iBGP2: a scalable iBGP redistribution mechanism leading to optimal routing

Marc-Olivier Buob
Nokia Bell Labs
marc-olivier.buob@nokia.com

Anthony Lambert
Orange Labs
anthony.lambert@orange.com

Steve Uhlig
Queen Mary University of London
steve.uhlig@qmul.ac.uk

Abstract—The Internet is made of almost 50,000 ASes exchanging routing information thanks to BGP. Inside each AS, information is redistributed via iBGP sessions. This allows each router to map a destination exterior to the AS with a given egress point. The main redistribution mechanisms used today, (iBGP full mesh, Route Reflectors and BGP confederations), either guarantee selection of the best egress point or enhance scalability, but not both. In this paper, we propose a new way to perform iBGP redistribution in an AS based on its IGP topology, conciliating optimality in route selection and scalability. Our contribution is threefold. First, we demonstrate the tractability of our approach and its benefits. Second, we provide an open-source implementation of our mechanism based on Quagga. Third, we illustrate the feasibility of our approach through simulations performed under ns-3 and compare its performance with full mesh and Route Reflection.

I. BACKGROUND

A. Context

The Internet is made of about 50,000 interconnected Autonomous Systems (ASes). Each AS consists in a set of networks and routers under the control of a given administrative authority (e.g., carrier, Internet Service Provider (ISP), Content Provider). To ensure reachability in the Internet, ASes exchange routing information about networks they can reach in the Internet. This is achieved by establishing exterior Border Gateway Protocol (eBGP) sessions between the AS border routers (ASBR) of neighboring ASes. The routing information learned by ASBRs is then redistributed inside the AS through internal Border Gateway Protocol (iBGP) sessions, established between the routers of the AS. This way, BGP populates routing tables of network equipment inside the AS with routes to destinations external to the AS. The key idea consists in selecting for each external destination an egress point called BGP next-hop. Each network equipment has to be able to reach BGP next-hops, which is usually achieved through the use of an Interior Gateway Protocol (IGP) such as OSPF [20] or IS-IS [22]. In such protocols, weights are assigned to physical links. Adjacency states are flooded between routers across the IGP network, enabling routers to build a map of the network and compute their shortest paths to any interior destination using Dijkstra’s algorithm.

In general, any exterior destination can be reached through several egress points. BGP routers therefore have to elect a single best egress point among all candidates. They do so by running their BGP decision process (see Figure 1). The first steps of the BGP decision process are concerned with interdomain metrics: AS economic policy and interdomain path length. The next steps focus on intradomain aspects: they allow to implement either cold-potato (using MED attribute) or hot-potato routing strategies (by selecting the closest egress point in terms of IGP costs).

BGP is an incremental protocol, i.e., a router only announces its preferred path to its neighbors for each destination. Moreover, routing information learned through an iBGP session should not be readvertised through another iBGP session. As a consequence, depending on the iBGP network topology, there is no guarantee that a router will learn its best possible egress point to a destination. The simpler solution to overcome this issue consists in establishing an iBGP session between every pair of routers, i.e., use an iBGP full mesh. Each router hence learns the best route selected by every other router for every destination. However, such an approach has poor scalability, in terms of configuration overhead, number of messages (each routing change is notified to all routers), and memory consumption (each router maintains an Adj-RIB-in and an Adj-RIB-out per neighbor). Consequently, an iBGP full mesh is traditionally only used in small ASes.

Large ASes, on the other hand, rely on approaches such as Route Reflection [4] or BGP confederations [28]. A Route Reflector (RR) is a router which is allowed to readvertise some iBGP routing information. More precisely, some of the RR’s iBGP peers can be configured as RR clients. A RR is allowed to redistribute a route learned from a RR client to every peer and non-client routes only to RR clients. BGP confederations consist in splitting an AS into a set of sub-ASes, exchanging routing information through eBGP. Those solutions allow to design scalable network topologies, but can lead to arbitrary filtering of routing information, and potentially to issues in route selection, e.g., oscillations [2], [13], non-optimality [10],
or non-determinism [6], contrary to an iBGP full mesh.

B. Proposed approach

[6] proved that if each router learns its best possible egress point, then none of the routing issues mentioned above can occur, because the network behaves as a full mesh (fm). Such a topology is said to be fm-optimal. [29] pointed out in Route Reflection, that it is important to set lower neighbor IP addresses to RR clients to make sure that all optimal routes are indeed learned.

In this paper, we present a route distribution mechanism that guarantees fm-optimality, for any network topology. The key idea of the mechanism is to make the iBGP announcements distribution follow the IGP topology. More precisely, the mechanism makes a router \( u \) only advertise a route towards an egress point \( s \) to a neighbor \( v \) if and only if \( u \) belongs to the shortest path from \( v \) to \( s \).

We show that such a mechanism discards most of the irrelevant BGP announcements while ensuring that each router is able to learn the route towards its best egress point. It therefore not only guarantees routing optimality and correctness, but also offers a good scalability. Moreover, since this approach is IGP-aware by design, each router updates its iBGP forwarding in case of IGP network topology change.

We propose an open-source implementation of our approach written in C++, based on the well-known ns-3 simulator [16] and the Quagga daemon routing suite [17], and simulate the behaviour of our approach on a reproducible network topology. Datasets, results, as well as the code of the simulator are made publicly available at [1].

C. Goals

Our main goal consists in finding a scalable approach guaranteeing routing optimality and routing correctness for any IGP topology. Those properties must remain satisfied even if the network topology changes, e.g., network failure, routing policy modification, addition of new equipment.

Additionally, we aim at simplifying the configuration of BGP routers, thus decreasing the risk of misconfiguration. To do so, we take advantage of the information provided by the IGP. Our approach remains backward-compatible with legacy routers, is incrementally deployable on a subset of the iBGP network, and can be emulated by Route Reflection [4].

Section II introduces the problem of route redistribution in iBGP. In Section III, we present a new simple and scalable iBGP redistribution mechanism guaranteeing routing optimality. In Section IV, we illustrate the behaviour of our approach through an implementation of our solution in ns-3. Section V lists the approaches implemented in current networks and proposed in the literature. Section VI discusses the backward-compatibility of our approach, in particular with respect to the ability to perform traffic engineering. Section VII concludes the paper.
elected as best among the ones it has learned. Those figures summarize some well-known issues with Route Reflection:

- **Sub-optimality** occurs when a stable iBGP propagation path does not exist between a router and its closest egress point (see Figure 2).
- **Oscillations** occur if propagation paths corresponding to optimal routes overlap, as shown on 3. Oscillations are a serious issue because routers never converge to a final routing state and permanently exchange routing messages, inducing unpredictable forwarding paths and unnecessarily stressing network equipment.
- **Deflection** may occur if some routers do not learn their optimal egress point, and traffic crosses a router that selected another egress point. Several consecutive route deflections may trap traffic in forwarding loops, as depicted in Figure 4. In practice, traffic trapped in a forwarding loop is dropped after multiple hops (among the network equipments involved in the forwarding loop), inducing traffic loss, bandwidth waste and stress on network equipment.

**C. Issues during reconvergence**

When the network topology changes (failure, route appearance or disappearance), routers may explore transient routing states. Depending on the arrival order of routing messages, routing inconsistencies may arise such as route deflections or even forwarding loops (called micro-loops). In particular, micro-loops may occur in the following situations:

- **Maintenance**: Operators may want to select another egress point without disrupting the reachability towards the corresponding destinations. This configuration change may cause a network reconvergence.
- **Next-hop failure**: The primary path is withdrawn (or the corresponding egress point fails). If a router has no backup path, it cannot route the traffic towards this destination and acts as a blackhole. Note that with a full mesh, if such a path exists, routers can switch to this new path fast. In approaches such as Route Reflection or confederations, reconvergence may be slow since the new route is propagated and adopted hop by hop.
- **IGP link failure**: The modification of IGP metrics can trigger the election of new egress points by routers and thus can cause traffic shifts [27].

Figure 5 illustrates a network reconvergence. We consider an AS using an iBGP full mesh where the arrival order of routing messages during reconvergence is as follows: at \( t_1 \), \( a \), \( b \), and \( c \) exit their traffic through \( a \). At \( t_2 \) a network failure occurs and \( d \) becomes the best egress point for \( b \), \( c \), and \( d \). Suppose that \( b \) updates its FIB before \( c \). Its new egress point is \( d \) and the corresponding forwarding IGP path is \( b \), \( c \), \( d \). However, \( c \) still exits the corresponding traffic through \( a \) and expects to use the shortest IGP path \( c \), \( b \), \( a \). As a consequence, traffic is trapped in a micro-loop between \( b \) and \( c \). At \( t_3 \), \( c \) finally installs its new BGP route and selects its new egress point \( d \). The micro-loop disappears.

To prevent micro-loops during maintenance, operators can perform a graceful shutdown [11], which consists in penalizing the obsolete route. The route corresponding to the new egress point will progressively be adopted by the router which used the former. More generally, micro-loops can be avoided by establishing an IP tunnel from a router towards its egress point, IP tunnels indeed enable to avoid route deflections and thus forwarding loops.

**D. Ideal iBGP redistribution**

Sections II-B and II-C pointed out the limitations of existing solutions. A full mesh offers desirable routing properties (optimal, deterministic, and stable routing) whereas Route Reflection and BGP confederations offer a good scalability to the detriment of these properties. Moreover, with the latter solutions, designing the iBGP network may be complex and may impact the final routing.

We aim at proposing a new iBGP redistribution mechanism that behaves like an iBGP full mesh but without its scalability limitations, i.e., offering the following properties:

- (A) **Feasibility**: The solution must be scalable and preserve hardware resources on the routers, such as CPU and RAM. This means that it should avoid keeping and redistributing irrelevant routing messages.
- (B) **Correctness**: The solution should lead to the same routing outcome as a full mesh, which guarantees routing optimality, determinism, and stability [6].
- (C) **Transparency**: The iBGP topology should be easy to configure and maintain. We should also not have to perform complex network verification to guarantee a scalable and robust iBGP network topology.
- (D) **Robustness**: The solution should be resilient to failures and IGP metric changes.
- (E) **Fast convergence**: The solution should react to network changes as fast as possible.

[12] presents a framework dedicated to network routing protocols design. Our correctness, transparency and robustness
goals are consistent with this framework. This framework also identified some other challenges such as expressiveness (which will be discussed in Section VI), policy opaqueness and autonomy (which are both always guaranteed in our case thanks to the AS subdivision of the Internet).

E. Fm-optimality in a nutshell

Fm-optimality is a property guaranteeing that any router is able to select its optimal egress point. It always holds in a full mesh, but not in Route Reflection. If an iBGP topology is not fm-optimal, some of the issues listed in Section II-B may arise.

In Figure 6, we consider two routes learned via eBGP sessions (ρ and ρ′) towards a given prefix p, which are equally good on steps 1 to 3 of the decision process (see Figure 1). Such concurrent routes are said to be quasi-equivalent and will be tied according to the next steps of the decision process. These preferences are depicted by the green and red areas in Figure 6. We assume that all the routers of the AS both run a BGP and an IGP daemon. We assume that BGP routes can only be re-advertised according to the dashed arrows. A BGP router only re-advertises the BGP routes it has selected as best.

Border router c₁ (resp. c₂) learns (and selects) ρ′ (resp. ρ). Note that the arrival order of ρ and ρ′ will not impact their final decision according to the 4th step of the BGP decision process. We denote by |x, y| the cost of the IGP shortest path from x to y. RR prefers ρ′ since |rr, c₁| < |rr, c₂|. Therefore, rr selects and forwards ρ′ to its peers, c₁ (resp. c₂) only learns ρ′ through rr. Thus, c₂ (resp. c₃) sends its traffic to c₂ which is farther than c₁. This suboptimal decision is due to rr, which has blocked the propagation of ρ′.

General case: We now formalize under which conditions routers may block relevant routes. We denote by S the set of egress points of the AS which may learn eBGP routes and by T the set of routers of the AS. Let s ∈ S be an egress point and t ∈ T a router. If t is always able to learn the route announced by s despite concurrent quasi-equivalent routes announced by egress points farther than s from t, then the pair (s, t) is said to be fm-optimal. If all the pairs of S × T are fm-optimal, the topology is said to be fm-optimal.

Suppose that s is the only optimal egress point, i.e., the other considered egress points are strictly farther than s from t. By definition of s, the concurrent egress points belong to:

\[ C(s, t) = \{ s' \in S \setminus \{ s \}, |t, s'| > |t, s| \} \]

**Definition:** Consider a router \( u \in T \). If there exists \( s' \in C(s, t) \) such that \( |u, s'| < |u, s| \), then \( u \) may prefer and select the route announced by \( s' \). Such a router \( u \) which may block the propagation of a BGP route \( ρ \) is said to be black (for pair \((s, t)\)). A non-black router is said to be white and is characterized by:

\[ W(s, t) = \{ w \in T \mid \forall s' \in C(s, t), |w, s| < |w, s'| \} \]

Figure 6 illustrates the concept of black and white routers. We see that the iBGP path from \( s = c₂ \) to \( t = c₄ \) crosses a black router (rr) which blocks the propagation of \( ρ \). More generally, if there exists an iBGP path from \( s \) to \( t \) such that each router of this path is white, then \( t \) learns the route announced by \( s \), and thus \((s, t)\) is fm-optimal.

If a router cannot elect its best egress point at the IGP cost step of the decision process, it will select one of these candidates according to its tie-break rule. If, while deflections might occur, packets will reach one of them via a loop-free shortest path.

III. iBGP2, AN OPTIMAL iBGP ROUTE REDISTRIBUTION

A. Big picture

The key idea of iBGP2 consists in guaranteeing the fm-optimality criterion for any pair \((s, t)\) at any moment and for any IGP topology while discarding most of the irrelevant routing announcements. According to Section II-E, it suffices to guarantee the existence of at least one white iBGP propagation path from \( s \) to \( t \) (for all \((s, t)\) \((s, t)\) \(S \times T \) pairs) to guarantee fm-optimality.

We show in Section III-B that any router \( u \) belonging to the shortest IGP path from \( t \) to \( s \) in \( S \) is always white for this \((s, t)\) pair. Therefore, allowing the announcement propagation from \( s \) along this path suffices to make this pair fm-optimal. For this very \( s \), if we extend this reasoning to any router of \( T \), our route redistribution mechanism thus consists in allowing the forwarding of the announcements from \( s \) along the RSPT (Reverse Shortest Path Tree) rooted at \( s \).

**Definition:** We call RSPT rooted at \( s \) the union of the shortest IGP paths from any router \( t \in T \) towards \( s \). Note that, in practice a RSPT may be a directed acyclic graph.

In our approach, the iBGP graph exactly matches the IGP graph (both in terms of routers and links). A router \( u \) propagates the route announced by \( s \) to an iBGP neighbor \( v \) if \( u \) belongs to a shortest path from \( v \) to \( s \), even if this route has been learned from iBGP. If so, we note \( u \in sp(v, s) \).

Figure 7 illustrates the iBGP2 route redistribution mechanism. We consider three concurrent routes collected by three border routers \( a, d \) and \( g \) (solid arrows). Each dashed arrow corresponds to an iBGP2 redistribution. The color of a router

---

1This assumption holds as network operators do so in practice.

2For instance by using Route Reflection.

3White routers are also called “safe routers” in [9].
corresponds to the egress point it selects once BGP has converged.

**Definition:** We call *zone* the set of routers electing a same given egress points (respectively depicted in orange, red, and purple on Figure 7).

We observe that each router learns and selects a route to its closest egress point: as routes are redistributed along RSPTs rooted at egress points, all routers closer to a given egress point than any other egress points will learn the routes redistributed along this very tree. Therefore, there is always at least one redistribution path from an egress point to the routers which prefer it. Moreover, this redistribution mechanism limits the propagation of routes in the AS to zones and their direct neighbors as depicted in Figure 7.

Indeed, a router having a link towards a router belonging to another zone benefits from the backup route provided by this neighbor. For example, router $f$ learns the route corresponding to egress points $d$ (from $d$), $a$ (from $b$) and $g$ (from $g$). This route diversity enables a graceful reconvergence in case of failure of the primary egress point, since these routes are learned through node-disjoint IGP paths and thus cannot be both impacted by a single internal failure.

Moreover, each router only collects its optimal routes most of the time only once. If several shortest paths exist from a router to its closest egress point, the corresponding route might be learned several times. In other words, iBGP2 discards most duplicated announcements. We will explain in Section III-C how to avoid such duplicated announcements.

**B. Proof**

In this section, we show that the iBGP2 redistribution criterion presented in Section III-A is sufficient to guarantee fm-optimality. To do so, we must prove the existence of a white iBGP propagation path from $s$ to $t$ for any $(s, t) \in S \times T$ pair. We assume that both routers belong to the same IGP connected component. Therefore, there is at least one shortest IGP path from $t$ to $s$. We denote by $\mu$ the corresponding directed iBGP2 propagation path from $s$ to $t$.

Let us consider a pair $(s, t) \in S \times T$, such that $s$ is the closest candidate egress point for $t$. Let $u$ be a router belonging to $\mu$. This leads to $|t, s| = |t, u| + |u, s|$. Suppose by contradiction that $u$ prefers a concurrent source $s' \in S$, i.e., $|u, s'| < |u, s|$. Then $|t, s| = |t, u| + |u, s| > |t, u| + |u, s'| \geq |t, s'|$, which contradicts that $s$ is the closest egress point of $s$. In other words, all the routers $u$ belonging to the shortest IGP path from $t$ to $s$ also prefer $s$, and thus are white routers.

We can easily prove by induction that any router of $\mu$ can always learn this route (since its predecessor both learns and selects the route announced by $s$).

**C. Properties of iBGP2**

We now discuss the properties and behavior of iBGP2 in different situations. We also discuss how it addresses the challenges we have identified in Section II-A.

1) Nominal case:

   (A) Feasibility: The number of iBGP2 sessions maintained by a router is equal to its IGP degree, which is assumed to be relatively small in practice. Moreover, each iBGP2 session only provides a subset of the Internet prefixes in general. Indeed, duplicated announcements (same route learned via several iBGP peers) only occur in case equally good IGP paths exist. For instance, on Figure 7, suppose that $\rho_a$ is the only eBGP route entering the AS. Then $f$ learns $\rho_a$ twice (via $(a, e, f)$ and $(a, b, f)$). More generally, a router $t \in T$ learns a route $\rho_s$ related to $s \in S$ at most $k$ times, where $k$ is the number of successors of $t$ in the RSPT rooted in $s$. Duplicated announcements could be avoided if these neighbors adopt a same strategy to decide which of them forwards $\rho_s$ to $t$ (for example the one having the lowest IGP router-id).

   Doing so, iBGP2 redistribution consists in allowing $\rho_s$ to be spread along this tree rooted in $s$, until reaching routers preferring another egress point. By construction, each router of the AS thus learns its optimal egress points exactly once. On the contrary, a given egress point may be learned via several iBGP propagation paths in Route Reflection depending on the topology. The concurrent egress points are learned at most once in iBGP2. We thus expect a reasonable memory consumption.

   iBGP2 also requires some extra computations. First, each BGP router has to compute shortest paths from each of its iBGP neighbors towards candidate egress points. This is achievable if the IGP protocol deployed is a link-state protocol such as OSPF [20] or IS-IS [22]. This assumption holds in practice because iBGP2 targets large ASes, which require fast IGP reconvergence and thus traditionally use a link-state IGP. This extra computation is reasonable. Indeed, it is for instance performed by the IP fast-reroute Loop Free Alternate (LFA) mechanism [26]. Second, iBGP2 redistribution mechanism consists in managing many BGP filters, and one may fear this additional complexity to be costly. But such filtering is today achieved very efficiently, for instance in a L3-VPN context by using Route Target constraint [18]. RT constraint is able to

---

Fig. 7. Example of iBGP2 redistribution involving three quasi-equivalent routes.
handle a large number of routes, of sessions and filters, which are in practice significantly greater than in our context and could therefore be used to implement iBGP2 filtering.

To sum up, iBGP2 seems to be feasible. It could be realized by using mechanisms supported by current commercial routers. The low number of messages induced by iBGP2 mechanism also contributes to improve the scalability of our solution.

(B) Correctness: Correctness is a consequence of the fm-optimality. Hence, iBGP2 leads to a stable, loop-free and optimal routing.

(C) Transparency: Route Reflection or confederations require to design a topology. The quality of the route distribution is dependent on this design. Previous works have even shown that computing a Route Reflection topology which preserves fm-optimality for any case of single failures [7], [9] is challenging. The design of the iBGP2 topology on the other hand is implicit, i.e., no design is necessary while guaranteeing fm-optimality. We call this property of iBGP2 transparency.

2) Reconvergence: Two classes of events may trigger a change in terms of iBGP decision.

First, the IGP topology has changed. In this case each iBGP2 router must update the way it reannounce iBGP information to conform to the new RSPTs. More precisely, consider an arbitrary egress point \( s \in S \) and \( u,v \) two neighboring routers of \( T \). If \( (v \rightarrow u) \) arc was in the RSPT rooted in \( s \) and still belongs to it, then nothing happens. If it was not in the former tree, but now belongs to it, routes related to \( s \) can now be propagated along this arc. If \( (v \rightarrow u) \) no longer belongs to the RSPT, the routes related to \( s \) can no longer be announced along this arc. This mechanism can be achieved at the speed of the IGP, but can suffer of the same issues as an IGP protocol (e.g. micro-loops and transient black holes). Note that this mechanism can be slightly enhanced. The ability to propagate routes related to \( s \) along the \( (u \rightarrow v) \) arc should be held for some time to let the new iBGP redistribution path towards \( v \) to be established. Doing so, no transient black hole would appear, \( v \) exiting its traffic towards \( s \) via \( u \) (if \( s \) is still reachable). The only drawback of this choice occurs if \( s \) is no longer the closest egress point of \( v \) for the considered prefix. In this case, until \( v \) do not learn its new optimal egress point, the corresponding traffic is exiting towards \( s \), which is suboptimal.

Second, an eBGP event has occurred and some BGP next-hops do not announce some prefixes anymore. For example consider in Figure 7 that next-hop \( s = d \) fails. Intuitively, a wave of withdraws will be propagated in the orange zone until it reaches its boundaries, and the routers adjacent to alternative zone(s) (i.e., the red or the purple ones) will elect and propagate their backup route. The red and purple zones will progressively grow until covering the routers initially inside the orange zone, leading to a new correct and optimal routing. While BGP converges, routers that used to belong to the orange zone may behave as black holes since they have no backup paths, that is if they are not adjacent to an alternative zone. At last, if a new concurrent egress point appears, its corresponding zone progressively grows and pushes back the other zones. By design iBGP2 guarantees that the router switching to this new egress points can indeed forward successfully the corresponding traffic via a shortest IGP path.

(D) Robustness: To sum up, the main routing issue occurs when an egress point stops announcing a prefix: in this case, a transient black hole will happen. Such an issue would also occur with Route Reflection. In case of IGP changes, routing can suffer from micro-loops and transient black holes as well as an IGP routing protocol. Once the network has converged, by construction of iBGP2, the new routing is correct and optimal. On the contrary, it is difficult to design a Route Reflection topology which remains correct in case of failures [7], [9]. For example, network operators have to design their iBGP topology to have two disjoint iBGP paths from a border router towards another router, leading to redundant BGP messages and more complex router configurations.

(E) Fast convergence: As explained earlier, thanks to iBGP2, iBGP could converge as fast as the underlying IGP, which in practice is very fast.

D. iBGP2 generic algorithm

iBGP2 consists in adapting the way a router redistributes an incoming route depending on BGP and IGP dynamics.

We consider a router \( u \). We denote by \( V_u \subseteq V \) its set of IGP neighbors and by \( S_u \subseteq S \) the set of BGP next-hops known to \( u \). For all \((s,v) \in S_u \times V_u \) pairs, we denote by \( R_u(s,v) \) the boolean variable such that \( u \) redistributes a route corresponding to next-hop \( s \) to neighbor \( v \) if and only if \( R_u(s,v) = 1 \). iBGP2 updates \( S_u, V_u \), and \( R_u \) according to two independent streams of events (BGP and IGP).

a) BGP events: The BGP stream allows \( u \) to discover egress points. When a BGP update is received, the corresponding next-hop \( s \) is stored into \( S_u. R \) is then initialized consequently.

- If \( s \notin S_u \)
  1) \( S_u \leftarrow S_u \cup \{s\} \)
  2) For each \( v \in V_u. R_u(s,v) \leftarrow (u \in sp(v,s)) \)

b) IGP events: The IGP stream allows to maintain \( V_u \) and to update \( R_u \) according to IGP dynamics (if needed).

- For each handled link state advertisement:
  1) If the impacted IGP link is incident to \( u \):
     a) If \((v \rightarrow u)\) disappeared : \( V_u \leftarrow V_u \setminus \{v\} \)
     b) Else if \((v \rightarrow u)\) appeared : \( V_u \leftarrow V_u \cup \{v\} \)
  2) For each \((s,v) \in S_u \times V_u \)
     a) If \( R_u(s,v) = 1 \land (|v,s| \text{ strictly increased}) \)
        i) \( R_u(s,v) \leftarrow (u \in sp(v,s)) \)
     b) Else if \( R_u(s,v) = 0 \land (|v,s| \text{ strictly decreased}) \)
        i) \( R_u(s,v) \leftarrow (u \in sp(v,s)) \)

Note that these IGP events correspond to appearances and disappearances of routers, links, networks, and to metric changes. Switching \( R(s,v) \) from 0 to 1 (resp. from 1 to 0) triggers the forwarding (resp. the withdrawal) of all the BGP routes corresponding to next-hop \( s \) and neighbor \( v \).
IV. SIMULATIONS

A. How to implement iBGP2

There are several ways to implement iBGP2. We present some of them, from the more intrusive (in terms of implementation) to the less intrusive:

1) **Patch BGP daemon:** we could first implement natively iBGP2 redistribution mechanism in the same way Route Reflection was implemented.

2) **Use Route Target constraint:** we could also rely on Route Target constraint optimized filtering capabilities which would indeed be a suitable solution as it is known to scale well.

3) **Mutual Route reflection:** finally we could establish an iBGP session between each neighboring IGP router of the AS, so that both routers are mutually route reflector, and configure BGP filters to mimic iBGP2 redistribution.

To evaluate our approach, we have implemented a prototype using ns3 [16], a well-known network simulator. By using ns3 direct code execution (ns3-dce), each node involved in the simulation can run Quagga [17] (which supports both OSPF and BGP). We excluded patching the BGP daemon because it highly depends on BGP implementations. We also discarded the Route Target constraint approach because Quagga does not support it. This is why we adopted the third approach. Mutual Route Reflection does not perform any filtering and thus induces an overhead in terms of BGP routes and messages compared to iBGP2. To mimic the iBGP2 redistribution mechanism, each router maintains a list of BGP filters as explained in Section III-D. A dedicated daemon sniffs IGP messages and issues the corresponding BGP commands to Quagga. It is also in charge of establishing and closing the iBGP sessions between the router and its IGP neighbors.

Since the correctness and the optimality of the routing computed by iBGP2 are guaranteed, the focus of our simulations is on demonstrating the feasibility of iBGP2 in a realistic environment and to compare the performance of the three considered solutions (full mesh (FM), Route Reflection (RR), and iBGP2).

B. Setup

We performed simulations based on the IGP topology depicted on Figure 7.

- **iBGP setup:** A full mesh consists in establishing iBGP sessions between each pair of \(|T| = \{a, \ldots, h\}\). For the RR use-case, we consider that \(b\) and \(f\) act as RR and share an iBGP session. The other routers are clients of these two RRs. This topology is realistic (robust to single failure, hierarchical, and consistent with CISCO best practices). iBGP2 sessions are configured automatically along each IGP link.

- **eBGP setup:** A router exterior to the simulated AS, named \(nh\), establishes an eBGP session towards each router of \(S \subseteq T\). To cover all possible scenarios, we consider that \(S = T\). For each value of \(n = \{1, \ldots, |S|\}\), we compute all the subsets of ASBRs \(C \in S\) and assign to each of them a given prefix. For each \(C\), we configure \(nh\’s\) eBGP filters to announce the assigned prefix only to the routers belonging to \(C\). We then perform for each value of \(n\) a dedicated simulation run.

The criteria listed in Section II-D are evaluated as follows:

- **(A) Feasibility:** CPU stress is estimated by counting how many BGP updates are sent. This is achieved by parsing `bgpd` logs. RAM usage is evaluated through the `show bgp memory` command, which details the memory usage of `bgpd` (including the share taken by the Adj-RIB-ins, the BGP RIB, and the Adj-RIB-outs).

- **(B) Correctness:** We test whether each router is able to route each input prefix, and if so, whether it is routed via an optimal egress point or not.

- **(D) Robustness:** We do not simulate each scenario of potential failures. Rather, we evaluate the route diversity [5] (i.e. the number of distinct egress points learnt by a router towards a destination prefix), which contributes to provide a backup path in case of failure of the primary path.

- **(E) Fast convergence:** The convergence time is measured by parsing `bgpd` logs. For each prefix, we retrieve the date of the last related iBGP update announced in the simulated AS.

(C) Transparency is not evaluated here, since it is a qualitative criterion. By design, iBGP2 redistributes routes according to the IGP topology. Thus, the route propagation is simple to understand. Redistribution is automatically adapted, making the iBGP topology easy to configure and maintain.

For more details regarding the way routers were configured, we refer the reader to our datasets and the corresponding results, publicly available at [1]. Note that we also provide the code of the simulator and the scripts to plot the curves presented in this article. Indeed, we want to allow the research community to challenge the three considered iBGP redistribution mechanisms on any topology, not only the one we used.

C. Results

Our plots depict the minimal, average and maximal results obtained for each value of \(|C|\).

a) **Feasibility:** Figure 8 and Figure 9 illustrate, as expected, that the full mesh is the most costly solution, while RR and iBGP2 exhibit similar a performance. Note that the cost observed for intermediate values of \(|C|\) are higher because \(\binom{|T|}{|C|}\) prefixes are involved in the simulation. For example, one can easily show that the number of BGP updates measured in the full mesh is exactly \(\binom{|T|}{|C|} \cdot |C| \cdot (|T| - 1)\).

b) Correctness: Figure 12 shows that iBGP2 behaves as anticipated, as a full mesh would, and hence guarantees the fm-optimality criterion, while in RR, some routers cannot learn their optimal route. Note that RR obtains good results because we consider an iBGP topology which is almost fm-optimal.

c) Robustness: As depicted on Figure 10, a full mesh provides the best diversity: each router learns all the possible egress points for all the prefixes (i.e., exactly \(|C|\) routes per prefixes). RR and iBGP2 offer a comparable route diversity.
d) Fast convergence: Figure 11 suggests that convergence times are close for all solutions. This is due to the length of iBGP paths which is almost the same. iBGP paths are slightly longer in iBGP2 compared to Route Reflection, explaining its higher convergence time.

Overall, our simulation results confirm that iBGP2 is a realistic solution since its results are most of the time comparable to Route Reflection. Moreover, by design, iBGP2 guarantees that routing is optimal as depicted on Figure 12.

V. RELATED WORK

A. Move BGP routing decisions outside the routers

RCP (Routing Control Platform) [8] proposes to centralize the routing plane into a single RCP node. The RCP node is omniscient and so can configure an optimized routing plane. The main drawback of this approach is its highly centralized nature. The RCP node has therefore to process a huge amount of information, and becomes a single point of failure.

oBGP [21] adopts a similar approach, but rather than assigning the whole IP space to a single node, network administrators split the IP space to multiple oBGP nodes. Each oBGP node knows the whole AS topology and can redistribute an optimal route to each router for the IP space it is in charge of, leading to fm-optimal routing. Prefix subsets assigned to each oBGP node may overlap to improve recovery in case of failure.

B. Enhancing RR capabilities

In [5], the authors propose to make RR more intelligent, by adapting the routes they redistribute based on various inputs, e.g., measurements. The exact redistribution mechanism is not specified, but the authors propose to redistribute routes to iBGP peers so as to balance the traffic load across the network.

ORR (Optimal Route Reflector) [24] aims at improving RR behaviour by making the router able to determine among the routes it learns the most relevant one for each of its neighbors. To do so, the ORR takes advantage of its knowledge of the IGP topology and runs a BGP decision process from its neighbors’ point of view. However, this overhead is costly in terms of resources. To tackle this issue, VRR (Virtual RR) aggregates client routers into pools. For each pool of client routers, a VRR runs a decision process for one of them, and apply the result to all of them. ORR/VRR might seem close to iBGP2, but the way the iBGP routes are redistributed is different. First, an ORR/VRR router determines route redistribution on a per prefix basis, whereas an iBGP2 router computes its redistribution rules on a per next-hop basis. Second, ORR/VRR still relies on an arbitrary hierarchy contrary to iBGP2 overlay which guarantees fm-optimality. This criterion is not always satisfied in ORR/VRR since decision are taken only 1 hop away while being constrained by the iBGP overlay. Thus, ORR/VRR may suffer from the same issues as standard RR.

C. LOUP/SOUP

[15] proves that a sufficient condition to avoid forwarding loops consists in guaranteeing that a router only adopts a BGP route if the corresponding IGP next-hop has also adopted this route. To do so, the authors of [15] propose a clean-slate approach, and design two new protocols (LOUP/SOUP) to distribute BGP routes over the AS. As explained in Section III-C2, iBGP2 guarantees by design the condition high-
lighted by [15]. Moreover, contrary to LOUP/SOUP, iBGP2 is backward compatible.

 VI. DISCUSSION

 a) Compatibility: We do not expect that all routers support iBGP2. As explained in Section IV-A, establishing mutual RR sessions and managing dynamically filters along these sessions can exactly reproduce the behavior of an iBGP2 session. If such filters cannot be managed (e.g., on legacy routers), the outcome is simply additional (suboptimal) routes. This overhead of routes will not impact the final routing since they will not be selected. Moreover, iBGP2 does not impact the structure of the packets (neither BGP nor IGP). In that sense, iBGP2 is backward-compatible with the current BGP. iBGP2 is of course compatible with MP-BGP [3].

 b) Traffic Engineering: ISPs often require to perform traffic engineering (TE) to improve the network performances. TE is usually achieved on ASBRs by changing BGP weights or target prefixes along eBGP sessions [23]. Such sessions are obviously not impacted by the adoption of iBGP2. TE along iBGP2 sessions, which is sometimes performed but not a best practice, could be achieved by running a legacy iBGP redistribution in parallel for those particular prefixes. Since iBGP2 follows the IGP graph, the choice of egress point for quasi-equivalent routes can be done by tweaking IGP weights.

 c) Additional benefits of iBGP2: The iBGP2 topology exactly matches the IGP topology, thus iBGP2 adjacencies could be established automatically. An iBGP2 topology is always fm-optimal, and therefore does not suffer from side-effects due to unwanted interactions between the IGP and the iBGP topologies. Improving the traffic matrix of the network only depends on the metrics chosen in the IGP topology.

 VII. CONCLUSION

 In this article we have proposed a new iBGP route redistribution mechanism called iBGP2, a scalable approach which not only guarantees routing optimality and correctness for any IGP topology, but also simplifies router’s BGP configuration, thus decreasing the risk of misconfiguration. We have released an open-source implementation of our mechanism. We have simulated and compared its behaviour to the current default options, namely the full mesh and Route Reflection. Our results confirmed that iBGP2 scales similarly to Route Reflection, while providing the routing optimality of a full mesh. iBGP2 is therefore a desirable alternative which moreover remains backward-compatible with legacy routers. iBGP2 could be extended to support several parallel IGP routing planes (relying on bandwidth, latency, etc.), and hence offers the opportunity to manage egress point selection according to different kind of metrics.

 The authors acknowledge Bruno Decraene and the reviewers for their valuable comments. This work has been partially carried out at Laboratory of Information, Networking and Communication Sciences (LINCS).

References