When the Cure is Worse than the Disease: the Impact of Graceful IGP Operations on BGP

Laurent Vanbever, Stefano Vissicchio, Luca Cittadini, and Olivier Bonaventure

***Abstract*—Network upgrade, performance optimization and traffic engineering activities often force network operators to adapt their IGP configuration. Recently, several techniques have been proposed to change an IGP configuration (e.g., link weights) in a disruption-free manner. Unfortunately, none of these techniques considers the impact of IGP changes on BGP correctness.**

In this paper, we show that known reconfiguration techniques can trigger various kinds of BGP anomalies. First, we illustrate the relevance of the problem by performing simulations on a Tier-1 network. Our simulations highlight that even a few link weight changes can produce long-lasting BGP anomalies affecting a significant part of the BGP routing table. Then, we study the problem of finding a reconfiguration ordering which maintains both IGP and BGP correctness. Unfortunately, we show examples in which such an ordering does not exist. Furthermore, we prove that deciding if such an ordering exists is NP-hard. Finally, we provide sufficient conditions and configuration guidelines that enable graceful operations for both IGP and BGP.

I. Introduction and related work

Routing protocols are traditionally classified as either intradomain or interdomain protocols. Intradomain protocols or Interior Gateway Protocols (IGPs) such as OSPF and IS-IS are responsible for the shortest-path forwarding of packets within an Autonomous System (AS), i.e., a network operated by a single administrative entity. In contrast, interdomain protocols such as BGP [1] are responsible for the forwarding of packets across multiple ASes. Conceptually, for a given destination prefix, a router uses BGP to find what is the best egress point inside its own AS, and then the IGP to find the best way to reach that egress point. Although they serve different purposes, the two routing protocols are tightly coupled. Indeed, when choosing between equally preferred egress points, a BGP router breaks ties based on lower IGP costs.

Network operators often need to change their IGP configuration. One of the primary goals of these adjustments is to perform intradomain traffic engineering. Indeed, network operators can optimize the traffic traversing their network by appropriately changing the link weights (e.g., [2], [3], [4]). To compute optimal link weights, network operators can rely on widely available tools (e.g., [5], [6]). By adapting link weights, network operators can also perform planned maintenance on a link or a node by first rerouting traffic around it [7], [8]. Besides traffic engineering and planned maintenance, operators may also need to perform larger IGP reconfigurations as the network grows or when upgrades or new services must be deployed. These reconfigurations include introducing (or removing) a hierarchy or changing the protocol (e.g., to benefit from a different features set) [9], [10], [11].

Given the practical relevance of IGP reconfiguration scenarios, the research community has devoted a lot of effort to prevent forwarding loops and congestion from appearing during IGP reconfigurations. In [12], Raza *et al.* propose a theoretical framework and a heuristic to minimize a certain disruption function (e.g., link congestion) when link weights have to be changed. François *et al.* propose protocol extensions to avoid transient forwarding loops after a link addition or removal [13]. Fu *et al.* [14] and Shi *et al.* [15] generalize these results by defining loop-free reconfiguration techniques encompassing any change in the forwarding plane and considering traffic congestion, respectively. In [16], Vanbever *et al.* propose techniques and tools to safely reconfigure IGP when routers can run two IGP processes simultaneously.

While prior work has striven to guarantee graceful reconfigurations to IGP destinations, the potential impact on BGP has not been analyzed. Unfortunately, due to the interplay between IGP and BGP, graceful IGP operations can affect BGP decisions and cause unexpected BGP-induced anomalies. Even worse, such BGP-induced anomalies can have a much more dramatic effect on traffic than the transient disruptions that graceful IGP operations are intended to avoid. In fact, with respect to IGP anomalies, BGP anomalies can affect a larger number of destinations, impact a larger fraction of the traffic, and last much longer [17], [18], [19].

This paper studies the impact of IGP reconfigurations on BGP correctness. It makes the following contributions:

- **Experiments:** We simulated several IGP reconfigurations of a Tier-1 network. We found that many BGP-induced anomalies can persist for large parts of the reconfiguration process, even if few link weights are changed (Section II).
- **Theoretical analysis:** We show that reconfiguring the IGP can introduce all possible kinds of BGP anomalies, *even using state of the art IGP reconfiguration techniques*. We also show that in some scenarios it is impossible to avoid BGP anomalies (Section III and Section IV).
- **Complexity analysis:** We prove that deciding whether an anomaly-free IGP reconfiguration will trigger BGP anomalies is $\mathcal{NP}$-hard (Section V).
- **Configuration guidelines:** We describe sufficient conditions and configuration guidelines that guarantee the absence of BGP-induced anomalies. When the sufficient conditions hold in both the initial and the final IGP topology, we show that an anomaly-free IGP reconfiguration strategy always exists (Section VI).

II. The Impact of IGP Reconfigurations on BGP

In this section, we study the impact of the interactions between graceful IGP operations and BGP by running several experiments on a Tier-1 network. In each experiment, we simulated the reweighting of few IGP links. The IGP reconfiguration is performed using the technique proposed in [16] which provably avoids forwarding loops to any IGP destination. The technique consists in reconfiguring the IGP on a per-router basis following a precise order. We provide more details about this reconfiguration technique in Section III.

The Tier-1 network we considered consists of more than 100 routers and more than 150 links. The IGP is a link-state protocol in which every router applies a shortest path algorithm to compute its routing decisions. BGP route reflection [20] is configured on the network, and BGP routers are arranged in a three-layer route reflection hierarchy. In addition to the configurations of all routers, our data set includes a dump of all the BGP routes received by the route reflectors at the top layer.

We simulated three reconfiguration scenarios. In the first scenario, we reweighted 5 links ($\approx 3\%$ of all links), in the second scenario we reweighted 10 ($\approx 6\%$) links, and in the third scenario we reweighted 15 links ($\approx 10\%$). These scenarios are meant to capture typical reconfigurations performed by network operators to achieve better traffic engineering while minimizing the number of reweighted links [3]. For each scenario, we performed several experiments. In each experiment, we randomly chose the reweighted links with uniform likelihood. The new weight to be assigned to the selected links was randomly chosen within the set of weights used in the initial configuration.

We used SimBGP [21] (a BGP simulator) to compute the forwarding tables during the reconfiguration. To reduce the number of BGP prefixes used in the simulation, we group BGP prefixes into virtual prefixes. Namely, we univocally map a virtual prefix to a combination of routers that, in our data set, injected BGP routes to the same BGP prefix. We stress that each virtual prefix typically corresponds to several BGP prefixes since several BGP prefixes can be injected by the same set of routers.

In each experiment, we performed per-router reconfigurations as proposed in [16]. After each router reconfiguration, we waited for BGP convergence and we analyzed the resulting forwarding tables to identify BGP forwarding loops towards virtual prefixes. We repeated experiments on each scenario 30 times, using a different ordering in each experiment.

We found that numerous BGP forwarding loops can appear during the reconfiguration process. Fig. 1 plots the fraction of experiments experiencing a given amount of BGP-induced loops. A data point (x, y) in the graph means that $(100 * y)\%$ of the experiments exhibited x forwarding loops. When reweighting 5 links, BGP-induced forwarding loops happened for at most 85 virtual prefixes in the worst case, and for at least 2 virtual prefixes in more than 40% of the experiments. We stress that this is significant as each virtual prefix can potentially map to a large number of actual BGP prefix. When reweighting 10 (resp. 15) links, the likelihood that at least one virtual prefix experienced a BGP-induced forwarding loop was more than 70% (resp. 90%) of the experiments. In the worst case, we observed forwarding loops for more than 1800 virtual prefixes, indicating that several virtual prefixes were affected by different forwarding loops at different stages of the reconfiguration.

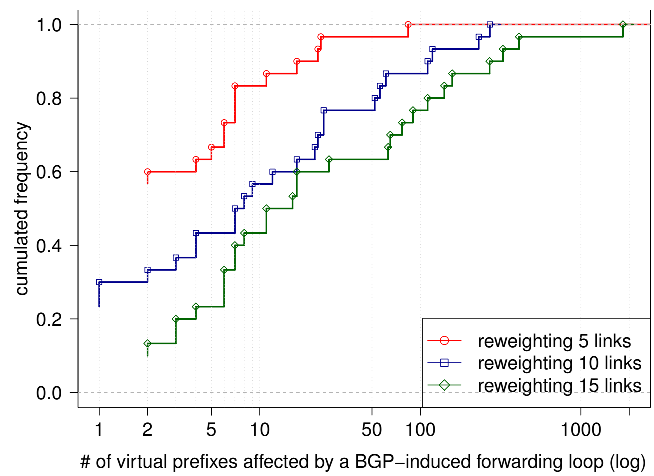

Fig. 1. Numerous BGP-induced forwarding loops can appear during IGP changes, even when state of the art techniques are applied.

	5 links	10 links	15 links
Average loop duration (% of process)	23.80	17.14	7.61
Average number of routers involved	5.33	5.33	7.90
Total number of routers involved	32.00	42.00	49.00
Maximum size of a loop (# routers)	2.00	2.00	8.00
Average size of RT impacted (%)	1.16	2.48	7.83
Total size of RT impacted (%)	15.00	31.00	97.00

TABLE I
BGP-induced loops are long-lived, involve multiple routers and impact a significant part of the BGP Routing Table (RT).

Besides potentially affecting a large number of prefixes, loops for BGP destinations can last during several steps of the reconfiguration process. In our experiments, some loops lasted for more than 20% of the entire reconfiguration process (Table I). While all the forwarding loops raised when reweighting 5 and 10 link involved 2 adjacent routers, we found some cases where as many as 8 routers were involved when 15 links are reweighted. For each impacted virtual prefix, we also accounted the number of real prefix impacted. We discovered that a significant fraction of the routing table can be impacted. Indeed, close to 8% of the RT was subject to at least one loop on average when 15 links were renumbered. Finally, we observed that different reconfiguration orderings created BGP loops toward different prefixes. For example, almost all virtual prefixes (97%) were impacted at least once in all the experiments we did for the 15 link reweighting scenario.

Step	Criterion
1	Prefer routes with higher `local-preference`
2	Prefer routes with lower `as-path` length
3	Prefer routes with lower `origin`
4	Prefer routes with lower lower `MED` (same next-hop AS)
5	Prefer routes learned via eBGP
6	**Prefer routes with lower IGP metric**
7	Prefer routes having the lowest `egress-id`
8	Prefer routes with shorter `cluster-list`
9	Prefer the route having the lowest `router-id`

TABLE II
BGP DECISION PROCESS.

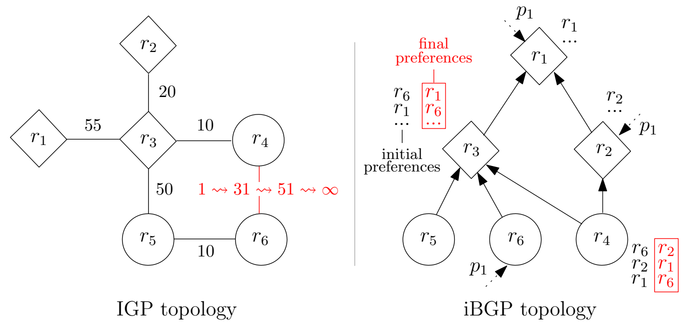

Fig. 2. SCISSORS GADGET: applying the metric-increment technique to avoid transient IGP loops cause forwarding loops to BGP destinations.

III. SHEDDING LIGHT ON BGP DISRUPTIONS

In this section, we analyze the coupling between IGP and BGP to gain a theoretical insight on BGP anomalies raised by state of the art IGP reconfiguration techniques. We also show that all IGP reconfiguration techniques can be responsible for BGP forwarding loops.

A. The interplay between IGP and iBGP

In a single AS, the route followed by a packet is determined by the interaction between the IGP and iBGP.

IGP controls packet forwarding between any pair of source and destinations belonging to the same AS. Most ISPs and enterprise networks deploy link-state IGPs (e.g., OSPF and IS-IS) as they scale better and converge faster. In the following, we focus on link-state IGPs, as previous work on graceful IGP operations did [13], [12], [16].

Internal BGP (iBGP) controls packet forwarding towards prefixes belonging to other ASes. Namely, iBGP keeps information about external destinations and Internet-wide route attributes. Based on this information, iBGP routers decide what is the last hop (or egress point) inside the AS to forward packets to a given external destination. Before installing the route in the forwarding table, iBGP relies on the IGP (by performing the so-called *recursive lookup*) to know the internal next-hop towards the selected egress point.

iBGP routers exchange routing information via iBGP sessions. As the original iBGP specification [1] mandates an iBGP full-mesh, a session between each pair of iBGP routers is required. For scaling reasons, two hierarchical mechanisms have been proposed: route reflection [20] and BGP confederations. In this paper, we focus on route reflection as it is the most widely adopted mechanism. With route reflection, the neighbors of each iBGP router are split into three sets: *clients*, *peers* and *route reflectors*. For each destination prefix, each iBGP router selects one best route among the routes it receives from its neighbors. Then, it propagates the best route according to the following rules: if the route is learned from a peer or from a route reflector, then it is relayed only to clients, otherwise it is reflected to all iBGP neighbors. In an iBGP full-mesh, all iBGP routers are peers. In general, however, a hierarchy of clients and route reflectors is established. We refer to the organization of iBGP sessions as *iBGP topology*.

The best route that each iBGP router selects and propagates is decided according to the BGP decision process [1] summarized in Table II. It consists of a sequence of rules. Whenever there are ties for a rule, the next rule is applied to break the tie. In the following, we only consider the BGP routes that are equally preferred according to the first four steps of the BGP decision process, as the other ones are discarded by all iBGP routers (on the basis of eBGP attributes). Observe that the sixth step of the BGP decision process takes into account IGP distances to egress points. Consequently, BGP routing decisions depends directly on the IGP configuration.

To summarize, the dependency between IGP and BGP is twofold. First, IGP metrics influence the BGP decision process. Second, IGP controls the forwarding paths used by each router to reach its selected BGP next-hop. In the following, we show how the dependencies between BGP and IGP produce undesired side effects on BGP routing and forwarding during IGP configuration changes.

B. BGP disruptions during graceful IGP reconfigurations

Recently, several techniques have been proposed to reconfigure IGP in a graceful manner, especially to serve traffic engineering purposes. We can roughly divide those techniques in two approaches. The first approach [22], [14], [15], [23], [12] consists in progressively changing routers' forwarding tables in such a way to minimize or avoid disruptions.The second approach consists in running two control-planes in parallel and applying a convenient operational ordering to switch from one control-plane to the other [10], [16]. In the following, we consider the metric-increment [22] and the ships-in-the-night (SITN) [16] techniques as representatives of the two approaches, respectively. We choose these two techniques as they are provably correct and require no modifications to current router implementation.

Metric-increment [22] is a reconfiguration technique that avoids transient loops during link reweighting. As an illustration, consider the IGP topology depicted on the left side of Fig. 2, where circles represent routers, diamonds represent route-reflectors, and link labels represent link weights. The distinction between circles and diamonds is only relevant for BGP. Now, assume that link (r_4, r_6) has to be shut down for maintenance reasons. To reduce convergence delay, network operators usually prefer to first reroute traffic out of the link by increasing its weight to a pseudo infinite value before actually shutting down the link [7]. However, if a network operator simply modifies the link weight in a single step, transient loops for IGP destinations might appear. For instance, depending on the message timing, a transient loop can arise between r_5 and r_6 for packets destined to r_3. Indeed, as soon as r_6 becomes aware of the link weight change, it starts forwarding to r_5 all the packets destined to r_3. If r_5 still relies on the old topological information, it will bounce back these packets as r_6 was on the shortest path from r_5 to r_3 before the link was reweighted.

The metric-increment technique consists in incrementing the link weight in progressive steps. At each intermediate step, the metric on the link is incremented in such a way that some of the routers that have shortest paths traversing the link will be able to select a better alternative without causing any loops. At the end of the sequence, no shortest path traverses the link and the reweighting process is complete. Interestingly, a loop-free weight increment sequence always exists [22]. In Fig. 2, the minimal sequence of weight assignment that prevents transient loops is $\{1 \rightsquigarrow 31 \rightsquigarrow 51 \rightsquigarrow \infty\}$. For example, setting the weight of link (r_4, r_6) to 31 prevents the previously described loop between r_5 and r_6. Indeed, this step forces r_5 to change its next-hop to r_3 *before* r_6 starts forwarding packets to r_5 as the shortest path from r_6 is still $(r_6\ r_4\ r_3)$.

Unfortunately, progressively incrementing link weights can create loops for BGP destinations. Even worse, this can happen even when both the initial and final configurations are known to be free from anomalies. Consider the iBGP topology on the left side of Fig. 2, where solid links represent iBGP sessions and are oriented from the client to the route-reflector. Dashed arrows represent external announcements received for a BGP destination prefix. The iBGP topology is a route reflection hierarchy in which r_1 is the top-layer route reflector, while r_1, r_2 and r_6 are egress points for prefix p_1. Each router is equipped with a list of egress points in descending order of preference. Some routers have two lists of egress points meaning that the IGP reconfiguration will change their egress point preferences. In this case, the boxed list represents the egress points preferences in the final IGP configuration.

We now describe the impact of the IGP reconfiguration process on BGP prefix p_1. As soon as the link weight is incremented to 31, a BGP-induced forwarding loop is created between r_3 and r_4. Indeed, r_4's best egress point for p_1 is now r_2. In contrast, r_3 does not learn r_2 due to iBGP propagation rules, hence it still uses r_6 as its egress point. Therefore, r_3 will forward packets to r_6 via r_4, while r_4 will send packets to r_2 via r_3, causing a forwarding loop. This loop disappears when the link weight is incremented from 31 to 51 as r_3 starts preferring r_1 over r_6.

Observe that a BGP-induced packet deflection persists in the final state as r_4 will send traffic to r_2 via r_3, while r_3 will deflect traffic to r_1. However, as this situation does not disrupt traffic, operators could be willing to tolerate it during the maintenance of link (r_4, r_6).

The main alternative to metric-increment is applying the **Ships-in-the-night (SITN)** technique. In addition to link reweighting, SITN can be also used in a variety of other scenarios including the replacement of protocol, the introduction of an IGP hierarchy or of route summarization [16]. With respect to metric-increment, SITN is especially convenient when several links have to be reweighted since it minimizes the number of transient routing states. SITN is based on the possibility of simultaneously running two IGPs. The reconfiguration then consists in waiting for the convergence of both IGP processes and then switch the process used for forwarding on a per-router basis. SITN also allows per-destination reconfigurations in which the forwarding of a router is reconfigured only for a single destination at each reconfiguration step [16].

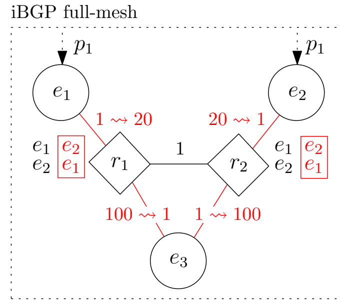

Fig. 3. VENDETTA GADGET: applying the SITN technique to avoid transient IGP loops cause forwarding loops to BGP destinations.

Since two routers could disagree about which IGP to use to forward a packet, SITN reconfiguration is prone to forwarding loops. Such forwarding loops can be avoided by reconfiguring routers in an appropriate order. Unfortunately a per-router ordering is not guaranteed to exist as there might exist contradictory ordering constraint for different destinations. In contrast, there always exists a *per-destination* ordering that guarantees the absence of forwarding loop towards any IGP destination [16]. Therefore, network operators can always trade traffic disruptions for the complexity of the reconfiguration process.

Another important property of SITN reconfiguration (see Prop. 1) is that reconfiguring a router has only a local impact since the initial and the final IGP configurations simultaneously run network-wide.

Property 1 (The SITN Locality Property)**.** *Assuming no network failures, migrating a router r only impacts r's forwarding choices.*

As an illustration of how SITN works, consider the network in Fig. 3. The reconfiguration scenario is such that links (e_1, r_1), (e_2, r_2), (r_1, e_3) and (r_2, r_3) have to be reweighted. The iBGP topology is a full-mesh and e_1 and e_2 are egress points for prefix p_1. The iBGP full-mesh guarantees that the initial and the final configurations are free from BGP anomalies. Consider now the reconfiguration process. To avoid forwarding loop towards the IGP destination e_3, r_1 must be reconfigured before r_2. Indeed, r_1 forwards traffic destined to r_3 via r_2 in the initial configuration while the opposite holds in the final one.

Unfortunately, Fig. 3 is an example of IGP reconfiguration in which the constraints to avoid IGP and BGP anomalies are contradictory. This means that respecting the constraint for IGP destination e_3 will result in a forwarding loop for BGP destination p_1. Indeed, r_2 forwards traffic destined to p_1 via r_1 in the initial configuration while r_1 forwards traffic destined to p_1 via r_2 in the final configuration. Hence, reconfiguring r_1 before r_2 to avoid loops to IGP destination e_3 will create a loop between r_1 and r_2 to BGP destination p_1.

Note that, in this example, the role of the iBGP topology is minimal as a full-mesh guarantees full route visibility (when a single control-plane is used). In fact, the BGP loop is not due to the partial route visibility, but to inconsistent states of the routers which rely on different IGP metrics.

IV. The extent of BGP disruptions

In Section II, we have presented several examples in which IGP reconfigurations created forwarding loops to BGP destinations. However, it is well-known that BGP configuration can also suffer from routing anomalies (e.g., oscillations) due to IGP dependency and the partial lack of visibility induced by route reflection [24], [25]. In this section, we show how IGP reconfigurations can also create any type of BGP routing anomalies. Moreover, we also describe an example in which no per-destination reconfiguration is graceful for both IGP and BGP. We focus on SITN as it is more general and less troublesome than metric-increment, but similar considerations apply to the metric-increment technique.

A. Any BGP anomaly can occur

Routing anomalies encompass two types of anomalies: *signaling* and *dissemination* anomalies. Signaling anomalies prevent a BGP network to settle to a stable state, forcing routers to continuously change their best route in a so-called *routing oscillation*. Dissemination anomalies consist in incorrect propagation of iBGP routes. Due to space constraints, we only focus on signaling anomalies, and we refer the reader to [26] for dissemination ones.

Consider the Evil-Twin gadget depicted in Fig 4 where the links (r_A, e_X), (r_B, e_3) and (r_B, e_4) are reweighted. In particular, the gadget contains two potentially oscillating structure known as Bad-Gadget [24]. Intuitively, a Bad-Gadget consists of three routers, called *pivot vertices*, which prefer the path provided by their clockwise neighbor to a more direct path to the destination. We refer to paths from one pivot vertex to another as *rim paths*, and to direct paths from each pivot vertex to the destination as *spoke paths*. In Fig. 4, a first Bad-Gadget Π exists between routers r_1, r_2 and r_3 for prefix p_1. Spoke paths in Π are $\vec{\mathcal{Q}} = ((r_1\ e_1)\ (r_2\ e_2)\ (r_3\ e_3))$, and rim paths are $\vec{\mathcal{R}} = ((r_1\ r_2)\ (r_2\ r_3)\ (r_3\ r_1))$. A second Bad-Gadget Π' concerns routers r_2, r_3 and r_4 for prefix p_2. Spoke paths are $\vec{\mathcal{Q}}' = ((r_2\ r_A\ e_x)\ (r_3\ r_B\ e_1)\ (r_4\ e_4))$, and rim paths are $\vec{\mathcal{R}}' = ((r_2\ r_4)\ (r_3\ r_2)\ (r_4\ r_3))$.

Observe that both the initial and the final configurations are oscillation-free. Indeed, in the initial configuration, r_2 steadily selects the routes announced by e_x for both p_1 and

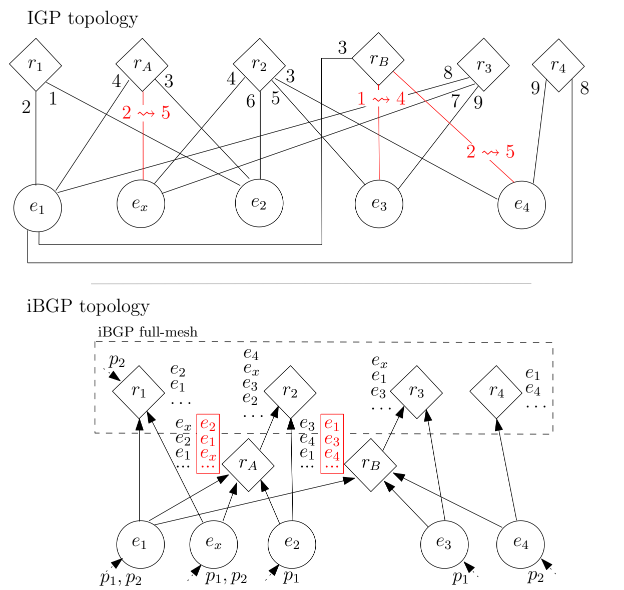

Fig. 4. Evil-Twin gadget: IGP reconfigurations can cause unavoidable BGP routing oscillations.

p_2, since it receives those routes from r_A. Thus, the spoke path $(r_2\ e_2)$ is never selected by r_2, preventing Π from oscillating. Symmetrically, r_B and r_3 are guaranteed to select the routes from e_x for p_1 and p_2, which prevents Π' from oscillating. In the final configuration, r_A is guaranteed to select the routes from e_2, path $(r_2\ r_A\ e_x)$ is never available at r_2. The absence of such a spoke path prevents Π' from oscillating. Symmetrically, r_B prefers e_1 to e_3, preventing Π from oscillating since the spoke path $(r_3\ r_B\ e_3)$ is never available at r_3.

During the reconfiguration process, however, a permanent oscillation is created in an intermediate configuration. Indeed, one of the following two cases applies.

1) *r_A is reconfigured before r_B.* Consider prefix p_1. After the reconfiguration of r_A, r_A starts selecting the route from e_2, and propagating that route to r_2. In this case, nothing prevents Π from permanently oscillating. Such an oscillation is interrupted only when r_B is migrated.
2) *r_B is migrated before r_A.* Consider prefix p_2. After the reconfiguration of r_B, r_B starts selecting the route from e_1, and propagating that route to r_3. Thus, nothing prevents Π' from permanently oscillating. Such an oscillation is interrupted only when r_A is migrated.

Similar examples of unavoidable route oscillations apply to other IGP reconfiguration scenarios (e.g., introducing an IGP hierarchy) [26].

B. Anomaly-free per-destination orderings do not always exist

In SITN reconfiguration, there always exist a *per-destination* ordering that guarantees the absence of forwarding loop towards any IGP destination. Unfortunately, this property does not hold anymore when BGP destinations are also considered. As an example, consider the Horizontal gadget illustrated

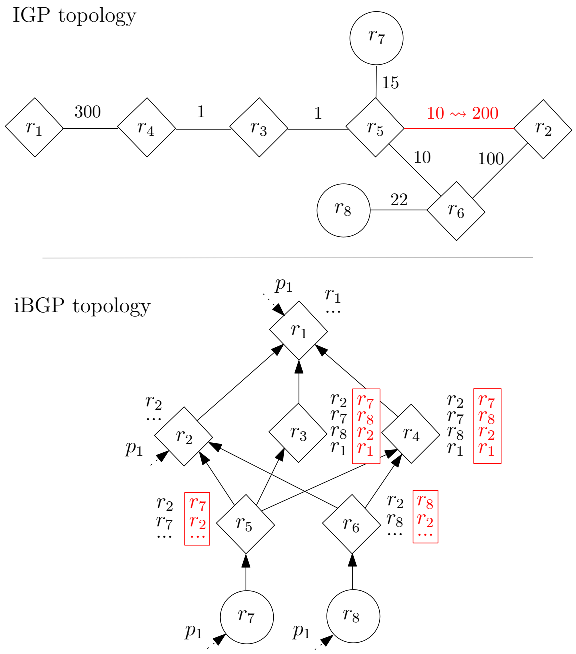

Fig. 5. HORIZONTAL GADGET. A per-destination ordering that is graceful for both IGP and BGP may not exist.

in Fig. 5, where the link (r_5, r_2) is reweighted from 10 to 200 and where the considered destination is r_2. Recall that in a SITN per-destination ordering, at each step, each router is reconfigured to start using the final forwarding path for the considered destination.

First, observe that the initial and the final configurations are loop-free. In the initial configuration, r_5 and r_6 receive and steadily select the route from r_2, while r_3 and r_4 only receive the route from r_1 and thus select it. In the final configuration, both r_5 and r_6 prefer the route propagated by their respective client r_7 and r_8, r_3 and r_4 also select the route from r_7 because of egress point preferences.

Consider now the reconfiguration process. To avoid an IGP-induced forwarding loop towards r_2, r_6 must be migrated before r_5. Indeed, r_5 forwards packets to r_6 in the initial configuration while the opposite holds in the final one. However, if r_6 is indeed migrated before r_5, then r_6 starts preferring the route R to p_1 announced by r_8 and sends it to r_4 which also selects it. Due to iBGP propagation rules, R is not propagated to r_3, which keeps selecting the route from r_1 as it is the only route r_3 receives. As a consequence, a forwarding loop occurs between r_3 and r_4. Indeed, r_4 forwards packets to r_3 to reach r_8 and r_3 bounces back packets to r_4 to reach r_1. The loop will last until when r_5 is reconfigured, allowing both r_3 and r_4 to both select the route from r_7.

V. Revisiting the complexity of IGP reconfigurations

It is known that reconfiguring an IGP protocol while guaranteeing the absence of forwarding loops is a hard problem in the general case [16]. In this section, we study the problem of performing an IGP reconfiguration avoiding undesired side effects induced by the interaction between BGP and IGP. More precisely, we focus on the following problem.

Problem 1 (Avoid Oscillation Problem - AOP)**.** *Given a BGP topology and two IGP topologies, decide if any IGP reconfiguration guarantees no BGP oscillations in all the intermediate configurations.*

We show that AOP is $\mathcal{NP}$-hard. This implies that it is computationally hard to decide if an IGP reconfiguration exists which is anomaly-free for both IGP and BGP. Even worse, since our proof can be adapted to dissemination and forwarding issues, deciding if IGP reconfigurations raise any specific type of BGP anomalies is also computationally hard.

Our proof consists of two parts. In the first part, we show that specific IGP reconfigurations can induce the change of the most preferred egress point on some iBGP routers. In the second part, we show that deciding if such changes can lead to BGP oscillations during the reconfiguration is $\mathcal{NP}$-hard.

A. IGP reconfigurations can cause BGP preference changes

Let $\mathcal{E}$ be the set of egress points of a given iBGP network. Let $\lambda_i^r(e)$ ($\lambda_f^r(e)$) be the position of egress point e in the initial (final) preference list of router r, where the most preferred egress point has position 1.

We now describe an IGP reconfiguration problem in which, at each step, a single BGP router swaps the positions of the two most preferred egress points. Namely, the IGP reconfiguration has three properties:

1) the initial (final) IGP topology is consistent with the initial (final) egress point preferences;
2) at each reconfiguration step, a single router r changes its preferences from λ_i^r to λ_f^r. Any other router $r' \neq r$ is not affected by the reconfiguration step; and
3) for some router r and egress points e_1 and e_2, $\lambda_i^r(e_1) < \lambda_i^r(e_2) \Leftrightarrow \lambda_f^r(e_2) < \lambda_f^r(e_1)$ if e_1 and e_2 are the two most preferred egress points of r, and $\lambda_i^r(e_1) < \lambda_i^r(e_2) \Leftrightarrow \lambda_f^r(e_1) < \lambda_f^r(e_2)$ otherwise. All the other routers have the same egress point preferences in the initial and final configurations.

We define the initial and final IGP topologies as follows. In both topologies, we have a link (r, e) between any router $r \notin \mathcal{E}$ and any egress point $e \in \mathcal{E}$. The weight of link (r, e) in the initial configuration is $w_i(r, e) = \lambda_i^r(e) + 3|\mathcal{E}|$. In the final configuration, $w_f(r, e) = 1 + 2|\mathcal{E}|$ if $\lambda_f^r(e) = 1$, and $w_f(r, e) = w_i(r, e)$ otherwise. This weight assignment directly ensures Property 3.

Also, such IGP topologies ensure that the shortest path between any router r and any egress point e is $(r\ e)$ in any intermediate configuration (including the initial and the final ones). Indeed, consider any path $P \neq (r\ e)$ between r and e. By definition, P must contain at least two links, hence its weight in any configuration i is $w_i(P) \geq 2 + 4|\mathcal{E}|$. Thus, $w_f(r, e) \leq w_i(r, e) \leq 4|\mathcal{E}| < 2 + 4|\mathcal{E}| \leq w_i(P)$, which also ensures Property 1.

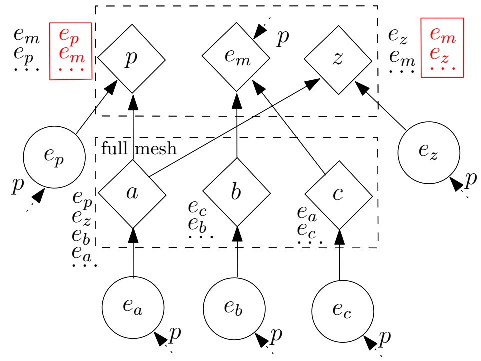

Fig. 6. Basic structure for our reduction.

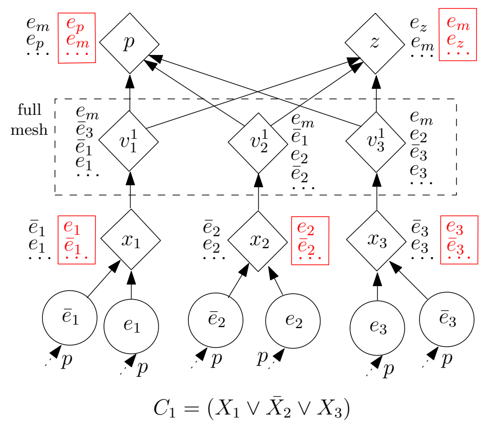

Fig. 7. Example of the translation of a 3-SAT clause.

Finally Property 2 holds since there is a one to one mapping between each edge and one shortest path, hence changing the weight of an edge affects the preferences of a single router.

B. AOP is $\mathcal{NP}$-hard

To prove that AOP is $\mathcal{NP}$-hard, we now reduce the 3-SAT problem [27] to AOP. Fig. 6 and 7 depict the reduction from a boolean formula F to a reconfiguration instance $B(F)$. Observe that $B(F)$ can be the result of an IGP reconfiguration, as described in the previous section.

The base BGP topology used in our reduction is represented in Fig. 6. Observe that a BAD-GADGET [28] Π' exists among a, b, and c. However, a's preferences are such that Π' is prevented from oscillating whenever a receives a route from e_p or e_z. Thus, Π' cannot oscillate in the initial nor in the final configuration. However, if z is reconfigured and p is not reconfigured yet, then a will not receive the routes to neither e_z nor e_p, and Π' will oscillate indefinitely. The presence of Π' hence forces any oscillation-free ordering to be such that p is reconfigured before z, which we denote as $p < z$.

The remaining part of $B(F)$ depends on the boolean formula F provided as input in the 3-SAT problem. Refer to Fig. 7. For each variable X_i in F, with $i = 1, \ldots, n$, we add one *variable router* x_i and two egress points e_i and $\bar{e}_i$. Egress point preferences are such that each x_i prefers $\bar{e}_i$ in the initial configuration and e_i in the final one. For each clause C_i, we add a *clause gadget* consisting of three *literal routers* v^i_j, with $j = 1, 2, 3$, representing the three literals in the clause. Observe that, since routers p and z can always reach one of their two most preferred egress points, literal routers belonging to different clauses cannot exchange paths. This allows us to consider clause gadgets separately.

For each clause C_i, a BAD-GADGET Π_i might exist among routers v^i_j. Indeed, the following property holds.

Property 2. *For each clause C_i, Π_i only exists if the variable routers corresponding to positive literals use their initial preferences, while the variable routers corresponding to negative literals use their final preferences.*

Moreover, since all literal routers prefer e_m over any other egress point, Π_i is prevented from oscillating when p is using its initial configuration or z is using its final configuration.

Intuitively, assigning X_i =TRUE (FALSE, resp.) corresponds to reconfiguring x_i before (after, resp.) p.

We now prove that the reduction is correct.

Theorem 1. *F is satisfiable if and only if an oscillation-free ordering exists on $B(F)$.*

Proof: We prove the statement in two steps.

- If F is satisfiable, then let $\mathcal{M}$ be a boolean assignment which satisfies F, and let $\mathcal{T}$ ($\mathcal{F}$, resp.) be the set of the variables that are set to TRUE (FALSE, resp.) in $\mathcal{M}$. Consider the ordering where we first reconfigure the routers corresponding to variables in $\mathcal{T}$ (in arbitrary order), then p, then z, and then the routers corresponding to variables in $\mathcal{F}$ (in arbitrary order). We now show that such an ordering is oscillation-free. Since $p < z$, BAD-GADGET Π' in Fig. 6 is prevented from oscillating. Also, for any migration step s, one of the following two cases applies: i) if p is not reconfigured yet or z is already reconfigured, then either p or z selects a path from e_m, preventing all BAD-GADGETs Π_i from oscillating; ii) s is the step in which p is reconfigured and z is still not. Consider any clause C_i and let l be one of the literals that satisfies C_i in $\mathcal{M}$. By construction of the reconfiguration ordering, if $l = X_i$ then router x_i is already migrated at step s. Otherwise, $l = \bar{X}_i$ and router x_i has not yet been migrated. In both cases, no BAD-GADGET Π_i exists at step s, because of Property 2. The same argument can be applied to all the clauses, so no oscillation can occur at s. Hence, an oscillation-free ordering exists.
- If F is not satisfiable, assume by contradiction that an oscillation-free ordering exists. The presence of Π' implies $p < z$ in the ordering. Consider any clause C_i and the migration step s immediately after the migration of p. Since neither p nor z select the route from e_m preventing Π_i from oscillating and we assumed that the migration ordering is oscillation-free, we conclude that Π_i does not exist at step s. Therefore, by Property 2, there must exist a router x_k such that either i) x_k corresponds to literal X_k in C_i and x_k is already migrated; or ii) x_k corresponds to

literal $\bar{X}_k$ in C_i and x_k has not been migrated yet. In the first case, we have $x_k < p$ which maps to $X_k =$ TRUE. Otherwise, we have $p < x_k$ which maps to $X_k =$ FALSE. In both cases, we are able to assign a truth value to X_k that satisfies C_i. Since the same argument can be applied to all the clause gadgets, then we are able to build a boolean assignment that satisfies F, yielding a contradiction. ■

Observe that by replacing all the BAD-GADGETs in the reduction with gadgets that trigger a dissemination anomaly or a forwarding loop, we derive similar reductions. This implies that guaranteeing that an IGP migration is free from any kind of BGP anomaly is $\mathcal{NP}$-hard.

Further, observe that BAD-GADGET Π' is used just to force $p < z$. However, it is easy to force $p < z$ by means of an IGP constraint rather than on a BGP constraint (e.g., by adding an IGP destination for which $z < p$ creates an IGP loop). Hence, with a similar proof we can show that avoiding IGP anomalies *and* BGP anomalies during an IGP migration is $\mathcal{NP}$-hard.

VI. BGP-FRIENDLY IGP RECONFIGURATIONS

In this section, we investigate viable approaches to perform reconfigurations that are disruption-free for both IGP and BGP destinations. In particular, we prove that anomaly-free reconfigurations can be achieved provided that the initial and the final configurations are correct and respect some conditions. We first focus on SITN, then we discuss metric-increment and other approaches.

A first condition enabling graceful reconfigurations for both IGP and BGP consists in ensuring that the egress point preferences in the initial and final configurations are the same.

Theorem 2. *If each router has the same egress point preferences in the initial and in the final configurations, no IGP reconfiguration can trigger BGP anomalies.*

Proof: In SITN, reconfiguring a router cause it to directly switch from considering the initial IGP topology to the final one [16]. Hence, at each reconfiguration step, the egress point preferences at each router coincide either with those of the initial or the final configuration which are the same by hypothesis. Since the BGP topology does not change, a BGP anomaly at a reconfiguration step implies that the same anomaly occurs in both the initial and the final configurations, contradicting our assumption on their anomaly-freeness. ■

As Theorem 2 applies in few practical cases, we now develop less constraining conditions.

Interestingly, the two main sufficient conditions for routing correctness, i.e. the prefer-client condition [24] and the no-spurious-over condition [25], are robust to IGP reconfigurations. Indeed, if the initial and final configuration comply with the sufficient conditions, then no IGP reconfiguration can invalidate them.

The prefer-client condition [24] requires that each route reflector prefer routes from its clients over routes from its iBGP peers or route reflectors. It has been shown [24], [25] that prefer-client is a sufficient condition to guarantee the absence of both oscillations and dissemination problems. We now show that the prefer-client condition is robust to IGP reconfigurations. In a sense, this means that the prefer-client condition is so strong that it constrains the impact that IGP topology changes have on the BGP decision process.

Theorem 3. *If the initial and final configurations both satisfy the prefer-client condition, then no IGP reconfiguration can trigger BGP routing anomalies.*

Proof: At each reconfiguration step, each router relies on either the initial or the final IGP weights independently from the configuration of the other routers. As the iBGP configuration does not change, each router has the same set of clients throughout the reconfiguration. Hence, a violation of the prefer-client condition at any intermediate step would result in a violation of the prefer-client condition in either the initial or the final configuration. The statement follows by noting that the prefer-client condition guarantees the absence of BGP routing anomalies. ■

The theorem applies to cases in which both the initial and the final configurations enforce the prefer-client condition by conveniently set IGP weights. Also, if the prefer-client condition is enforced at the BGP level (e.g., as proposed in [29], [30]), then IGP and BGP are decoupled enough to guarantee no BGP oscillations during IGP reconfigurations.

The no-spurious-over condition [25] guarantees the absence of dissemination anomalies, and requires that only top-layer route reflectors have iBGP peering relationships, while every other pair of routers must have a client-reflector relationship. The following theorem holds.

Theorem 4. *If both the initial and the final configurations comply with the no-spurious-over condition, no IGP reconfiguration can trigger BGP dissemination anomalies.*

Proof: The statement follows by noting that no IGP reconfiguration adds nor removes any iBGP session, hence it cannot invalidate the no-spurious-over condition at any reconfiguration step. ■

Unfortunately, sufficient conditions for forwarding correctness (e.g., [24]) are less robust. Intuitively, this is because they impose strong congruence between the IGP and the iBGP topologies, hence changing IGP can lead to temporary violations. However, forwarding issues can be avoided by relying on packet encapsulation (e.g., using MPLS or IP tunnels). Intuitively, packet encapsulation breaks the dependency between IGP and BGP in the forwarding plane. Note that encapsulation mechanisms like MPLS are commonly deployed in many ISP networks.

Theorem 5. *If packet encapsulation is used network-wide, no IGP reconfiguration can trigger BGP forwarding anomalies.*

Proof: If packet encapsulation is deployed, then each packet from any source router r to any BGP destination is guaranteed to reach the egress point e that r selects in BGP. Because of the BGP decision process, e will forward the packet outside the network (provided that eBGP routes are stable), hence the statement. ■

With respect to SITN, the metric-increment case is harder to tackle as it does not comply with the SITN Locality Property (see Property 1), i.e., it does not guarantee that any IGP change will only have a local effect. During the IGP reconfiguration, some routers can therefore have egress points preferences that do not reflect neither the initial nor the final ones. Thus, the prefer-client condition can be violated in some intermediate configurations if it is enforced through IGP weights. In contrast, Theorems 4 and 5 continue to hold. Observe that, besides avoiding forwarding anomalies, encapsulation mechanisms mitigate the impact of routing anomalies, since packets are guaranteed to be delivered outside of the AS even during routing oscillations.

A cleaner way to solve the reconfiguration problem would be to decouple BGP from the IGP. Recently, research proposals have proposed to loosen the interaction between IGP and BGP by decoupling BGP route selection and route propagation (as in an iBGP full-mesh) [31], [32]. While such a decoupling prevents BGP routing anomalies, it does not prevent forwarding anomalies, as testified by cases in which forwarding loops can arise even with an iBGP full-mesh (see Section III). Other research proposals propose to delegate both BGP route selection and propagation to a centralized component [33]. Whether centralized approaches enable graceful reconfigurations that are also practical (fast, reliable, and able to deal with failures and external routing changes) is an open problem.

In [23], Alimi *et al.* proposed an improved version of the SITN approach in which multiple configurations are run simultaneously on routers in an isolated way. By replicating both the IGP and the BGP configurations, this technique seems promising to achieve graceful reconfigurations. Unfortunately, it is not yet supported by current router implementations.

VII. Conclusions

In this paper, we highlighted the importance of considering the dependency between network protocols even for problems that seem to be restricted to a single protocol. In particular, we showed that state of the art IGP reconfiguration techniques should be revisited in the presence of BGP. Indeed, such techniques can create any type of BGP routing and forwarding anomalies even when a few changes are made.

In our opinion, this paper has the potential to spur new research effort regarding graceful network operations. As a fundamental step, we already discovered some sufficient conditions which makes BGP correctness robust to IGP reconfigurations. Interestingly, these conditions relate to static configuration allowing network operators to focus their attention on the initial and the final configurations. In the future, we plan to extend our study of the impact of IGP operations to other protocols like multicast protocols.

References

[1] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271, Jan. 2006.

[2] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. INFOCOM*, 2000.

[3] ——, "Optimizing OSPF/IS-IS weights in a changing world," *IEEE JSAC*, vol. 20, no. 4, pp. 756 –767, May 2002.

[4] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Comm. Mag.*, pp. 118–124, 2002.

[5] C. Filsfils, A. Maghbouleh, and P. Lucente, "Best Practices in Network Planning and Traffic Engineering," NANOG52, 2011.

[6] M. Horneffer, "IGP Tuning in an MPLS Network," NANOG33, 2005.

[7] R. Teixeira and J. Rexford, "Managing Routing Disruptions in Internet Service Provider Networks," *IEEE Comm. Mag.*, vol. 44, no. 3, pp. 160 – 165, 2006.

[8] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of Failures in an IP Backbone," in *Proc. INFOCOM*, 2004.

[9] V. Gill and M. Jon, "AOL Backbone OSPF-ISIS Migration," NANOG29, 2003.

[10] G. Herrero and J. van der Ven, *Network Mergers and Migrations: Junos Design and Implementation.* Wiley, 2010.

[11] "Results of the GEANT OSPF to ISIS Migration," GEANT IPv6 Task Force Meeting, 2003.

[12] S. Raza, Y. Zhu, and C.-N. Chuah, "Graceful Network Operations," in *Proc. INFOCOM*, 2009.

[13] P. Francois and O. Bonaventure, "Avoiding transient loops during the convergence of link-state routing protocols," *Trans. on Netw.*, vol. 15, no. 6, pp. 1280–1932, 2007.

[14] J. Fu, P. Sjodin, and G. Karlsson, "Loop-Free Updates of Forwarding Tables," *Trans. on Netw. and Serv. Man.*, vol. 5, no. 1, pp. 22–35, 2008.

[15] L. Shi, J. Fu, and X. Fu, "Loop-Free Forwarding Table Updates with Minimal Link Overflow," in *Proc. ICC*, 2009.

[16] L. Vanbever, S. Vissicchio, C. Pelsser, P. Francois, and O. Bonaventure, "Lossless Migrations of Link-State IGPs," *IEEE/ACM Trans. Net.*, 2012, to appear. Available at http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6179350.

[17] P. Francois, P.-A. Coste, B. Decraene, and O. Bonaventure, "Avoiding disruptions during maintenance operations on BGP sessions," *Trans. on Netw. and Serv. Man.*, vol. 4, no. 3, pp. 1–11, 2007.

[18] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed Internet routing convergence," *IEEE/ACM Trans. Netw.*, vol. 9, pp. 293–306, 2001.

[19] R. Teixeira, A. Shaikh, T. Griffin, and J. Rexford, "Impact of Hot-Potato Routing Changes in IP Networks," *IEEE/ACM Trans. Net.*, vol. 16, no. 6, pp. 1295 –1307, dec. 2008.

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

[21] J. Qiu, "SimBGP: Python Event-driven BGP simulator," http://www.bgpvista.com/simbgp.php.

[22] P. Francois, M. Shand, and O. Bonaventure, "Disruption-free topology reconfiguration in OSPF Networks," in *Proc. INFOCOM*, 2007.

[23] R. Alimi, Y. Wang, and Y. R. Yang, "Shadow configuration as a network management primitive," in *Proc. SIGCOMM*, 2008.

[24] T. Griffin and G. Wilfong, "On the Correctness of IBGP Configuration," in *Proc. SIGCOMM*, 2002.

[25] S. Vissicchio, L. Cittadini, L. Vanbever, and O. Bonaventure, "iBGP Deceptions: More Sessions, Fewer Routes," in *Proc. INFOCOM*, 2012.

[26] L. Vanbever, "Handling network reconfiguration," Ph.D. dissertation, Université catholique de Louvain, June 2012.

[27] C. Papadimitriou, *Computational complexity*. Addison-Wesley, 1994.

[28] T. G. Griffin and G. T. Wilfong, "An analysis of BGP convergence properties," in *Proc. SIGCOMM*, 1999.

[29] R. Musunuri and J. Cobb, "A complete solution for iBGP stability," in *Proc. ICC*, 2004.

[30] L. Cittadini, G. Di Battista, and S. Vissicchio, "Doing don'ts: Modifying BGP attributes within an autonomous system," in *Proc. NOMS*, 2010.

[31] I. Oprescu, M. Meulle, S. Uhlig, C. Pelsser, O. Maennel, and P. Owezarski, "oBGP: an Overlay for a Scalable iBGP Control Plane," in *Proc. IFIP Networking*, 2011.

[32] R. Chen, A. Shaikh, J. Wang, and P. Francis, "Address-based Route Reflection," in *Proc. CoNEXT*, 2011.

[33] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and implementation of a routing control platform," in *Proc. NSDI*, 2005.