently.

*2) Ordered Export Transformation:* The ordered export transformation enables the use of multiple routes to destinations, without requiring that the routes have the same weights or AS-path lengths. This is accomplished by requiring that the route reported by a router in AS $k$ for destination $d$ must be the path corresponding to the maximum label among all the routes in $\mathcal{L}_d^k$.

The constraint imposed by the ordered export transformation for a router in AS $k$ to inform **all or only some of its neighbor routers** of a new route $P_d^k[r]$ for destination $d$ (depending on whether they are in provider, consumer or peer ASes) is:

$$\boldsymbol{OB}_e : \ell_d^k[r] = k\ \ell_{dmax}^k. \tag{4}$$

A router in AS $k$ sends an update message with a new route record for destination $d$ if the value of $\ell_d^k[r]$ changes. Furthermore, if $(\ell_{dq}^k = \ell_\infty)\ \forall \ell_{dq}^k \in \mathcal{L}_d^k$ at a router in AS $k$, then $\ell_d^k[r] \leftarrow \ell_\infty$ and the router must send an update message with a route withdrawal for destination $d$, because the router does not have a route to $d$ guaranteed to be loop-free.

The non-intuitive approach adopted in $\boldsymbol{OB}_e$ of having nodes communicate their longest paths to allow nodes to use multiple routes locally while having them communicate to their neighbors only a single path per destination was first proposed by van Beijnum et al. [42].

*3) Multi-Path Local-Preference Function:* E-OBGP allows routers to choose among accepted ordered routes according to local preferences defined by the OBGP local preference function.

The local preference function of OBGP includes the same steps as the steps taken during Phase 2 of the BGP Decision Process (Section 9.1.2.2 of RFC 4271) if the IBGP used in the AS is fully-meshed. Otherwise, the steps described in the next subsection should be adopted.

In addition to the steps needed to determine preferred totally-ordered paths, routers must maintain the set of locally-available routes for each destination, and determine the route that has the maximum path label as defined previously.

Hence, a router in AS $k$ must take two steps for each destination $d$:

1) Maintain the set of labels $\mathcal{L}_d^k$.
2) Update $\ell_{dmax}^k$ to be the maximum label in $\mathcal{L}_d^k$ each time an update is made to $\mathcal{L}_d^k$.

From Eq. (2) and the definition of $\boldsymbol{OB}_e$, it follows that

$$\forall\ \ell_d^k(i) \neq \ell_d^k[r]\ \left(\ \ell_d^k(i) \prec_\ell \ell_d^k[r]\ \right).$$

### D. Internal OBGP (I-OBGP)

Given the specification of E-OBGP in the previous subsection, I-OBGP can be described by means of the following additional changes needed to the ordered import transformation,

ordered export transformation, and multi-path local preference function introduced for E-OBGP.

*1) Designated Reflector:* Each AS organized into clusters with route reflectors, and a single route reflector is elected or configured to serve as the designated reflector for the AS.

If the designated reflector is elected, the election can be made very simple by choosing, for example, the reflector with the smallest identifier or the smallest. cluster identifier as the designated reflector. This can be done very quickly given that reflectors should be fully meshed with one another.

*2) Ordering of Internal Paths through Designated Reflector:* Each OBGP speaker orders the valid routes it receives giving preference to the routes that include the designated reflector of its own AS. Hence, the routes that are reflected across clusters in an AS are all based on the choices made by the designated reflector, rather than the local choices of clients or reflectors in different clusters, which have different local preferences and hence lead to conflicts.

To implement this ordering with minimum changes to the Route Reflection Method for IBGP (in RFC 4456, Section 8) and without changing IBGP signaling, the designated-reflector identifier is set to equal the identifier of the cluster where the designated reflector resides, and this identifier is then included in the cluster list defined in IBGP [4].

In I-OBGP, an OBGP speaker reports a single path to any destination and there is a single designated reflector in an AS. A path reported in I-OBGP consists of a component within the AS and an external component.

The methods defined for I-OBGP focus on the internal components of paths. To avoid confusion between internal and external paths, we denote by $I_d^k$ the internal component of a path from an OBGP speaker in cluster $k$ to a remote destination $d$.

We denote by $\lambda_d^k$ the cluster list carried in an I-OBGP update from an OBGP speaker in cluster $k$ reporting path $P_d^k$, and we use $\delta$ to denote the identifier of the cluster in which the designated reflector resides.

*Definition 5:* **Internal Label Ordering:** An OBGP speaker in cluster $a$ is ordered along internal path $I_d^a = aI_d^b$ with respect to its next hop in cluster $b$ of the same AS if

$$\begin{aligned} \boldsymbol{I}: \ \lambda_d^b \prec_\ell \lambda_d^a \ \equiv \qquad\qquad (5)\\ \Big( \ a \notin \lambda_d^b \ \Big) \ \wedge \Big( \ \Big[(\delta \in \lambda_d^b) \wedge (\delta \notin \lambda_d^a)\Big] \vee \\ \Big[\Big(( \ (\delta \in \lambda_d^b) \wedge (\delta \in \lambda_d^a) \ ) \vee ( \ (\delta \notin \lambda_d^b) \wedge (\delta \notin \lambda_d^a) \ )\Big) \wedge \\ \Big([|\lambda_d^b| < |\lambda_d^a|] \vee [(|\lambda_d^b| = |\lambda_d^a|) \wedge (b < a)]\Big)\Big] \ \Big) \end{aligned}$$

Eq. (5) states that a router in cluster $a$ can accept a route reported by a router in cluster $b$, provided that: (a) no loops occur based on the cluster list of the route, and (b) the route from cluster $b$ either includes the designated reflector or the route from $b$ is better in terms of length while neither or both of the routes in the two clusters include the designated reflector.

The reason for the ordering condition in Eq. (5) is that all routes traversing the clusters of an AS should be based on what the designated reflector of the AS perceives to be the best choices, rather than what individual OBGP speakers in different clusters perceive to be the best choices in their own clusters of the AS.

*3) Ordered Route-Reflection Method:* This method modifies RFC 4456, Section 8. When a route reflector reflects a route, it adds the local cluster identifier to the cluster list carried in the update or creates the cluster list if the routes does not carry one. The route reflector uses Eq. (5) to determine whether or not to accept the route.

In terms of RFC 4456, Eq. (5) simply restricts the way in which a route reflector *accepts routes* to establish ordering centered around the designated reflector.

*4) Multi-path Local-Preference Function:* Once ordering condition $\boldsymbol{I}$ is used to accept or reject routes in I-OBGP, the method used to implement preferences are the same as in the multi-path local-preference function discussed for E-OBGP.

Exemplary lists of of steps representing a valid preference function are stated in [9], [37] for BGP, and work correctly with the modifications needed to implement the ordering conditions introduced in E-OBGP and I-OBGP.

## IV. Correctness of OBGP

OBGP constitutes a routing etiquette in which a node gives a neighbor node enough information to maintain ordering among the routes selected by the same node or different nodes based on private policies. The following theorems show that OBGP is loop-free at every instant and converges deterministically to loop-free paths within a finite time, and the two definitions that follow are used in these theorems.

*Definition 6:* **Feasible Path:** A path to destination $d$ is said to be feasible if it does not contain any routing loop.

*Definition 7:* **Stability** (Convergence to Feasible Routes): A routing protocol is said to converge to feasible routes for a given destination $d$ after topology changes stop occurring at time $T$ if:
**(1)** For any destination $d$ that a router $k$ can reach, router $k$ obtains at least one path $P_d^k(n)$ within a finite time after $T$, such that $\ell_d^k(n)$ does not include any node identifier more than once and $\omega_d^k(n) < \omega_\infty$;
**(2)** for any unreachable destination $d$ for router $k$, router $k$ sets $\ell_d^k(1) = \ell_\infty$ and $\omega_d^k(1) = \omega_\infty$ within a finite time after time $T$; and
**(3)** router $k$ does not change the value of any $\mu_d^k(n)$ within a finite time after time $T$.

*Theorem 1:* E-OBGP is guaranteed to be loop-free if the ordering condition $\boldsymbol{L}$ is satisfied at every instant by every router for any destination $d$.

*Proof:* Assume that $\boldsymbol{L}$ is true but E-OBGP is not loop-free and a loop $L$ of $h$ hops is created at some point in time with $L = \{n(1) \rightarrow n(2) \rightarrow ... \rightarrow n(h-1) \rightarrow n(1)\}$.

Without loss of generality, assume that each node has a single path to $d$. Because $\boldsymbol{L}$ is true, it must be true that the following is true:

$$\ell_d^{n(1)} \prec_\ell \ell_d^{n(h-1)}; \ \ell_d^{n(i)} \prec_\ell \ell_d^{n(i-1)} \text{ for } 1 < i \leq h-1;$$

However, this is a contradiction, because it implies that $\ell_d^{n(i)} \prec_\ell \ell_d^{n(i)}$ for $1 \leq i \leq h-1$, which cannot be true because of the irreflexivity property of $\prec_\ell$. Therefore, the theorem is true. ■

*Theorem 2:* If E-OBGP ensures convergence to feasible routes for any destination $d$, the ordering condition $\boldsymbol{L}$ must be satisfied by every node within a finite time after topology changes stop occurring.

*Proof:* The proof is by contradiction. Assume that E-OBGP has converged to feasible routes at time $T$ but $\boldsymbol{L}$ is not satisfied.

From Definition 7, no node can change the path label of any path after time $T$ and no node can transmit a signaling message to update a path label. Hence, node $k$ cannot change the routing metric $\mu_d^k(n)$ of path $P_d^k(n)$ after time $T$.

Let $q$ be the next hop along path $P_d^k(n)$. Router $k$ must have used the routing metric reported by $q$ to select $q$ as its next hop along $P_d^k(n)$, and that routing metric corresponds to a path $P_d^q(m)$ from $q$ to $d$. Furthermore, $\mu_d^q(m)$ cannot change after time $T$.

Because $\boldsymbol{L}$ is not satisfied at time $T$, node $k$ can use $q$ as its next hop along path $P_d^k(n) = kP_d^q(m)$ at time $T$ while node $q$ uses node $n$ as its next hop along path $P_d^q(m) = qP_d^k(n)$ at time $T$. This is a contradiction, because then $P_d^k(n)$ and $P_d^q(m)$ cannot be feasible paths. ■

*Theorem 3:* If the ordering condition $\boldsymbol{L}$ is satisfied by every node for any destination $d$ within a finite time after topology changes stop occurring, then E-OBGP ensures convergence to feasible routes.

*Proof:* The proof is by contradiction. Let $T_s$ be the time when topology changes stop occurring. Because $\boldsymbol{L}$ must be satisfied within a finite time $T_o \geq T_s$, it must be true that Eq. (1) is satisfied at time $T_o$ by each node $k$ and its next hop along any path to any destination $d$ that is reachable. From Theorem 1, it follows that the preferred paths to $d$ at each node are feasible. On the other hand, because each node computes routes according to E-OBGP, no node needs to update any route to destination $d$ after time $T_o$ with each route being feasible, which is a contradiction to the assumption that some node is unable to converge to a feasible route to $d$. ■

The following corollary is a direct consequence of the previous theorems.

*Corollary 1:* E-OBGP is guaranteed to be stable if the ordering condition $\boldsymbol{L}$ is satisfied at every instant by every router for any destination $d$.

Given Theorems 1 to 3 and Corollary 1, the following theorem implies that E-OBGP is loop-free and that it must converge to loop-free routes to destinations if they exist, without ever creating a loop.

*Theorem 4:* Ordering along loop-free paths ($\boldsymbol{L}$) is satisfied at every instant if E-OBGP is executed correctly.

*Proof:* The proof is by contradiction, i.e., by showing that having both E-OBGP executed correctly and $\boldsymbol{L}$ not being satisfied by a router in an AS $k$ for a given destination $d$ at some point in time $T$ is a contradiction.

According to the correct operation of E-OBGP that is assumed, a router in AS $k$ either has no route to a destination $d$ and $\ell_d^k[r] = \ell_\infty$, or it has a route with $\ell_d^k[r] \prec_\ell \ell_\infty$. A router cannot negate the ordering constraint $\boldsymbol{L}$ in the first case, because it does not have any path to $d$. Therefore, the rest of the proof can focus on the second case.

Assume that a router $y$ in AS $k$ computes a finite route $P_d^k(n)$ to destination $d$ at time $T$ executing E-OBGP correctly but $\boldsymbol{L}$ is false. Because E-OBGP is executed correctly, it follows from the execution of the local-preference function at router $y$ that $\ell_d^k(n) = q\ell_{dq}^k$ with $q \in A^k$ and $\ell_{dq}^k \in \mathcal{L}_d^k$. Because router $y$ stores route $\ell_{dq}^k$, it follows from the execution of the ordered import transformation (Eq. (3)) that $\ell_{dq}^k = \ell_d^q[r] \prec_\ell \ell_d^k[r]$ when router $y$ accepts the route with label $\ell_d^q[r]$.

If router $y$ updates $\ell_{dmax}^q$as a result of the new route it accepts with label $\ell_d^q[r]$, it follows from the correct execution of the ordered export transformation (Eq. (4)) that either $\ell_d^q[r] \prec_\ell \ell_{dmax}^k \prec_\ell \ell_d^k[r]$ or $\ell_d^q[r] = \ell_{dmax}^k \prec_\ell \ell_d^k[r]$.

The previous three facts constitute a contradiction to the assumption that ordering along loop-free paths given by Eq. (1)) is not true at some point in time when router $y$ computes a new finite route $P_d^k(n)$. Therefore, the theorem is true. ■

Proving that ordering along loop-free paths ($\boldsymbol{L}$) is satisfied at every instant if I-OBGP is executed correctly follows much the same argument as in Theorem 4 and is omited due to space limitations Intuitively, we observe that condition $\boldsymbol{I}$ further restricts $\boldsymbol{L}$, which has been shown to render loop-free routing. Accordingly, OBGP (E-OBGP and I-OBGP) is loop-free and stable.

## V. Comparing OBGP with BGP

### A. Eliminating EBGP Looping and Route Oscillations

We illustrate the benefits of E-OBGP over EBGP using well-known examples of looping and route-oscillation problems in BGP for routing across ASes.

*1) BAD-GADGET System* [14]*:* This is a classic example of an unsolvable BGP system, with no execution of BGP being capable of arriving to a stable routing state.

Figure 1 illustrates the operation of E-OBGP in the BAD GADGET system using circles to represent ASes and capital letters to denote the AS identifiers, such that $A < B < C < D$ to correspond to the original example in [14]. An intended destination $d$ is located at AS $A$. In the BAD-GADET system, each AS has a local preference for the counter-clockwise route of two AS hops over all other routes to AS $A$. Hence, absent any ordering constraints, AS $D$ would prefer route $DCA$, AS $C$ would prefer route $CBA$, and AS $B$ would prefer route $BDA$. As it is described in [14], this leads to temporary routing-table loops and non-convergence in EBGP.

The initial updates communicated in E-OBGP by OBGP speakers are shown in Figure 1(a), with routers in ASes $B$, $C$ and $D$ announcing routes of one AS hop to AS $A$. Links that belong to paths that are not announced by nodes are shown with dashed lines, and links that are part of paths being announced by nodes are shown with solid lines.

Figures 1(b) and 1(c) show the routes announced by each AS after nodes process updates from neighboring ASes. The path announced by each node is shown in bold letters next to the node, and locally known valid paths are listed below the announced paths. Links corresponding to paths that are only locally known in an AS are indicated in dashed lines.

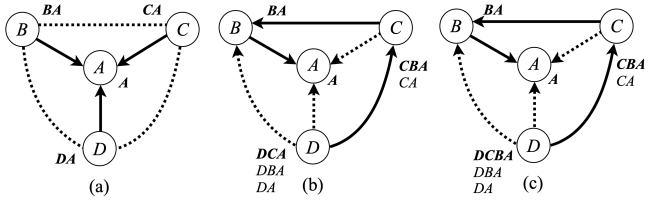


Figure 1: E-OBGP converges in the BAD-GADGET system

In E-OBGP, routers in AS $B$ are unable to enact the local preference of using the route initially announced by AS $D$ because $BA = \ell_d^B \prec_\ell \ell_d^D = DA$. However, routers in AS $D$ can use routes announced by routers in AS $C$ because $CA \prec_\ell DA$, and can also use routes announced by routers in $B$ if local preferences allow because $BA \prec_\ell DA$. Similarly, routers in AS $C$ can use the route announced by routers in AS $B$ because $BA \prec_\ell CA$. As a result, the system *converges deterministically* to one or multiple routes to the final state shown in Figure 1(c). This convergence is independent of how fast updates are propagated and no routing-table loops is ever created.

Because routers announce their largest preferred paths, routers in AS $B$ announce path $BA$, routers in AS $C$ announce path $CBA$, and routers in AS $D$ announce path $DCBA$. As the figure shows, routers in AS $D$ have three routes to $d$, and routers in AS $C$ have two routes to $d$.

*2) SURPRISE System* [14]*:* Some systems are solvable (i.e., can converge) in EBGP based on the initial topology on which activation sequences occur. However, as pointed out in [14], link or router failures may result in non-convergent systems in BGP. The SURPRISE system is an example of this case, and Figure 2 shows how E-OBGP converges deterministically in this system without creating routing loops. The figure shows in dashed lines links that are not part of paths announced by attached nodes, and in solid lines those links that are part of routes announced by nodes. Each subfigure shows one step taken by the nodes, which consists of processing all messages received in the previous step and announcing a new route.

Figure 2(a) shows the state of routers in all the ASes when the session between ASes $E$ and $F$ fails. The routes at ASes $E$, $D$, $B$, and $C$ are impacted by this event. Figure 2(b) illustrates the fact that routers in AS $E$ do not have any loop-free route to $d$ because none of the local choices satisfies $\boldsymbol{L}$. Accordingly, such routers must send updates with $\ell_d^E = \ell_\infty$.

Figure 2(c) shows that routers in ASes $C$, $B$ and $D$ determine that their reported paths to $d$ must be updated because they contain route $EF$ as part of their own reported routes; however, the routers in these ASes have alternate routes with labels that satisfy $\boldsymbol{L}$ and send the corresponding updates stating the routes with the maximum labels among those locally available.

As Figure 2(d) to 2(f) show, routers ASes $C$, $D$, and $E$ continue updating the largest paths they can announce while routers in ASes $B$ and $A$ have no new choices that satisfy $\boldsymbol{L}$. Eventually, all ASes converge to one or multiple routes to destination $d$ as shown in Figure 2(f). Even though the path information that routers in different ASes have may be inconsistent, routing loops are never created prior to convergence. In the worst case, routers in an AS (e.g., AS $E$ in this example) do not have valid routes to an intended destination for a short period of time, which is preferable to sending data traffic along loops and causing congestion across ASes.

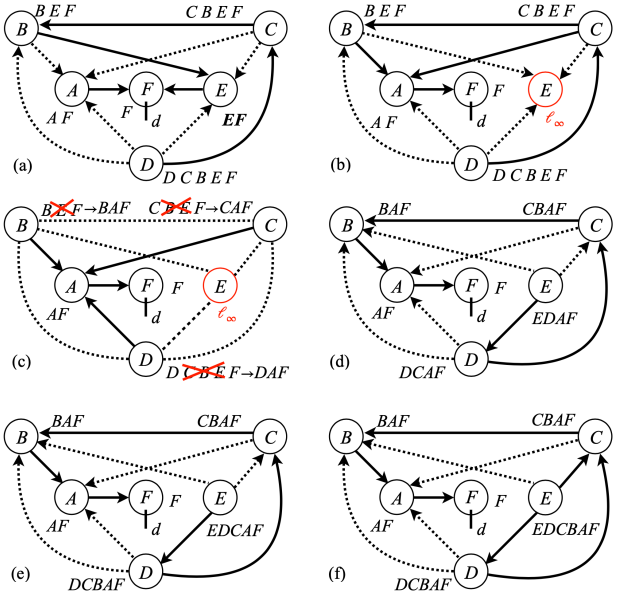


Figure 2: E-OBGP convergences after failures

### B. Eliminating IBGP Looping and Route Oscillations

We illustrate the fact that I-OBGP does not incur looping and oscillations using the system described in [3], [29].

Figure 3 illustrates a system in which IBGP with route reflection oscillates as described in Section 3 of [3]. Arrowheads indicate next hops along valid paths to destination $d$ at OBGP speakers in $AS0$. Solid arrowheads indicate the routes preferred by the designated reflector or reflected to other clusters. Arrowheads in dashed lines indicate valid paths known locally at different clusters.

As Figure 3 illustrates, I-OBGP converges deterministically to multiple paths at each OBGP speaker. The reason for this is that total ordering is maintained among routes reflected across clusters, and reflectors and clients of reflectors are required to adopt routes that include the routes chosen by the *designated reflector* $A$. As illustrated by the solid arrowheads in Figure 3, this results in route reflectors establishing a directed tree towards the cluster of the designated reflector, which then points out to one or multiple paths to destinations in remote ASes. OBGP speakers that are not border routers and are not in the same cluster of the designated reflector know only of paths to remote ASes that go to that cluster. Border routers may know local valid paths to remote ASes that do not involve the designated reflector but do not propagate them.

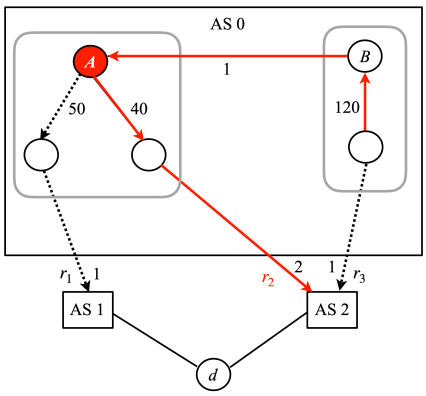


Figure 3: I-OBGP convergences deterministically and is loop-free in ASes with route reflectors

## VI. Conclusions

OPERA-based BGP (OBGP) provides stable, loop-free multi-path routing across ASes by means of modifications to the policy-routing methods used in EBGP and IBGP. OBGP ensures that ordering is always maintained among paths to destinations and all OBGP speakers in the same AS agree to use routes based on the routes that have been reflected by the designated reflector of the AS. A major advantage of I-OBGP and E-OBGP over all prior proposals attempting to solve the looping and convergence problems of BGP is that I-OBGP and E-OBGP can be deployed incrementally, because they do not change any of the signaling of BGP.

The routes used within an AS and across ASes in OBGP to ensure total ordering need not use available resources efficiently. However, the ordering conditions used in OBGP allow the use of path weights in addition to path labels. This enables a larger redesign of BGP in which path weights can be used on a transitive way, rather than just for local preferences.

## References


[1] ANSI, "Intermediate System to Intermediate System Inter-Domain Routing Information Exchange Protocol," ANSI Doc. X3S3.3/90-132, 1990.

[2] J.S. Baras and G. Theodorakopoulos, "Path Problems in Networks," *Synthesis Lectures on Communication Networks* (J. Walrand, Ed.), Lecture 3, Morgan & Claypool Pubs., 2010.

[3] A. Basu et al., "Route Oscillations in I-BGP with Route Reflection," *Proc. ACM SIGCOMM '02*, Aug. 2002.

[4] T. Bates, E. Chen, and R. Chandra, "BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)," RFC 4456, , April 2006.

[5] A. Bremler-Barr, Y. Afek, and S. Schwarz, "Improved BGP Convergence via Ghost Flushing," *Proc. IEEE INFOCOM 2003*, April 2003.

[6] B. Carre, *Graphs and Networks*, Clarendon Press, 1979.

[7] D. Estrin et al., "A Protocol for Route Establishment and Packet Forwarding across Multidomain Internets," *IEEE/ACM Trans. on Networking*, Feb. 1993.

[8] A. Flavel and M. Roughan, "Stable and Flexible iBGP," *Proc. ACM SIGCOMM '09*, Aug. 2009.

[9] Cisco Systems, "BGP Best Path Selection Algorithm," Document ID 13753, Sept. 2016.

[10] L. Gao and J. Rexford, "Stable Internet Routing without Global Coordination," *IEEE/ACM Trans. Networking*, 2001.

[11] J.J. Garcia-Luna-Aceves, "Loop-Free Routing Using Diffusing Computations," *IEEE/ACM Transactions on Networking,* Feb. 1993.

[12] J.J. Garcia-Luna-Aceves, "Stable, Loop-Free, Multi-Path Inter-Domain Routing Using BGP," *Proc. IEEE ICC '22 NGN)*, May 2022.

[13] M. Gouda and M. Schneider, "Maximizable Routing Metrics," *IEEE/ACM Trans. Networking*, Aug. 2003.

[14] T.G. Grifin and G. Wilfong, "An Analysis of BGP Convergence Properties," *Proc. ACM SIGOMM '99*, Aug. 1999.

[15] T.G. Griffin F. Bruce, and G. Wilfong, "Policy Disputes in Path-Vector Protocols," *Proc. IEEE ICNP '99*, Oct. 1999.

[16] T.G. Griffin, and G. Wilfong, "On the Correctness of iBGP Configuration," *Proc. ACM SIGCOMM '02*, Aug. 2002.

[17] T.G. Griffin, and G. Wilfong, "Analysis of the MED Oscillation Problem in BGP," *Proc. IEEE ICNP '02*, Nov. 2002.

[18] T.G. Griffin and J.L Sobrinho, "Metarouting," *Proc. ACM SIGCOMM '05*, Aug. 2005.

[19] C. Huitema, *Routing in the Internet*, Prentica-Hall, 1995.

[20] N. Kushman et al., "R-BGP: Staying Connected in a Connected World," *Proc. USENIX NSDI '07*, 2007.

[21] C. Labovitz et al., ,"Delayed Internet Routing Convergence," *Proc. ACM SIGCOMM 2000.* .

[22] C. Labovitz, et al., "The Impact of Internet Policy and Topology on Delayed Routing Convergence,' *Proc. IEEE INFOCOM 2001*, April 2001.

[23] K. Lougheed and Y. Rekhter, "A Border Gateway Protocol (BGP)," RFC 1105, June 1989.

[24] K. Lougheed and Y. Rekhter, "A Border Gateway Protocol (BGP)," RFC 1105, June 1990.

[25] K. Lougheed and Y. Rekhter, "A Border Gateway Protocol 3 (BGP-3)," RFC 1105, Oct. 1991.

[26] J. Luo et al., "An Approach to Accelerated Convergence for Path Vector Protocol," *Proc. IEEE Globecom 2002*, Nov. 2002.

[27] Z. Mao et al., "Route Flap Damping Exacerbates Internet Routing Convergence," *Proc. ACM SIGCOMM 2002*, Aug. 2002.

[28] J. Mauch, J. Snijders, and G. Hankins, "Default External BGP (EBGP) Route Propagation Behavior without Policies," RFC 4271, July 2017.

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

[30] D.L. Mills, "Exterior Gateway Protocol Formal Specification," RFC 904, April 1984.

[31] R. Musunuri and J.A. Cobb, "A Complete Solution for iBGP Stability," *Proc. IEEE ICC '04*, June 2004.

[32] D. Pei et al., ,"Improving BGP Convergence Through Consistency Assertions," *Proc. IEEE INFOCOM 2002*, June 2002.

[33] D. Pei et al., "BGP-RCN: Improving BGP Convergence through Root Cause Notification," *Computer Networks*, 2004.

[34] A. Rawat and M.A. Shayman, " Preventing Persistent Oscillations and Loops in IBGP Configuration with Route Reflection," *Computer Networks*, Dec. 2006.

[35] Y. Rekhter, "BGP Protocol Analysis," RFC 1265, Oct.1991.

[36] Y. Rekhter, "Experience with the BGP Protocol," RFC 1266, Oct.1991.

[37] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2005.

[38] E. Rosen, "Exterior Gateway Protocol (EGP)," RFC 827, Oct. 1982.

[39] E. Rosen, "Stub Exterior Gateway Protocol," RFC 888, Jan. 1988

[40] J. L. Sobrinho, "Network Routing with Path Vector Protocols: Theory and Applications," *Proc. ACM SIGCOMM '03*, Aug. 2003.

[41] J. L. Sobrinho, "An Algebraic Theory of Dynamic Network Routing," *IEEE/ACM Trans. Networking*, Oct. 2005.

[42] I. van Beijnum, J. Crowcroft, F. Valera, and M. Bagnulo "Loop-Freeness in Multipath BGP through Propagating the Longest Path," *Proc. IEEE ICC '09 Workshops*, 2009.

[43] K. Varadhan, R. Govindan, and D. Estrin, "Persistent Route Oscillations in Inter-Domain Routing," *Computer Networks*, Jan. 2000.

[44] D. Walton, D. Cook, A. Retana, and J. Scudder, "BGP Persistent Route Oscillation Solution," IETF Internet draft, May 2002.

[45] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain Routing," *Proc. ACM SIGCOMM '06*, 2006.