Attaining Stable and Loop-Free Inter-Domain Routing without Path Vectors

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

ABSTRACT

A sufficient condition for loop-free routing is introduced based on path labels. A path label consists of the identifier of the first node and hop-count length of a path to a destination. This condition is applied to the policy mechanisms used in BGP, which results in BGP-ELF (BGP Enhanced for Loop Freedom). BGP-ELF uses updates, queries, and replies based on path labels to attain multi-path loop-free and stable routing across autonomous systems without the need for path vectors.

CCS CONCEPTS

• **Networks** → **Network protocol design**;

KEYWORDS

inter-domain routing, BGP, path vectors, loop-free routing

ACM Reference Format:
J.J. Garcia-Luna-Aceves. 2022. Attaining Stable and Loop-Free Inter-Domain Routing without Path Vectors. In *ACM SIGCOMM 2022 Workshop on Future of Internet Routing & Addressing (FIRA '22), August 22, 2022, Amsterdam, Netherlands.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3527974.3545718

1 INTRODUCTION

The focus of this paper is on routing across ASes, and the only two protocols have been implemented for this purpose: The Exterior Gateway (EGP) [26] and the Border Gateway Protocol (BGP) [25].

EGP was the first protocol implemented for routing across ASes, and required ASes to be singly connected through a backbone (e.g., the ARPANET in 1983) in a way that the AS topology was acyclic. This was the case because routing loops and counting-to-infinity would occur in EGP otherwise. As the Internet grew in size and complexity, the engineered topology needed for EGP could not be sustained, and led to the development of BGP, which addressed the limitations of EGP by introducing path vectors. Today, BGP-4 [25] 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.

BGP allows 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, and its routing updates state the ASes along paths to address ranges that allow loop detection. However, BGP is known to have non-termination and route oscillation problems for which only partial remedies exist.

Section 2 provides a summary of prior work addressing the looping and convergence problems of BGP, and shows that this prior work assumes the use of path vectors.

Several routing protocols provide loop-free routing based on destination sequence numbers (e.g., DSDV [21]) or multi-hop router coordination (e.g., EIGRP [27]). However, these protocols are intended for routing within ASes and require the use of network-wide optimality criteria. Section 3 introduces a new sufficient condition for loop-free routing that enables the distributed selection of routes based on private policies that do not require routing metrics based on global optimality criteria.

Section 4 presents **BGP-ELF** (*BGP Enhanced for Loop Freedom*) based on the new sufficient condition for loop-free routing. BGP-ELF eliminates route oscillations and looping in BGP by replacing path vectors with labeled path lengths stating the lengths of paths and the identifier of the first ASes in the paths. In addition to updates, queries and replies are used to ensure that the new sufficient condition is always satisfied based on labeled path lengths. BGP-ELF allows route selection to be based on local preferences as in BGP, and BGP speakers can report a single route even when they use multiple routes locally. The proofs in Appendix A show that BGP-ELF is loop-free at every instant and that it converges deterministically to valid routes, without the need for complex AS policy configurations.

Section 5 discusses a well-known case of route oscillation and non-deterministic convergence in BGP to illustrate the major advantages of BGP-ELF. Section 6 summarizes our results.

2 RELATED WORK ON BGP

Several studies have examined the dynamic behavior of inter-AS routing based on BGP and path-vector routing protocols in general (e.g., [9, 10, 14, 15, 30]). 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. Grifin and Wilfong

FIRA '22, August 22, 2022, Amsterdam, Netherlands

ACM ISBN 978-1-4503-9328-7/22/08…$15.00
https://doi.org/10.1145/3527974.3545718

[9] 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.

Dynamic schemes include the use of such features as sender side loop detection (SSLD) [14], withdrawal rate limiting (WRATE) [14], consistency assertions [22], notifying the cause and origin of route changes [16, 23] , expediting the propagation of updates regarding deleted routes [3], attempting to limit route flapping [17], and propagating more than one route [5]. 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.

Many studies have addressed the oscillations and looping problems of IBGP (e.g., [1, 11, 12, 20, 24, 31]) in large ASes that are not fully meshed. The proposals to solve these problems have focused on either properly configuring ASes (e.g., [24]), or requiring BGP speakers to communicate much more path information that may induce excessive overhead [1, 20].

Recently, van Beijnum, Crowcroft, Valera, and Bagnulo [29] presented a counter-intuitive approach to support multi-path routing in BGP while allowing routers to announce a single path to a destination. The approach relies on the loop-detection mechanism of BGP; however, it requires each BGP router to communicate the route with the longest AS-path among the routes it considers to be valid to reach a given destination.

OBGP [8] extends the approach in [29] by introducing a new sufficient condition for loop-free routing based on the largest AS paths advertised by BGP routers. According to this condition, AS x can accept an AS path reported by AS y if: (a) AS x is not part of that AS path; and (b) either the new path has fewer AS hops than the path used by AS x, or the two paths have the same length but y is lexicographically smaller than x.

This review of prior approaches addressing the looping and non-convergence problems in BGP shows that all of them have focused on using path vectors. By contrast, this paper focuses on an approach that eliminates path vectors.

3 LOOP-FREE ROUTING WITH PRIVATE POLICIES

We first introduce some terminology to present our sufficient condition for loop-free routing with private policies.

N is a set of nodes, and a node corresponds to the routers executing BGP-ELF in an AS and E is the set of edges, with each edge connecting two nodes. A node in N is denoted by a lower-case letter, a link between nodes n and m in N is denoted by (n, m), nodes n and m are said to be immediate neighbors of each other, the set of nodes that are immediate neighbors of node k is denoted by N^k, and the nth path from node k to destination node d is denoted by $P_d^k(n)$. Path $P_d^k(n)$ can be viewed as the sequence of links along the path or the sequence of nodes along the path, and can be denoted as the augmentation of a path $P_d^q(i)$ with link (k, q) to node q; therefore, $P_d^k(n) = (k, q)P_d^q(i) = kP_d^q(i)$.

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

Definition 1. **Labeled Path Length:** *The labeled path length of $P_d^k(n)$ is denoted by $\ell_d^k(n)$, is assigned by the routers in AS k, and is defined to be the tuple $(k, h_d^k(n))$, where $h_d^k(n)$ is the number of AS hops in $P_d^k(n)$.*

This definition transforms a path vector to its hop length and the identifier of the first node along the path. By definition, $\ell_o = (D, 0)$ is the initial labeled path length associated with a known reachable destination d, where D is the AS of destination d, and $\ell_\infty = (nil, \infty)$ is the labeled path length for an unreachable destination.

Definition 2. **Ordering on Labeled Path Lengths:** *Node a is ordered along path $P_d^a(n)$ with respect to its next-hop node b along that path if*

$$\mathbf{L}: \ \ell_d^b(m) \prec_\ell \ell_d^a(n) \ \equiv \ [h_d^b(m) < h_d^a(n)] \vee [(h_d^b(m) = h_d^a(n)) \wedge (b < a)] \quad (1)$$

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 2:

(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 both the number of AS hops of a path and a and node identifier are positive integers.

The importance of the three properties of $\prec_\ell$ is that labeled path lengths can be used to induce a total ordering among the routes reported by nodes with no need to use a routing metric based on some optimality criteria (e.g., minimum distance). Hence, routers in an AS are free to select routes based on private policies defined for that particular AS, provided that their selections are constrained by $\boldsymbol{L}$. This total ordering eliminates the policy disputes that may occur in BGP and, as Theorem 1 shows, enables the design of a policy-based routing protocol that is provably loop-free.

Theorem 1. *A routing protocol is guaranteed to be loop-free if it ensures that the labeled-path ordering condition $\mathbf{L}$ is satisfied at every instant by every node for any destination d.*

Proof. Assume that $\boldsymbol{L}$ is true but the routing protocol 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: $\ell_d^{n(1)} \prec_\ell \ell_d^{n(H-1)}$ and $\ell_d^{n(i)} \prec_\ell \ell_d^{n(i-1)}$ 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. □

4 BGP-ELF

4.1 Overview

BGP-ELF uses the same signaling and policy mechanisms defined for BGP; hence, we only discuss the changes needed to transform BGP into BGP-ELF.

Due to space limitations, we assume that all routers in an AS k advertise the same routes to destinations in other ASes. This is the case if all ASes are fully meshed. 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 [2, 18, 25].

The three policy mechanisms for routing used in BGP are also used in BGP-ELF and can be viewed as:

- An import transformation with which routes are accepted for consideration.
- A preference function with which valid routes are compared and preferred routes are selected.
- An export transformation with which preferred routes are announced.

In a nutshell, BGP-ELF replaces path vectors with labeled path lengths described in Definition 1, and uses updates, queries and replies to ensure that $\boldsymbol{L}$ is always satisfied.

Nodes simply send updates with their labeled path lengths as long as they have neighbor nodes that satisfy $\boldsymbol{L}$ for a given destination. Otherwise, a node sends a query stating its current labeled path length to a destination and a *requested label*. This label is equal to the value of its own labeled path length prior to the input event that prompted the query.

BGP-ELF supports multi-path loop-free routing while allowing ASes to announce a single route to each destination without requiring the adoption of the same optimality criteria in all ASes. To accomplish this, BGP-ELF adopts the non-intuitive idea of communicating the longest route to a destination among all the valid routes available locally at a node first proposed by van Beijnum et al. in the context of BGP [29]. The resulting export transformation is modified by the use of labeled path lengths instead of path vectors.

A node that receives a query sends a reply if its next hop along the path corresponding to its reported labeled path length satisfies $\boldsymbol{L}$ with the value of the requested label stated in the query. The reply from the node states its own labeled path length and the requested label in the query. Otherwise, the node propagates the query specifying its own labeled path length and the requested label in the query it received.

A query is propagated towards the destination along the path corresponding to the reported labeled path lengths.

A node that forwards a reply states its own labeled path length and the requested label in the response it receives. Updates and replies are sent to all neighbors, and queries may be sent to all neighbors or a single neighbor.

The theorems proving that BGP-ELF is loop-free at every instant and that it converges deterministically to loop-free paths within a finite time are presented in Appendix A. In a nutshell, the theorems first establish that the signaling of BGP-ELF ensures that ordering on labeled path lengths is satisfied at every instant by every node, which renders loop freedom. Given that, they show that BGP-ELF disseminates valid values of labeled path lengths along loop-free paths, which renders deterministic convergence.

4.2 BGP-ELF Signaling

As it is the case for BGP, each router in BGP-ELF 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.

The labeled path length for destination d reported by the routers in AS k is denoted by $\ell_d^k[r]$, and defined to be $\ell_d^k[r] = (k, h_d^k)$, where h_d^k is the number of AS hops in the path to d. For simplicity, $\ell_d^k[r]$ is called the **reported label** by node k for destination d.

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. We denote by ℓ_{dq}^k the reported label for destination d sent by a router in AS q and maintained at the routers in AS k.

Conceptually, each node k maintains a Neighbor Table (NT^k) and a Routing Table (RT^k). NT^k stores the reported labels sent by each neighbor of node k. RT^k lists an entry for each destination d and states: The reported label ($\ell_d^k[r]$), a reference label (r_d^k), the set of next hops (S_d^k), and the next hop (s_d^k) along the path corresponding to $\ell_d^k[r]$. If there is no next hop to d, then $S_d^k = \emptyset$ and $s_d^k = 0$.

The value of the **reference label** r_d^k equals the value of $\ell_d^k[r]$ when node k has valid next hops to destination d, or the smallest value of a requested label stated in a query created or forwarded by the node. How a router in AS k uses the data in RT^k to populate its forwarding information base is outside the scope of this paper.

An update for destination d is denoted by $U(d, \ell_d^k[r])$; a query is denoted by $Q(d, \ell_d^k[r], \rho_d^k)$, where ρ_d^k is a labeled path length stated by the AS from which the the query originated; and a reply is denoted by $R(d, \ell_d^k[r], \rho_d^k)$, where ρ_d^k is copied from the query being answered. For simplicity, we refer to ρ_d^k as a **requested label**.

4.3 BGP-ELF Import Transformation

The import transformation of BGP is such that an AS k accepts a path reported by a neighboring AS q if k is not part of the path from AS q to the destination. BGP-ELF modifies this by applying $\boldsymbol{L}$ rather than the loop-detection mechanism based on path vectors. More specifically, routers in an AS are allowed to accept routes for destinations in other ASes only if they are ordered according to $\boldsymbol{L}$, and also order the routes they store locally according to $\boldsymbol{L}$.

When a router in AS k receives an update, query or reply from a neighbor router in AS q with a reported label $\ell_d^q[r]$ for destination d, the import transformation of BGP-ELF consists of accepting $\ell_d^q[r]$ only if the reported label is totally ordered with respect to the current value of its own reported label $\ell_d^k[r]$, which can be stated as follows:

$$\boldsymbol{BE}_i : \ell_d^q[r] \prec_\ell \ell_d^k[r]. \tag{2}$$

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

Once node k updates NT^k, it updates RT^k and takes different steps depending on its routing state. The routing state of routers in AS k is determined by the following condition:

$$\mathcal{T} : \left(\exists\, q \in N^k \left[\ell^k_{dq} \prec_\ell r^k_d\right]\right) \vee \left(\forall q \in N^k \left[\ell^k_{dq} = \ell_\infty\right]\right) \tag{3}$$

Condition $\mathcal{T}$ states that node k either has neighbors with reported labels smaller than its reported label, or all its neighbors have declared the destination to be unreachable.

A node is said to be **passive** if $\mathcal{T}$ is true and is **active** otherwise.

If node k is passive, then $r^k_d \leftarrow \ell^k_d[r]$. If it is active, then r^k_d is not updated and equals the last value of $\ell^k_d[r]$ when node k was passive. Node k sends an update, a query, or a reply depending on the the input event and whether it is passive or active after its routing table is updated.

Node k takes the following steps to process an update $U(d, \ell^q_d[r])$ or after detecting a change in the state of its link with neighbor q:

(1) Sends $U(d, \ell^k_d[r])$ if it remains or becomes passive.
(2) Originates $Q(d, \ell^k_d[r], \rho^k_d = r^k_d)$ if it becomes active.
(3) Sends $Q(d, \ell^k_d[r], \rho^k_d = r^k_d)$ if it remains active after the input event, at least one neighbor v has reported a finite distance, and h^k_d was updated.

Node k takes the following steps to process a reply $R(d, \ell^q_d[r], \rho^q_d)$ from neighbor q:

(1) Sends $R(d, \ell^k_d[r], \rho^k_d = \rho^q_d)$ if it either becomes passive and $\rho^q_d \leq r^k_d$, or it remains passive and either $\rho^q_d < r^k_d$ or the value of h^k_d was updated.
(2) Originates $Q(d, \ell^k_d[r], \rho^k_d = r^k_d)$ if it becomes active as a result of the reply from q; however, if $q \in S^k_d$ before the reply made node k become active, it updates $S^k_d \leftarrow \{q\}$, $h^k_d \leftarrow h^q_d + 1$.
(3) Stays silent if it was active before the reply from q and remains active.

Node k takes the following steps to process a query $Q(d, \ell^q_d[r], \rho^q_d)$ received from neighbor q:

(1) Sends $R(d, \ell^k_d[r], \rho^k_d = \rho^q_d)$ if it is passive and has a neighbor v such that $\ell^k_{dv} \leq \rho^q_d$.
(2) Forwards $Q(d, \ell^k_d[r], \rho^k_d = \rho^q_d)$ to its next hop s^k_d if it remains passive and has no neighbor v such that $\ell^k_{dv} \leq \rho^q_d$.
(3) Forwards query $Q(d, \ell^k_d[r], \rho^k_d = \rho^q_d)$ to all its neighbors if it becomes active or remains active and $\rho^q_d < r^k_d$, and sets $r^k_d \leftarrow Min\{r^k_d, \rho^q_d\}$.
(4) Stays silent if it is active before the query from q is received and all its neighbors have sent ℓ_∞ for destination d.

4.4 Multi-Path Local-Preference Function

BGP-ELF allows routers to choose among accepted routes according to local preferences defined by the local preference function, which consists of the same steps as those taken during Phase 2 of the BGP Decision Process (Section 9.1.2.2 of RFC 4271).

Let W be the set of link weights in which each link weight describes performance or policy-based characteristics of the link. The weight of the link from router i to router j is denoted by $w(i, j)$, and we make the restriction that $w(i, j) \in \mathbb{R}$ and $w(i, j) > 0$.

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 for BGP-ELF in terms of a sequence of attributes as stated below.

Definition 3. **Path Weight:** *The weight $\omega^k_d(n)$ of path $P^k_d(n)$ is defined to be a tuple with a finite number of attribute values associated with the path.*

The ordered sequence of the 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 jth attribute of path $P^a_d(n)$ is denoted by $a_j[P^a_d(n)]$.

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.

Definition 4. **Path-Weight Preference:** *A path $P^b_d(m)$ is preferred over path $P^a_d(n)$ if the following path-preference condition is satisfied:*

$$\omega^b_d(m) < \omega^a_d(n) \equiv \exists j \leq |A| \left[\left(a_j[P^b_d(m)] < a_j[P^a_d(n)] \right) \wedge \left(\forall i < j \left[a_i[P^b_d(m)] = a_i[P^a_d(n)] \right]\right)\right]$$

Definition 4 simply reflects Phase 2 of the BGP Decision Process stated in Section 9.1.2.2 of RFC 4271. Algorithm 1 in Appendix B shows a concrete example of a local-preference function in BGP-ELF that slightly modifies the local-preference function in BGP with the use of reported labels.

4.5 BGP-ELF Export Transformation

To support policy-based multi-path routing, routers maintain the set of locally-available routes for each destination. Furthermore, because at most one path to each destination can be shared across ASes, routers in an AS must determine the route with the longest labeled path length among all valid routes available locally according to Definition 5 stated below.

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

Definition 5. **Longest Labeled Path Length:** *The longest labeled path length in $\mathcal{L}^k_d$ is denoted by ℓ^k_{dmax} and is such that*

$$\forall\, \ell^k_{dq} \in \mathcal{L}^k_d - \{\ell^k_{dmax}\} \left(\ell^k_{dq} \prec_\ell \ell^k_{dmax} \right) \tag{4}$$

A router in AS k takes the following two steps for destination d:

(1) Maintains the set of labels $\mathcal{L}^k_d$ and update S^k_d (as next hops to d) to include those neighbors with labels in $\mathcal{L}^k_d$.
(2) Updates ℓ^k_{dmax} to be the longest label in $\mathcal{L}^k_d$ each time an update is made to $\mathcal{L}^k_d$.

The export transformation allows routers to share a single route to a destination and use multiple routes to destinations locally without creating routing loops. 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 $\mathcal{L}_d^k$.

In BGP-ELF, the constraint imposed by the export transformation for a router in AS k to inform **all or only some of its neighbor routers** of a new route for destination d (depending on whether they are in provider, consumer or peer ASes) is:

$$\boldsymbol{BE}_e : \ell_d^k[r] = (k,\ 1 + |\ell_{dmax}^k|) \tag{5}$$

where $|\ell_{dmax}^k|$ is the number of hops in ℓ_{dmax}^k.

The steps and signaling described in Section 4.3 are used together with the local use of multiple routes to destinations without incurring routing loops, because the test that $\boldsymbol{L}$ is satisfied at all times by any route used in any AS to reach any destination is done on the basis of the reported labels.

4.6 Extending BGP-ELF To Support Safe Route Filtering based on AS Classes

The reader familiar with RFC 7454 [4] may wonder how BGP-ELF can support route filtering based on the ASes in AS paths, given that path vectors are not used. Clearly, without AS paths included in updates, routes cannot be filtered on the basis of specific AS identifiers. However, it is important to point out that BGP implementations that allow BGP speakers to use multiple routes locally cannot enforce safe route filtering based on AS identifiers.

Figure 1 illustrates the safety problem of route filtering in BGP based on AS identifiers when multi-path routing is allowed. In the figure, circles represent ASes and capital letters denote AS identifiers. Destination d is located in AS F. The AS path advertised by each AS is stated next to the AS. Solid arrowheads correspond to AS hops that are part of advertised AS path, and dashed arrowheads represent AS hops in AS paths known locally at various ASes. Assume that AS D is required to filter out AS paths to destination d that include AS E or AS A as relays. Data from D to d can still traverse ASes A and E depending on the forwarding steps taken by ASes C and B.

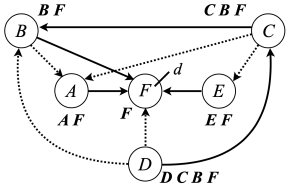

Figure 1: Unsafe route filtering in BGP

AS path filtering is really intended for the filtering of AS paths on the basis of the type of ASes (neighboring ASes or relays) that may be part of AS paths. This type of filtering can be supported in BGP-ELF by adopting a system-wide approach to the classification of ASes into classes, and by making minor changes in the signaling of BGP-ELF based on that approach. We summarize one of possibly many approaches.

ASes can be classified according to a globally-defined list of AS classes. Each AS class is denoted by an integer value from 1 to $|C|$, where $|C|$ is the total number of AS classes defined in the system. Accordingly, a class vector with a bit for each AS class can be used to denote the fact that an AS belongs to one or multiple AS classes.

The class vector of a given AS consists of the ordered sequence of bits $\{c_1, c_2, ..., c_{|C|}\}$, where $1 \le i \le |C|$ and $c_i = 1$ if the AS belongs to the ith AS class defined in the system.

Node k has a list of unwanted types of ASes for each destination d, which is denoted by the unwanted class vector u_d^k of with $|C|$ bits. The ith bit of this vector is denoted by $u_d^k(i)$ and $u_d^k(i) = 1$ if node k does not want routes to destination d that contain ASes belonging to AS class i.

Node k also has a class vector v_d^k with $|C|$ bits. The ith bit of this vector is denoted by $v_d^k[r](i)$ and $v_d^k[r](i) = 1$ if an AS in any of the AS paths used by node k to reach destination d belongs to AS class i.

The simplest way to modify the export transformation of BGP-ELF to account for filtering of routes based on AS classes consists of having a node k report all its routes to all its neighbors, and to let those neighbors filter out unwanted routes themselves based on their own preferences. This way, nodes sending updates do not have to keep track of the unwanted class vectors of all their neighbors.

The reported label from node k to destination d, $\ell_d^k[r]$, is augmented with an associated unwanted class vector u_d^k and a class vector v_d^k. An update, query or reply from node k regarding destination d includes only $\ell_d^k[r]$ and v_d^k, because nodes need not know the unwanted class vectors of their neighbors.

Node k stores the reported label and the class vector for destination d reported by each neighbor node. The class vector stored at node k and reported by node q for destination d is by v_{dq}^k, with $v_{dq}^k \leftarrow v_d^q$.

We use $u_d^k \cap v_{dq}^k = \bar{0}$ to denote the fact that $u_d^k(i) \cap v_{dq}^k(i) = 0$ for $1 \le i \le |C|$. Using this notation, the import transformation of BGP-ELF is extended as follows to filter out routes corresponding to AS paths that include ASes of unwanted AS classes:

$$\boldsymbol{BE}_i : \left(\ell_d^q[r] \prec_\ell \ell_d^k[r]\right) \wedge \left(u_d^k \cap v_{dq}^k = \bar{0}\right) \tag{6}$$

When node k accepts a route from neighbor q for destination d based on Eq. (6), it updates v_d^k with the bitwise OR of its own class vector and the class vector of the new route, i.e., $v_d^k \leftarrow v_{dq}^k \cup v_d^k$. This way, the updates, queries and replies sent by node k for destination d contain the most recent class vector associated with the reported label for the destination.

5 EXAMPLES OF BGP-ELF OPERATION

We illustrate how BGP-ELF operates and its benefits over BGP using a well-known example of looping and route-oscillation problems in BGP for routing across ASes.

5.1 BAD-GADGET System [9]

BAD GADGET is an example of an unsolvable BGP system, with no execution of BGP being capable of arriving to a stable routing state. Figure 2 illustrates BAD GADGET using the same type of depiction of ASes used in Figure 1. In this example, the lexicographic values

of AS identifiers are such that $A < B < C < D$, which follows the example in [9]. Destination d is assumed to be located at AS A.

In the BAD-GADET system, each AS has a local preference for the counter-clockwise route of with two 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 [9], this leads to temporary routing-table loops and non-convergence in BGP.

The reported labels that routers in one AS communicate to routers in neighboring ASes are indicated in Figure 2 by tuples (F, h) next to the ASes, where F is the first AS along the route to destination d and h is the number of AS hops traversed in the route. We also indicate the path corresponding to each reported label. In this example, $\mathcal{T}$ is always satisfied and hence routers only exchange updates for destination d that state their reported labels. For simplicity, the updates sent between ASes are not shown.

The initial updates communicated among routers are shown in Figure 2(a), with routers in ASes B, C and D announcing routes of one AS hop to AS A. Figures 2(b) and 2(c) show the routes announced by each AS after routers process updates from neighboring ASes, and routes that are only locally known in an AS are indicated in dashed lines.

Routers in AS B are not able to enact the local preference of using the route announced by AS D because $BA \equiv (B, 1) \prec_\ell (D, 1) \equiv DA$. As a result, AS B must continue to use the direct route to AS A. On the other hand, 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 and C if local preferences allow because $BA \prec_\ell DA$. Similarly, routers in AS C can use the route announced by routers in B because $BA \equiv (B, 1) \prec_\ell (C, 1) \equiv CA$.

The same argument stated above holds for the routes announced in Figure 2(c). The end result is that BGP-ELF *converges deterministically* to the final state shown in Figure 2(c) independently of how fast updates are propagated and without routing-table loops ever being created.

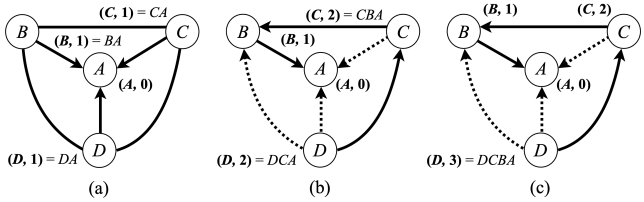

Figure 2: BGP-ELF in the BAD-GADGET system

5.2 Link Failure in BAD-GADGET System

BGP-ELF does not suffer from any non-termination problems resulting from resource failures, because it is loop-free at every instant and its use of total ordering among reported and stored routes leads to deterministic convergence. Figure 3 illustrates this point using a link failure in the BAD-GADET system as an example.

Figure 3(a) shows the initial state of the system when link (B, A) fails. The reported labels are indicated as in Figure 2, and updates, queries, and replies are denoted without indicating destination d.

Figure 3(b) shows that routers in AS B must become active and send query $Q(d, (nil, \infty), (B, 1))$ because $[(C, 2) \not\prec_\ell (B, 1)] \wedge [(D, 3) \not\prec_\ell (B, 1)]$ and hence $\mathcal{T}$ is not satisfied.

As Figure 3(c) shows, AS D remains passive after receiving the query from B and forwards the query to $s_d^D = C$, because its reported label $(D, 3)$ is not smaller (according to $\prec_\ell$) than the requested label $(B, 1)$. On the other hand, AS C sends a reply to AS B because its next hop to destination d is AS A and $(A, 0) \prec_\ell (B, 1)$.

The reply from AS C allows AS B to become passive. Figure 3(d) shows that AS B sets its reported label to be $(B, 2)$, which corresponds to the AS path BCA, and sends an update. Concurrently, AS D forwards the reply from C to all its neighbors, because its reported label is updated, and the reply states $R(d, (D, 2), (B, 1))$.

As Figure 3(e) shows, the reply from AS D does not provide an additional valid route for AS B, but the update from AS B allows AS D to acquire a larger valid route, which causes AS D to send an update with its new reported label $(D, 3)$. The second reply from AS C does not cause any updates in neighboring ASes.

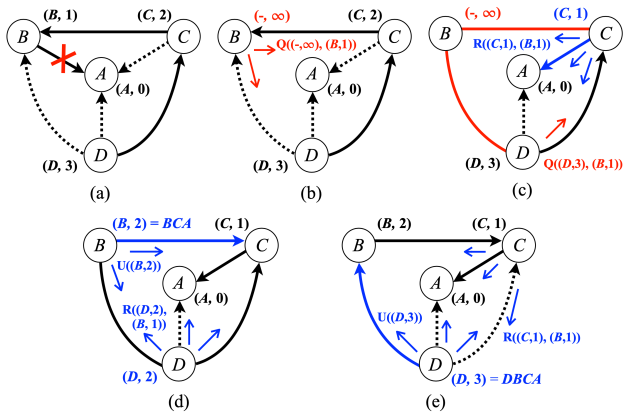

Figure 3: BGP-ELF convergence after failures

6 CONCLUSIONS

BGP-ELF is the first protocol for inter-AS routing that is provably stable and loop-free without the use of path vectors or the need to engineer routing policies.

Eliminating loops without using path vectors in BGP-ELF required the introduction of queries and replies, which may remind the reader of diffusing computations used in EIGRP [7], [27]. However, the signaling in BGP-ELF is far more efficient than signaling based on diffusing computations, because an AS can become passive with the first reply it receives, rather than having to wait for all neighbor ASes to reply, and queries can be forwarded towards destinations.

Given that AS paths are not communicated among routers using BGP-ELF, an alternative mechanism was introduced to enable route filtering based on the type of ASes included in the paths used to reach destinations. In contrast to the use of multi-path routing in today's BGP implementations, this approach is safe.

Due to space limitations, our description of BGP-ELF assumed the case in which ASes are fully meshed, but ASes need not be fully meshed. Our forthcoming work describes what could be called "Internal BGP-ELF," i.e., the additional mechanisms used to ensure that BGP-ELF works correctly when route reflectors are used within an AS.

APPENDIX A: PROOF OF BGP-ELF CORRECTNESS

The proofs of the theorems presented in this Appendix are based on the following two definitions, which reflect the fact that policy-based routing across ASes does not seek to attain optimum routes.

Definition 6. **Feasible Route:** *A route to destination d is said to be feasible if it does not involve a 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 routers in AS k can reach, the routers obtain at least one route through a neighbor AS q within a finite time after T, such that $\ell_d^k[r] < \ell_\infty$.*
(2) *For any unreachable destination d for routers in AS k, the routers set $\ell_d^k[r] = \ell_\infty$ within a finite time after time T.*
(3) *Routers in AS k do not change the value of $\ell_d^k[r]$ within a finite time after time T.*

Theorem 2. *An AS path in which $\mathcal{T}$ is satisfied at every AS along the path cannot be a loop*

Proof. Assume that $\mathcal{T}$ is true at every AS along a path L. For the sake of contradiction, assume that L is a routing loop that excludes destination d at time t and let this loop be $L = \{v_1 \rightarrow v_2 \rightarrow ... \rightarrow v_h \rightarrow v_{h+1}\}$, where $v_{h+1} = v_1$.

Each AS $v_i \in L$ informs its neighbor ASes of its reported label to d at a time denoted by t_i, where $t_i < t$, and its neighbors in L use that value at a subsequent time to determine whether $\mathcal{T}$ is satisfied.

The time when router $v_i \in L$ makes router $v_{i+1} \in L$ a next hop to d is denoted by t_i^+ and $t_i^+ \leq t$, which implies that $s_d^{v_i}(t) = s_d^{v_i}(t_i^+)$, $\ell_d^{v_i}[r](t) = \ell_d^{v_i}[r](t_i^+)$, and $r_d^{v_i}(t) = r_d^{v_i}(t_i^+)$ for all $v_i \in L$.

The following results are a consequence of the fact that $\mathcal{T}$ must be satisfied at each router $v_i \in L$:

(a) $r_d^{v_i}(t_i^+) = r_d^{v_i}(t) > \ell_{dv_{i+1}}^{v_i}(t)$.
(b) $\ell_{dv_{i+1}}^{v_i}(t) = \ell_{dv_{i+1}}^{v_i}(t_i^+) = \ell_d^{v_{i+1}}[r](t_{i+1})$.
(c) $\ell_d^{v_{i+1}}[r](t_{i+1}) = r_d^{v_{i+1}}(t_{i+1}) = r_d^{v_{i+1}}(t)$.

It follows from (a), (b) and (c) that $r_d^{v_i}(t) > r_d^{v_{i+1}}(t)$. However, this constitutes a contradiction, because it implies that $r_d^{v_i}(t) > r_d^{v_i}(t)$ for all $v_i \in L$; therefore, the theorem is true. □

Theorem 3. *No routing loop can be created in BGP-ELF when nodes transition from passive to active state.*

Proof. The proof follows from the fact that a node that transitions to the active state either has no next hop or must keep its current next hop. The first case negates the existence of a routing loop. In the second case, the current next hop was part of a path established by nodes in passive state, which negates the existence of a routing-table loop because of Theorem 2. □

Theorem 4. *BGP-ELF is loop-free for any destination d.*

Proof. If $\mathcal{T}$ is always satisfied at every node, then it follows from Theorem 2 that no routing loops can form. Therefore, if follows from Theorem 3 that the proof needs to show that no routing-table loop can be created when a router transitions from active to passive state.

For a node k to become passive once it is active, it must receive an update or a response such that $\mathcal{T}$ is satisfied, and a node $n \in N^k$ can send an update or a reply to router k only if it is passive itself. The path from n to d either consists of nodes that are passive, or consists of both active and passive nodes. In the first case, it follows from Theorem 2 that node k cannot create a loop by setting $n = s_d^k$ because then the path from n to d is loop-free and extending that path with link (k, n) cannot create a loop. In the second case, the path from n to d is the concatenation of subpaths, each consisting of one or more nodes that are all passive or are all active, and it follows from Theorems 2 and 3 that such subpaths are loop-free and hence extending the path from n to d with link (k, n) cannot create a loop. □

Theorem 5. *BGP-ELF converges to correct feasible routes for all reachable destinations within a finite time after topology changes stop occurring in a finite system.*

Proof. Assume that routers in every AS execute BGP-ELF correctly but routers in AS k converge incorrectly with $\ell_d^k[r] = \ell_\infty$ after topology changes stop occurring at time T.

Given that the system is finite and there are physical AS paths to d, all loop-free paths in the system are finite and it takes a finite time for BGP-ELF messages to propagate along any loop-free path. Furthermore, because BGP-ELF is executed correctly, the reported label $\ell_d^k[r]$ must correspond to a loop-free path from AS k to destination d through some neighbor AS $s \in N^k$ such that $\ell_d^k[r]$ has the largest hop count among all feasible routes available.

Updates, queries, and replies must propagate over loop-free paths because BGP-ELF is loop-free (Theorem 4). This implies that routers in AS k must receive an update or a reply from routers in AS s reporting $\ell_d^s[r]$ within a finite time after T, which makes $\ell_{ds}^k = \ell_d^s[r]$. However, this is a contradiction to the assumption that BGP-ELF is executed correctly, because this would require routers in AS k to update $\ell_d^k[r]$ with $h_d^k = 1 + h_d^s$ and hence $\ell_d^k[r] < \ell_\infty$. □

Theorem 6. *BGP-ELF converges to ℓ_∞ for all unreachable destinations within a finite time after topology changes stop occurring in a finite system.*

Proof. Assume that an AS n_k belongs to a connected component from which destination d is unreachable starting at time t_0, and that no topology changes occur after time $t_n \geq t_0$. For the sake of contradiction, assume that AS n_k converges to $\ell_d^{n_k}[r] \prec_\ell \ell_\infty$ at time $t_k \geq t_n$.

If AS n_k has a reported label $\ell_d^{n_k}[r] \prec_\ell \ell_\infty$ at time t_k, then it must have a next-hop neighbor $n_{k-1} \in N^{n_k}$ such that $\ell_{dn_{k-1}}^{n_k} \prec_\ell r_d^{n_k}$.

Because BGP-ELF is loop-free (Theorem 4), there must be an originating AS n_o that sent an update or a reply with a reported label $\ell_d^{n_o}[r] \prec_\ell \ell_\infty$ that allowed updates or replies with finite reported labels to be sent to AS n_k before time t_k, and such that no AS along the path from AS n_k to d changes its reported label to d after some time t_k; furthermore, n_o must be a neighbor of the AS of destination d. This is a contradiction, because d is not in the connected component of AS n_k starting at time $t_0 < t_k$, and no AS

in the connected component can consider itself being a neighbor of the AS of d a finite time after t_0. □

APPENDIX B: AN EXAMPLE LOCAL PREFERENCE FUNCTION IN BGP-ELF

Algorithm 1 Local Preference Function in BGP-ELF

Step 1: If multiple routes have the same weight, prefer the route with the highest local preference, where the local preference is the same for all routers in the same AS.

Step 2: If multiple routes have the same local preference, prefer the route that was originated by the local router.

Step 3: If none of the routes were originated by the local router, prefer the route with the smallest reported label (i.e., shortest AS-path and the smallest first AS identifier among the routes with the shortest AS-path).

Step 4: If the reported label is the same, prefer the lowest origin type (IGP < EGP < incomplete).

Step 5: If all origin codes are the same, prefer the route with the lowest multi-exist discriminator (MED).

Step 6: If the routes have the same MED, prefer external routes (EBGP) over internal routes (IBGP).

Step 7: Prefer the route with the lowest IGP metric to the BGP next hop.

Step 8: For EBGP paths, select the oldest route to minimize the effect of routes going up and down (flapping).

Step 9: Prefer the route with the lowest neighbor BGP-ELF router ID value.

Step 10: Prefer the route shortest cluster-list (if AS is not fully meshed)

Step 11: If the BGP-ELF router IDs are the same, prefer the route with the lowest neighbor IP address.

REFERENCES

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

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

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

[4] J. Durand, I. Pepelnjak, and G. Doering, "BGP Operations and Security," RFC 7454, Feb. 2015.

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

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

[7] J.J. Garcia-Luna-Aceves, "Loop-Free Routing Using Diffusing Computations," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 1, February 1993.

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

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

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

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

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

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

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

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

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

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

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

[19] 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.

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

[21] C. E. Perkins and P. Bhagwat, "Routing over Multihop Wireless Network of Mobile Computers," *Proc. ACM SIGCOMM '94*, 1994.

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

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

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

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

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

[27] D. Savage et al., "Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)" RFC 7868, 2016.

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

[29] 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.

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

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