The Art of Detecting Forwarding Detours

Julián M. Del Fiore*, Valerio Persico†, Pascal Mérindol*, Cristel Pelsser* and Antonio Pescapè†
*ICube, University of Strasbourg, France, †University of Napoli Federico II, Italy

***Abstract*—The full Internet feed, reaching ∼867K prefixes as of March 2021, has been growing at ≈50K prefixes/year over the last 10 years. To counterbalance this sustained increase, Autonomous Systems (ASes) may filter prefixes, perform prefix aggregation and use default routes. Despite being effective, such workarounds may result in *routing inconsistencies*, i.e., in routers along a forwarding route mapping the same IP addresses to different IP prefixes. In turn, the exit AS border routers associated with these distinct prefixes may potentially differ. For some prefixes, *forwarding detours* (FDs) may occur, i.e., traffic may deviate from best IGP paths. In this work we investigate the phenomenon of FDs and derive a methodology to detect them. In particular, our tool is able to pinpoint cases where multiple prefixes are subject to FDs. We run measurements from 100 vantage points of the NLNOG RING monitoring infrastructure and find FDs in 25 out of 54 ASes. We see that FDs are heterogeneous, i.e., the number of prefixes and AS border routers in between which we detect FDs strongly depend on the studied AS. Finally, we discover a remarkable binary effect such that either all transit traffic traversing between two border routers of an AS detours, or none does.**

***Index Terms*—Forwarding Information Base; Forwarding Detours; Load Balancing; Traffic Engineering; Network management; Routing Inconsistencies; Scalability**

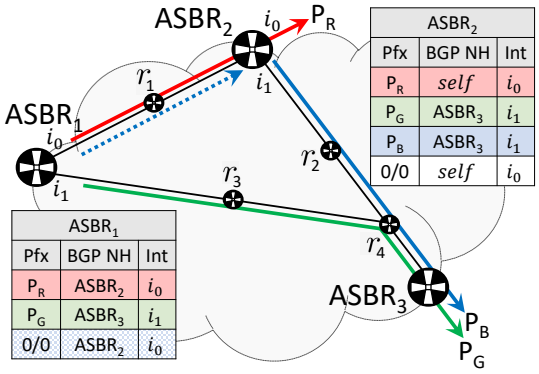

Fig. 1: From routing inconsistencies to FDs. The default route of $ASBR_1$, that has a partial-FIB, leads to a routing inconsistency between this router and $ASBR_2$ for the blue prefix P_B. Since $ASBR_2$ redirects traffic concerning P_B towards $ASBR_3$, the resulting route does not match the best IGP from $ASBR_1$ to $ASBR_3$. Hence, we say that P_B is subject to FDs. Moreover, as P_G is not subject to FDs, a multi-path routing pattern appears between $ASBR_1$ and $ASBR_3$.

I. Introduction

Over the last 8 years, the full Internet feed has doubled in size, reaching ∼867K prefixes as of March 2021 [1]. The sustained increase in the number of prefixes advertised on the Border Gateway Protocol (BGP) has led Autonomous Systems (ASes) to exchange more update messages [2]–[4], and to suffer from scalability issues. Indeed, considering the current trend, maintaining a full Forwarding Information Base (FIB) may be challenging, specially for ASes incapable of upgrading their network devices regularly [5], [6], [8].

In this context, networks operators have found alternatives to endure with legacy routers unable to maintain a complete FIB in memory. For example, in a BGP-free core, tunneling techniques reduce the size of the FIB on core routers [9]. In addition, partial iBGP dissemination relying on route-reflector hierarchies may also boost scalability [10]. This technique allows routers to maintain less BGP peers and, in some rare cases, may even prevent the full redistribution of BGP prefixes within the AS [11]. In addition, memory-constrained routers may aggregate routes to limit the number of FIB entries [12]. Other type of workarounds consist in storing a partial-FIB [13], [14], and redirecting traffic via default routes towards more capable routers (e.g. having a full-FIB). Some network operators even apply this technique on switches with IP capabilities [15].

While the aforementioned workarounds may look effective at first glance, ASes relying on them may suffer from *routing inconsistencies*. In such cases, inside those ASes, routers along a route may map the same destination IP address to distinct (most specific) prefixes. Since these prefixes may be associated to discrepant AS border routers (ASBRs), *forwarding detours* (FDs) may occur, i.e., for some prefixes, traffic may not traverse the network through best IGP paths. Due to this, we refer to such prefixes as *prefixes subject to FDs*. In general, the simultaneous existence of prefixes subject and not subject to FDs generates *multi-path routing* patterns. However, contrary to hot-potato routing, FDs increase the IGP distance required to traverse an AS and may generate loops [16], arguably resulting in waste of resource utilization inside the network. Attempting to suppress FDs, network operators may implement tunneling techniques [17], with Label Distribution Protocol (LDP) [18] or Segment Routing (SR) [19]. However, these mechanisms only allow to avoid FDs within each tunnel/segment (for BGP-free core routers in particular) but may fail to do so between endpoints of an AS.

Fig. 1 illustrates how routing inconsistencies may produce FDs. In this example, $ASBR_1$ has a partial-FIB and, relying on its default route, forwards traffic destined to prefix P_B towards $ASBR_2$ (blue dotted line). There exists a routing inconsistency for P_B since $ASBR_2$ disagrees with $ASBR_1$ regarding the BGP exit point; indeed, rather than itself, $ASBR_2$ considers $ASBR_3$ as the best BGP next-hop for P_B. Hence, $ASBR_2$ redirects traffic targeting P_B towards $ASBR_3$. While the best IGP path from $ASBR_1$ to $ASBR_3$ is $(ASBR_1, r_3, r_4, ASBR_3)$, and is

used for P_G, the forwarding route for P_B differs, being $(ASBR_1, r_1, ASBR_2, r_2, r_4, ASBR_3)$. Consequently, P_B is subject to FDs, but P_G is not, thus generating a multi-path routing pattern between $ASBR_1$ and $ASBR_3$. Moreover, even if tunnels mechanisms were used between ASBRs, e.g. $ASBR_1$ and $ASBR_2$, after exiting the tunnel, traffic concerning P_B would still be redirected towards $ASBR_3$.

In this study we take a close look at the phenomenon of FDs. As discussed before, FDs may result as a side effect of scalability workarounds. However, misconfigurations [20] or bugs in router's software such as BGP zombies [21] may also create routing inconsistencies leading to FDs. Consequently, network operators may ignore FDs occur on their AS, and provide degraded performance to customer ASes. Prior work has focused on detecting routers relying on backup default routes [22], or identified them as a possible cause of BGP lies [23]. However, no study has focused on the impact of such techniques on the forwarding inside ASes. In that sense, to the best of our knowledge, we are the first to tackle the problem of detecting FDs, indistinctly of the underlying causes generating them. Our methodology allows network operators to check the sanity of the routing inside their own network, and customer ASes to check whether their provider ASes suffer from FDs. The detection of FDs is the first step towards the ultimate goal of systematically quantifying the effect of FDs on traffic. In a nutshell, we make the following contributions:

- We discuss the difficulty of detecting FDs, a problem without recipe, in Sec. II. This is particularly challenging when ASes deploy load balancing and traffic engineering techniques, that also produce multi-path routing patterns.
- We discuss the different load balancing flavors that exist in Sec.III. In particular, we describe one not yet presented in the literature that, different from others, as long as the destination prefixes remain constant, the same routes are consistently used. This concept also holds for FDs and traffic engineering. We refer to these three as prefix-based mechanisms.
- We design a novel algorithm, our main ingredient, able to detect prefix-based mechanisms in Sec. IV. Our methodology consists in studying the correlation between measured prefixes and the set of forwarding routes that are revealed when tracing them.
- We propose an FD-detector, the final dish, in Sec. V. Our solution adds a last spice to the previous algorithm: it applies a verdict allowing to discriminate FDs from the other prefix-based mechanisms. For this, we focus on cases where FDs affect numerous external prefixes. Our tool relies only on IP-to-AS mapping data and data-plane information collected with `traceroute`.
- We analyze the FD-phenomenon in the wild in Sec. VI, running our FD-detector from 100 nodes of the NLNOG RING monitoring infrastructure, and find FDs in 25 out of 54 ASes. We validated the behavior of the FD-detector with emulations and on a network where we have ground truth.
- We release the dataset we collected, the emulations setups and our code to foster replicability and reproducibility[1].

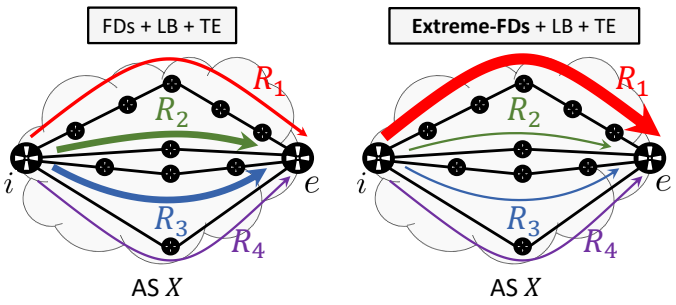

Fig. 2: Forwarding patterns when FDs ($\mathcal{R}_X^{FD}(i,e) = \{R_1\}$), LB ($\mathcal{R}_X^{LB}(i,e) = \{R_2, R_3\}$) and TE ($\mathcal{R}_X^{TE}(i,e) = \{R_4\}$) coexist. The size of every arrow is proportional to the number of prefixes for which each route is used. While the forwarding pattern inside the AS on the left case undergoes no major change due to FDs, on the right case it is largely modified by the occurence of extreme-FDs, i.e., FDs for most prefixes.

In addition, we present related work in Sec. VII, discuss the robustness of the FD-detector we implemented in Sec.VIII, and draw final remarks in Sec. IX.

II. Challenge: Finding a Recipe

In this section we show why detecting FDs, a problem for which there is currently no recipe, is challenging. In particular, this task is not trivial since load balancing (LB) and traffic engineering (TE) techniques also produce multi-path routing patterns.

In practice, observing a multi-path routing pattern between any two routers i and e of an AS X is not enough to declare the occurrence of FDs: the use of LB and TE can also produce the same effect. With LB methods such as equal-cost multi-path (ECMP), the strict notion of best IGP path is generalized to a set of paths $\mathcal{R}_X^{LB}(i,e)$ sharing the same IGP distance. The purpose of ECMP is to evenly spread the load across such set of best parallel IGP paths. On the other hand, TE allows to create sets of constrained paths $\mathcal{R}_X^{TE}(i,e)$ that are commonly used for specific usages regarding a limited number of external prefixes, but not for best-effort traffic. Finally, Fig. 1 illustrates a simple scenario with a unique detouring route, however, between i and e, a set of detouring routes $\mathcal{R}_X^{FD}(i,e)$ might exist if prefixes are subject to FDs due to different underlying causes. Considering the left side of Fig. 2, where $\mathcal{R}_X^{FD}(i,e) = \{R_1\}$, $\mathcal{R}_X^{LB}(i,e) = \{R_2, R_3\}$, $\mathcal{R}_X^{TE}(i,e) = \{R_4\}$, the question we aim to address is, by simply collecting routes with `traceroute`, how can we distinguish FDs?

A first attempt to solve this problem would be to assume that hop count is used as the IGP metric, compare routes by their length, and conclude for FDs when routes of different lengths are discovered between i and e. However, for other IGP metrics, such heuristic may lead to misclassify ECMP as FDs, e.g. R_3 in Fig. 2, and vice-versa. On the other hand, TE routes are not restricted to be shortest paths between two endpoints.

[1] See https://github.com/julian10m/FD-detector.git and https://zenodo.org/record/4458140

Hence, this highlights that, to avoid both false positives and negatives in the detection of FDs, the designed method should be valid for any IGP metric and contemplate TE.

Another naive solution would be to assume that, inside an AS, transit traffic traverses exactly two ASBRs. Under this assumption, we could first learn the ASBRs launching traces, and then pinpoint FDs looking if three or more ASBRs of the same AS were traversed in any trace. For example, in Fig. 1, the blue path that detours traverses $ASBR_1$, $ASBR_2$ and $ASBR_3$. Though apparently effective, this technique only works in specific network topologies where ASBRs are never used as transit core routers. For example, if router r_3 in Fig. 1 was also used as ASBR for some prefixes, then prefix P_G would incorrectly look as subject to FDs. In short, this technique cannot be used since, in practice, it is likely that traces will usually traverse multiple ASBRs of the same AS, even in the absence of FDs.

To correctly detect FDs, rather than computing misleading metrics for each route and/or comparing them one at a time, we propose to analyze the *forwarding pattern* for (i, e) in AS X. In other words, we propose to closely study which routes of X, leading from i to e, are used depending on the targeted prefixes. For this, multiple traces traversing i and e need to be collected for as many prefixes and destinations as possible, and the distribution of prefixes per set of routes analyzed. On the left case of Fig 2, few prefixes are subject to FDs, and thus differentiating them from TE and LB might not be simple. The main bulk of prefixes evenly spreads over $\mathcal{R}_X^{LB}(i, e)$, and only a reduced number of prefixes are forwarded across $\mathcal{R}_X^{TE}(i, e)$. In contrast, in the cases involving *extreme-FDs*, i.e., scenarios where most prefixes are subject to FDs, we expect to see a remarkably distinct forwarding pattern in which a large fraction of external prefixes is aggregated on $\mathcal{R}_X^{FD}(i, e)$. This is exemplified on the right side of Fig. 2, where traffic traversing from i to e is aggregated on $\mathcal{R}_X^{FD}(i, e) = \{R_1\}$ for multiple prefixes. Note how this case notoriously contrasts with that on the left side of Fig. 2.

The proposal of studying forwarding patterns focusing on the detection of extreme-FDs, though more promising than the previous heuristics, still does not explain how to actually identify the existence of $\mathcal{R}_X^{FD}(i, e)$, i.e., how we can conclude that the routes on which most prefixes are aggregated do not represent LB or TE routes. In addition, the aforementioned analysis does not model the effect of different LB flavors. This is particularly important since, actually, there exists an LB flavor that defines flows at the prefix granularity. As such, this LB flavor generates a forwarding pattern in which the route in use may vary depending on the prefix that is considered. This is similar for FDs and TE. Moreover, LB and FDs can interfere with each other, since ECMP can also apply on detouring routes. Overall, to understand how FDs can be detected, having a clear understanding of the distinct forwarding patterns that LB, TE and FDs produce is imperative.

III. Load Balancing and Forwarding Patterns

In this section we study the current LB techniques, and their impact on data plane information collected with `traceroute`. In Sec. III-A we present the different LB flavors that exist. On the other hand, in Sec. III-B we discuss the distinct forwarding patterns that these LB flavors produce, and their similarity with that generated by FDs and TE.

A. Load balancing in a nutshell

With the use of LB techniques, for any two routers i and e inside an AS X, multiple LB routes connecting them, denoted $\mathcal{R}_X^{LB}(i, e)$, might exist. This LB set results from the presence of load balancers, i.e., routers that may use different next-hops towards the same destination IP address. To balance packets across next-hops, these LB routers take into account either (some) packet header fields, or none at all [24], [25].

The simplest mode of LB, namely *per-packet* LB [26], [27], assigns packets to next-hops blindly, in a round-robin fashion. Consequently, with this approach, packets exchanged in a TCP connection are subject to reordering, a fact known to degrade the performance of TCP [28]–[30]. Moreover, faced to this LB flavor, any traceroute implementation may fail to reveal some links, and even infer false ones [24], [25]. Fortunately, per-packet LB is rarely found in practice [31], [32].

Other more sophisticated LB methods, which we call hash-based, decide next-hops relying on the use of a hash function, rather than blindly. More precisely, load balancers apply a hash on packet header values, and use the outcome of such computation to choose one among the available next-hops. As a consequence, in contrast with per-packet LB, packets belonging to the same TCP connection are always forwarded to the same next-hop. Due to this, such packets are said to belong to the same flow, and to have a similar flow-identifier (or simply flow-ID). Depending on the fields used to compute the hash, hash-based LB methods have historically been subdivided in two types: *per-destination* LB, or in short *per-dest* LB [26], and *per-flow* LB [26], [27], [33]. While the source and destination IP addresses are used as input in per-dest LB, the source and destination transport ports are additionally taken into consideration in per-flow LB.

Previous work has mainly focused on per-dest and per-flow LB, that are the two most widespread LB flavors [34], however, there exists a third hash-based LB flavor that has been systematically omitted in the literature, known as *per-prefix* LB [33], [35]. With per-prefix LB, the hash function is evaluated on the most specific prefix associated with the destination IP address of each packet. Note how this LB flavor contrasts with the other two hash-based LB methods, where the destination IP address is hashed at once. Due to this, we say that per-prefix LB is a *coarse-grained* LB type, while per-dest and per-flow LB are *fine-grained* LB types. We often indicate fine-grained LB types as per-dest/flow LB.

Finally, to mimic distinct hashing functions, load balancers also rely on additional parameters, such as the router-id or a configured seed value, to determine next-hops. Note that these complementary inputs neither depend nor are extracted from the packets being forwarded. This allows to avoid polarization effects, that prevent the use of redundant routes [36], but has also been observed to produce next-hops re-mapping events often mistakenly attributed to routing changes [32].

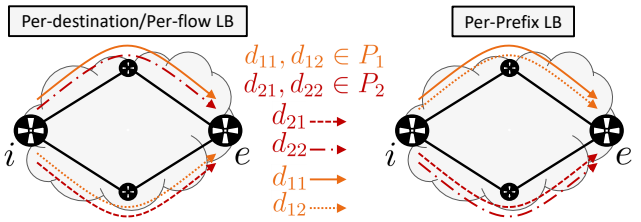

Fig. 3: Forwarding patterns for per-dest, per-flow and per-prefix LB. While for per-dest/flow LB all routes of $\mathcal{R}_X^{LB}(i,e)$ are used for both prefixes, for per-prefix LB, traffic targeting P_1 and P_2 flows through different routes. Indeed, for the latter, different routes can only be revealed tracing different prefixes, but the opposite is not true. This also holds for FDs and TE.

B. Forwarding patterns: LB might resemble FDs and TE

We are interested in the forwarding patterns that the different hash-based LB flavors produce inside an AS, in order to be able to discriminate them from FDs[2].

For both per-dest and per-flow LB, the route that each traceroute reveals may vary as the destination changes. This possible variation of route also applies even when the destinations traced belong to the same prefix. This property is illustrated on the left side of Fig. 3, where per-dest/flow load balancer i uses its 2 available next-hops for traces targeting both P_1 and P_2. As a consequence, for fine-grained LB types, exploring one prefix is enough to reveal all routes of $\mathcal{R}_X^{LB}(i,e)$.

On the other hand, per-prefix LB discriminates packets on a prefix basis and thus, for each prefix, the same next-hop is consistently chosen. Hence, each route of $\mathcal{R}_X^{LB}(i,e)$ is used only to forward traffic destined to the specific set of prefixes for which the same next-hop is chosen. As an example, on the right side of Fig. 3, per-prefix load balancer i chooses different next-hops for prefixes P_1 and P_2, but always the same and unique one for traces belonging to the same prefix. Indeed, with coarse-grained LB types, there is no route variation for different destinations belonging to the same prefix.

From this analysis, we can derive a critical concept: the forwarding pattern of per-prefix LB is similar to that of FDs and TE. This occurs since the three of them are *prefix-based mechanisms*. Indeed, in the same vein as the route used in per-prefix LB may change or not depending on the prefix that is considered, so does the occurrence of FDs, and the use of constrained TE paths. Hence, we say that per-prefix LB, FDs and TE produce *prefix-based forwarding patterns*.

IV. The Main Ingredient: A Detector of Prefix-Based Forwarding Patterns

In this section we build a framework that investigates the forwarding pattern inside ASes, and determines whether they are prefix-based. To tackle this problem, we propose an analysis in four steps, referred to as exploration, prefix-grouping, multi-route discovery and merging phases, respectively. [3] The exploration phase collects traces and identifies *ASBR-couples* of each AS, i.e., the ingress-ASBR and egress-ASBR of an AS that are simultaneously traversed by a trace.[4] For these ASBR-couples, we determine their associated *internal routes*, i.e., the routes inside the AS that connect each couple. Then, the prefix-grouping phase looks for multi-path routing patterns across different ASes, i.e., whether depending on the traced prefix, the internal route revealed for an ASBR-couple varies. For each couple where such pattern is found, we continue the study with the multi-route discovery phase. This step extends the probing, aiming to reveal all internal routes that are used for each of the prefixes for which an ASBR-couple is observed. Finally, the merging phase discriminates between per-dest/flow LB and prefix-based mechanisms for each ASBR-couple. Next, we detail these steps relying on the following notation: R is used to denote a route, $\mathcal{R}$ a set of routes, and $\mathbb{R}$ a set of sets of routes. The same convention is used for prefixes, i.e., we use P, $\mathcal{P}$ and $\mathbb{P}$, respectively. We postpone the explanation of how this methodology can be turned into an FD-detector to Sec. V.

A. Exploration phase

This step collects ASBR-couples and internal routes across ASes. For this, we perform a lightweight traceroute campaign, launching traces for some random prefixes (e.g. /24 subnets). An IP-to-AS mapping tool is used to determine ASBR-couples, and the internal routes inside each AS. According to the prefixes that are probed, it could happen that few traces traverse some couples. To enlarge the set of routes that are gathered for each of them, we collect a special internal route, that we call the *direct internal route (DIR)*. The DIR of each ASBR-couple is obtained by tracing the egress-ASBR, and is the internal route that starts in the ingress-ASBR and finishes in the egress-ASBR. As we detail in Sec. V, the DIR has a key role in the detection of FDs, hence we discard those couples for which the DIR cannot be determined (see Sec. V-B).

As a last step, we annotate the prefixes for which each internal route was revealed, i.e., the /24 subnet (usual longest BGP prefix [37]) covering the destination IP of the trace from which the internal route was extracted. The only exception is the DIR, which we consider associated to a /32 prefix, e.g. for a couple (i,e), then $e/32$. In the left table of Fig. 4 we show the outcome of the exploration phase for a couple (i,e): tracing the prefixes of the left column $\{P_1, \dots P_7, e/32\}$, the routes on the right column $\{R_1, R_2, R_3, R_4\}$ are revealed.

B. Prefix-grouping phase

For the ASBR-couples that remain at this stage, we seek for a multi-path routing pattern by grouping the prefixes for which the same internal route was revealed. The outcome of the prefix-grouping phase for an ASBR-couple (i,e) is illustrated in the middle matrices of Fig. 4, for both prefix-based mechanisms and per-dest/flow LB. Indeed, the prefixes for which the same route is observed, e.g. $\mathcal{P}_1 = \{P_1, e/32\}$, $\mathcal{P}_2 = \{P_3, P_7\}$ are respectively associated with R_1 and R_2, etc. As highlighted on the figure, the prefix-grouping phase

[2]Recall that per-packet LB is rarely found in practice, see Sec. III-A.

[3]While in the following we pay special attention to the intuition and general objective behind each phase, the repository of our tool also includes a pseudo-code highlighting implementation details for the interested readers.

[4]To ease the reading, we often refer to ASBR-couples simply as *couples*.

may return the same result for per-dest/flow LB and prefix-based mechanisms. Thus, to be able to differentiate between both of them, further analysis is required.

Finally, note that for each ASBR-couple (i, e) of each AS X, two sets are stored: (i) a set of prefixes $\mathbb{P}_X(i, e)$ grouping the sets of prefixes for which the same internal route in X from i to e is observed; (ii) a set of corresponding internal routes $\mathcal{R}_X(i, e)$, one for each set of prefixes in $\mathbb{P}_X(i, e)$. At this stage, $\mathbb{P}_X(i, e) = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_r\}$ is a set of sets of prefixes, whereas $\mathcal{R}_X(i, e) = \{R_1, R_2, \ldots, R_r\}$ is a set of routes, such that $r = |\mathbb{P}_X(i, e)| = |\mathcal{R}_X(i, e)|$. In particular, for the couples where $r = 1$, no multi-path routing pattern is observed and, therefore, there is no need to continue exploring them. On the contrary, when $r > 1$, then $\mathbb{P}_X(i, e)$ and $\mathcal{R}_X(i, e)$ are transferred to the multi-route discovery phase. This is the case in Fig. 4, where $r = 4$.

C. Multi-route discovery phase

This block extends the probing for the ASBR-couples delivered from the prefix-grouping phase. Our aim is to determine all the internal routes associated with each set of prefixes for which traces traverse an ASBR-couple. In other words, for each ASBR-couple (i, e) in any AS X, for each $\mathcal{P}_j \in \mathbb{P}_X(i, e)$, we look whether routes inside AS X other than $R_j \in \mathcal{R}_X(i, e)$ can be revealed probing destinations in $\mathcal{P}_j$. For this, we replace each route R_j with a set of routes $\mathcal{R}_j$ where we keep track of all internal routes in AS X from i to e that are found probing $\mathcal{P}_j$. As a result, note that while r remains constant, $\mathcal{R}_X(i, e)$ becomes a set of sets of routes $\mathbb{R}_X(i, e)$, i.e. $\mathbb{R}_X(i, e) = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_r\}$. The unaltered set of prefixes $\mathbb{P}_X(i, e)$ and $\mathbb{R}_X(i, e)$ are then passed to the merging phase.

The right matrices of Fig. 4 show the result of the multi-route discovery phase run for the couple (i, e) with $\mathbb{P}_X(i, e) = \{\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3, \mathcal{P}_4\}$ and $\mathcal{R}_X(i, e) = \{R_1, R_2, R_3, R_4\}$ as delivered from the prefix-grouping phase. Contrary to what was observed in the previous step, and recalling the analysis in Sec. III-B, the outcome of the multi-route discovery phase is different for prefix-based mechanisms and per-dest/flow LB. For the first, each set of $\mathbb{R}_X(i, e)$ ends up containing a unique route, the one discovered in the exploration phase, i.e., $\forall j$, $\mathcal{R}_j = \{R_j\}$. Indeed, for prefix-based mechanisms, the route observed for any set of prefixes $\mathcal{P}_j$ remains constant indistinctly of the IP target inside $\mathcal{P}_j$ that is traced. On the other hand, for per-dest/flow LB, additional internal routes are discovered for each set of prefixes, e.g., $\mathcal{R}_1 = \{R_1, R_2, R_4\}$, $\mathcal{R}_2 = \{R_2, R_3, R_4\}$, etc. This happens because per-dest LB and per-flow LB are fined-grained LB types, meaning that the destination IP address is part of their flow-ID. Consequently, probing several IP addresses included in $\mathcal{P}_j$, it is likely that $\mathcal{R}_j$ will include more routes than just R_j. In an ideal case, for fine-grained LB types, it holds that for $\forall j \in \{1, 2, \ldots, r\}$, $\mathcal{R}_j = \mathcal{R}_X^{LB}(i, e)$, as what happens for $\mathcal{P}_3$ in Fig. 4.

D. Merging phase

For each ASBR-couple (i, e), this step analyzes $\mathbb{P}_X(i, e)$ and $\mathbb{R}_X(i, e)$ to determine whether the forwarding pattern observed between i and e inside AS X corresponds to that of per-dest/flow LB or prefix-based mechanisms. During the multi-route discovery phase, while the sets composing $\mathbb{R}_X(i, e)$ do not change for prefix-based mechanisms, it is likely that they are enlarged and contain internal routes in common for fined-grained LB. Hence, we (always) proceed to convert $\mathbb{R}_X(i, e)$ into a partition, i.e., we repeatedly merge the intersecting sets of routes until no more overlaps exist among the merged sets. In this process, we also merge the subsets of $\mathbb{P}_X(i, e)$ accordingly. This operation results in $s \leq r$ sets composing $\mathbb{R}_X(i, e)$ and $\mathbb{P}_X(i, e)$.

The merging phase outputs different results for fine-grained LB flavors and prefix-based mechanisms, and thus allows to determine if a prefix-based forwarding pattern is observed

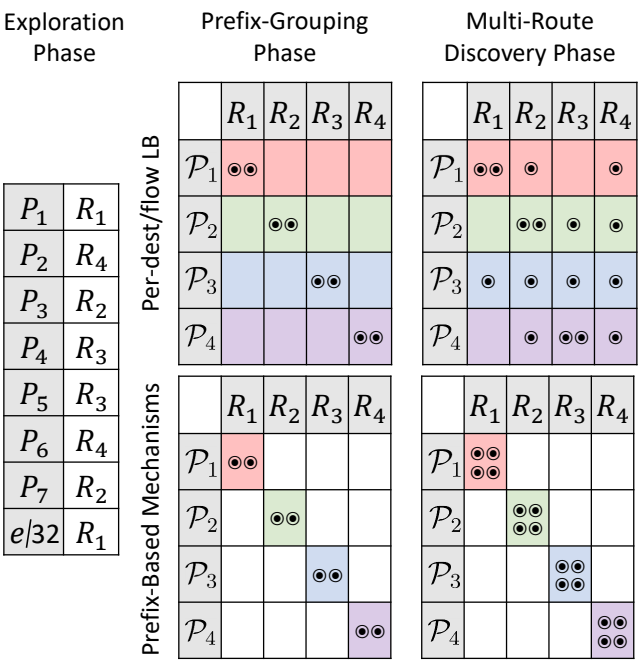

Fig. 4: Detecting the type of forwarding pattern for an ASBR-couple (i, e). While the colored cells represent the routes associated with each set of prefixes, the dots show those revealed while tracing. The exploration phase runs `traceroute` and reveals one internal route per measured prefix. The prefix-grouping phase then groups those prefixes for which the same route was revealed. At this stage, the result is the same for per-dest/flow LB and prefix-based mechanisms. The multi-route discovery phase extends the measurements to find the complete set of routes associated with each set of prefixes. For per-dest/flow LB we see that routes in common emerge across the different sets of prefixes. However this does not occur for prefix-based mechanisms. Ultimately, the merging phase will expose the nature of the forwarding pattern, merging all routes and prefixes into a unique set for fine-grained LB, but failing to do so for prefix-based mechanisms. Therefore, in the cases where more than one set remains at the final step, we can conclude that the forwarding pattern for (i, e) is prefix-based.

for an ASBR-couple (i, e) inside AS X. [5] For per-dest/flow LB, it holds that $s = 1$, such that $\mathbb{R}_X(i, e) = \{\mathcal{R}_X^{LB}(i, e)\}$ and all prefixes in $\mathbb{P}_X(i, e)$ are also grouped into a unique set. In the example of Fig. 4, all sets overlap[6], and thus the merging phase outputs $\mathbb{R}_X(i, e) = \{\{R_1, R_2, R_3, R_4\}\}$ and $\mathbb{P}_X(i, e) = \{\{P_1, \ldots, P_7, e/32\}\}$. On the other hand, for prefix-based mechanisms, since the sets do not overlap, as shown in the bottom-right matrix in Fig. 4, the composition of $\mathbb{P}_X(i, e)$ and $\mathbb{R}_X(i, e)$ does not change, thus it holds that $s = r > 1$, and $s = 4$ in this particular example. [7]

In the next section we show how our detector of prefix-based forwarding patterns can be refined, and turned into an FD-detector. Indeed, to allow the detection of FDs even when per-prefix LB and TE are jointly present, looking at the number of sets composing $\mathbb{P}_X(i, e)$ and $\mathbb{R}_X(i, e)$ is not enough. The size and content of their merged subsets need to be analyzed.

V. The Resulting Dish: An FD-Detector

In this section we present the FD-detector we designed, our final dish. In particular, Sec. V-A shows how the detector of prefix-based forwarding patterns can be turned into an FD-detector by adding a last spice: an FD-verdict seeking for a lonely DIR to infer whether extreme-FDs occur or not. On the other hand, Sec. V-B describes how we implemented our FD-detector based on current probing tools.

A. FD-verdict: the key spice is a lonely DIR

To detect FDs for an ASBR-couple (i, e) of AS X, we propose looking at the set of prefixes associated with the DIR, the special internal route introduced in Sec. IV-A. Recall that the DIR, denoted $D_X(i, e)$, is the route inside X from i to e obtained by tracing e. This internal route is particularly important since it must hold that

$$D_X(i, e) \in \mathcal{R}_X^{LB}(i, e)$$

The networking rationale for this assumption is that, presumably, internal prefixes of ASes, such as the internal destination e of AS X, are not subject to FDs. In other words, regarding internal destinations, it is reasonable to assume that all devices are full-FIB routers.[8] Hence, $D_X(i, e)$ is not expected to detour, and always to represent a best IGP path, which by definition is included in $\mathcal{R}_X^{LB}(i, e)$.[9]

When we conclude for a prefix-based forwarding pattern relying on the detector of Sec. IV, i.e., $s \geq 2$, then we declare that extreme-FDs occur only if we see a *lonely DIR*, i.e., when

$$D_X(i, e) \in \mathcal{R}_j \ \wedge \ |\mathcal{P}_j| < t(Z, \mathbb{P}_X(i, e))$$

$$t(Z, \mathbb{P}_X(i, e)) = \frac{Z}{|\mathbb{P}_X(i, e)|} \sum_{\forall \mathcal{P}_k \in \mathbb{P}_X(i,e)} |\mathcal{P}_k| = Z \cdot \frac{1}{s} \sum_{k=1}^{s} |\mathcal{P}_k|$$

where $t(Z, \mathbb{P}_X(i, e))$ is an adaptive threshold, $0 < Z \leq 1$ is an adjustable parameter and $\frac{1}{s}\sum_{k=1}^{s} |\mathcal{P}_k|$ is the number of prefixes that each set of prefixes $\mathcal{P}_m \in \mathbb{P}_X(i, e)$ should contain assuming a uniform distribution. Note that, for each ASBR-couple (i, e), the total number of prefixes $\sum_{k=1}^{s} |\mathcal{P}_k|$ for which the couple is revealed, and the number of sets s conforming the partitions $\mathbb{P}_X(i, e)$ and $\mathbb{R}_X(i, e)$ generally change. On the other hand, the value of Z can be used to tune the precision and recall of the FD-verdict, i.e., to adjust how cautious we are to declare that FDs occur. The lower Z, the stricter the condition.

The reasoning for the threshold we compute is as follows. In the absence of FDs, while the constrained routes composing $\mathcal{R}_X^{TE}(i, e)$ may carry the traffic of a limited number of prefixes, the LB routes $\mathcal{R}_X^{LB}(i, e)$ evenly distribute the load of the main bulk of prefixes. When FDs occur, some prefixes are forwarded across the routes in $\mathcal{R}_X^{FD}(i, e)$. This can strongly modify the usual distribution of prefixes across routes: fewer prefixes are associated with LB routes. The more prefixes subject to FDs, the less the IGP routes are used to carry transit traffic. In particular, in the event of extreme-FDs, most prefixes are subject to FDs. Hence, looking at the set containing the DIR, we can infer whether the LB set is associated with few or no external prefixes, and we argue that this is a strong hint revealing the occurrence of extreme-FDs.

To illustrate the behavior of the FD-verdict, let us recall the example of Fig. 4, and assume that while tracing a complementary set of prefixes $\mathcal{P}_5 = \{P_9, P_{10}, \ldots, P_q\}$ a new detouring route R_5 was always revealed. Note that, in the updated example, in total q prefixes are measured, 8 from Fig. 4, and the remaining included in $\mathcal{P}_5$. Hence, the higher q, the more prefixes subject to FDs. Since R_5 was not revealed before, then s increases by one for both per-dest/flow LB and prefix-based mechanisms. Indeed, for the first, instead of $s = 1$, we would now have $s = 2$: the new set $\mathcal{P}_5$, and $\{P_1, P_2, \ldots, P_7, e/32\}$, the previously merged one. A uniform distribution would thus require finding $q/2$ prefixes in each set. Assuming $Z = 0.1$, our FD-verdict concludes for extreme-FDs if less than $0.1 \cdot q/2$ prefixes are associated with the DIR, i.e., if $q > 20 \cdot 8$. On the other hand, for the prefix-based mechanisms, we would go from $s = 4$ to $s = 5$, each set containing 2 prefixes, except for $\mathcal{P}_5$. In this case, following the same reasoning as before, the condition to declare extreme-FDs is $q > 50 \cdot 2$. In particular, these examples highlight that, for the FD-verdict to be robust, the number of prefixes analyzed per ASBR-couple needs to be high, e.g. at least 100 prefixes.

B. The FD-detector: a tool deployed in the wild

In this section we describe how we turned the algorithm of Sec. IV, incorporating the FD-verdict, into a tool able to

[5] Note that per-dest/flow LB and prefix-based mechanisms may interfere with each other, generating more complex forwarding patterns. As we discuss in Sec. VIII, our method remains valid in all cases.

[6] This condition is sufficient, but not necessary for $s = 1$ to hold.

[7] Indeed, for the multi-route discovery and merging phases to be applied on any ASBR-couple, a multi-path routing pattern must have been discovered in the prefix-grouping phase, meaning $r > 1$.

[8] Since the IGP does not suffer from similar scalability issues as BGP does, all internal prefixes are expected to be installed in all routers. In addition, IGP prefixes constitute the backbone of an AS and removing them from the FIB of any router would represent a minor scalability gain while letting BGP running on top of a flawed IGP network.

[9] Topologies involving BGP confederations may lead the DIR to be a concatenation of best IGP paths across the sub-ASes into which an AS is divided. Though the collected DIR may not represent the optimal path that could be used between the ASBRs of the AS, it is a best path across the IGPs, and hence still belongs to the LB set.

detect FDs in the wild.

a) Measurement infrastructure: We run our FD-detector leveraging 100 vantage points (VPs) of the NLNOG RING monitoring infrastructure [38] on May 26th 2020. We choose this platform since, besides benefiting from geographically-spread VPs hosted across various tier-1, transit and stub ASes, we are able to run our own scripts to carry out the required measurements. In addition, opposite to RIPE ATLAS [39], we are able to tune the probing rate and number of concurrent measurements. We selected our set of VPs aiming to evenly distribute them across continents and type of ASes, randomly re-assigning their location when the number of available VPs in a continent, or a kind of AS, is not enough to achieve a fair distribution.

b) Collecting traces: We used `scamper` [40] to run ICMP-Paris `traceroute` [41] at 200 pps towards a list of IP addresses extracted from the Internet Address Hitlist provided by the USC/ISI ANT project [42], that covers every allocated /24 IPv4 prefix. In particular, we randomly selected 100K IP addresses in distinct /24 prefixes, where the last byte of each IP address was also randomly chosen. For any destination d_j, the trace $T(d_j)$ is associated to the /24 prefix P_j containing d_j. Our method requires the destination d_j to reply only when collecting DIRs, otherwise they cannot be determined, as we study next. In all remaining traces, we are not sensitive to this, since we are only interested in gathering internal routes of ASes traversed by transit traffic, that thus do not own the traced IP addresses. Note that to check the sanity of the routing inside a specific AS, the destinations can be chosen by leveraging historical measurements or systems as those proposed in [43], [44], to ensure that the collected traces traverse this AS.

c) Identifying robust ASBR-couples and extracting internal routes: For each trace $T(d_j)$, for each AS X that is traversed, we identify the ASBR-couple (i, e) of X as the first and last hop with an IP address mapping to X, and extract the internal route $R_X(d_j)$. We remove (i, e) if either the previous hop of i or next hop of e in $T(d_j)$ fails to be correctly mapped to an AS (e.g. '$\star$', a missing hop). In other words, we only keep unambiguous ASBR-couples. To map from IP-to-AS, we use `bdrmapIT` [45], configured on top of CAIDA's IP-to-AS mapping dataset [46]. Internal routes $R_X(d_j)$ including loops or hops mapping to an AS distinct from X are discarded. While we keep internal routes traversing explicit MPLS tunnels, those where i and e are directly connected are discarded as invisible MPLS tunnels [47] may be obscuring intermediate hops. Finally, recall that for every identified ASBR-couple (i, e) in any AS X, we keep track of $\mathbb{P}_X(i, e)$, the prefixes for which (i, e) is revealed, and $\mathbb{R}_X(i, e)$, the observed internal routes for traces targeting those prefixes. To mitigate outliers or undersampled evidences influencing the outcome of the FD-verdict, we discard all ASBR-couples for which $\mathbb{P}_X(i, e)$ contains less than 100 prefixes, i.e., $\sum_j |\mathcal{P}_j| < 100$.

d) Determining the DIRs: To collect the DIR of each ASBR-couple, from the list of all couples in our dataset, we extract a list of unique egress-ASBRs, and collect a trace for each of them. When the target IP address does not reply, i.e., the trace does not reach the egress-ASBR, we consider that the DIR cannot be determined. All couples in our data collection where the egress-ASBR does not reply are then discarded. On the other hand, when the egress-ASBR replies, we then look for the ingress-ASBR. In this case, the couples associated to the same egress-ASBR but with an ingress-ASBR other than the one observed in the trace are discarded. Indeed, for these ASBR-couples, the DIR cannot be determined since packets enter the AS through another ingress-ASBR. Note that even re-tracing the egress-ASBR, the same mismatching ingress-ASBR would be repeatedly seen. For example, if the couples $(i, e), (i, e'), (i', e)$ and (i', e') are revealed in AS X, then we trace e and e' once. If e replies but e' does not, we delete (i, e') and (i', e'). If in the trace targeting e we find i as ingress-ASBR of X, then we have collected the DIR for (i, e). At the same time, the DIR for (i', e) simply cannot be collected since when we trace e, we reveal i and not i' as ingress-ASBR. Hence, we also discard (i', e), thus only keeping (i, e) at the end of the process. Finally, note that if we had encountered a third ingress-ASBR i'', all couples would have been removed.

e) Managing the probing cost: In the multi-route discovery phase, for each ASBR-couple (i, e) in any AS X, we explore 4 random prefixes for each set of prefixes $\mathcal{P}_j \in \mathbb{P}_X(i, e)$, 64 IP addresses per each. The rationale for this is as follows. Recall that, for each set of prefixes $\mathcal{P}_j$, the same route R_j was observed at the exploration phase. The multi-route discovery phase aims to determine if rather a set of routes $\mathcal{R}_j$ is associated to $\mathcal{P}_j$, instead of only R_j. As discussed in Sec. III-B, and illustrated in Fig. 3 and Fig. 4, the outcome largely depends on the forwarding pattern for the ASBR-couple analyzed. For prefix-based mechanisms, probing different destinations inside a fixed set of prefixes $\mathcal{P}_j$ does not alter the traced prefixes, thus it is likely that the same route is repeatedly seen. On the other hand, since per-flow and per-dest LB are fine-grained LB types, then varying the traced destination would allow to reveal all LB paths even for a unique prefix. In theory, thus, tracing only one prefix per set of prefixes $\mathcal{P}_j$ can seem enough to reveal all routes in $\mathcal{R}_j$. However, to avoid corner cases, e.g., the prefix picked is an outlier and is subject to TE practices, we are conservative and trace 4 prefixes. Finally, note that measuring 64 IP addresses per prefix, the total for each set of prefixes is $256 = 4 \times 64$. Taking into account results of previous research on LB, this value is conservative, as discussed in Sec. VIII. In any case, the prefix-grouping phase greatly reduces the number of prefixes to be probed, thus allowing for the concession of 64 traces per prefix.

f) Dealing with missing hops: The internal routes collected may include missing hops, that appear as '$\star$'. When comparing whether two sets of routes $\mathcal{R}_j, \mathcal{R}_k \in \mathbb{R}_X(i, e)$ intersect or not in the merging phase, we consider all missing hops as wildcards that may be matched to any IP address, but never replace them. Since the FD-verdict declares that a couple (i, e) is subject to FDs when the set containing the DIR is associated to less than $t(Z, \mathbb{P}_X(i, e))$ prefixes, then treating missing hops as wildcards relaxes the condition allowing to merge sets, and thus increases the chances of not finding a lonely DIR. Consequently, this results into a stricter condition to declare FDs, i.e., this is the most conservative approach to deal with missing hops: we may introduce false negatives, but

no false positives.

VI. Capturing Forwarding Detours In The Wild

In this section we discuss the results we obtained running our FD-detector in the wild. First, Sec. VI-A shows results concerning the underlying probing campaigns we performed. We detect FDs in 25 ASes out of 54, across 168 ASBR-couples and 65 ingress-ASBRs. Then, in Sec. VI-B we explore the forwarding patterns we found for each ASBR-couple. We discover a **binary effect** around FDs, i.e., **either all the observed transit traffic traversing a couple detours, or none does**. Then, in Sec. VI-C, we quantify the amount of extreme-FDs we capture per AS and per ASBR-couple. Our results depict the heterogeneity of the FD-phenomenon: from ASes with none or very few couples subject to FDs, to others where thousands of prefixes, across multiple couples suffer from forwarding detours. Moreover, in Sec. VI-D, we investigate the relationship between ingress-ASBRs and FDs. A priori, we do not observe a clear correlation between the ingress-ASBR through which traffic enters any AS and the occurrence of FDs. Finally, we make an attempt to infer the most likely root cause generating the FDs we collect, i.e., with the observed binary characteristics, in Sec. VI-E, and present the efforts we invested in validating our results in Sec.VI-F.

A. Measurement campaigns and coverage

We run measurements from 100 NLNOG RING's VPs, however, we experienced technical issues in 8 of them that did not allow us to complete the measurements required by the FD-detector. In the following, the results refer to the 92 VPs where we could complete the analysis.

In the exploration phase, out of the 100K traces we run, we extracted on average 3 internal routes per trace distributed across 7500 ASes. From those internal routes with unambiguous borders, we see that we traverse from 1405 up to 2205 distinct ingress-ASBRs (except one VP where the value raises up to 2335), between 5662 and 8758 unique egress-ASBRs, and from 6475 to 11590 different ASBR-couples. However, our results indicate that most couples are not commonly encountered: at least 50% appear only once, and 96% are traversed at most for 30 traces. Hence, while the requirement of finding 100 prefixes per couple has a limited effect on the final dataset we analyze, it allows us to be conservative, avoiding to introduce false positives/negatives (see Sec.V-B). On the other hand, when tracing the egress-ASBRs to collect DIRs, we had a success rate usually between 50% and 60%.

Our FD-detector was able to analyze 3963 ASBR-couples spanning 54 ASes. Fig. 5 reports the marginal utility of extending the set of NLNOG RING's VPs in terms of couples covered and traversed ASes. Initially, the tendency shows almost a linear increase with the number of VPs. However, the decreasing slope of the curve and the plateau on the right side of the figure suggest that the gain after 70 VPs is negligible. Indeed, beyond that point, we are able to investigate only 138 additional couples. In the end, we find extreme-FDs in 25 ASes, across 168 ASBR-couples and 65 ingress-ASBRs.

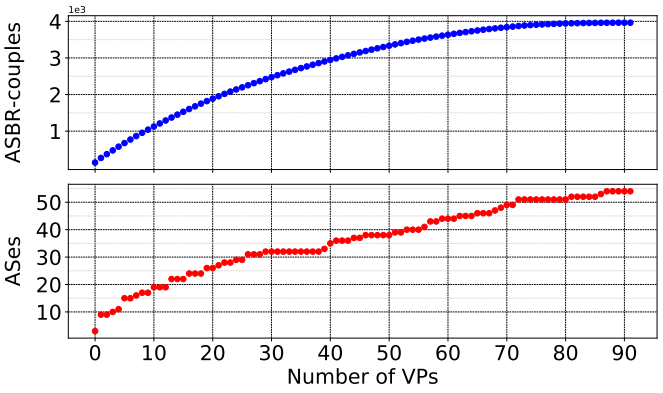

Fig. 5: Marginal utility of adding NLNOG RING's VPs in terms of distinct ASBR-couples (top) and unique ASes (bottom). For more than 70 VPs, the gain in negligible.

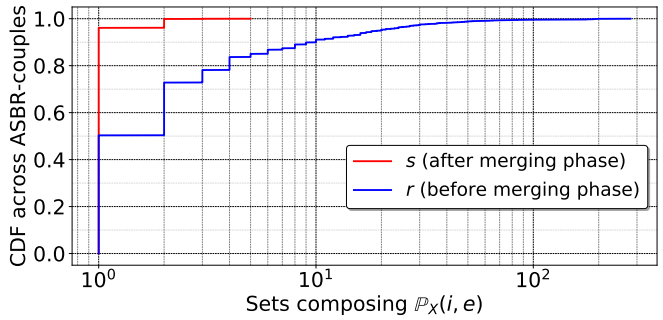

Fig. 6: Cumulative number of sets composing $\mathbb{P}_X(i,e)$ across ASBR-couples before (r) and after (s) the merging phase. When $r = 1$, no multi-path routing pattern was observed. The difference with $s = 1$ relates to cases where we find a forwarding pattern that corresponds to that of per-dest/flow LB. Finally, when $s \geq 2$ a prefix-based forwarding pattern is observed. In these cases, in general, $s = 2$, and they are FDs.

B. Forwarding patterns and the binary effect of FDs

We are interested in determining the forwarding patterns we found for the ASBR-couples in our dataset. In this sense, Fig. 6 reports the CDF of the number of sets composing $\mathbb{P}_X(i,e)$ across couples before and after the merging phase (blue and red curve, respectively). Notably, while multiple sets of routes are visible in half of the couples we explore (blue distribution), less than 5% of them are not eventually merged in the final partition (red distribution). In more detail, observing the blue curve, we see that $r = 1$ in 50% of the cases. These are ASBR-couples for which no multi-path routing pattern was observed. In these cases, we conservatively conclude that these couples are not subject to FDs only running the exploration phase. For the remaining 50% of couples, the other phases are enforced since $r > 1$. At the end of the process, we observe that $s = 1$ for 96% of the couples. The difference in the value between $s = 1$ and $r = 1$ is 46% of the total, and are the cases where we discovered the forwarding pattern of per-dest/flow LB. In other words, for most ASBR-couples e.g. (i,e), the multi-route phase enlarged the sets composing $\mathbb{R}_X(i,e)$, and then the

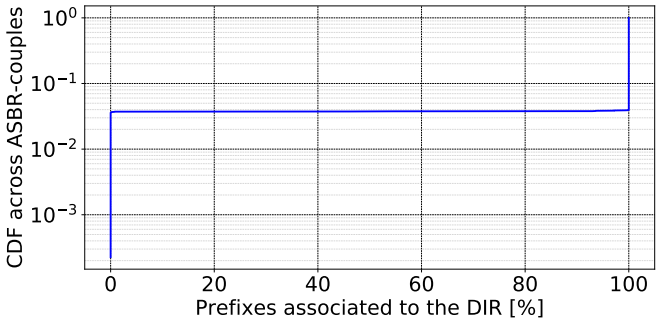

Fig. 7: Cumulative number of prefixes associated to the DIR across ASBR-couples. We observe a clear binary pattern: for any couple, either all traffic detours (left side, ∼4%), or none does (right side, ∼96% of the cases). Hence, our FD-detector is not sensible to the value of the threshold $t(Z, \mathbb{P}_X(i, e))$.

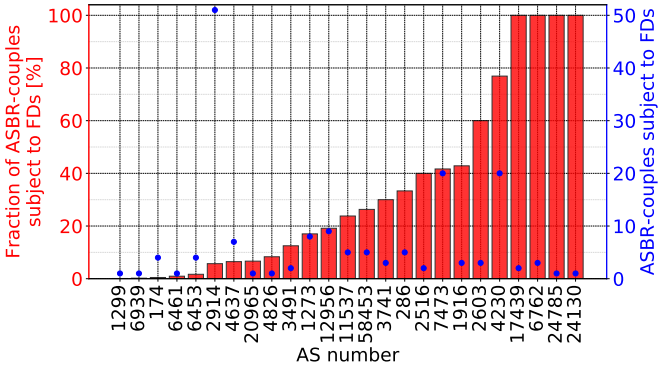

Fig. 8: Quantification of ASBR-couples subject to FDs per AS. While most ASes have less than 10 couples subject to FDs (blue dots), the fraction they represent out of the total in their AS (red bars) largely varies. This indicates that the problem of FDs is AS-dependent.

merging phase was able to group them, since they had routes in common. This highlights the effectiveness of the multi-route discovery and merging phases. Moreover, recalling that we only measured 4 prefixes across the sets of $\mathbb{P}_X(i, e)$, this also shows the potential of the prefix-grouping phase. Finally, for the remaining 4% of ASBR-couples, we find a prefix-based forwarding pattern where, except for a few exceptions, $s = 2$.

From the cases where $s = 2$, we then extract the number of extreme-FDs. Fig. 7 shows the share of prefixes associated with the DIR for all ASBR-couples. Recall that the FD-verdict concludes that a couple (i, e) in AS X is subject to FDs when less than $t(Z, \mathbb{P}_X(i, e))$ prefixes are associated to the DIR $D_X(i, e)$ (see Sec. V-A). The curve in Fig. 7 reveals a remarkable on/off pattern indicating that all measured transit traffic that traverses any ASBR-couple either always detours, or never does. The right side of Fig. 7 relates to the ∼96% of the ASBR-couples for which $s = 1$ and all prefixes are forwarded along best IGP paths. On the other hand, the ∼4% remaining in Fig. 7 are those ASBR-couples for which $s = 2$ in Fig. 6. Since the rate of prefixes associated to the DIR is always 0%, then all these couples are subject to FDs, i.e., the rate of prefixes subject to FDs is of 100% (except for the DIR, of course). This shows that our FD-detector is not sensitive to any calibration issue concerning the adaptive threshold $t(Z, \mathbb{P}_X(i, e))$ in the FD-verdict. In other words, there are no gray regions: when $s = 2$, no false negatives can occur since it always holds that 100% of the prefixes are not associated with the DIR, i.e., lonely DIRs are always completely alone.

C. Distribution of FDs per AS and ASBR-couples

Fig. 8 shows the breakdown per AS of the 168 ASBR-couples subject to FDs, sorted by increasing relative fraction across ASes. We observe no general trend, indicating that the prevalence of FDs is AS-specific, e.g. depending on both router's hardware and OSes in use. This analysis is supported by the fact that, even though most ASes have few measured couples with FDs, less than 10 in general, the relative values spawn from as low as almost 0% to up to 100%. Moreover, while one could argue that the left side of the Fig. 8 seems to be populated with ASes with a high AS Rank [48], the same holds for example for AS6762, that has all of its measured couples with FDs. In addition, it is interesting to mention the case of AS2914, with a relative value around 10%, but more than 50 couples for which traffic detours; and those of AS7473 and AS4230, both with 20 couples exhibiting FDs, but that represent 40% and 80% respectively of the total measured. These three cases emphasize the lack of a general tendency among ASes, i.e., the FD-phenomenon seems to depend on configurations specific to each AS.

More in depth, considering the granularity of the ingress-ASBR, across the 168 ASBR-couples subject to FDs, we observe that they span (only) 65 ingress-ASBRs. Fig. 9 complements Fig. 8 offering this detailed view: for each AS (color), the couples and prefixes subject to FDs (bars) are grouped per ingress-ASBR (separated by dash lines). In general, FDs affect multiple prefixes in many ASes, and are sometimes distributed across numerous ingress routers (at least relying on an IP level view) as it is the case in AS2914. The same variability we already discuss at the AS-scale occurs for ASBR-couples. Indeed, while some ingress-ASBRs exhibit many prefixes subject to FDs, other expose few. The same occurs even more clearly across different egress-ASBRs of any fixed ingress-ASBR.

D. Correlation between ingress-ASBRs and FDs

In this section we question whether the ability to detect FDs largely depends on the ingress-ASBR we traverse on each AS. In other words, we aim to determine whether transit traffic always detours if a given ingress-ASBR is traversed, indistinctly of the egress-ASBR through which traffic exits the AS under study. According to Fig. 9, there exist multiple ASBR-couples (i, e) subject to FDs for which the same ingress-ASBR i appears associated to different egress-ASBRs. e.g. e and e'. However, this does not imply that there does not exist another distinct egress-ASBR e'' for which the couple (i, e'') is not subject to FDs. To clarify this aspect, Fig. 10 shows the fraction of egress-ASBRs subject to FDs associated to each

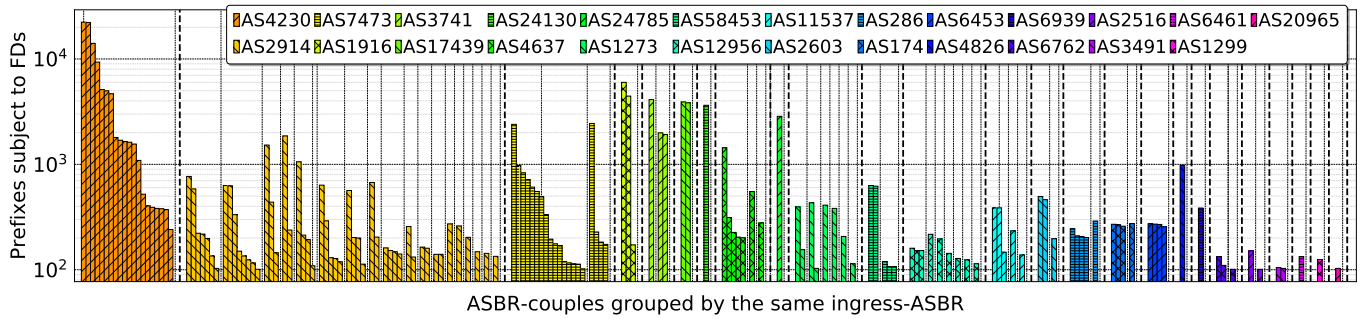

Fig. 9: Number of prefixes subject to FDs per ASBR-couple. The bars are separated by dashed lines to emphasize a distinct ingress-ASBRs. The number of ingress-ASBRs, ASBR-couples and prefixes subject to FDs strongly depends on the AS studied.

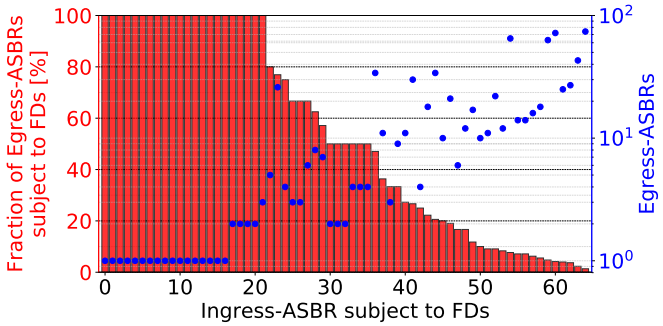

Fig. 10: Fraction of egress-ASBRs that are subject to extreme-FDs (red bars) out of the total (blue dots) for each ingress-ASBR. The tendency shows that the more egress-ASBRs per ingress-ASBR, the less the fraction subject to FDs. However, for 17 ingress-ASBRs we cannot conclude anything since they only appear in one ASBR-couple.

ingress-ASBR, e.g. the case comprising i, e, e' and e'' would result into a red bar of height $66,6\%$, and a blue dot indicating the value of 3. We see a tendency that indicates that, the more egress-ASBRs that we find for an ingess-ASBR, the fraction subject to FDs is less. However, there are still cases where we observe that an ingress-ASBR is associated to multiple (2 or 3) egress-ASBRs, and we always find FDs. In addition, there are $17/65$ ingress-ASBRs for which we cannot derive any conclusion since they are only seen in a unique ASBR-couple. Hence, for the moment, we can only conservatively state that a relationship between FDs and ingress-ASBRs is not clear, and would like to better study this in future work.

E. Speculating on the root causes generating FDs

Based on previous results, this section elaborates an explanation of what may have generated the FDs we observed. Despite risky since the root causes behind forwarding detours may be multiple (see Sec. I), we argue this is valuable since the patterns observed seem clear cut. Indeed, even if the core contribution of this work is our methodology to detect FDs, the binary effect we found (Fig. 7) makes us believe that we are also able to pinpoint the most likely reason behind the FDs we collected. In short, the FDs we detect seem to result from scenarios involving partial-FIB routers, i.e., where routers keep IGP prefixes but delete a large fraction (if not all) of BGP prefixes from the FIB. Note that this is emphasized by the binary effect, that is even more severe that what we previously labeled as extreme-FDs.

A partial-FIB router x with no BGP prefixes installed and relying on a default route, systematically sends traffic towards a default gateway y. A priori, if y considers itself the best exit point of the AS for *all* BGP prefixes then, no FDs occur. However, depending on the best covering prefix of the destination IP address of the packets being forwarded, y may likely redirect transit traffic towards another ASBR z. This is similar to what happens with prefixes P_R and P_B in the example shown in Fig. 1 for $x = ASBR_1$, $y = ASBR_2$ and $z = ASBR_3$, where traffic for P_B detours, but that of P_R does not. More generally, in all cases where the best IGP path from x to z does not go through y, FDs occur.

The proportion of red in each bar of Fig. 10 could then be considered a measure of how bad it was to choose y as default gateway for x. In particular, the cases of complete red bars are of interest, since in them y never chooses itself as exit point of the AS, and all traffic detours. This could be the case, for example, if y was not an ASBR, but rather a core router. On the other hand, the shortest red bars also represent an interesting case of study that may result from multiple causes. A trivial explanation could be that the default gateway was well chosen. However, other causes, more complex, are possible. For example, it could happen that traffic exited the AS before reaching the gateway, hence avoiding FDs for these egress-ASBRs. Another plausible explanation could be that the ingress-ASBR i was actually not the partial-FIB router, but rather a core router x on which i relies. In such a scenario, only those prefixes for which traffic ingresses via i, and then x is traversed, will lead to few ASBR-couples subject to FDs.

We believe that these last examples highlight well the difficulty in finely validating the root causes generating FDs, which besides being many, may be distributed across the AS. This is also emphasized by the heterogeneous patterns found in the results of Fig. 8, 9 and 10, which imply that ASes may employ multiple partial-FIB routers located at different positions in the network and resulting in many ASBR-couples identified as subject to FDs for varying number of prefixes.

F. Validation: emulations and ground truth

Relying on GNS3, we reproduce by emulation all the forwarding patterns we describe in this paper, specially that of per-prefix LB. To mimic FDs, we rely on a static default route having a higher priority than other FIB entries. In addition, we run our FD-detector on each LB flavor independently or combined with FDs and TE to corroborate its potential and correctness on all the scenarios discussed in our work.

In addition, we corroborated the performance of our tool from a VP where we had previously discovered the presence of a partial-FIB router. The example of Fig. 1 accurately describes the network hosting such router. While for some prefixes the router was generating BGP lies [23], i.e., traceroute AS-level forwarding routes to differ from BGP paths, for others it was introducing FDs. Our tool was able to detect these FDs, probing its usefulness in a real life experiment.

Finally, at this stage, we cannot fully validate the origin of FDs for all cases. Despite this, we claim that similarly to LB tools tested on controlled environments such as GNS3, our FD-detector has proven to be valid. In any case, we believe our analysis opens a door to develop a better understanding of the FD-phenomenon, that may be deepened in future research.

VII. Related Work

Back in 2004, when full FIBs only had 100K entries, compared to more than 800K nowadays, Bu et al. [49] studied the increase in BGP tables caused by what they called an explosive growth of the Internet. While their study focused on the reasons behind this increase, we focus on the consequences; more precisely, on their impact on the forwarding inside ASes. Several proposals aim to reduce routing tables sizes by aggregating routes [12] and sometimes redirecting traffic to more knowledgable routers [13]. The growth of the FIB indeed favors the use of workarounds like partial-FIBs and default routes, that may in turn lead to FDs.

Deflections are a known phenomenon that has been studied from different angles, however, none are run at the same scale, nor with the same objective as ours. Elena et al. [50] pinpoint AS-wide deflections, though their goal is to detect path diversity on the Internet. They conclude that intra-domain LB was not well deployed at the time. Secci et al. [51] study end-to-end deflections created by BGP. While they also investigate intra-domain deflections, they focus on the dynamics and oscillations due to the MED attribute. Agarwal et al. [52] analyze BGP routing changes as deflections. They try to detect intra-domain deflections to build accurate traffic matrices. Bush et al. [22] investigate the use of safety net default routes ensuring reachability upon routing events. For this, they poison routes and then test whether associated prefixes are still reachable. Different to these studies, our work focuses on detecting FDs inside ASes, not focusing on any particular cause that might generate them, and only using `traceroute`, i.e., without interfering with the routing. Moreover, our FD-detector can complement the work of Del Fiore et al. [23], that pinpointed partial-FIB routers as a reason for discrepancies between BGP paths and traceroute-AS paths.

On the other hand, there have been multiple studies concerning LB. Augustin et al. [24] introduce Paris-traceroute, a per-flow load-balancing-aware version of traceroute allowing to avoid erroneous inference of links, loops and cycles seen in the standard traceroute, as further studied by Viger et al. [25]. Based on the principles of Paris-traceroute, Augustin et al. [53] develop the Multipath Detection Algorithm (MDA), allowing to detect per-flow and per-packet load balancers. In subsequent studies, they extend the MDA also to detect per-destination load balancers [54], [55]. Veitch et al. [31] refine the stopping points of the MDA to bound the failure probability of full multipath discovery. Vermeulen et al. [56] propose the MDA-Lite, a lite version of the MDA that requires less probes, but may fail to discover all nodes and links. Later, they propose Diamond Miner [32], a system able to produce Internet-wide multipath topology maps in less than 3-day long snapshots [32]. Diamond-Miner implements the MDA with a stateless probing fashion relying on Yarrp [57], a randomized high-speed prober. Almeida et al. [34] generalize the MDA and propose the Multipath Classification Algorithm (MCA). In general, all these works show that per-flow and per-destination LB are the most widespread LB flavors. Except for Diamond Miner, they run measurement campaigns in the order of 10K and no more than 70K destination IP addresses from at most 32 VPs. In particular, they seek for multipath routing patterns and implicitly assume they result from LB techniques. Our analysis complements these works: we study per-prefix LB, a flavor not discussed in the literature, and we show that FDs also produce multi-path routing patterns. In addition, the coverage of our campaign is larger: we use 100 VPs, and an IP list of 100K destinations on the exploration phase. Moreover, we propose a novel prefix-grouping step, that may allow to decrease the probing cost of LB discovery campaigns.

VIII. Discussion: Robustness of the FD-detector

In this section we analyze how our FD-detector performs face to complex forwarding patterns in Sec. VIII-A, explain why routing changes and IP-to-AS mapping errors do not induce the results we obtained in Sec. VIII-B and VIII-C, illustrate why the probing cost of the multi-route discovery phase was sufficient in Sec. VIII-D and discuss why our analysis does not require alias resolution techniques in Sec. VIII-E.

A. An FD-verdict handling all interactions of FDs and LB

The LB types studied in Sec. III, fine-grained and coarse-grained, may be mixed to produce *hybrid* LB flavors. When per-prefix LB is applied upstream of per-dest/flow LB, this combination results in a generalization of per-prefix LB. For traces concerning a fixed set of prefixes $\mathcal{P}_j$, instead of a unique route R_j, a set of routes $\mathcal{R}_j$ are repeatedly revealed. In fact, the routes in $\mathcal{R}_j$ are only used to forward traffic concerning the prefixes in $\mathcal{P}_j$. Consequently, this hybrid flavor is coarse-grained, meaning that the property $s > 1$ still holds. On the other hand, when load balancers are applied in the reverse order, that is, with per-dest/flow followed by per-prefix LB, each prefix is not anymore forwarded thorough all routes of $\mathcal{R}_X^{LB}(i, e)$ like with per-dest/flow LB. Indeed, in these cases, tracing a set of prefixes $\mathcal{P}_j$, the same sub-set of routes $\mathcal{R}_j$ is consistently found. However, different to the previous hybrid

LB flavor, it can be shown that $\forall j,k, \mathcal{R}_j \cap \mathcal{R}_k \neq \emptyset$. These intersections usually contain multiple routes, and thus the merging phase would likely output the same as for fine-grained LB flavors, i.e., $s = 1$. Hence, our FD-detector is not affected by the hybrid flavors, resulting in $s \geq 2$ for the first, and $s = 1$ for the latter, as with the simpler LB flavors they generalize.

Finally, note that detouring traffic may traverse a load balancer, thus FDs may be subject to LB. In particular, if the load balancer applies per-dest/flow LB, then no major changes occur since the FD-detector will be able to group the detouring routes into a unique set of routes during the merging phase. On the other hand, if the load balancer uses per-prefix LB, then the prefixes subject to FDs would be evenly distributed across $\mathcal{R}_X^{FD}(i,e)$. Our FD-detector will not be able to merge these load balanced FDs into a unique set of routes. However, we designed the FD-verdict to take this case into account: rather than searching for a set that presumably is the one resulting from FDs, we look at the one containing the DIR, that is associated to LB. When extreme-FDs occur, a lonely DIR is found, indistinctly of whether the FDs are load balanced or not, and thus we are still able to detect FDs.

B. A binary effect that unlikely results from routing changes

To avoid issues related to routing events, since our study is performed at the scale of ASBR-couples (i,e), we only require the routing to remain stable within the studied AS (while we are measuring each couple). Even if routing changes occured inside the AS, since we always request to find i and e on the paths, such changes would affect the collection of routes only if they occurred on links or routers in the paths between i and e. Overall, our measurement campaign lasts less than one day; this period, being lower than typical topology discovery campaigns, seems short enough to limit the impact of IGP routing changes. In addtition, we collect again the DIR during the multi-route discovery phase. Hence, we consider it is very unlikely that IGP routing changes may have generated the binary effect wee detected. Indeed, for this to happen, it would mean that only the DIR got affected, but not the other internal routes that were collected at the same time.

C. On the (in)sensibility of flawed ASBR detection

While we expect the IP-to-AS mapping tool in use to be accurate, here we discuss why our analysis should not be significantly impacted even if `bdrmap-it` [45] failed to work properly. Let us assume an example where (i,e) is the real ASBR-couple, and (i',e') are the borders identified in the mapping process. First, even though our FD-detector specifically checks whether FDs occur between ASBR-couples, our methodology remains valid for any two IP addresses belonging to the same studied AS. Hence if i' and e' are actually core routers in the same AS as i and e, we may only lose the opportunity to detect some FDs. Indeed, this happens because we overlook the subpaths between i and i', and e and e'. On the other hand, when e' actually belongs to a peering AS, as long as the prefix used in the point-to-point link between e and e' is redistributed within the IGP of the targeted AS, our methodology remains valid. This holds because the DIR towards e' still represents a valid IGP route associated with LB, thus we can continue to use it in the FD-verdict. Finally, when i' belongs to a peering AS, this could potentially generate more problems since i' may forward traffic to ingress-ASBRs in the studied AS other than i. While we argue that this is not a common practice, we acknowledge that this could be perceived as a limitation. However, in these cases in particular and for all mapping errors in general, we expect the FD-verdict to strongly mitigate their impact: finding a lonely DIR still implies a case likely resulting from FDs.

D. Measurement stopping points

While the MDA uses adaptive measurement stopping points (see Sec. VII), we decided to launch a static number of traces per prefix (i.e., 64). The MDA works on a hop-by-hop fashion: as measurements are being carried, it adaptively updates its probing stopping points according to the probability of achieving the full discovery of all routes. In our case, to ease the management of vantage points, we opted to feed all nodes with a fixed set of destinations to probe. This not only grants predictability of the full probing cost of the campaign and so its duration, but also allows measurements to run faster than with the MDA, similar to the stateless fashion of Diamond-Miner [32]. Note that the number of traces we consider per group of prefixes (4×64) largely exceeds 11 and 96, the number of traces required to reveal 2 and 16 next-hops of a load balancer [53]. Indeed, 2 and 16 represent the largely most common and the maximum number of next-hops usually found in practice, respectively [34], [55], [58]. As discussed in Sec. VI-B, the patterns we observe in our results highlight the effectiveness of the merging and multi-route discovery phases.

E. Alias resolution: a nice, but dangerous additional feature

Similar to the LB studies presented in Sec. VII, our methodology performs its analysis at the IP-level. However, alias resolutions techniques (e.g. MIDAR [59]) would allow us to produce a router-level view of the problem. In particular, by identifying IP addresses belonging to the same ASBR, we would be able to refine our analysis of forwarding patterns. In other words, this would allow us to detect all paths ending at the same ASBR, for all IP addresses of the ASBR, and thus better quantify the number of prefixes subject to FDs. Despite this, alias resolution techniques are known to be error prone and to require extensive probing. Consequently, we are cautious, and leave this feature for future work.

IX. Conclusion

With routing tables beyond 800K routes, not all devices are able to handle such load. In these circumstances, ASes may deploy offloading workarounds to cope with these scalability issues, e.g. some BGP entries, if not the vast majority, may not be pushed in the FIB of some routers. However, such workarounds increase the risk of introducing FDs inside these networks, thus losing the IGP optimality. Besides the use of partial-FIB routers and default routes, other reasons like bugs or prefix aggregation can also lead to the same phenomenon.

At the same time, ASes usually rely on ECMP load balancers and TE to increase and control the distribution of traffic in their network, respectively. With FDs, LB and TE, multi-path routing patterns emerge. While exposing such multi-path routing patterns only requires extensive probing, determining the underlying cause generating them is challenging.

In this paper, we propose a method to detect FDs within an AS. More precisely, we show that studying the forwarding pattern between ASBRs of an AS, it is possible to discriminate LB and TE from FDs in the cases when multiple prefixes are subject to FDs. To the best of our knowledge, we are the first to tackle this problem. We build an FD-detector and, using large-scale measurement campaigns, we show that almost half of the ASes in our dataset suffer from FDs. Our results indicate that FDs are usually visible from few ingress points of ASes, and can be revealed depending on the particular egress point that is observed. In addition, our analysis provides a notable takeaway: FDs look to be more extreme than we what we imagined, i.e., we systematically observe a binary effect such that, between two ASBRs of an AS, either all prefixes we measured were subject to FDs, or none were. Though beyond the scope of this paper, we argue that the root cause behind such FDs may be due to the use of partial-FIB routers. Finally, our study allows to refine previous work on topology discovery. Indeed, not only we consider an LB flavor omitted in the literature, i.e. per-prefix LB, but also propose a novel probing methodology that can be directly plugged into LB discovery techniques to improve their probing cost. In future work, we would like to adapt the current implementation of the FD-detector to turn it into an online tool. In addition, we aim to shed light on the quantification of the detrimental effects that FDs have on routing performance.

X. Acknowledgments

We are grateful to Tom Trassoudaine, who run the emulations on GNS3, and tested our tool on such environment. We thank Job Snijders for proving us access to the NLNOG RING monitoring infrastructure. This work has been published under the framework of the IdEX Unistra and benefited from a funding from the state managed by the French National Research Agency as part of the "Investments for the future" program. This project has been made possible in part by a grant from the Cisco University Research Program Fund, an advised fund of Silicon Valley Foundation. This work is partially funded by the Italian Research Program "PON AIM *Attraction and International Mobility*, Azione I.2 Linea 1, *Mobilità dei Ricercatori*" (Codice proposta attivitá AIM1878982-2 CUP E56C19000330005)

References

[1] "*CIDR-REPORT* Status summary." [Online]. Available: https://www.cidr-report.org

[2] A. Elmokashfi and A. Dhamdhere, "Revisiting bgp churn growth," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 1, pp. 5–12, 2013.

[3] "BGP in 2018 – BGP Churn," https://blog.apnic.net/2019/01/22/bgp-in-2018-bgp-churn/.

[4] D. Hauweele, B. Quoitin, C. Pelsser, and R. Bush, "What do parrots and bgp routers have in common?" *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 3, p. 2, 2018.

[5] X. Zhao, D. J. Pacella, and J. Schiller, "Routing scalability: an operator's view," *IEEE Journal on Selected Areas in communications*, vol. 28, no. 8, pp. 1262–1270, 2010.

[6] "What caused today's Internet hiccup," https://www.bgpmon.net/what-caused-todays-internet-hiccup/.

[7] "768k Day. Will it Happen? Did it Happen?" https://labs.ripe.net/Members/emileaben/768k-day-will-it-happen-did-it-happen.

[8] F. Coras, D. Saucez, L. Jakab, A. Cabellos-Aparicio, and J. Domingo-Pascual, "Implementing a BGP-free ISP core with LISP," 2012.

[9] S. Vissicchio, L. Cittadini, and G. Di Battista, "On ibgp routing policies," *IEEE/ACM Transactions on Networking*, vol. 23, no. 1, pp. 227–240, Feb 2015.

[10] S. Vissicchio, L. Cittadini, L. Vanbever, and O. Bonaventure, "iBGP deceptions: More sessions, fewer routes," 03 2012, pp. 2122–2130.

[11] J. L. Sobrinho, L. Vanbever, F. Le, A. Sousa, and J. Rexford, "Scaling the Internet routing system through distributed route aggregation," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3462–3476, Dec. 2016.

[12] H. Ballani, P. Francis, T. Cao, and J. Wang, "Making routers last longer with viaggre." in *NSDI*, vol. 9, 2009, pp. 453–466.

[13] E. Karpilovsky, M. Caesar, J. Rexford, A. Shaikh, and J. Van Der Merwe, "Practical network-wide compression of ip routing tables," *IEEE Transactions on Network and Service Management*, vol. 9, no. 4, pp. 446–458, 2012.

[14] "*Slimming down the Internet routing table* by tore anderson." [Online]. Available: https://www.redpill-linpro.com/sysadvent/2016/12/09/slimming-routing-table.html

[15] J. Fu, P. Sjodin, and G. Karlsson, "Loop-free updates of forwarding tables," *IEEE Transactions on Network and Service Management*, vol. 5, no. 1, pp. 22–35, 2008.

[16] Y. Vanaubel, P. Mérindol, J.-J. Pansiot, and B. Donnet, "MPLS Under the Microscope," in *the 2015 ACM Conference*. New York, New York, USA: ACM Press, 2015, pp. 49–62.

[17] B. Thomas, L. Andersson, and I. Minei, "LDP Specification," RFC 5036, Oct. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc5036

[18] A. Bashandy, C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, and R. Shakir, "Segment Routing with the MPLS Data Plane," RFC 8660, Dec. 2019. [Online]. Available: https://rfc-editor.org/rfc/rfc8660.txt

[19] S. Deshpande, M. Thottan, and B. Sikdar, "An online scheme for the isolation of bgp misconfiguration errors," *IEEE Transactions on Network and Service Management*, vol. 5, no. 2, pp. 78–90, 2008.

[20] R. Fontugne, E. Bautista, C. Petrie, Y. Nomura, P. Abry, P. Gonçalves, K. Fukuda, and E. Aben, "Bgp zombies: An analysis of beacons stuck routes," in *International Conference on Passive and Active Network Measurement*. Springer, 2019, pp. 197–209.

[21] R. Bush, O. Maennel, M. Roughan, and S. Uhlig, "Internet optometry: Assessing the broken glasses in Internet reachability," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '09. ACM, 2009.

[22] J. M. Del Fiore, P. Merindol, V. Persico, C. Pelsser, and A. Pescapé, "Filtering the noise to reveal inter-domain lies," in *2019 Network Traffic Measurement and Analysis Conference (TMA)*, June 2019, pp. 17–24.

[23] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, 2006, pp. 153–158.

[24] F. Viger, B. Augustin, X. Cuvellier, C. Magnien, M. Latapy, T. Friedman, and R. Teixeira, "Detection, understanding, and prevention of traceroute measurement artifacts," *Computer Networks*, vol. 52, no. 5, pp. 998 – 1018, 2008.

[25] "How does load balancing work?" https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html.

[26] "Per-flow and per-packet load balancing," https://support.huawei.com/enterprise/en/doc/EDOC1100055041/ebc8ad42/per-flow-and-per-packet-load-balancing.

[27] K.-C. Leung, V. O. Li, and D. Yang, "An overview of packet reordering in transmission control protocol (tcp): problems, solutions, and challenges," *IEEE transactions on parallel and distributed systems*, vol. 18, no. 4, pp. 522–535, 2007.

[28] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On load distribution over multipath networks," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 3, pp. 662–680, 2011.

[29] J. Bellardo and S. Savage, "Measuring packet reordering," in *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*, 2002, pp. 97–105.

[30] D. Veitch, B. Augustin, R. Teixeira, and T. Friedman, "Failure control in multipath route tracing," in *IEEE INFOCOM 2009*, 2009, pp. 1395–1403.

[31] K. Vermeulen, J. P. Rohrer, R. Beverly, O. Fourmaux, and T. Friedman, "Diamond-miner: Comprehensive discovery of the internet's topology diamonds," in *17th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 20)*, 2020, pp. 479–493.

[32] "Understanding the algorithm used to load balance traffic on mx series routers," https://www.juniper.net/documentation/en_US/junos/topics/concept/hash-computation-mpcs-understanding.html.

[33] R. Almeida, R. Teixeira, D. Veitch, C. Diot *et al.*, "Classification of load balancing in the internet," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*. IEEE, 2020, pp. 1987–1996.

[34] "Configuring per-prefix load balancing," https://www.juniper.net/documentation/en_US/junos/topics/usage-guidelines/policy-configuring-per-prefix-load-balancing.html.

[35] "Cef polarization," https://www.cisco.com/c/en/us/support/docs/ip/express-forwarding-cef/116376-technote-cef-00.html.

[36] S. D. Strowes, "Visibility of ipv4 and ipv6 prefix lengths in 2019." [Online]. Available: https://labs.ripe.net/Members/stephen_strowes/visibility-of-prefix-lengths-in-ipv4-and-ipv6

[37] "*NLNOG RING* monitoring infrastructure," https://ring.nlnog.net.

[38] "RIPE Atlas - User-Defined Measurements," https://atlas.ripe.net/docs/udm/#rate-limits.

[39] "Scamper," https://www.caida.org/tools/measurement/scamper/, [Online; accessed January 2021].

[40] "Paris Traceroute," https://paris-traceroute.net/, [Online; accessed January 2021].

[41] X. Fan and J. Heidemann, "Selecting representative IP addresses for Internet topology studies," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 411–423.

[42] I. Cunha, P. Marchetta, M. Calder, Y.-C. Chiu, B. Schlinker, B. V. A. Machado, A. Pescapé, V. Giotsas, H. V. Madhyastha, and E. Katz-Bassett, "Sibyl: A practical Internet route oracle," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16)*, 2016.

[43] V. Giotsas, T. Koch, E. Fazzion, Í. Cunha, M. Calder, H. V. Madhyastha, and E. Katz-Bassett, "Reduce, reuse, recycle: Repurposing existing measurements to identify stale traceroutes," in *Proceedings of the ACM Internet Measurement Conference*, 2020, pp. 247–265.

[44] A. Marder, M. Luckie, A. Dhamdhere, B. Huffaker, J. M. Smith *et al.*, "Pushing the boundaries with bdrmapit: Mapping router ownership at internet scale," in *Proceedings of the Internet Measurement Conference 2018*. ACM, 2018, pp. 56–69.

[45] "CAIDA's prefix2AS dataset," http://www.caida.org/data/routing/routeviews-prefix2as.xml.

[46] J. Luttringer, Y. Vanaubel, P. Mérindol, J. Pansiot, and B. Donnet, "Let there be light: Revealing hidden mpls tunnels with tnt," *IEEE Transactions on Network and Service Management*, pp. 1–1, 2019.

[47] "As rank," https://asrank.caida.org.

[48] T. Bu, L. Gao, and D. F. Towsley, "On characterizing BGP routing table growth." *Computer Networks ()*, 2004.

[49] E. Elena, J. L. Rougier, and S. Secci, "Characterisation of AS-level path deviations and multipath in internet routing," in *6th EURO-NGI Conference on Next Generation Internet*, 2010.

[50] S. Secci, J.-L. Rougier, A. Pattavina, M. Marinoni, G. Maier, and E. M. T. Elena, "Detection of bgp route deflections across top-tier interconnections," 2009.

[51] S. Agarwal, C.-N. Chuah, S. Bhattacharyya, and C. Diot, "The impact of BGP dynamics on intra-domain traffic." *SIGMETRICS*, 2004.

[52] B. Augustin, T. Friedman, and R. Teixeira, "Multipath tracing with paris traceroute," in *2007 Workshop on End-to-End Monitoring Techniques and Services*. IEEE, 2007, pp. 1–8.

[53] B. Augustin, T. Friedman, and R. Teixeira, "Measuring load-balanced paths in the internet," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, 2007, pp. 149–160.

[54] B. Augustin, T. Friedman, and R. Teixeira, "Measuring multipath routing in the internet," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 830–840, 2010.

[55] K. Vermeulen, S. D. Strowes, O. Fourmaux, and T. Friedman, "Multi-level mda-lite paris traceroute," in *Proceedings of the Internet Measurement Conference 2018*, 2018, pp. 29–42.

[56] R. Beverly, "Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery," in *Proceedings of the Sixteenth ACM SIGCOMM/USENIX Internet Measurement Conference (IMC)*, Nov. 2016.

[57] P. Marchetta, A. Montieri, V. Persico, A. Pescapé, A. Cunha, and E. Katz-Bassett, "How and how much traceroute confuses our understanding of network paths," in *2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*, 2016, pp. 1–7.

[58] K. Keys, Y. Hyun, M. Luckie, and k. claffy, "Internet-Scale IPv4 Alias Resolution with MIDAR," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 383–399, Apr 2013.

Julián M. Del Fiore received the PhD degree from the University of Strasbourg, France, in 2021. Previously, he obtained with honors the Electronics Engineer degree from the University of Buenos Aires, Argentina. He initially worked in the field of wireless networks, developing link-layer protocols for Industrial-IoT applications. Currently, his work aims to extend the results of his PhD, therefore focusing on the detection of routing anomalies, and Internet measurements.

Valerio Persico is an Assistant Professor at DIETI, University of Napoli Federico II, where he received the PhD in Computer and Automation Engineering in 2016. His work concerns network measurements, cloud-network monitoring, and Internet path tracing and topology discovery. He has co-authored more than 30 papers within international journals and conference proceedings.

Pascal Merindol received the Ph.D. degree from the University of Strasbourg, France, in 2008. He was a Postdoctoral Researcher with the Université catholique de Louvain, Belgium, for two years. He is currently an Associate Professor with the Networks Research Group, ICube Laboratory, University of Strasbourg. His main research topics are routing and Internet measurements.

Cristel Pelsser Cristel Pelsser received the Ph.D. degree from the Université catholique de Louvain (UCLouvain), Belgium. She has spent nine years working for ISPs. She has been a Professor with the University of Strasbourg since November 2015. She leads a team of researchers focusing on core Internet technologies. Her aim is to facilitate network operations, and to avoid network disruptions and, when they occur, pinpoint the failure precisely in order to quickly fix the issue.

Antonio Pescapé (SM'09) is a Full Professor of computer engineering at the University of Napoli Federico II. His work focuses on measurement, monitoring, and analysis of the Internet. He has co-authored more than 200 conference and journal papers, he is the recipient of a number of research awards. Also, he has served as an independent reviewer/evaluator of research projects/project proposals co-funded by a number of governments and agencies.