Designing optimal iBGP route-reflection topologies

Marc-Olivier Buob[†], Steve Uhlig[*] and Mickael Meulle[†]
[†] Orange Labs
{marcolivier.buob,michael.meulle}@orange-ftgroup.com
[*]Delft University of Technology
S.P.W.G.Uhlig@ewi.tudelft.nl

Abstract. The Border Gateway Protocol (BGP) is used today by all Autonomous Systems (AS) in the Internet. Inside each AS, iBGP sessions distribute the external routes among the routers. In large ASs, relying on a full-mesh of iBGP sessions between routers is not scalable, so route-reflection is commonly used. The scalability of route-reflection compared to an iBGP full-mesh comes at the cost of opacity in the choice of best routes by the routers inside the AS. This opacity induces problems like suboptimal route choices in terms of IGP cost, deflection and forwarding loops. In this work we propose a solution to design iBGP route-reflection topologies which lead to the same routing as with an iBGP full-mesh and having a minimal number of iBGP sessions. Moreover we compute a robust topology even if a single node or link failure occurs. We apply our methodology on the network of a tier-1 ISP. Twice as many iBGP sessions are required to ensure robustness to single IGP failure. The number of required iBGP sessions in our robust topology is however not much larger than in the current iBGP topology used in the tier-1 ISP network.
Keywords: BGP, route-reflection, iBGP topology design, optimization.

1 Introduction

The Internet consists in a collection of more than 25,000 interconnected domains called Autonomous Systems (ASs). Inside a single domain, an Interior Gateway Protocol (IGP) [1] such as IS-IS or OSPF is used to ensure the routing between each router of the domain. In a domain, each IGP router computes its shortest path (according to the IGP metric) to each other router of the domain. Between neighboring ASs, routers exchange their routing information thanks to BGP [2]. External BGP (eBGP) sessions are established over inter-domain links, i.e. links between two different ASs (BGP peers), while internal BGP (iBGP) sessions are established between the routers inside an AS.

Typically, BGP routers do not forward iBGP messages to their iBGP peers, to reduce the amount of routing messages. Thus, each router has to establish an iBGP session with each other BGP router of its AS to diffuse its own routes. Such a topology is called a *iBGP full-mesh* (fm) and requires $n(n-1)/2$ iBGP sessions where n is the number of BGP routers in the AS. This solution is commonly used in small ASs but does not scale. To achieve a scalable iBGP topology in a larger AS, network administrators have to use BGP confederations[1] [3] or route-reflection[2].

[1] It consists in splitting the AS into smaller domains.
[2] Some BGP routers, namely route-reflectors, are authorized to forward iBGP messages to their iBGP clients.

Because route-reflection is the commonly used approach today in large ASs, we only focus on it for the rest of the paper. Despite its wide adoption, route-reflection may suffer from problems in two particular cases:

1. Some routers select their route according to the Multi Exit Discriminator (MED) attribute. These routing problems have already been studied in [4] and can easily be avoided with *always-compare-med* or *set-deterministic-med*.
2. The best routes are selected according to the *hot-potato routing steps* of the BGP decision process: *"prefer a route learned by eBGP over a route learned by iBGP"*, and *"prefer a route with a closest BGP next-hop"* (see [5]).

In large ASs, the hot-potato steps are of particular importance as they may account for up to 70% of the BGP best routes [6]. Furthermore, an iBGP topology with route-reflectors does not guarantee that each router systematically selects its closest possible egress point in the AS towards each destination. Indeed, route-reflectors reduce routing diversity because they only forward their best BGP route for each destination. Ideally, the network should converge to the same final state as in a full-mesh. Such a topology is said to be *fm-optimal* [5].

Before presenting how we address the iBGP topology design problem, let us summarize the the main advantages and drawbacks of full-mesh iBGP and route-reflection. A full-mesh has optimal routing and is deterministic. Convergence is fast and the network is as robust as possible. It is however not scalable, as many iBGP sesssions are necessary, and adding or removing routers imply significant configuration overhead. Furthermore, a change in a best route triggers updates to all other routers. In route-reflection on the other hand, scalability is improved in terms of configuration overhead, convergence, and size of routing information bases. However, it comes with loss of route diversity [7], which may induce suboptimal routing and non-deterministic routing [8], routing oscillations [4, 9, 10], route deflection or forwarding loops [11–13]. Furthermore, the behavior of route-reflection under failures [14] and IGP topology changes is unclear.

In this paper, we aim at building iBGP topologies that respect simultaneously several essential requirements:

– **Fm-optimality:** route-reflection is a scalable alternative to an iBGP full-mesh. We do not compromise the optimal route selection under an iBGP full-mesh while using route-reflectors. As we show in this paper, keeping the best of both worlds, while not trivial, is possible.
– **Correctness:** checking for correctness is proved to be NP-hard. But thanks to *fm-optimality*, we can nonetheless ensure that a network is both loop-free (because deflection-free) and deterministic, and therefore correct.
– **Reliability:** We design an iBGP topology that follows the IGP graph as much as possible [14, 15]. We allow multi-hop sessions[3] only if necessary.
– **Robustness:** We build a topology robust to IGP link failures and router maintenance. Furthermore, even after a single link failure or the removal of a router, the topology remains *fm-optimal*.

[3] when iBGP sessions cross other BGP routers than the two BGP end-points of the session.

– **Scalability:** We build a topology having as few iBGP sessions as possible.

As far as we know, this paper is the first attempt to design route-reflection topologies that respect all previously mentioned requirements, and in particular *fm-optimality* robust to failures. Using an algorithm based on the Benders' decomposition framework, we manage to solve efficiently the problem on real-world network topologies.

The remainder of this paper is structured as follows. Section 2 presents the terminology. Section 3 proposes different approaches to solve the problem. In Section 4 we evaluate our solutions on both real-world transit ISP networks and generated topologies. The related work is presented in Section 5. Finally, Section 6 concludes and discusses further work.

2 Terminology

IGP graph Let be $G_{igp} = (V_{igp}, E_{igp})$ the physical topology of the network. Each vertex of V_{igp} represents a router and each weighted (u, v) arc of E_{igp} characterizes a physical link and its IGP metric. We denote by $dist : V_{igp} \times V_{igp} \longrightarrow \mathbb{N}$ the function which returns the weight of the shortest path between two routers.

BGP graph We denote by $\mathcal{N}$ the set of possible BGP next-hops in routes learned by the AS, and by $\mathcal{R}$ the set of routers running BGP inside the AS. The graph $G_{bgp} = (V_{bgp}, E_{bgp})$ describes the route-reflection topology. $V_{bgp} = \mathcal{R} \cup \mathcal{N}$. E_{bgp} denotes the set of BGP sessions between routers. When two routers share an iBGP session, we add two edges between the routers labeled with UP from a client to one of its route reflectors, $DOWN$ from a route-reflector to one of its clients, or $OVER$ between peers (see Figure 1 and section 2). We use the same notations as [8, 15].

We assume that the border routers of the AS (ASBR) are the BGP next-hops in the BGP routes (i.e. $\mathcal{N} \subseteq \mathcal{R}$). We denote by $\mathbb{L} = \{UP, OVER, DOWN\}$ the types of BGP sessions and by $label : E_{bgp} \longrightarrow \mathbb{L}$ the label of a given link. We also denote by $sym : \mathbb{L} \longrightarrow \mathbb{L}$ the function that returns the symmetric label of a given label: $sym(UP) = DOWN$, $sym(DOWN) = UP$, $sym(OVER) = OVER$

A BGP path in G_{bgp} is a *valid path* if composed of zero or more UP arcs, followed by zero or one $OVER$ arc, followed by zero or more $DOWN$ arcs. Any valid iBGP label sequence verifies the regular expression $(UP)^*(OVER)?(DOWN)^*$.

Let $(n, r) \in \mathcal{N} \times \mathcal{R}$ be a given next-hop router pair. When considering this pair (n, r), we assume that:

– there exists a prefix p and several routes (called concurrent routes) to this prefix in the AS that are tie-breaked on the IGP cost to the next-hop,
– n is the closest BGP next-hop of r in the IGP graph.

We try to ensure that r will always be able to learn the route advertised by its closest BGP next-hop n. We only have to consider the following concurrent next-hop set:

$$\mathcal{N}(n, r) = \{n' \in \mathcal{N}, dist(r, n') > dist(r, n)\}.$$

If there exists a valid iBGP path from n to r such that each router w of that path selects the route advertised by n, then r learns the route advertised by n. We call *white-router* a router verifying this property. The set of white routers related to a given pair (n, r) is defined by:

$$\mathcal{W}(n, r) = \{w \in \mathcal{R} | \forall n' \in \mathcal{N}(n, r), dist(r', n) < dist(r', n')\}$$

Note that n and r always belong to $\mathcal{W}(n, r)$. Moreover $\mathcal{N}(n, r) \cap \mathcal{W}(n, r) = \emptyset$.

We call *white-path* any iBGP path only made of *white-routers*. If for each $(n, r) \in \mathcal{N} \times \mathcal{R}$ there exists at least one valid white path, then the topology is said *fm-optimal*. Note that *fm-optimality* is prefix independent. This criterion ensures a good "behaviour" of the network for *any* set of concurrent BGP routes.

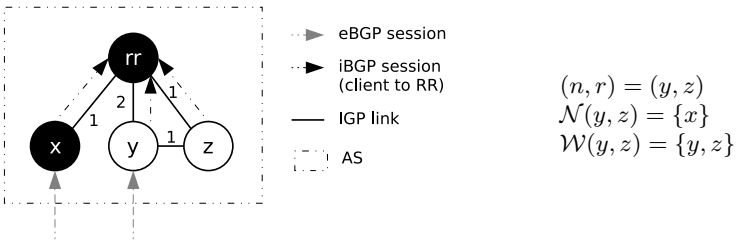

Fig. 1. An example of suboptimal routing: the traffic sent by z follows the IGP path (z, rr, x) instead of (z, y).

Figure 1 provides a brief example illustrating *fm-optimality*. In this topology, (y, rr, z) is a valid iBGP path from y to z. However $rr \notin \mathcal{W}(y, z)$. (y, rr, z) is not a valid white-path. In fact, rr may select the route advertised by x and z is thus unable to learn the route announced by its closest BGP next-hop y.

3 How to build fm-optimal iBGP topologies

We now introduce our approach to solve the iBGP design problem. As input, we need the BGP next-hops set ($\mathcal{N}$), the BGP routers set ($\mathcal{R}$), and the IGP topology (G_{igp}). We use a mixed-integer program (MIP) to model the problem. It is impractical to enumerate the whole set of constraints because the networks using route-reflection are typically very large. That is why we use a Benders' decomposition to generate dynamically a reduced set of constraints.

1. In a first step (section 3.1), we present the approach to solve the iBGP design problem when no failure happens, called *nominal case*. For each pair $(n, r) \in \mathcal{N} \times \mathcal{R}$, we build a satellite problem which will be satisfied if and only if at least one *fm-optimal* path exists from n to r.

2. In a second step (section 3.2), we detail how to introduce constraints to be robust to failures. We build a satellite for each triple (n, r, f) with a given failure f. We aggregate during a presolve step redundant satellites to reduce the problem size.

We do not consider $OVER$ sessions for Benders' decomposition to avoid the problem degenerescence, i.e. having too many equivalent solutions. Each $OVER$ session may be turned into an UP or $DOWN$ session without invalidating any iBGP path.

3.1 Nominal case

Master problem

Variables For each candidate iBGP session (u, v), $(u, v) \in \mathcal{R}, u \neq v$, we define two 0-1 variables: $up(u, v)$ (equal to 1 if $label(u, v) = UP$, 0 otherwise), and $down(u, v)$ (equal to 1 if $label(u, v) = DOWN$, 0 otherwise).

Objective function We try to design an iBGP topology as close as possible to the IGP topology while minimizing the number of installed iBGP session. We denote by F the objective function defined by:

$$F = min(\sum_{(u,v)\in\mathcal{R}} (R(u, v).(up(u, v) + down(u, v))))$$

where $R(u, v)$ characterizes the number of IGP hops needed[4] to establish an iBGP session from u to v.

Constraints The master problem is made of two sets of constraints:

- **Domain constraints:** Each pair $(u, v) \in \mathcal{R} \times \mathcal{R}$ is connected by 0 or 1 iBGP session, and $label(u, v) = sym(label(v, u))$. This leads to the following linear constraints:
  - $\forall u, v \in \mathcal{R},\ up(u, v) + down(u, v) \leq 1$,
  - $\forall u, v \in \mathcal{R},\ up(u, v) = down(v, u)$.
- **Max-flow Min-cut constraints:** At the beginning, this set of constraints is empty. These constraints will be described in Sections 3.1 and 3.2.

At each iteration it, the master problem queries many satellites problems to test the ***fm-optimality*** of their corresponding pair (n, r). Each queried unsatisfied satellite problem inserts a new max-flow min-cut constraint into the master problem. The set of max-flow min-cut constraints ensures the route propagation between any router pair (n, r) through the iBGP graph. If all satellite problems are satisfied, the MIP resolution returns a ***fm-optimal*** solution which minimizes the objective function F.

Satellite problems

[4] This value is equal to the length of the shortest IGP path from u to v.

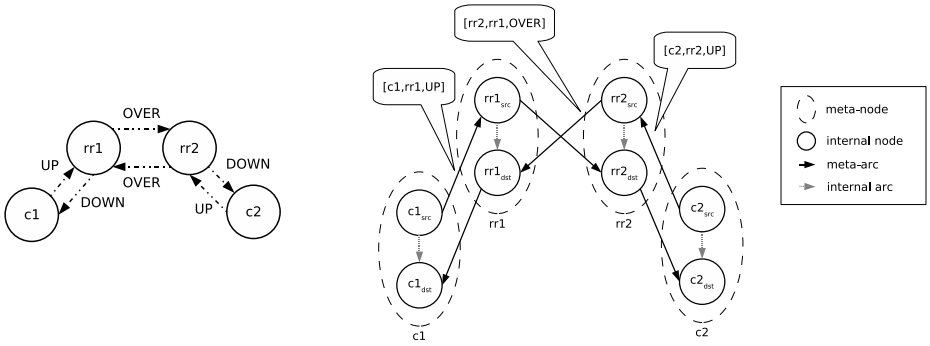

Fig. 2. An example of iBGP graph and its corresponding extended graph. The valid path (c_1, rr_1, rr_2, c_2) in G_{bgp} is mapped to $(c1_{src}, rr1_{src}, rr2_{src}, rr2_{dst}, c2_{dst})$ in G^{ext}_{bgp}.

Extended graph concepts To guarantee that only valid iBGP paths can be built, we use the graph transformation introduced in [5]. We transform each vertex of V_{bgp} into a *meta-node* composed of two nodes (called *source-node* and *target-node*) and an arc (called *internal arc*) according to Figure 2. The way we link two *meta-nodes* depends on the iBGP relationship between the two related routers. In the extended graph we can only build valid iBGP paths. We call *meta-arc* each arc that connects two vertices belonging to different *meta-nodes*. In this graph, each *meta-arc* is mapped to the iBGP session and the two routers establishing this iBGP session. We denote by $[u, v, rel]$ the *meta-arc* which is mapped to meta-nodes u and v, and to iBGP session rel. Each valid path of G_{bgp} from $s \in V_{bgp}$ to $t \in V_{bgp}$ is thus mapped to exactly one path in the extended graph from s_{src} to t_{dst}, where s_{src} is the *source-node* of s and t_{dst} the *target-node* of t.

Satellite graphs To ensure the *fm-optimality* of a given pair (n, r), we build a satellite problem. For each satellite problem we build a satellite graph $G_w(n, r)$. Each vertex of this graph belongs to $\mathcal{W}(n, r)$ (see Section 2). To reduce the number of candidate iBGP sessions, we only consider the sessions (u, v) which verify the following properties:

1. $u, v \in \mathcal{W}(n, r)$: the BGP route sent by n only goes through routers that never hide this route;
2. $dist(n, u) \leq dist(n, v)$ and $dist(v, r) \leq dist(u, r)$: a BGP message crossing the arc (u, v) increases its distance to n and decreases its distance to r.

The iBGP sessions that do not verify point 1 might prevent the *fm-optimal* routes from being propagated, so we do not want to use those sessions. Points 2 prevents iBGP messages announced by n to follow too long paths from n to r. Note that the iBGP full-mesh remains a possible solution because the direct session between n and r is allowed.

Thus, the graph $G_w(n, r) = (\mathcal{W}(n, r), E_w(n, r))$ gathers the candidate iBGP white-paths able to satisfy the (n, r) pair. If for all $(n, r) \in \mathcal{N} \times \mathcal{R}$, there exists at least one valid path from n to r in $G_w(n, r)$, then the iBGP topology is *fm-optimal*.

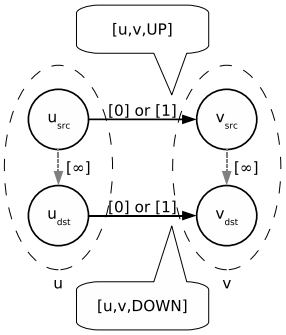

Fig. 3. Two candidate iBGP sessions: $label(u, v) = UP$ or $label(u, v) = DOWN$.

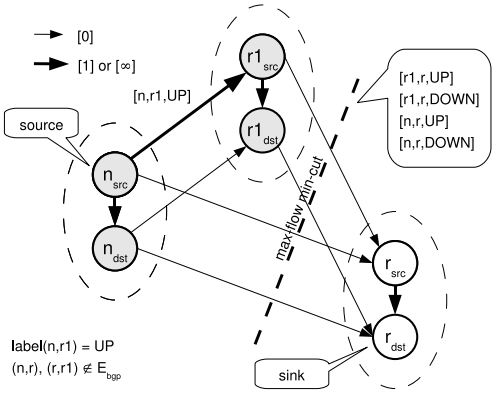

Fig. 4. This min-cut max-flow inserts the following constraint into the master problem: $up(r_1, r)+down(r_1, r)+up(n, r) + down(n, r) \geq 1$.

Satellite problems For each pair (n, r) we build the extended graph $G_w^{ext}(n, r)$ from $G_w(n, r)$ and all the candidate *meta-arcs*. We denote by n_{src} the *source-node* of *meta-node* n and r_{dst} the *target-node* of *meta-node* r. We install for each edge $(i, j) \in G_w^{ext}(n, r)$ an arc capacity, as shown in Figure 3.

- If i and j belong to the same *meta-node*, we install on (i, j) an infinite capacity.
- Otherwise (i, j) is a *meta-arc*. Let $rel \in \{UP, DOWN\}$ be the iBGP relationship mapped to (i, j), r_i the *meta-node* mapped to i, and r_j the *meta-node* mapped to j: If an iBGP session rel is set from r_i to r_j, we install on the arc (i, j) a capacity equal to 1 (0 otherwise). Therefore, there is at most one *meta-arc* from *meta-node* r_i to *meta-node* r_j with a capacity equal to 1.

If the max-flow sent from n_{src} (the source) to r_{dst} (the sink) is greater or equal to 1 then the pair (n, r) is satisfied. Otherwise, no flow unit can reach the sink. We search the max-flow min-cut in this graph. We denote by $C(n, r, it)$ the set of *meta-arcs* that intersect this cut during the current iteration it. We insert the following max-flow min-cut into the master problem:

$$\sum_{[r_i, r_j, rel] \in C(n,r,it)} (rel(r_i, r_j)) \geq 1.$$

Figure 4 provides an example of max-flow min-cut. In this example $\mathcal{W}(n, r) = \{n, r_1, r\}$ and the previous MIP resolution has lead to $label(n, r_1) = UP, label(r_1, r) = label(n, r) = NOT$. When the satellite related to (n, r) is queried, the max-flow min-cut inserts the linear constraint $up(r_1, r) + down(r_1, r) + up(n, r) + down(n, r) \geq 1$ into the MIP.

3.2 IGP failures

We now detail how to take into account IGP failures. A BGP router uses its IGP shortest path to establish the sessions with its BGP neighbors. When an IGP failure occurs,

the BGP router updates its IGP shortest path tree and re-establishes the broken BGP sessions according to its new path tree. If the IGP connectivity is not working between the two BGP peers, the BGP session will be down. We denote by $f = (V_{igp}^f, E_{igp}^f)$ an IGP failure, where $V_{igp}^f \subseteq V_{igp}$ stands for the set of involved routers and $E_{igp}^f \subseteq E_{igp}$ for the set of involved IGP links. We denote by ϕ the empty failure.

Methodology An IGP failure f consists in recomputing the IGP cost between each router pair belonging to the same connected component. For each considered input IGP failure, we apply the "nominal case" reasoning in each connected component. Let us consider a (n, r) pair such as n and r belong to a same IGP connected component C. We only consider in $G_w(n, r, f)$ the white-vertices belonging to C. An iBGP session can only be mounted if both BGP routers sharing the session belong to the same IGP connected component. Thus, we only have to consider IGP failures such that n and r remain in the same IGP connected component and $n \notin V_{igp}^f$,$r \notin V_{igp}^f$.

Satellite aggregation If we construct a satellite (n, r, f) for each IGP failure (including ϕ), we notice that many satellites are redundant. For example, if f does not affect a given pair (n, r), it is useless to build the satellite (n, r, f) as the (n, r, f) and (n, r, ϕ) satellites will insert the same flow constraints. Let us consider two failures f and f' for a given pair (n, r). Let be $G_w(n, r, f)$ and $G_w(n, r, f')$ the two related flow graphs. If $G_w(n, r, f) \subseteq G_w(n, r, f')$, $G_w(n, r, f)$, the constraints induced by $G_w(n, r, f)$ will be more restrictive than the $G_w(n, r, f')$ ones. Thus we can safely omit the (n, r, f') satellite.

Failure case study In the next part we will only consider the single IGP node and link failure cases, which is the most common case of network failure. To be as generic as possible, we assume that an IGP router failure also provokes the corresponding iBGP router failure. We also remove the unmountable iBGP sessions (if the two iBGP routers are not in the same IGP connected component anymore).

4 Building optimal iBGP topologies

To illustrate how the different approaches perform on different topologies, we present in this section results for real and generated topologies. We start in Section 4.1 with relatively small topologies of less than 30 nodes. Then we tackle the problem on a large tier-1 network in Section 4.2.

4.1 Small topologies

We first compare our two approaches on five topologies.

– The topology of the GEANT network from 2004[5].

[5] `http://www.geant.net`

		GEANT	NA-D	NA-2T	W-D	W-2T
Input graph	$\|V_{bgp}\| = \|V_{igp}\|$	22	25	25	25	25
	$\|E_{igp}\|$	72	128	96	130	96
	$\|E_{bgp}\|$ in f.m.	462	600	600	600	600
Without failure	$\|E_{bgp}\|$	74	80	72	100	64
With failures	$\|E_{bgp}\|$	172	168	146	194	126

Table 1. Solutions found on small topologies.

– Four topologies generated by the iGen topology generator[6]. iGen allows to generate random points in one or any continent, and then to connect the nodes using network design heuristics [16]. We generated four small topologies made of 25 nodes. Two of them for Northern America (NA) and two for the whole world (W). Two network design heuristics were used to generate the physical connectivity between nodes: Delaunay triangulation (D) and the Two-Trees algorithm (2T).

We assume that each router is a border router ($\mathcal{N} = \mathcal{R} = V_{igp}$), i.e. any BGP router may receive external routes towards any arbitrary prefix. This is the most constrained form of the problem. Indeed, the computed iBGP topology remains *fm-optimal* for all subset of $\mathcal{N}$. Thus, a smaller next-hop set would lead to an iBGP topology made of less iBGP sessions. Table 1 shows the results of the Benders approach with and without failures (last two rows), for the five topologies described above. The number of iBGP sessions required for an iBGP full-mesh are also indicated in the third row.

The real GEANT network is configured with a full-mesh of iBGP sessions, i.e. 462 directed $OVER$ sessions between its 22 routers. According to Benders' decomposition results, GEANT would only need 74 iBGP sessions under route-reflection to ensure *fm-optimality*, and 170 iBGP sessions to ensure *fm-optimality* under single failures.

Delaunay topologies (NA-D and W-D) are more connected than the Two-Trees ones that are made of two disjoint spanning trees. Benders finds iBGP topologies with strictly less iBGP sessions than IGP links, for the four iGen topologies, as those graphs are well-connected. The World-Delaunay iGen topology requires multi-hop sessions to reach *fm-optimality*, like for GEANT.

When IGP failures are considered (last row of Table 1), the number of required iBGP sessions roughly doubles. This is still about 3 times less sessions than an iBGP full-mesh, while *fm-optimality* being guaranteed.

To illustrate the properties of the topologies computed by the Benders approach, we use three indicators. Figures are shown for the GEANT network only because we observe a similar behaviour for the iGEN topologies.

1. *Degree distribution:* when more iBGP sessions are established by a router (higher degree), it requires more memory because of larger RIB-Ins. In the nominal topology (top left part of Figure 5), each router has iBGP sessions with at most 7 BGP routers, while with up to 15 when failures are considered.
2. *White path length distribution:* for each pair (n, r), we compute the iBGP path length in terms of iBGP hops. The more iBGP hops are needed for a router to get

[6] `http://www.info.ucl.ac.be/~bqu/igen/`

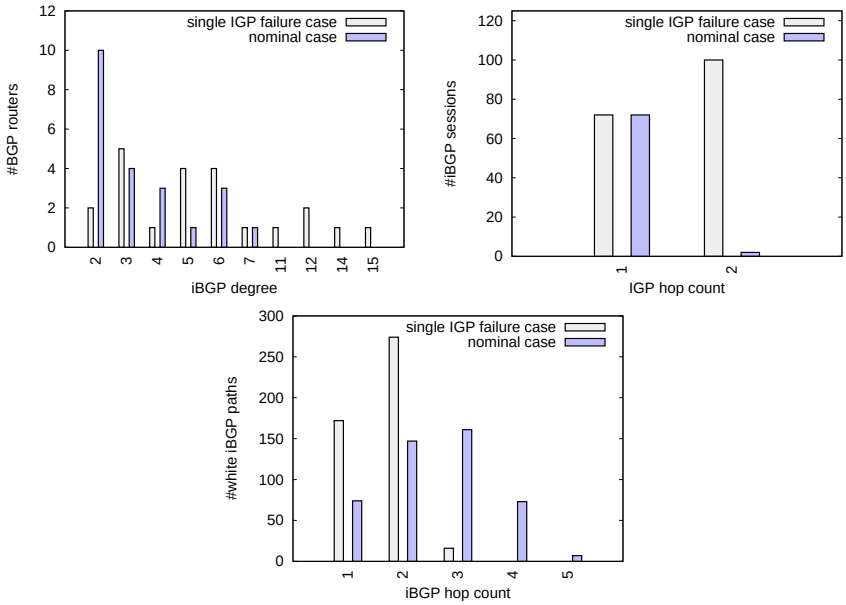

Fig. 5. Properties of iBGP topologies generated by the Benders' approach (GEANT network).

its BGP route, the slower the convergence time inside the AS will be. The top right of Figure 5 shows that in the nominal case most white paths have not more than 1 or 2 iBGP hops, while tending to be longer when IGP failures are considered.

3. *Matching with the IGP topology:* for each iBGP session, we look at its IGP hops length. Ideally, the iBGP topology should be as close as possible to the IGP topology [14, 15]. The lower part of Figure 5 shows that most of the iBGP sessions go over a single IGP link in the nominal case. However, a majority of iBGP sessions cross two IGP links when failures are considered.

4.2 Tier-1 ISP network

To show how our approach scales to large networks, we rely on a tier-1 provider network topology having hundreds of nodes and iBGP sessions. Due to confidentiality reasons, results are presented differently than for the small topologies. We denote by $G_{bgp}^{original} = (V_{bgp}, E_{bgp}^{original})$ the original iBGP topology of the tier-1 AS and by $G_{bgp}^{benders} = (V_{bgp}, E_{bgp}^{benders})$ the iBGP topology computed by the Benders approach. Table 4.2 shows that many modifications have to be done to migrate from the original topology to the computed topologies. The classical design rules used today (e.g. a 3-level route-reflection hierarchy made of intercontinental, continental, and national level) are simple, but do not lead to a reliable and efficient iBGP topology.

The nominal-case topology is 45% smaller than the original topology and is *fm-optimal*. Each router has to establish fewer iBGP sessions. However, the average white

$E_{bgp}^{original}$		$E_{bgp}^{benders}$	
Removed	73 %	Added	52 %
Modified	6 %	Modified	11 %
Kept	20 %	Kept	36 %

Table 2. Nominal topology

$E_{bgp}^{original}$		$E_{bgp}^{benders}$	
Removed	59 %	Added	67 %
Modified	27 %	Modified	21 %
Kept	12 %	Kept	10 %

Table 3. Topology robust to IGP failures

iBGP path length is a bit longer than in the original input topology, so BGP convergence might be slower.

The topology when considering IGP failures requires 25% more iBGP sessions than the original topology, but remains close to the original one in terms of iBGP degree distribution and white path length distribution.

5 Related work

[13] was the first work to notice the possibility of occurrence of forwarding loops in route-reflection. Loop-free forwarding in iBGP was studied in [15]. [15] proved that checking the correctness of an iBGP graph is NP-complete, and showed that two conditions ensure a correct (loop-free) iBGP graph: 1) route-reflectors should prefer client routes to non-client routes, 2) every shortest path should be a valid signaling path. Those two conditions are actually too restrictive, designing correct iBGP topologies does not require the first condition [11].

[14] provides an iBGP design problem formulation. The authors aim to optimize a kind of network robustess defined through two reliability criterion (Expected Lifetime and Expected Session Loss). The computed topology is made of two hierarchical levels.

[11] relied on this meshing of the top-level reflectors to design more scalable iBGP topologies that are robust to IGP failures. The approach of [11] relies on a hierarchy of route-reflectors that ensures the correctness of the iBGP propagation.

[12] details the common routing problems encountered in networks under route-reflection. This paper provides conditions to avoid route deflection and MED oscillations. The computed topology is a two-level reflection hierarchy and minimizes the IGP cost between two iBGP peers.

The last three approaches do not guarantee that the iBGP topology remains valid if some IGP failure occurs. Thus suboptimal routing or forwarding loops may occur.

6 Conclusion

In this work we proposed solutions to design optimal iBGP route-reflection topologies, i.e. route-reflection topologies that will lead to the same routing as with an iBGP full-mesh (*fm-optimal topology*). We showed that it is possible to build *fm-optimal* route reflection topologies with minimal number of iBGP sessions robust to IGP failures.

Applying our method on both real-world and generated networks revealed that guaranteeing *fm-optimality* is possible by using a number of iBGP sessions in the order of the number of physical links, if no IGP failures occur. When IGP failures are considered, the minimal required number of iBGP sessions roughly doubles compared to the

situation without failures, but still remains 5 times smaller than an iBGP full-mesh in the case of our tier-1 ISP.

iBGP aims at diffusing routing information inside an AS. This diffusion depends on the diffusion graph, and the protocol that drives the diffusion of the routing information. Our approach in this paper was to optimize the iBGP diffusion graph, without changing the rules of route-reflection. Another way to think about iBGP route diffusion is to change the iBGP protocol itself instead of designing an iBGP graph. We plan to explore this second way of designing iBGP in the future.

Acknowledgments We would like to thank Olivier Klopfenstein and Jean-Luc Lutton for their help on the Benders' approach.

References

1. B. Halabi and D. Mc Pherson, *Internet Routing Architectures (2nd Edition)*, Cisco Press, January 2000.
2. Yakov Rekhter and Tony Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, Mar. 1995.
3. P. Traina, D. McPherson, and J. Scudder, "Autonomous System Confederations for BGP," RFC 3065, February 2001.
4. Timothy Griffin and Gordon T. Wilfong, "Analysis of the med oscillation problem in BGP," in *ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols*, Washington, DC, USA, 2002.
5. M. Buob, M. Meulle, and S. Uhlig, "Checking for optimal egress points in iBGP routing," Proc. of the 6th IEEE International Workshop on the Design of Reliable Communication Networks (DRCN 2007), October 2007.
6. R. Teixeira, T. Griffin, G. Voelker, and A. Shaikh, "Network sensitivity to hot potato disruptions," in *Proc. of ACM SIGCOMM*, August 2004.
7. Steve Uhlig and Sébastien Tandel, "Quantifying the impact of route-reflection on BGP routes diversity inside a tier-1 network," in *Proc. of IFIP Networking*, Coimbra, Portugal, May 2006.
8. Nick Feamster, Jared Winick, and Jennifer Rexford, "A Model of BGP Routing for Network Engineering," in *ACM Sigmetrics - Performance 2004*, New York, NY, June 2004.
9. D. McPherson, V. Gill, D. Walton, and A. Retana, "BGP persistent route oscillation condition," March 2001.
10. A. Basu, L. Ong, B. Shepherd, A. Rasala, and G. Wilfong, "Route oscillations in i-BGP with route reflection," in *ACM SIGCOMM*, 2002.
11. Mythili Vutukuru, Paul Valiant, Swastik Kopparty, and Hari Balakrishnan, "How to construct a correct and scalable iBGP configuration," in *IEEE INFOCOM*, Barcelona, Spain, April 2006.
12. Anuj Rawat and Mark A. Shayman, "Preventing persistent oscillations and loops in iBGP configuration with route reflection," *Computer Networks*, pp. 3642–3665, December 2006.
13. Rohit Dube, "A comparison of scaling techniques for BGP," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 3, pp. 44–46, 1999.
14. L. Xiao, J. Wang, and K. Nahrstedt, "Optimizing iBGP route reflection network," in *IEEE INFOCOM*, 2003.
15. Timothy G. Griffin and Gordon Wilfong, "On the correctness of iBGP configuration," in *Proc. of ACM SIGCOMM*, August 2002.
16. R. Cahn, *Wide area network design: concepts and tools for optimization*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.