UNIVERSITÉ DE STRASBOURG

ÉCOLE DOCTORALE MATHÉMATIQUES, SCIENCES DE L'INFORMATION ET DE L'INGÉNIEUR

Laboratoire ICube — UMR7357

THÈSE présentée par:

Julián Martín DEL FIORE

soutenue le : **08 février 2021**

pour obtenir le grade de: **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité: **Informatique**

Detecting Hidden Broken Pieces of the Internet: BGP Lies, Forwarding Detours and Failed IXPs

THÈSE dirigée par :
Mme. PELSSER Cristel Professeur, Université de Strasbourg

THÈSE co-encadré par :
M. MERINDOL Pascal Maître de conférence, Université de Strasbourg

RAPPORTEURS :
M. DONNET Benoit Professeur, Université de Liège
M. URVOY-KELLER Guillaume Professeur, Université Nice Sophia Antipolis

---

AUTRES MEMBRES DU JURY :
Mme. MAGNIEN Clémence Directrice de recherche, Sorbonne Université

Julián Martín DEL FIORE

Detecting Hidden Broken Pieces of the Internet: BGP Lies, Forwarding Detours and Failed IXPs

Résumé

L'objectif de cette thèse est de détecter des éléments défaillants d'Internet. Tout d'abord, nous étudions le déploiement des points d'échange Internet (IXP) en Amérique latine et constatons que certains pays sont en situation d'échec dans leur déploiement IXP, c'est-à-dire aucun IXP du tout, ou bien que l'IXP n'a pas réussi à attirer suffisamment de membres. Deuxièmement, nous étudions BGP, le protocole de routage utilisé sur Internet, et en particulier s'il existe des mensonges BGP, c'est à dire si les routes par lesquelles les paquets circulent réellement sur Internet divergent des chemins que les systèmes autonomes (AS) annoncent. Nous trouvons effectivement des cas où les chemins ne correspondent pas. Enfin, nous étudions comment le trafic circule à l'intérieur des AS et nous nous concentrons sur la détection des détours d'acheminement, c'est-à-dire les cas où les itinéraires d'acheminement ne correspondent pas aux meilleurs itinéraires disponibles, selon le protocole de routage utilisé. Nous mettons ainsi en évidence des détours dans plusieurs AS.

Mots clés : BGP, IP-to-AS mapping, IXPs, Latin America, IGPs, Forwarding Detours, Load Balancing, Traffic Engineering, Multipath Routing, Scalability, Forwarding Information Base

Abstract

The objective of this thesis is to detect hidden broken pieces of the Internet. First, we study the deployment of Internet exchange points (IXPs) in Latin America and find that while some IXPs across the region have managed to proliferate, some countries have failed IXPs, i.e., no IXP at all, or the IXP has not succeeded to attract members. Second, we focus on the border gateway protocol (BGP), the routing protocol used on the Internet, and study whether ASes carry on BGP lies, i.e., if the forwarding routes through which packets actually flow on the Internet diverge from the AS-paths that ASes advertise on BGP. We find cases where the paths indeed mismatch. Finally, we study how traffic flows inside ASes and focus on the detection of forwarding detours, i.e., cases in which the forwarding routes do not match the best available routes, according to the internal gateway protocol (IGP) in use. We reveal such forwarding detours in multiple ASes.

Keywords : BGP, IP-to-AS mapping, IXPs, Latin America, IGPs, Forwarding Detours, Load Balancing, Traffic Engineering, Multipath Routing, Scalability, Forwarding Information Base

*The turtle is **slow**, **painfully** slow...*
*...but **wise**, it chooses its steps **thoughtfully**...*
*...the seek of **perfection** drives it, it will never settle with less...*
*...it does not need **motivation**, **recognition** nor **validation**, the words of **fools**.*

Acknowledgments

A journey of more than 3 years is coming to an end, at least in the paper. The things I have learned, and the title of doctor will carry on with me from now onwards. Wherever I go, whatever I do, I hope I will be able to work with the same passion as I have during my PhD. The future is to come, but now is rather the moment to express some gratitude.

I thank Cristel Pelsser, la directrice de ma thèse, for all the effort she put in helping me during this process. From the initial discussions while I was discovering the field, to her very precise and accurate feedback usually leading to the improvement of our articles. In addition, all the recommendation letters she wrote for me come to my mind: they allowed me to participate in two TMA PhD schools, extend my contract, etc. I was Cristel's first PhD student, and even though many bureaucratic requirements may have been initially unknown to her, she never hesitated in fighting the system side-by-side with me, always finding solutions to the enigmatic situations that every now and then came up. I will always remember that she also let me take her place in the AIMS-KISMET workshop, how she always tried to find opportunities to make me go present my work and make it more known. In addition, I appreciate how she is now helping me to find a path to continue my research. Lastly, I am very grateful for the speech she dedicated to me the day that I defended my PhD: it makes me happy to see that she values all the effort that I put in my work, and that she enjoyed working together all this time. Thank you for saying that publicly and for not saving words to express it.

I am very grateful to Pascal Merindol, my co-encadrant, for so many reasons that it is hard to put it into words, though I will still try. My best souvenirs of the PhD will always be the outstanding amount of endless meetings that we had to discuss our work. To me, the memorable intellectual fights we had and the way we pushed each other to the limits of their knowledge are what I hope every PhD student experiences with its advisors, i.e., research in its pure state. We were the "kings" of the "what if?", the key question in the manual of perfectionists, and the one that should also often appear in the mind of any (good) researcher. I also gave lessons at the university with Pascal, where again we gave our best, but this time to teach our students, even when the covid-era begun. Thanks for all what you taught me, your advice, your commitment to help, the passion with which you work. Thank you also for being such a good person, starting and finishing every email with a smiley, and for all the informal discussions we had about life. To conclude, I want to express

that the fact that you started calling me a researcher collegue long before my PhD defense was held meant a lot to me. Thank you for trusting my work, choices and capacity.

All in all, I also thank my two advisors because I appreciate that back in 2017, when they had to make the choice, they considered me the best candidate for this PhD, even when they knew that (i) I worked well according to a collegue, but they did not have a certain way to know who I really was, and; (ii) my background, though very solid, was from a different area. In any case, I thank them for being able to recognize my potential in the interview we had, and for having the courage to invest on me. Being chosen made me happy back then, and it still does, so I am grateful to them for that. Considering the work we have done together, I am convinced that they do not regret it neither.

Moving forward, I want to thank the complementary part of the jury in my PhD defense, i.e., Guillaume Urvoy-Keller, Benoit Donnet and Clémence Magnien, for unanimously deciding I that deserved the PhD title, as written in the report of my defense. I thank them for reading the manuscript, and for expressing that they felt that my work improved the state-of-the-art. To me it is nice that they also noticed how enthusiastic and didactic I am when I talk about my work. In particular, I am specially thankful to Guillaume, who wrote a report that helps to quantify the amount of work I did, and highlights the passion that guides the research I carry on. Finally, I thank Clémence for taking care of the extra work she had as president of the jury.

In addition, besides my advisors, I want to thank all the remaining co-authors of the publications I worked in, with a special mention for Esteban Carisimo and Valerio Persico. I have known Esteban for more than 10 years, and it makes me very happy that our PhDs have brought our paths closer, making us meet in Buenos Aires, Vienna and Paris. I particularly enjoy that he and I were able to shed some light on the status of the Internet in Latin America, a region deeply attached to our hearts. I would say that most of the work we have done with Esteban remains yet unpublished, and I hope that we will be able to honor our efforts in the future. On the other hand, I am happy to have met Valerio more than two years ago when he stayed as a research fellow in Strasbourg for some months. Thank you Valerio for being my touristic guide in Napoli and for making sure that I made it to all the good pizzerias, coffee and pasta places. I will always remember returning from the lab with you, and this view of the Vesuvio appearing in the back after one turn. The moments that Melanie and I spent with you and Marianna in Napoli and Barcelona will turn into, and in fact already are, anecdotes we will never forget. In conclusion, I think of both Esteban and Valerio as examples to follow, both as researchers and humans overall. I deeply thank the two of them for being so thorough in their work, and for making a difference in my daily life as a PhD student.

I want to thank the permanent people that make the Network Research Team in the ICube Laboratory for being nice co-workers. I wish we had been able to spend more time together over the last year, but the covid-19 reduced our chances. I will always remember the lunches at 11.45, the barbecue at the lab, the Christmas dinner in 2017 and 2018, and the nice talk we had in Academie de la Biere with Stéphane Cateloin and Julien Montavont. In particular, thanks to Fabrice Theoleyre for the multiple occasions where he helped me to deal with bureaucratic stuff, Guillaume

Schreiner for the technical support setting up VMs, Pierre David for the orange chocolate bars of each monday, and Thomas Noel for replying my emails in record time. A very special thank goes to Quentin Bramas, who was a main speaker in the seminar I organized in 2018, with whom I enjoyed a lot concurrently giving lessons, and with whom I look forward to carry on some research in future work.

Further, I want to thank the PhD students that were in the same working team as me chronologically in the period that goes from 2017 to 2021. Thanks Jean Romain Luttringer for the interesting research discussions, Philippe Pittoli for the very interesting talks about the meaning of life and the purpose of research, Loic Miller for the chess games, and the same for Jean-Philippe Abegg that, despite stealing my bench in the PhD students room when he started his PhD (I recognize that I was away in Barcelona at that time, so he did not know), also made me smile when he shared his knowledge about Argentina and talked to me about Fernet and Rodrigo. I thank Rodrigo Teles Hermeto for his help before and right after I arrived to Strasbourg, for being the delegated to prepare coffee at midday, for bringing board games to play on the pauses, for the Christmas day we spent together and with our wives in 2018, etc. To continue with the Brazilian side, I thank Renato Juacaba Neto for always bringing happiness with his characteristic laugh, for our jokes comprising AR » BR or the opposite (we both know the truth), for coming to visit me in Barcelona in 2019, for his special sense of humor, for always being present when needed as a friend, etc. I am grateful to Sebastián Lucas Sampayo and his wife that together gave the Argentinian touch that made me feel closer to home during my stay in France, for all the nice moments we spent together in bars and restaurants, and for all the help, advice and information you always provided me. I am also grateful to Andreas Guillot since he was always trying to help with any problem I might have had, hosting the gatherings of the PhD students in which we had such a fun and were able to enjoy pure friendship, visiting me in Barcelona in 2019, and also for essentially listening to me when I needed someone to talk with. I also want to thank Amine Falek because, despite the fact that he is difficult to reach, he is a good friend and as perfectionist as I am; he always listened when I explained him the problems I was trying to solve in my PhD and gave me useful feedback. I hope I will work with him in the future, and also play some guitar and chess as we did before.

In addition, I want to thank Georgios Z. Papadopoulos, my former advisor in the research lab in Rennes who recommended me to Cristel and Pascal as a good candidate for the PhD. I also want to thank Renzo E. Navas for making me feel like if I was in Argentina while I stayed in Rennes, for still being present during my PhD despite the distance, and for the invaluable feedback he gave me while I was making the final arrangements in the slides I presented on my defense. I want to thank Andra Lutu for taking me as an intern in Telefonica I+D for 3 months in Barcelona, an experience I will always remember. Similarly, I thank Antonio Pescape for receiving me as a research fellow in his lab at the University of Napoli Federico II for almost during 1 month.

There are also special friends I made over these years that I want to thank for making my life happier during my PhD. I have many positive things to say about all of them, but I will be short because, as good friends always do, they already know what I think about them. I do not have enough words to express

my love towards Nadja Groysbeck, Antonella Zerpa, Odnan Ref Sanchez and Ana Alice Torres. The same applies to the Serbian team composed by Jelica Vasiljević, Mihailo Obrenovic and his wife Aleksandra. Thanks for all the friends I made in Spain during my internship in Telefonica: Mariona Caros Roca, Gabriele Castellano, María Lara Gauder, Guillermo Cámbara. I also enjoyed meeting with Santiago Pascual, Marcos Paulucci, Patricio Pavón, Federico Longhi and Lynelle Sigona. I am also very grateful to my friends in Argentina, that sent me their support while I was away: Gonzalo J. Figueroa, Cecilia Osorio, Mathias E. Garcia, Patricio Olaberría, Philippe Clavier.

I also want to thank my family, and that of my wife, for sending their love to both of us across the ocean, from Argentina straight to France, and also for embracing us every time we went back to visit them. In particular, I enjoyed each and every talk I shared with my grandparents and my aunt. I will always be grateful to my parents, that gave me everything I could have asked for and more, without whom I would not have been able to come all along this way. I miss them all every day of my life, but the effort has paid back, and it will continue to do so.

Finally, the person I want to thank the most is my wife, Melanie Belen Castagno. She is an extraordinary person that, from the big heart that she has, irradiates love 24/7. Through these more than 3 years, I have done the research you will soon discover in the manuscript, but it is actually her that has done the hard job. She has had extraordinary patience with me, she has listened to my complaints, encouraged me when I felt out of energies, helped me to deal with my anxiety close to the deadlines, heard me telling her every day about a new pain that I had, provided me with her company when I felt sad. In short, it is her love that has been next to me all this time, and without which my PhD would not have been the same. I will always be grateful to her. I hope one day I will be able to be there for her, the same way she has been there for me.

Summary

The Internet is an interconnection of independent networks known as Autonomous Systems (ASes). Given that ASes are built on top of hardware and software, and that network operators, i.e., humans, manage ASes, then the Internet is constrained to some limitations. For example, humans are error-prone and eventually take arbitrary decisions, enterprises are generally greedy from a revenue point of view, and hardware may fail and require maintenance or replacement. All these factors may lead the Internet to have broken pieces, i.e., malfunctioning components, networks facing limitations and even selfish networks prioritizing their own revenue rather than the better performance of the Internet.

The objective of this thesis is to detect broken pieces of the Internet. First, we study the deployment of Internet exchange points (IXPs) in Latin America, a region that has previously received little attention in Internet studies. We construct the most comprehensive dataset of the status of the Internet in Latin America and characterize the AS ecosystem in the region. We find that while some IXPs across Latin America have managed to proliferate, some countries have **failed IXPs**, i.e., no IXP at all, or the IXP has not succeeded to attract members. Second, we focus on the border gateway protocol (BGP), the routing protocol used on the Internet, and study whether ASes carry on **BGP lies**, i.e., if the forwarding routes through which packets actually flow on the Internet diverge from the AS-paths that ASes advertise on BGP. In practice, performing this comparison is complex since besides the multiple levels at which data needs to be synchronized, missing hops, third-party addresses and AS siblings may introduce errors by wrongly triggering the detection of BGP lies. In particular we develop a methodology allowing to filter this noise, and run measurements in the wild. We find cases where after sanitizing the dataset with our framework, paths still mismatch. Finally, we study how traffic flows inside ASes and focus on the detection of **forwarding detours**, i.e., cases in which the forwarding routes do not match the best available routes, according to the internal gateway protocol (IGP) in use. We develop a formalism explaining when forwarding detours occur, and implement a detector allowing to differentiate forwarding detours from load balancing and traffic engineering techniques. We run measurements with our detector and find detours in multiple ASes with a remarkable binary pattern such that transit traffic traversing between two border routers of an AS either never detours, or always does.

List of publications during the PhD

Journals

- **Julián M. Del Fiore**, Valerio Persico, Pascal Merindol, Cristel Pelsser and Antonio Pescapè. *The Art of Detecting Forwarding Detours*, to appear in IEEE Transactions on Network and Service Management (IEEE TNSM) 2021.
- Esteban Carisimo, **Julián M. Del Fiore**, Diego Dujovne, Cristel Pelsser, and J. Ignacio Alvarez-Hamelin. 2020. *A first look at the Latin American IXPs*, in SIGCOMM Comput. Commun. Rev. 50, 1 (January 2020), 18–24.

Conferences

- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *Filtering the Noise to Reveal Inter-Domain Lies*, in 2019 Network Traffic Measurement and Analysis Conference (TMA), pages 17–24, 2019, IEEE.

Posters

- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *Routing Inconsistencies at the FIB level*, in 2019 Network Traffic Measurement and Analysis Conference (TMA), IEEE.
- Esteban Carisimo, **Julián M. Del Fiore**, Diego Dujovne, Cristel Pelsser, J. Ignacio Alvarez-Hamelin, *Country-level influence of IXPs in Latin America*, in Latin American Student Workshop on Data Communication Networks (LANCOMM) 2019.
- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *A BGP-lying Tale: Stop Blamming the Mapping*, in 2018 Network Traffic Measurement and Analysis Conference (TMA), IEEE.

Contents

Chapter 1

Research Questions and State-of-the-Art

The Internet appears as an infallible system that never fails, however, this is not the case. Actually, the Internet is simply an interconnection of independent networks, known as Autonomous Systems (ASes). Given that ASes are built on top of hardware and software, and that network operators, i.e., humans, manage ASes, then the Internet is constrained to some limitations. First, humans not only are naturally error-prone, but also sometimes take arbitrary decisions that are not always the best. Moreover, for the same problem, different people may consider discrepant constraints as the most relevant, and thus propose diverging solutions as the best one, contributing another level of randomness surrounding the behavior of the Internet. In addition, from a business point of view, we could argue that enterprises are generally greedy, thus turning the Internet into a profit-driven ecosystem. Besides this, human-built systems are usually not perfect: they tend to comprise modules that may fail and require maintenance, or that after a given time may become obsolete and need replacement. All these factors may lead the Internet to have broken pieces, i.e., malfunctioning components, networks facing limitations and even selfish networks prioritizing their own revenue rather than the better performance of the Internet. The objective of this thesis is to detect problems such as these. This task is challeniging since the phenomena we want to uncover may be hidden anywhere on the Internet, which counts with approximately 70K ASes as of November 2020.

1.1 Internet Exchange Points in Latin America

The first component of the Internet we study on this thesis are Internet exchange points (IXPs), the interconnection facilities commonly used by ASes. The structure of the Internet, i.e., how ASes establish connections with each other, was largely modified by the irruption of IXPs in the 2000s [11]. IXPs allow ASes to establish connections at a larger scale, and to produce monetary savings. However, the popularity of IXPs varies across regions. In some cases, countries may have **failed IXPs**, i.e., no IXP at all, or an IXP that has not succeeded to attract members. Whereas there exist large IXPs in Europe, such as DE-CIX, LINX

and AMS-IX that have been subject of study [**ager2012anatomy**], there are no reports showing a similar story of success in North America. Recent studies have focused on the role of IXPs in the African AS ecosystem [**fanou2017investigating**, **fanou2015diversity**, **fanou2017reshaping**]. In other regions, in contrast, little is known about IXPs and even about the Internet as a whole.

In particular, the case of Latin America (LatAm), corresponding with the *Regional Internet Registry* (RIR) named LACNIC, is an interesting case of study. LatAm covers 20 million km^2 [**worldbank1**] and comprises 20 countries: right after North America, it has the largest urban population rate (80%) [**worldbank2**]. Moreover, LatAm is home of 652 million people [**un1**] and has three out of the four largest metropolitan areas in the Americas (Sao Paulo, Mexico City and Buenos Aires with populations of 21.3M, 21.2M and 15.3M habitants respectively) [**un2**]. The region also has appealing numbers regarding to Internet considering it contributes to the global AS ecosystem with 14.5% (9,988/68,912) of the active ASes. Furthermore, 6458 ASNs have been delegated by NIC.br (Brazilian national Internet registry) to Brazilian-based organizations. Between 2005-2015, LatAm experienced significant progress in fixed and mobile penetration rates, reaching 40.57% and 57.41% of the population, respectively [**katz2018accelerating**]. Moreover, several countries of the region have recently benefited from the creation of domestic IXPs [**galperin2016localizing**].

Despite the progressing development of the Internet in LatAm, the shape of the Latin American network remains relatively unmapped. This encourages us to explore its interconnection and structure. Given that IXPs contributed to flatten the Internet in the 2000s, it is natural to wonder if 20 years later, these peering infrastructures are also benefiting developing countries, many with much larger surface, to embrace the Internet. This brings us to our first research question.

Research Question

Have all IXPs in LatAm managed to proliferate or are there failed IXPs? If some have failed, why?

A priori, LatAm has remained quite unexplored, presumably due to a historical scarcity of representative Internet data. For example, the footprint of RIPE Atlas and Ark CAIDA in LatAm is composed of 285 out of 11,142 (2.56%), and 12 out of 190 (6.32%) probes, respectively. The numbers decrease considering IPv6: just 2.22% (101/4,556) of RIPE's IPv6-capable probes are located in Latin America. On the other hand, it is known that the lack of BGP data allows to draw a fairly incomplete representation of AS ecosystems [**lakhina2003sampling**]. In that sense, Routeviews [1] and RIPE RIS [2] have only respectively deployed two and one BGP data collectors in the region, two being redundant since they are placed at the same Brazilian IXP in Sao Paulo.

[1] `http://www.routeviews.org/`

[2] `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`

Despite the aforementioned limitations, some Internet studies have focused on this region. Berenguer *et al.* [**berenguer2016hidden**] applied graph-theoretical metrics to evaluate dataset augmentation when routes collected from local looking glasses are added to RIPE RIS and RouteViews BGP dumps. Brito *et al.* [**brito2016dissecting**] gathered one BGP dump per each looking glass co-located in each regional IXP of IX.br, the Brazilian IXP network, and then compared them with IXPs from other regions in terms of connected networks and peering policies prevalence. A complementary study of the same authors included an analysis of IPv6 deployment [**brito2016analysis**], however, it is limited to IPv6 prefix size and the number of IPv6 entries in routing tables. Muller *et al.* [**muller2019challenges**] relied on sFlow data gathered at a regional IXP of IX.br to infer spoofed traffic traversing the IXP. Formoso *et al.* [**formoso2016looking**] relied on RIPE Atlas probes deployed in LatAm to measure inter-country latency, inferring asymmetric paths and poorly interconnected countries.

In particular, Chapter 3 studies the deployment of IXPs in LatAm, and shows the first broken piece of the Internet we find. Indeed, IXPs are a story of either success or failure depending on the country considered in Latin America. While Argentina, Brazil and Chile count with IXPs that have managed to proliferate, other countries have failed IXPs. We dive into the reasons of why some IXPs are able to gather a large number of members that announce multiple IP addresses while others not. We find a negative correlation between success of IXPs and the presence of monopolistic ASes concentrating the IPv4 address space delegated to countries. In addition, since LatAm has never been closely analyzed, we take the opportunity and also characterize the AS ecosystem in the region. We see that IXPs in LatAm, similar to others in developing regions, are mainly populated by domestic or regional players, whereas those internationally renown in Europe rather behave as international hubs.

1.2 Seeking for BGP Lies

The second element of the Internet we look at in this thesis is the border gateway protocol (BGP), the routing protocol used on the Internet. BGP dictates how ASes exchange reachability information concerning the IP prefixes each of them owns. With BGP, each AS announces the prefixes it owns to its neigbouring ASes, and in turn these relay the message to other ASes. In this process, the exchanged routing messages keep track of the AS-path that was followed, i.e., a list of the ASes, from the first to the last, that advertised the prefix. In this thesis, we refer to the AS-path as the control path (CP), since it is built on the control plane of BGP. On the other hand, we refer to the set of ASes that packets actually traverse towards their destinations as data paths (DPs), since this occurs at the data plane of BGP. An analogy of an AS advertising a prefix with a given CP to a neighbouring AS, is the offer of a contract. More precisely, the BGP announcement plays the role of the contract, and the service that gets offered is that, to reach a given prefix, the DP will replicate the ordered set of ASes expressed in the CP. Hence, DPs are expected to match the CPs advertised for all prefixes.

The underlying assumption that CPs match DPs for all prefixes advertised in BGP is not trivial to verify: the current troubleshooting tools, e.g. traceroute,

usually allow to recover IP-paths, but not the forwarding AS-level route that was followed. As a consequence, the implicit trust that ASes advertise the paths they use for packet forwarding may be misplaced. Network operators may manipulate CPs [27] and DPs [28, 29] potentially leading them to mismatch. Whenever the CP and DP for a prefix mismatch, we say that a BGP lie occurred. We choose this name because if the paths mismatch, then the ASes that advertise the CP are lying, since the DP differs. Note that this nomenclature applies irrespectively of whether CPs and/or DPs are tweaked, or if this results from a deliberate or unintended behavior.

The objective behind BGP lies may be multiple. An AS may try to redirect and intercept traffic, or hinder its tracking with consequences on the ability to troubleshoot connectivity issues. Moreover, BGP lies may lead to the violation of agreements between adjacent ASes, with potential subsequent legal retaliation. These lies may be deliberate to obfuscate traffic interceptions or be driven by economical interests, e.g. attracting traffic by promising interesting routes but using cheaper alternatives. On the other hand, they can result from incongruent logical and physical topologies, in particular when BGP sessions are not set on simple point-to-point inter-domain links. Others may be rooted in technical limitations, such as limited memory on the routers hampering the storage of the full routing table, i.e. resulting in a partial-FIB. In conclusion, BGP lies generate a broken piece of the Internet.

We set our goal to detect BGP lies, however, carrying out this task requires addressing a considerable practical challenge: the CPs and DPs that need to be gathered with measurements and the IP-to-AS mapping tools usually used to transform the DPs from IP-level to AS-level paths are noisy, hence, BGP lies may be misinterpreted as noise, or vice versa. Therefore, to draw representative conclusions, the developed framework should filter the noise affecting the measurements. This motivates our second research question.

Research Question

Can we develop a framework that, filtering the noise affecting the comparison of CPs and DPs, allows to quantify the daily rate of BGP lies that occur on the Internet?

There exist many papers that have focused on the comparison of CPs and DPs, and in characterizing different souces of noise affecting this task. Mao *et al.* [70] find that IXPs, AS siblings, and ASes announcing IP prefixes for which they are not the real originating AS (OAS) are predominant causes for mismatches among CPs and DPs. In a follow-up work [71], they develop a systematic approach to correct inaccurate IP-to-AS mappings by reassigning the OAS of prefixes. Hyun et al. [72] also analyze the discrepancies between CPs and DPs and conclude that insertions of IXPs in the DPs and of ASes under the same ownership are the main cause that leads to mismatches. However, in their study, incomplete traces are discarded and the comparison does not rely on the latest BGP updates, i.e. CPs and DPs are not well time-synchronized. Zhang et al. [73] extract the mismatching fragments

of CPs and DPs and show that the main pitfall of using traceroute in AS-level topology measurements originates from the appearance of IP addresses assigned from AS neighbors. However, their measurement platform suffers from the inability to ensure that the data and control plane vantage points are co-located, i.e., that the DPs exit the AS (or at least traverse) the BGP speaker sharing the CPs. On the other hand, Hyun et al. [74] introduced the concept of third-party addresses (TPAs), i.e., IP address appearing in traceroute that be mapped to an AS that is off-path. Their study concludes that TPAs are not common and that they do not distort AS maps significantly. In addition, according to the authors, finding multiple TPAs in a row mapping to the same AS is unlikely, although possible. A later analysis of Marchetta et al. [75] using IP timestamp options states the contrary. They find that consecutive TPAs are common, and may even entirely hide an AS from an AS-level path. However, a subsequent study from Luckie et al. [76] reports that most observed IPs in traceroute traces are from in-bound interfaces, thus on-path. They argue that techniques using IP timestamps are not reliable to detect TPAs. Ahmed et. al [77] propose an offline method that tags up to two IPs that appear in a row as possible TPAs if they introduce an AS that either violates valley-free paths or translate into new AS relationships. Further work concerning detection of TPAs for correctly determining AS boundaries include bdrmap [78], MAP-IT [79] and bdrmapIT [80]. In the case of bdrmap, inter-AS link interface addresses between a network with a traceroute vantage point and directly connected networks are inferred relying on alias resolution probing techniques, and AS relationship inferences. On the other hand, MAP-IT attempts to achieve the same for all connected ASes based on traceroute results collected from multiple vantage points distributed in different ASes. Finally, bdrmapIT combine the previous two, improving the inference of router-ownership and identification of links between ASes.

In Chapter 4 we model how ASes may introduce BGP lies and we propose a methodology to detect them. In particular, we present a framework allowing to filter noise, i.e., mapping inaccuracies introduced by AS siblings, TPAs or IXPs and missing hops, that may affect the comparison between CPs adverised in BGP and the DPs that packets follow on the Internet. In particular, our methodology relies on heuristics to estimate whether noise may be affecting the paths we collect, and then attempts to correct them. Our framework is modular, i.e., the user can select among multiple filters that vary how they estimate what results from noise or not, and thus allow to implement different noise-filtering models. We run long-term measurements on the Internet and sanitize the dataset with our framework. Our results show that, even relying on the most conservative noise-filtering model, some mismatches between the CPs and DPs we collect still remain, likely representing BGP lies. Compared to the literature, our study not only deploys more vantage points, but also goes beyond what had previously been done by providing results based on daily-analysis. Comparing the results with basic IP-to-AS mapping tools and with our framework, we see that in the vantage points where our framework is effective to filter the noise, eliminating numerous false positives, the results are stable in time. On the other hand, when both methods output a high and comparable number of inferred discrepant paths, we see that results have a larger variation over time. While most of the related work essentially blames the IP-to-AS mapping for the observed discrepancies between CPs and DPs, our work relies on conservative

heuristics that remove the noise in the measurements and the mapping errors, attempting to minimize false positives in the detection of BGP lies. In other words, different to previous studies, our aim is to detect "real" BGP lies, hence our framework is designed to provide a lower bound of mismatches between CPs and DPs. The mismatches we find after applying our filters show that the IP-to-AS mapping is not the only culprit for them.

1.3 Modeling and Detection of Forwarding Detours

The last component of the Internet we analyze are internal gateway protocols (IGPs), the routing protocols that ASes use inside their networks. IGPs characterize for resulting in a forwarding scheme such that traffic traversing ASes flows through best paths that minimize the distance or cost according to a metric of choice.

In this case, the broken piece we are interested in are *forwarding detours*, i.e., cases where traffic flows through forwarding routes that divert or diverge from the expected best IGP paths. Contrary to hot-potato routing, FDs increase the IGP distance required to traverse an AS and arguably result in waste of resource utilization inside the network. Attempting to suppress FDs, network operators may implement tunneling techniques. However, these mechanisms only allow to FDs within each tunnel/segment (for BGP-free core routers in particular) but may fail to do so between endpoints of an AS.

In particular, FDs may result from side effects of scalability workarounds. Indeed, the full Internet feed, reaching ∼867K prefixes as of February 2021, has been growing at ≈50K prefixes/year over the last 10 years. The sustained increase in the number of prefixes advertised on the BGP has led ASes to exchange more update messages [31, 32, 33], 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 [**zhao2010routing**, 34, 35, 36]. 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 [37]. In addition, partial iBGP dissemination relying on route-reflector hierarchies may also boost scalability [38]. 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 [39]. In addition, memory-constrained routers may aggregate routes to limit the number of FIB entries [40]. Other type of workarounds consist in storing a partial-FIB [41], 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 [42]. While the aforementioned techniques may look effective at first glance, ASes relying on them may suffer from forwarding detours. This may happen when routers along a route choose different exit ASBRs for the same prefix.

In addition, besides side effects of scalability workarounds, misconfigurations and bugs in router's software, such as BGP zombies [**fontugne2019bgp**], may also create FDs. Consequently, network operators may ignore FDs occur on their AS, and provide degraded performance to customer ASes. This brings us to a third research question.

Research Question

Can we formally model the root causes that produce forwarding detours, and design a methodology to detect them?

Concerning related work, back in 2004, when full FIBs only had 100K entries, compared to more than 800K nowadays, Bu et al. [81] 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 [40] and sometimes redirecting traffic to more knowledgable routers [41]. The growth of the FIB indeed favors the use of workarounds like partial-FIBs and default routes, that may lead to FDs.

On the other hand, 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. [82] pinpoint AS-wide deflections, though their goal is to detect path diversity on the Internet. They conclude that intra-domain load balancing was not well deployed at the time. Secci et al. [83] 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 BGP multi-exit discriminator (MED) attribute. Agarwal et al. [84] analyze BGP routing changes as deflections. They try to detect intra-domain deflections to build accurate traffic matrices. Bush et al. [85] 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.

Finally, there have been multiple studies concerning the multi-path routing patterns that load balancing generates. This relates to FDs since the simultaneous existence of prefixes subject and not subject to FDs also generates multi-path routing patterns. Augustin et al. [22] 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. [50]. Based on the principles of Paris-traceroute, Augustin et al. [65] develop the multi-path 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 [66, 67]. Veitch et al. [56] refine the stopping points of the MDA to bound the failure probability of full multi-path discovery. Vermeulen et al. [68] 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 [57], a system able to produce Internet-wide multi-path topology maps in less than 3-day long snapshots [57]. Diamond-Miner implements the MDA with a stateless probing fashion relying on Yarrp [**yarrp-imc16**], a randomized high-speed prober. Almeida et al. [58] generalize the MDA and propose the Multi-path Classification Algorithm (MCA). In general, all these works show that per-flow and per-destination load balancing are the most widespread load balancing flavors. Except for Diamond Miner, they all run measurement campaigns in the order of 10K and no more than 70K

destination IP addresses from at most 32 vantage points.

In particular, Chapter 5 describes why detecting FDs is complex, and shows the tool we develop to tackle this objective. While the related work has focused on the effect of one particular cause that could potentially create FDs, our solution allows for a systematic analysis that can be applied to detect FDs inside ASes of any kind. The methodology we propose closely analyzes how traffic flows inside ASes and does not require privileged knowledge concerning the networks being analyzed, e.g. knowing the IGP metric in use, to conclude whether FDs occur not. Note that detecting FDs is a particularly challenging task under these circumstances, i.e., when no assumption is made about the IGP metric that ASes use, since forwarding detours and best IGP paths in the end are simply forwarding routes. Moreover, similar to FDs, techniques such as load balancing and traffic engineering also generate multi-path routing patterns. Opposed to forwarding detours, these are still considered optimal in the sense of the IGP cost or for the specific needs of the AS deploying them, respectively. Our analysis may complement the studies focusing on load balancing since we also contemplate per-prefix load balancing, a flavor not discussed in the literature. Our detector not only addresses all these challenges, but also uses novel prefix-grouping step, that may allow to decrease the probing cost of load balancing discovery campaigns, and to discover additional next-hops of per-destination load balancers. We test our FD-detector in the wild and find forwarding detours in multiple ASes, with a remarkable binary pattern in which transit traffic traversing between two border routers of an AS either never detours, or always does. Finally, the concept of forwarding detours focuses on the consequences, i.e., on paths differing from best IGP paths, but does not tell anything on how these are generated, i.e., on the root causes. To shed light on this previously unexplored topic, we develop a formalism around forwarding detours allowing to formally describe the phenomenons that lead to the occurrence of forwarding detours.

Chapter 2

Background

Contents

In this chapter we interpret background knowledge and condense it into the set of basic topics on top of which the research of this thesis is built, particularly providing complementary knowledge and insights in Sec. 2.2.2 and 2.3.3. First, Sec. 2.1 presents how the Internet works, i.e., how ASes are able to communicate with BGP. In addition, we study IXPs, the peering facilities allowing multiple ASes to peer in a single location. Then, Sec. 2.2 zooms into ASes and studies the characteristics of the routing protocols that networks operators may deploy for intra-domain routing, i.e. IGPs. Moreover, we explain how load balancing and traffic engineering techniques can be deployed to modify the forwarding routes used inside ASes. Lastly, Sec. 2.3 focuses on traceroute, the main troubleshooting and debugging tool allowing to discover the routes that packets traverse on the Internet. Moreover, we shou how the output of traceroute, lists of traversed IP addresses, can be translated into AS-level forwarding paths with the use of an IP-to-AS mapping. We study the standard version of traceroute and the Paris implementation. Finally, we present the multi-path discovery algorithm that allows to convert Paris traceroute into a tool uncovering multi-path routing, essentially load balanced paths, inside ASes.

Expression	Definition
Internet	**Interconnection** of autonomous systems
Autonomous system	**Independent** network, e.g. belonging to an ISP

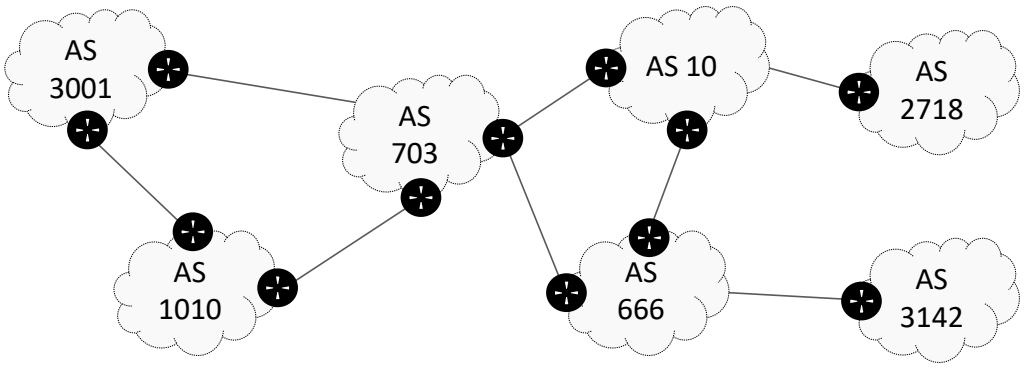


Figure 2.1: Simplified representation of the Internet. The Internet is an interconnection of ASes that establish links among their ASBRs. Each of these independent networks is identified with a unique ASN.

2.1 Internet

In Chapter 3 and 4 we study the success of IXPs in Latin America and the occurrence of BGP lies in the wild, respectively. Both topics require understanding how the Internet works. To shed light on this, Sec. 2.1.1 explains BGP and Sec. 2.1.2 presents IXPs, the routing protocol and the peering facilities commonly used on the Internet, respectively.

2.1.1 BGP

The Internet is an interconnection of independent networks, known as Autonomous Systems (ASes). To internconnect, ASes establish links among AS border routers (ASBRs). In practice, ASes may be Internet Service Providers (ISPs), networks that belong to private companies, universities, governmental agencies, etc. On the Internet lingo, each AS is considered a domain, thus, ASes have their own intra-domain network. In addition, to differentiate among ASes, each is identified by an AS number (ASN). A simplified representation of the Internet is shown in Fig. 2.1.

The Internet has been running over more than 20 years, and has been subject to an unsupervised growing process in which ASes willing to participate in the Internet simply have to connect to other domains, i.e., ASes. In these cases, ASes are said to peer. To communicate, ASes rely on BGP [3], commonly accepted as the de-facto inter-domain routing protocol.

BGP is a path-vector routing protocol where routing messages propagate across ASes. More specifically, BGP routing messages are exchanged between routers that are said to be BGP speakers. In general, ASBRs are BGP speakers. As an Origin AS (OAS), each AS announces the IP prefixes it owns to its peering ASes, and in turn, these re-advertise these prefixes to their own neighboring ASes. In this

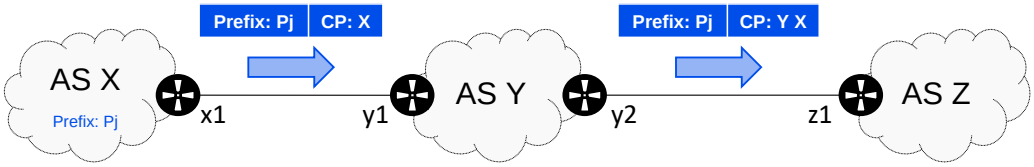


Figure 2.2: Example of the basic operation of BGP. AS X announces prefix P_j as the OAS, and the prefix is further advertised by AS Y that updates the CP including itself in the path.

process, routing messages get updated, e.g. the path they follow is tracked in a field known as BGP AS-path, and often referred to as control path (CP) in this thesis. In general, if an ASBR finds a loop in the CP, i.e., sees the ASN of the AS to which it belongs already in the CP, then the message is dropped. This basic operation of BGP is exemplified in Fig. 2.2.

As BGP speakers receive multiple announcements concerning the same prefix, they select the best route running a BGP decision process. This selection mechanism takes into account the value of 7 attributes, in decreasing order of importance, until a tie-breaker is found [3]. Among these, the local preference is the most relevant one. ASes tune the value of the local preference to express preference of some routes depending on the peer that announced them the prefix. The higher this metric is, the more appealing the route is considered. Following the local preference, the length of the control path is the attribute that matters the most. In this case, routes are only discriminated based on the length of the path, the shortest ones being the preferred ones. In some cases, ASes use AS prepending, adding multiple times their ASN to the CPs, in order to influence the likelihood that they may end up being chosen as best paths. An exhaustive list of the remaining attributes can be found in [3]. Among them, we highlight that the IGP cost appears as one of such attributes, coupling BGP with intra-domain routing protocols. This compels with hot-potato routing in which the potato that burns, i.e., transit traffic traversing ASes, flows through best paths and exits the AS as soon as possible, according to a IGP metric of choice.

BGP speakers run two different sessions, namely external-BGP (eBGP) and internal-BGP (iBGP), as illustrated in Fig 2.3. In particular, eBGP and iBGP concern the exchange of information across ASBRs of different ASes and BGP speakers within the same AS, respectively. With eBGP, each ASBR in an AS announces to each peering ASBR belonging to another AS only the best path to each prefix. Thorough iBGP, the situation replicates between the BGP speakers of an AS. This allows each BGP speaker to find, among all routes considered the best by the different BGP speakers within the AS, the overall best route.

BGP speakers maintain a BGP routing table called BGP Routing Information Base (RIB). As routing messages are received, routers populate the RIB. There exist different fields that are completed for each entry: the prefix that was advertised in BGP, the BGP next-hop, i.e., the ASBR that announced the prefix, and the attribute values included in the message in which the prefix was advertised. For each prefix, the overall best path is chosen relying on the aforementioned BGP decision process.

Session	Acronym	Description
Internal	iBGP (-)	Run among BGP speakers **within** the same AS
External	eBGP (-)	Run among ASBRs in **different** ASes

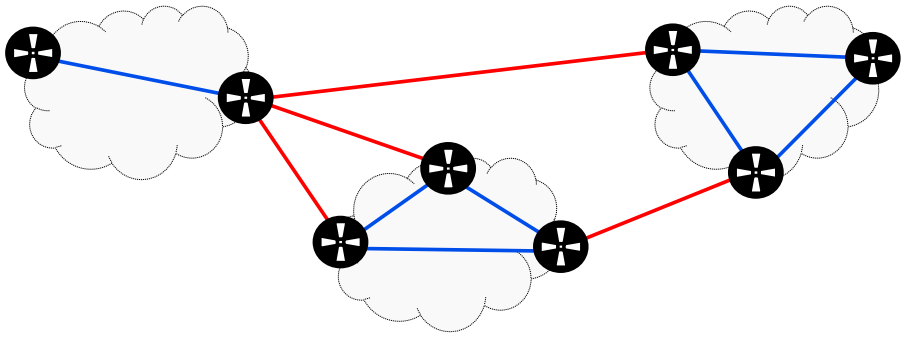


Figure 2.3: BGP sessions: external BGP (eBGP) and internal BGP (iBGP). The first are run among ASBRs in different ASes while the latter between BGP speakers in the same AS.

A file containing the RIB of a BGP speaker at a given point in time is usually referred to as BGP snapshot, BGP table or BGP dump among others, and usually used as a source of BGP data by researchers. There exist BGP data collection projects such as Routeviews, RIPE RIS, PCH, Isolario Project, etc.[1]. In general, these projects make publicly available the results of passive measurements, i.e., measurements that comprise dumping BGP data, in many cases aggregating BGP tables of multiple BGP speakers. According to the BGP decision process, the chosen overall best route may differ from BGP speaker to BGP speaker. As a consequence, the BGP data that can be collected from one to another may vary. This aspect is critical for the study we carry on in Chapter 4 since the analysis requires having a traceroute vantage point from which BGP data needs to be also available.

As a routing protocol, BGP is said to have two planes. The process we have described, concerning the construction of routing knowledge, conforms the control plane. This plane dictates the paths that should be reflected when packets exit an AS via a given ASBR, procedure that represents the data plane. In general, traffic traverses a ordered set of ASes until the destination is reached. We refer to these inter-domain forwarding paths as data paths (DPs). In other words, whereas the control plane determines best paths for each prefix, or the CPs, the data plane relies on this information to forward packets, which then flow through DPs. In particular, in Chapter 4 we study whether DPs and CPs usually match in practice, and find cases where this does not hold, i.e., where BGP lies occur. This topic could be clustered among others dealing with security in BGP such as distributed denial of service (DDoS) attacks [**mirkovic2004taxonomy**, **zargar2013survey**], BGP hijacks [**zhang2007practical**, **testart2019profiling**], etc. More on this can be found in [**butler2009survey**]. Even though there exist a secure version extending BGP, namely BGPsec [**rfc8205**], it has never been really adopted. On the

[1]Routeviews: `www.routeviews.org/routeviews/`; RIPE RIS: `https://www.ripe.net`; PCH: `https://www.pch.net/`; Isolario Project: `https://www.isolario.it/`

other hand, the Resource Public Key Infrastructure (RPKI) [**rfc6480**], allowing to validate the OAS of prefixes, is nowadays becoming more extensively used.[2]. Arguably, this partially occurs since RPKI is central in in the mutually agreed norms for routing security (MANRS). [3]

Another important aspect of BGP is that it allows ASes to apply filtering policies, and choose to which ASes they announce the prefixes they learn. In general, this depends on the business relationships that ASes establish [4, 5]. These relationships are dictated by who pays to whom when traffic gets exchanged: an AS pays its provider ASes, peers freely with sibling ASes and peer ASes, and gets paid by customer ASes. Hence, when choosing among available routes for the same prefix, the order of preference grows from last to first, and ASes set up the local preference accordingly. In other words, ASes apply import policies in which an AS chooses paths:

- advertised by customers as priority since this generates revenue;
- learned via peer-to-peer links that have a shared-cost, if no customer offers transit and;
- involving links to providers where the AS has to pay as the last option.

These policies, aiming to maximize the revenue, partially explain why BGPsec did not succeed. Indeed, BGPsec requires prioritizing security above local preference in the BGP decision process to be effective against routing attacks [**lychev2013bgp**], something that numerous network operators are not willing to do [**gill2013survey**]. In addition, when advertising prefixes, ASes apply export policies making sure control paths are valley-free, meaning that they do not provide transit for free to providers or peers. In practice, ASes enforce this by:

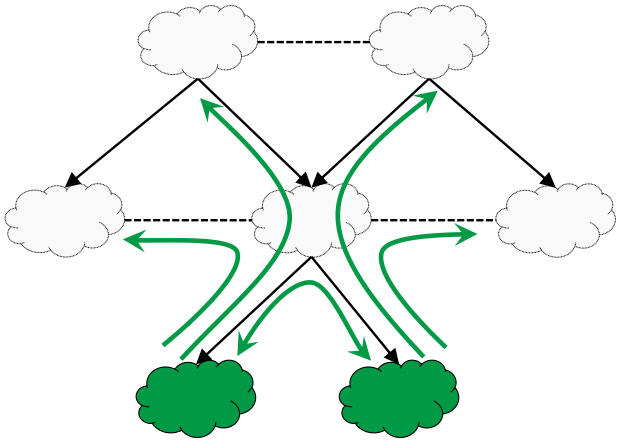
- advertising all paths they learn to customers and siblings;
- announcing to providers only those paths learnt via customers or siblings, and;
- filtering paths learnt via providers when advertising paths to peers.

These rules are usually referred to as Gao-Rexford policies. This study, and in particular the proposed rules, form part of research concerning BGP convergence or BGP safety, i.e., the conditions that guarantee that BGP will eventually converge to a stable routing outcome [**griffin1999analysis**, **griffin2002stable**, **gao2001inherently**, **griffin2003design**, 26]. As shown in Fig. 2.4, the Gao-Rexford rules imply that:

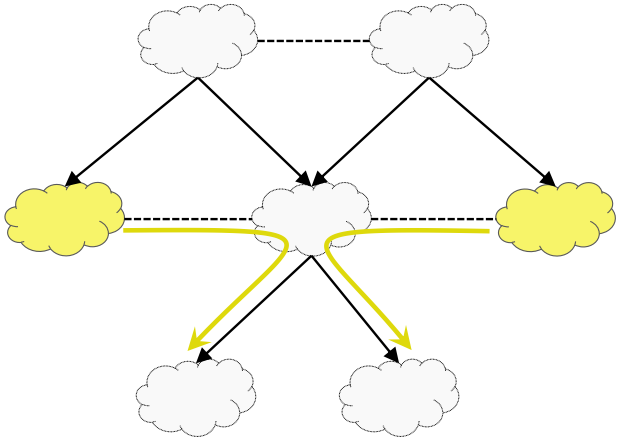
- prefixes learned via customers and siblings are advertised to all ASes (Fig. 2.4a);
- prefixes learned via peers and providers are only announced to customers and siblings (Fig. 2.4b and 2.4c, respectively).

Though these policies are still considered the norm, studies have reported on cases associated with the application of more complex policies [7, 8, 9]. As we show in Chapter 4, BGP lies may relate to ASes attempting to avoid paying for using customer-to-provider links by sending traffic towards peer-to-peer links.

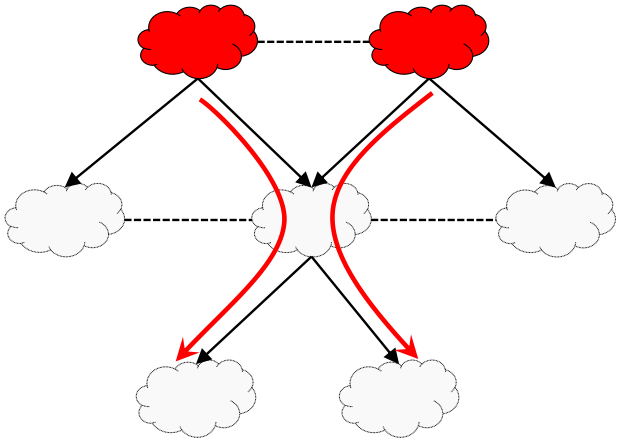
[2]See `https://rpki-monitor.antd.nist.gov/` and `https://rov.rpki.net/`
[3]https://www.manrs.org/



(a) Customers to everyone

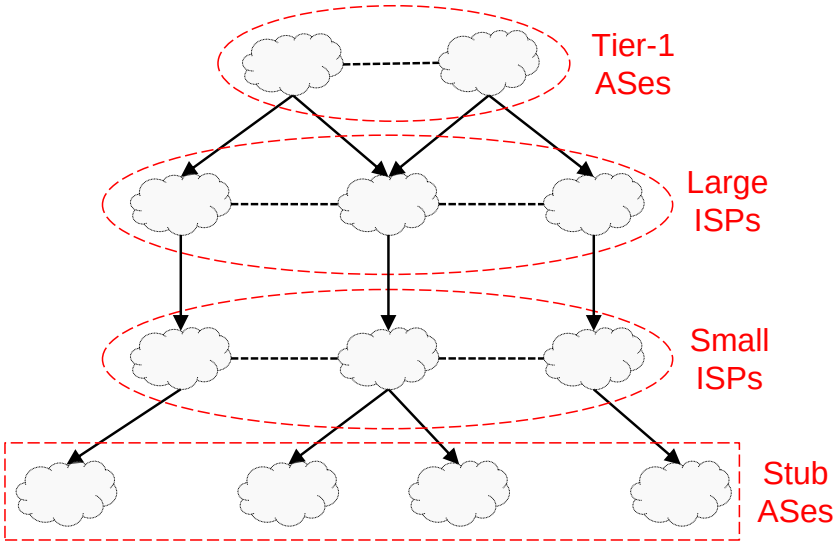


(b) Peers to customers

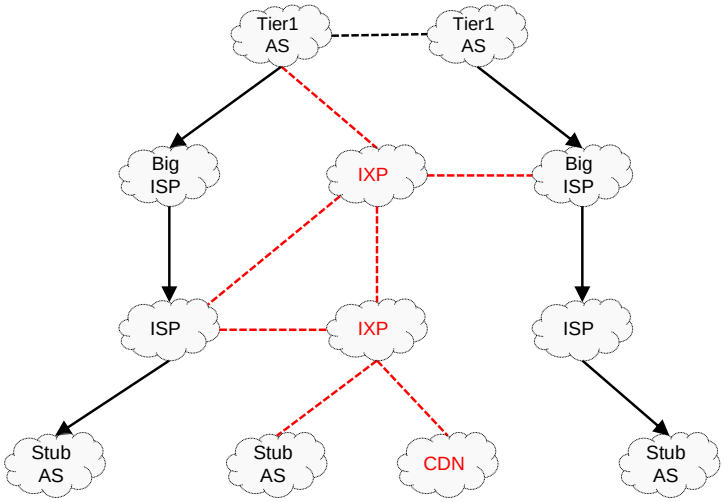


(c) Providers to customers

Figure 2.4: BGP policies following Gao-Rexford rules. The AS in the center announces all prefixes learnt via customers (green ASes) to all remaining ASes, but those learned via peers (yellow ASes) and providers (red ASes) are only exported to customer ASes.



(a) Internet structure before 2000s.



(b) Current shape of the Internet.

Figure 2.5: Evolution of the (simplified) structure of the Internet. Lines with arrows indicate provider-to-customer links, and those dashed peer-to-peer links. While before the Internet had a clear pyramidal structure, the appearance of IXPs increased the number of peer-to-peer links and the Internet flattened. In addition, multiple content providers developed their own CDNs, which further accelerated this process.

2.1.2 IXPs

As a result of the business relationships that ASes tend to establish, i.e., provider, customer and peer ASes, the Internet naturally adopted a hierarchical structure, as illustrated in Fig. 2.5a. The ordering among ASes is still usually described as a pyramid in which, from base to tip, three types of ASes can be distinguished: stub ASes, transit ASes and tier-1 ASes. The set of stub ASes is composed by

those ASes without customers, whereas transit and tier-1 ASes are mainly ISPs. Since stub ASes do not have customers, they only forward traffic either destined to or originated from themselves or siblings. On the other hand, transit ASes have customers ASes of which they forward traffic, but also have provider ASes. In general, transit ASes may be described as those ranging from small to large/big ISPs, usually providing a service constrained to a specific region or continent, whereas tier-1 ASes usually characterize for being ISPs with a geographical footprint covering multiple continents, thus not needing to buy transit service from any other AS.[4] In general, a communication between stub ASes located in different continents required packets to traverse up and then down the pyramid.

The structure of the Internet begun a flattening process in the early 2000s, largely motivated by the massive irruption of Internet Exchange Points (IXPs) [10, 11]. A closer representation of what the Internet looks like nowadays is shown in Fig. 2.5b, where we also highlight the appearance of content distribution networks (CDNs), i.e., networks belonging to content providers. IXPs are facilities that allow multiple ASes to peer at a single location. To interconnect ASes, IXPs use layer-2-devices, i.e., switches. In addition, IXPs usually count with route servers to which ASes may opt to connect. An AS willing to participate in an IXP establishes a unique session with the IXP and, accessing the route server, is able to simultaneously peer with all the other participants in the IXP, known as IXP members or connected networks. In general, IXP members establish peer-to-peer relationships and only announce their customer cone [**customercones**] at IXPs, i.e., prefixes with AS-paths leading to customer ASes, and customers of these. Besides peering with the route server, IXPs usually allow ASes to peer privately, i.e., carry out announcements that are not advertised in the route server. In short, IXPs allow ASes to peer at a larger scale, and thus are known as peering fabrics. Indeed, the use of IXPs broke the mechanics of the hierarchical pyramid, and terms such as tier-1 ASes could arguably be considered as rather legacy. Nowadays, metrics such as the AS-rank [12] are considered more informative to give a sense of the size of the business an AS runs.

The major success of IXPs is backed by many reasons. First, IXPs allow to offload upstream traffic, resulting in monetary savings for IXP members. Second, IXPs allow to exchange local traffic within the same region, shortening end-to-end paths and reducing latency [**carisimo2019studying**, **hoiland2016measuring**]. Third, IXPs attract CDNs [10], and CDNs attract members. Indeed, CDNs place caches in IXPs to get to (multiple) eyeballs from a single peering point [**ager2012anatomy**], thus allowing IXPs to offer low-cost access to content [13].

In particular, in Chapter 3 we study the multiple IXPs deployed across Latin America, a region that has received little attention in Internet studies. Besides shedding light on the status of the Internet in Latin America, we investigate whether IXPs have had a similar success than in the 2000s in the region.

[4] The way I describe the geographical footprint of small and large/big transit and tier-1 ASes is only an interpretation that provides a fast way of classifying ASes into these different categories, but should not be considered strict definitions to which no exceptions exist.

2.2 Intra-domain networks

In Chapter 5 we develop a methodology to detect forwarding detours inside ASes, i.e., whether traffic traversing an AS does not flow through the best available paths in the network. Therefore, understanding how the routing works inside ASes is necessary. To build a strong foundation on such topic, while Sec. 2.2.1 studies the main characteristics of IGPs, Sec. 2.2.2 and 2.2.3 show how load balancing and tunneling techniques allow to modify the standard behavior of the usual intra-domain routing protocols.

2.2.1 IGPs

Besides BGP, ASes additionally use an Internal Gateway Protocol (IGP), i.e., a dedicated routing protocol for intra-domain routing. In particular, renown IGPs include Open Shortest Path First (OSPF) [14] and Intermediate System to Intermediate System (IS-IS) [15].

OSPF and IS-IS are link-state routing protocols, meaning that routers build a graph of the intra-domain network. This graph represents the connectivity status of the network, i.e., shows which nodes are connected to which other nodes.

To construct their routing knowledge, routers exchange routing messages indicating the links they have with other routers. In addition, every advertised link is weighted with a given IGP cost or IGP distance, according to a pre-defined IGP metric. As an example, a hop-count metric assigns a constant IGP cost of 1 to every link, but other metrics may for example vary the cost depending on the bandwidth available on each link.

With OSPF and IS-IS, each router considers that, for every destination advertised in the IGP, the shortest path according to the IGP metric in use represents the best IGP path. To determine such paths, routers run Dijkstra's algorithm [16]. Once the best path to reach a prefix is decided, routers take note of the router that announced the path, and use it as IGP next-hop or next-hop towards that prefix. When the considered prefix is actually a BGP prefix, a recursive lookup is performed, and the next-hop matches the neighboring router offering the best path towards the BGP next-hop of that prefix. In general, the next-hops are further installed in a table known as Forwarding Information Base (FIB), that is optimized to perform fast the actual forwarding of packets. As we study in Chapter 5, some routers sometimes are not able to keep a full-FIB in memory, and the workarounds implemented in these scenarios may lead to forwarding detours.

The concept of best paths across IGPs and BGP do not imply the same, there are two key differences: (i) the information that routers or ASes receive to actually decide which is the best path, and; (ii) the conceptual meaning of the metrics that are taken into account to choose best paths.

While IGP best paths are globally optimal, this may not be the case in BGP. According to IGPs, since all routers in the network receive the same routing messages, all subpaths of any best path are also best paths. This property results from the fact that routers have the same view of the network: what a router considers the best, is also seen as the best for the remaining routers. We refer to this agreement among the devices of a network as the existence of routing consistency in the network. As

we study in Chapter 5, when this property is not met, forwarding detours may likely occur. In contrast, in BGP, the opportunity to apply filtering policies breaks down this property: not all ASes receive the same information, so ASes choose best paths from a set of routes that may be just a subset of all those actually available. In other words, contrary to IGPs, the best paths BGP chooses are locally optimal instead of globally [26].

On the other hand, IGPs attempt to maximize the efficiency of the routing and forwarding, however, BGP rather minimizes the monetary cost of handling traffic. The purpose of using hop-count or bandwidth-available metrics in IGPs is to enforce "fastest" paths as best paths, i.e., those through which traffic traverses the network with the least required IGP distance. On the contrary, for BGP, the importance of end-to-end delay is secondary. This results from the fact that the BGP decision process weights the local preference more than any other attribute, and this metric only reflects business relationships, as previously explained. For BGP, comparing the length of the control paths comes as the second criteria, hence, shorter paths via providers will usually be neglected even when faced against longer paths, given that these were advertised by peers or customers. In fact, since not necessarily all ASes comply to the same policies, and thus may take conflicting decisions, this has lead to the research concerning BGP convergence, a concept introduced in Sec. 2.1.1.

2.2.2 Load balancing

Load balancing (LB) is a technique that network operators may deploy to both enhance and increase the resource utilization of their network. LB provides a means to distribute the load across multiple paths, named LB paths. In general, LB relies on the equal-cost multi-path (ECMP) routing feature of the intra-domain routing procols OSPF [14] and IS-IS [15], such that all parallel paths that are used share the same IGP cost [47]. Note, however, that LB may also be used across inter-domain links, for example, with the use of multi-path BGP [48].

The routers that enable ECMP and apply LB are known as load balancers or LB routers. LB routers have the capability to choose among different next-hops towards a destination. Every time load balancers have to forward a packet, they send it to the next-hop they select out of a set of available next-hops towards each destination. As we show in Chapter 5, similar to LB, forwarding detours generate multi-path routing patterns between endpoints of ASes and, therefore, differentiating among these two, requires understanding the details of how LB operates. Indeed, LB can be deployed in different LB flavors, or flavors [49]. To balance packets across next-hops, load balancers take into account either (some) packet header fields, or none at all [22, 50]. In the following we describe the usual LB flavors.

The simplest mode of LB, namely per-packet LB [51, 52], 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 [53, 54, 55]. Moreover, faced to this LB flavor, tools aiming to retrieve the forwarding paths used on the Internet may fail to reveal some links, and even infer false ones [22, 50]. Fortunately, per-packet LB is rarely found in practice [56, 57, 58].

Other more sophisticated LB methods, which we call hash-based LB, 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 [51], and per-flow LB [51, 52, 62]. While the source and destination IP addresses are used as input in per-dest LB, the source and destination transport ports or the ICMP checksum are additionally taken into consideration in per-flow LB for UDP and TCP, and ICMP packets respectively. In addition, the IP type of service (TOS) or DSCP and ICMP code fields have also been identified as fields that load balancers may take into account to choose among next-hops [22, 58]. Moreover, a per-application LB scheme, only relying in the transport port numbers has also been proposed [64]. However, results in [58] suggest that, as of today, these flavors are not as widespread as per-dest and per-flow LB.

Previous work has mainly focused on per-dest and per-flow LB, however, there exists a third hash-based LB flavor that has been systematically omitted in the literature, known as per-prefix LB [62, 93], which we consider in this thesis. 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, in this thesis we propose a new nomenclature: we say that per-prefix LB is a coarse-grained LB type (C-LB), while per-dest and per-flow LB are fine-grained LB types (F-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, preventing the use of redundant routes, concentrate traffic on a subset of the available LB routes [63]. On the other hand, the rebooting of routers, and the consequent recompute of the seed value, has been suggested to produce next-hops re-mapping events often mistakenly attributed to routing changes [57].

2.2.3 Tunneling mechanisms

In an ISP, transit traffic usually traverses the AS from ingress-ASBRs to egress-ASBRs. For this to be handled efficiently and correctly, all routers in the network would be required to speak BGP, or at least to know the best gateway for each external prefix. To reduce the load on devices, ISPs and ASes in general may opt to run a network with a BGP-free core, and rely on tunneling mechanisms [44]. With this alternative option, traffic is tunneled between edge routers, that are the only devices that require having a full BGP feed. On the other hand, core devices take forwarding decisions based not on the destination IP addresses, but rather depending on the ingress point through which traffic enters the AS and the egress point to which it needs to be redirected.

There exist different tunneling protocols, e.g. IP in IP, generic routing encapsulation (GRE), but multi-protocol label switching (MPLS) [**rfc3031**] is usually considered the de-facto standard. With MPLS, traffic is tunneled via label-switched paths (LSPs), where labels are appended to packets by the ingress node of the path, referred to as label switched router (LSR). The forwarding is then performed based on the labels values, that get updated at each hop.

Network operators may implement MPLS over the core with the combination of an IGP and the label distribution protocol (LDP) [45], or segment routing (SR) [46]. In general, this option encompasses scalability purposes related to running BGP-free cores, ensuring traffic traverses the network through best IGP paths between LSRs. As we study in Chapter 5, networks operators may use these techniques to avoid suffering from forwarding detours in their networks, but still may fail to achieve this goal.

On the other hand, tunneling mechanisms may be used for traffic engineering, with the resource reservation protocol for traffic engineering (RSVP-TE) [**rfc3209**]. With this option, network operators can implement LSPs with constrained requirements concerning bandwidth, jitter, etc. and hence use it for traffic engineering purposes. In general, this solution is used for a reduced set of prefixes since RSVP-TE suffers from well-known scalability issues, that increase with the quantity of TE paths that are deployed [**filsfils2015segment**]. As a variant to RSVP-TE, segment routing implements a lightweight control plane that allows it to scale better, and thus is now considered the new state-of-the-art TE and fast-reroute technology deployed in most ISPs. As we model in Chapter 5, TE generates multi-path routing patterns. As such, constrained paths resulting from TE may be confused with forwarding detours. In particular, in Chapter 5 we address the difficulty of differentiating between both.

2.3 Traceroute

Traceroute is the most widespread tool allowing to collect IP forwarding paths. In addition, we make extensively use of it in Chapter 4 and 5, thus present its functioning principle in detail in this section. In particular, Sec. 2.3.1 explains how the original version of traceroute works and a basic method allowing to translate the IP path that this tool outputs into an AS-level forwarding path. Then, Sec. 2.3.2 presents *Paris* traceroute, an updated load-balancing-aware version of traceroute. In addition, Sec. 2.3.3 shows how Paris traceroute can be converted into a multi-path detection tool.

2.3.1 Standard version

Van Jacobson introduced traceroute, a tool allowing a host on the Internet to discover the forward IP path towards a destination IP address [17].

Traceroute manipulates the Time-To-Live (TTL) field in the IP header of packets to elicit responses from the intermediate network-layer devices that are traversed in the path towards the destination IP address. This field expresses the number of times that a packet can still be forwarded before it needs to be dropped, and is decreased by routers at each hop [18] (though this has been shown not to hold in

practice sometimes [22, 50, 87]). As a packet moves along towards the destination, each intermediate router that receives it first checks whether the destination IP address coincides with any of the IP addresses assigned to its interfaces. If this is not the case, the router may proceed in two different ways depending on the value in TTL field of the packet header. If the value is greater than one, the router decreases it by one unit and forwards the packet to the next-hop towards the destination. Instead, if the value equals one, the packet is said to have expired, and is discarded. However, before doing so, the router handling the packet sends an ICMP Time Exceeded message back to the source. This message encapsulates the IP header of the probe packet, and the first 64 bits that follow it [19]. Since traceroute does not use IP options, then these 8 bytes comprise, depending on the probe packet protocol, the complete UDP or ICMP Echo Request headers, or partially the TCP header. The router uses as source IP address of this message that of the outgoing interface it uses to forward the packet [20, 21]. When the routing is symmetric, this IP address coincides with that of the incoming interface that received the probe packet.

Traceroute sends packets in succesive rounds towards a destination IP address, with increasing TTL values: in the first round packets have a TTL = 1, the second one TTL = 2, and so on. As these packets expire at increasing hop distances, all the intermediate routers finish replying with ICMP Time Exceeed messages. The entity running traceroute, by inspecting the source IP address of each of the messages it receives, is thus able to learn the IP addresses that are traversed towards the destination IP address. However, at this point, two questions remain to be addressed: (i) how is traceroute able to determine the order in which the IP address it learns were crossed? and; (ii) how does traceroute know when to stop sending packets?

Concerning (i), besides continuously changing the TTL values, traceroute encodes additional data in other packet header fields, allowing it to reconstruct the traversed IP path in the correct order. To retrieve this data, traceroute wisely chooses the encoding fields, from either the IP header, or the first 8 bytes of IP payload. This way, traceroute makes sure that the fields it uses are quoted in the ICMP Time Exceeded messages it receives. The exact fields depend on the transport-protocol that traceroute uses. For UDP probes, TCP SYN and ICMP Echo Request messages, traceroute systematically varies the destination port, IP ID and ICMP sequence number fields, respectively [22]. Using a unique value per packet that is sent, traceroute is thus able to match each response message to each original packet it sent, and thus to infer the correct IP path.

Regarding (ii), the basic assumption is that, as the source is continuously launching probes with an increasing TTL value, at a given moment a packet will arrive to the destination IP address. When the destination actually receives the packet, it will corroborate that the destination IP address in the IP header field coincides with one of the IP addresses in its interfaces. In this case, rather than an ICMP Time Exceeded message, the destination will reply to the source IP address with another type of message. For UDP probes, TCP SYN and ICMP Echo Request messages, these replies will usually be an ICMP Destination Port Unreachable, TCP ACK+SYN or Reset and ICMP Echo Reply messages, respectively. Upon reception of this different type of packet, the source is thus able to determine that the tracing

can be stopped. Besides the options enumerated above, where traceroute ends with a successful status, there exist cases where traceroute (partially) fails. These include cases where, the host, network or protocol is unreachable, the communication is administratively prohibited, etc. A more exhaustive list can be found in [23]. In other words, traceroute may halt with a failing status, and only provide a partial IP path towards a destination.

Traceroute also keeps a timer for each packet it sends, and thus it is able to estimate the round-trip time (RTT) towards each IP address it finds in the path towards the destination. In addition, traceroute also uses these timers to declare packets as lost. Indeed, this is assumed to have happened once a timer exceeds a given threshold that can usually be pre-configured. In these cases, traceroute will include missing hops, which are usually indicated as '*'. Whether the packet traceroute sent or the reply itself was lost cannot be known. In addition, if the router received the packet, but applied ICMP rate limiting, i.e., could not reply at that moment, or will never do so, requires additional analysis [24, 25].

Traceroute is a tool that runs active measurements, i.e., that allows to obtain data by actively sending packets towards a host. There exist looking glasses and measurements platforms such as RIPE ATLAS, NLNOG RING infrastructure, the PEERING testbed, CAIDA's Ark, among others that allow to perform traceroute from nodes distributed around the globe.[5]. In addition, it is important to know that even though traceroute is the state of the art tool to gather data paths, there exist other tracing tools to obtain forwarding IP paths. This is the case of packets with IP options, such as the IP record route [18]. These type of packets were seen to be dropped in edge networks, and their practical usefulness questioned [**fonseca2005ip**]. However, lately, other studies have argued the opposite [**goodchild2017record**] and even constructed a reverse IP path tracer relying on them [**katz2010reverse**]. A limitation of the record route option is that, contrary to traceroute, it is constrained in the number of IP addresses it can report, to no more than 9 [18]. Besides this, my personal experience is more aligned with [**fonseca2005ip**].

Finally, the output of traceroute is an IP-level path, however, in multiple occasions it is valuable to translate this into AS-level paths, i.e, understand which ASes were traversed in the path towards the destination IP address. This is usually done with an IP-to-AS mapping tool. To map from IP paths to AS-level paths, the standard method is to rely on BGP snapshots, and map each IP address to the OAS announcing its best covering prefix. As we study in Chapter 4, this process is error-prone, and usually requires refining. Moreover, this chapter presents a new framework we propose to filter inaccuracies resulting from the basic IP-to-AS mapping method.

2.3.2 Paris traceroute

The standard version of traceroute has the limitation that, in the presence of load balancers, it may likely provide incorrect route inferences [22]. This problem re-

[5]RIPE ATLAS: `https://atlas.ripe.net/`; NLNOG RING infrastructure: `https://ring.nlnog.net/`, PEERING testbed: `https://peering.usc.edu`; CAIDA's Ark: `https://www.caida.org/projects/ark/`

sults from the fact that load balancers may likely forward the multiple packets that traceroute sends across different next-hops. This may lead to both missing links and false links, and has be shown to explain most anomalies, such as loops and cycles, seen in the standard traceroute [22, 50].

The aforementioned problem only occurs for per-packet and per-flow LB. For per-packet LB, this issue appears due to the non-deterministic nature in the selection of next-hops. Consquently, a priori, this problem cannot be solved for this flavor. On the other hand, for hash-based LB flavors, the difficulty originates from the fact that the fields that load balancers use to ascribe packets to different flows are modified by traceroute. Indeed, this happens because these fields are either (i) the same encoding fields that traceroute uses to be able to reconstruct the IP paths in the correct order, or; (ii) indirectly modified by the encoding traceroute uses. For a fixed source IP address, since traceroute also keeps the destination IP address constant per trace, these situations can only arise in the presence of per-flow load balancers.

The artifacts that the presence of per-flow load balancers produce depends on the type of packets that traceroute uses. The problem exists for UDP and ICMP probes, since traceroute deliberately changes the destination port number and indirectly the ICMP checksum when updating the ICMP sequence number, respectively. For TCP, instead, traceroute keeps the port numbers constant, in particular choosing the value 80 for the destination port number to emulate web traffic. In these cases, traceroute relies on the IP ID field as the encoding field, thus TCP probes launched by traceroute are not subject to load balancing artifacts.

Paris traceroute was introduced in 2006 as a per-flow load-balancing-aware version of traceroute [22, 50]. Basically, Paris traceroute only changes the encoding fields that the standard traceroute uses to other assumed not to be used for LB. Hence, for all transport-layer protocols, Paris traceroute manages to keep the fields that are used by per-flow load balancers with a fixed value. This ensures that load balancers forward all packets issued in a trace to the same next-hop, and fixes most artifacts described in [22, 50]. For UDP probes, Paris traceroute fixes the destination port number, and uses the transport-layer checksum as the encoding field. For this, Paris traceroute carefully crafts the payload these UDP packets carry such that the UDP checksum, besides increasing by a unit each time a packet is sent, is also valid. On the other hand, for ICMP probes, the ICMP sequence number is still used as the encoding field. However, Paris traceroute additionally modifies the ICMP identifier field to offset the change in the ICMP sequence number, ensuring that the ICMP checksum remains constant in all packets. Finally, Paris traceroute also implements a TCP version that, compared to that of the standard traceroute, simply replaces the encoding field from the IP ID to the TCP sequence number. The motivation behind this TCP implementation, however, is only to uniformize the use of the IP ID field across all probe packet protocols as the process encoding field, i.e., the field that relates to the running process identifier (PID) of each traceroute instance that is launched.

2.3.3 Multi-path detection algorithm

To detect the set of load-balanced paths between a source and a destination on the Internet, the multi-path detection algorithm (MDA) was introduced in 2007 as a variation of Paris traceroute [65]. In short, the MDA is a stochastic probing algorithm that can be used to find, with a pre-defined high-confidence, all interfaces and links at every hop towards a destination. In particular, MDA is able to gather all these results when only per-dest or per-flow load balancers are traversed, but may fail to reveal true links faced to per-packet load balancers.

The MDA has been subject to multiple refinements over time [66, 67, 56]. However, the basic working principle always remains the same. This section focuses on the latest implementation, that relies on the assumptions that: (i) load balancers are independent among themselves, i.e., the next-hop a load balancer chooses does not affect the one that another load balancer may choose; (ii) load balancers choose any of their next-hops with equal probability (for per-packet or per-dest/flow load balancers, this should be interpreted at the packet or flow level, respectively); (iii) MPLS is not deployed; (iv) no routing change occurs while the MDA is tracing a destination, and; (v) routers reply to probe packets. When the aforementioned hypothesis are not met, the MDA may provide incomplete or inaccurate results.

The MDA has two modes of operation, either at the node level or end-to-end path level, that mainly differ in the probing that each requires. While the first is able to conclude with a high significance level α that all next-hops of one incoming interface have been discovered, the latter bounds, for a given theoretical maximum number of interfaces Q_I along the path, the failure probability β of discovering all next-hops of all interfaces across the path. In practice, the arbitrarily chosen values $\alpha = \beta = 0.05$ and $Q_I = 30$ are often used. The effects of varying α or Q_I have never been studied, while that of changing β from 0.01 up to 0.5 has shown impact in the probing cost, but not on performance [56]. In other words, the end-to-end path model does not provide a tight bound to the probability of failure. As a consequence, the actual interpretation of the parameter β becomes blurred, being rather an indicator of how many extra measurements the user may afford to waste. I believe that tuning Q_I differently may allow not only to consider more realistic scenarios, but also will likely translate into a significant save of probing cost.

In general terms, the underlying behavior of the MDA is similar regardless of the mode of operation. Indeed, the MDA works in multiple rounds that are repeated at every interface that is revealed, until the probing reaches the destination. In the following, we will analyze the procedure of the MDA considering an arbitrary hop h at which an interface r_h with N next-hops is being analyzed. The MDA sends packets through r_n and declares the interfaces that are revealed at hop $h+1$ as the next-hops of r_h. The MDA studies all interfaces at hop h, and then continues with those discovered at hop $h+1$. The notation we use is summarized in Table 2.1

The MDA initially assumes load balancers are per-flow load balancers, and seeks at least n flow-IDs $\phi_1, \phi_2, \ldots \phi_n$ for which r_h is traversed. The flow-IDs are varied by choosing different destination IP addresses. This is done progressively increasing the size of the IP prefix from which the IP addresses are obtained, until the /24 granularity is reached. Beyond that point, [66] and [67] do not describe what should be done, however it has been recently proposed to continue varying flow-IDs by

h	Hop number
r_h	Interface at hop h
ϕ_i	Flow-ID i
α	Significance level that all next-hops of one interface have been discovered
Q_I	Theoretical maximum number of interfaces along the path
β	Failure probability of discovering all next-hops of all interfaces across the path.
N	Number of next-hops of interface r_h
$\hat{N}$	Number of next-hops estimated for of interface r_h by the MDA
$n \triangleq n_{\hat{N}}$	Stopping point the MDA uses given $\hat{N}$ next-hops have been discovered

Table 2.1: Notation used to describe the functioning principle of the MDA.

.

changing port numbers [57]. Though effective for per-flow load balancers, this strategy may fail to detect additional next-hops in case per-destination load balancers are traversed. In this thesis, we propose an alternative that may help to solve this problem, as discussed in Chapter 5. In any case, since r_h is at least the next-hop of one interface r_{h-1} previously analyzed, then the flow-IDs for which both r_{h-1} and r_h were revealed, can be reused to discover next-hops of r_h. In a more general sense, all of the flow-IDs $\phi_1, \phi_2, \ldots \phi_j$ associated with paths for which r_h is reached can be reused. If $j < n$, then the MDA searches the $n - j$ missing ones in a trial-and-error fashion, i.e., sending successive probes carrying a random flow-ID.

Relying on hypothesis (ii), i.e., that load balancers uniformly distribute flow-IDs across their next-hops, the MDA calculates the value of n to meet the requirements established by the mode of operation, i.e., according to α or β and Q_I. In particular, the values are calculated for the scenario with least chances of revealing all interfaces, that corresponds to the case where $\hat{N}$, the number of next-hops already discovered, equals $N - 1$. In other words, the question that the MDA addresses is "after n measurements and $\hat{N}$ next-hops that have been respectively carried and discovered, could it be that we are still missing one next-hop?". The reason of why asking if only one and not more, intuitively, is that the less next-hops remain to be discovered, the harder it becomes to find them. As a consequence, when estimating the additional probing required to ensure that the probability of missing any number of next-hops is low (or at least bounded below a certain value), this represents the most conservative approach. Adopting this criteria, n can be shown to depend only on $\hat{N}$ and α or β and Q_I. Moreover, considering the latter parameters are pre-defined by the user, n is usually indicated as $n_{\hat{N}}$ and referred to as stopping point. While the values of the stopping points implemented by the end-to-end path level version of the MDA can be found in [56], for the node level version, after re-arranging the formulas and conditions in [65], $n_{\hat{N}}$ is the minimum value such that the following constraint holds

$$\sum_{i=0}^{\hat{N}} (-1)^{\hat{N}-i} \binom{\hat{N}+1}{i} \left(\frac{i}{\hat{N}+1} \right)^{n_{\hat{N}}} < \alpha$$

Proof. I find the explanation in [65] quite hard to follow, therefore I will provide

the complete demonstration. The question to answer is "*given that an interface has N next-hops, what is the probability P_f that only one might not be revealed, i.e., $N = \hat{N} + 1$, after the MDA uses $n_{\hat{N}}$ probes?*". We can model the scenario as the well-known problems involving *balls* that are thrown into *urns*. The parallelism dictates that urns are next-hops, and balls are the probes that the MDA sends. Therefore the question can be reformulated as: *Given that there are N urns, what is the probability that after throwing $n_{\hat{N}}$ balls, one urn might be empty?*". Defining the random variables

$$U_i : \text{number of balls inside urn } i$$

what we want to find is the probability P_f that any U_i might be null, i.e.,

$$P_f = P(U_1 = 0 \ \cup \ U_2 = 0 \cup , \ldots , \cup U_N = 0) = P\left(\bigcup_{i=1}^{N} U_i \right)$$

To solve this, first we can use the inclusion-exclusion principle that generalizes the property stating that $P(A \cup B) = P(A) + P(B) - P(A \cup B)$, which gives

$$P_f = \sum_{k=1}^{N} (-1)^{k-1} \sum_{\substack{I \subset \{1,\ldots,N\} \\ |I|=k}} P(U_I)$$

such that $U_I \triangleq \bigcap_{i \in I} U_i$. Since the probability of the events U_I depends on the cardinality of I, then we can rewrite P_f as

$$P_f = \sum_{k=1}^{N} (-1)^{k-1} \binom{N}{k} \left(\frac{N-k}{N} \right)^{n_{\hat{N}}}$$

where $\binom{N}{k}$ is the number of events that have cardinality k, and $(\frac{N-k}{N})^{n_{\hat{N}}}$ is the probability of the event that k given urns might be empty after throwing $n_{\hat{N}}$ balls. Considering the property that $\binom{N}{k} = \binom{N}{N-k}$ and defining $i = N - k$, then

$$P_f = \sum_{i=0}^{N-1} (-1)^{k-1} \binom{N}{i} \left(\frac{i}{N} \right)^{n_{\hat{N}}}$$

and recalling that $N = \hat{N} + 1$, we have

$$P_f = \sum_{i=0}^{\hat{N}} (-1)^{\hat{N}-i} \binom{\hat{N}+1}{i} \left(\frac{i}{\hat{N}+1} \right)^{n_{\hat{N}}}$$

Once P_f is known, the question that remains is: *what should be the value of $n_{\hat{N}}$ so that P_f is less than α?* The value of $n_{\hat{N}}$ has to be calculated so that

$$\sum_{i=0}^{\hat{N}} (-1)^{\hat{N}-i} \binom{\hat{N}+1}{i} \left(\frac{i}{\hat{N}+1} \right)^{n_{\hat{N}}} < \alpha$$

■

$\hat{N}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$n_{\hat{N}}$	6	11	16	21	27	33	38	44	51	57	63	70	76	83	90	96

Table 2.2: Stopping points used by the node level version of the MDA given that $\alpha = 0.05$. Each column indicates that if $\hat{N}$ next-hops have been discovered for one interface, once $n_{\hat{N}}$ are sent and no new next-hop appears, the MDA stops probing this interface and continues with another one. Note that the table in this manuscript differs with that in [65] since they show N instead of $\hat{N}$, where $N = \hat{N} + 1$.

.

There is actually no closed form to express the values that $n_{\hat{N}}$ should take, so they need to be manually calculated to comply with the aforementioned constraint. Table 2.2 displays the values $n_{\hat{N}}$ takes for different values of $\hat{N}$. As measurements are carried and more next-hops of r_h are discovered, the value of $\hat{N}$ increases, and thus that of $n_{\hat{N}}$ needs to be iteratively updated. The process continues until, eventually, $n_{\hat{N}}$ flow-IDs are tested and no new interface is discovered, and thus the probing on the interface halts. When this happens, the discovered $\hat{N}$ next-hops of r_h are assumed to be all the available next-hops of r_h.

The methodology that the MDA carries out can be interpreted as follows. Initially, except for the destination, we always have $\hat{N} = 1$ for all interfaces, even if the next-hops for each of them are still unknown. Consequently, the first stopping point $n_1 = 6$, can be considered as the MDA testing if the interface r_h belongs to a load balancer. If r_h is not a load balancer, then $N = 1$. In this case, for the 6 flow-IDs that the MDA generates, the same unique next-hop of r_h is always discovered. Consequently, with a relatively low probing cost, the MDA is able to rapidly conclude that r_h does not belong to a load balancer, and thus continues to investigate other interfaces. On the other hand, when r_h is a load balancer, then $N > 1$. If $N = 2$, for example, then the MDA begins sending 6 probes with different flow-IDs. The first probe reveals one of the 2 available next-hops, and the successive ones may either also discover the remaining next-hop or fail to do so. In the first case, then $\hat{N} = 2$ and the stopping point is updated. Hence, the probing continues until $n_2 = 11$ probes are sent. Indeed, even though $\hat{N} = N$ before the $n_{\hat{N}}$ probes are sent, MDA has no means to know at this moment that all next-hops have already been discovered, and thus some extra probing is wasted. In the latter case, instead, since $\hat{N} = 1$, then once MDA sends $n_1 = 6$ probes, the probing stops early, saving probing cost, but failing to reveal all next-hops. Note, however, that the probability of this event occurring is bound by design when selecting the value of α or β and Q_I. The outcome $N = 2$ and $\hat{N} = 1$ is an interesting case since MDA fails to detect that r_h belongs to a load balancer. For $N > 2$, though the same may apply, it is more likely that $\hat{N} > 1$, i.e., that MDA will detect the load balancer and thus may fail to reveal some additional next-hops of r_h. Finally, it is important to notice that as $\hat{N}$ increases, $n_{\hat{N}}$ also becomes larger. Therefore, it is likely that, beyond a given $n_{\hat{N}}$, MDA will also need to search, by randomly probing, for new flow-IDs meeting the requirement of traversing r_h.

As a last step, the MDA infers which LB flavor the discovered load balancers apply. The MDA tests this one interface at a time, first sending k packets that carry

the same fixed flow-ID for which r_h is known to be traversed. If different next-hops are revealed in the k trials, then MDA concludes that the interface belongs to a per-packet load balancer. If, on the contrary, the same next-hop is always revealed, then the MDA proceeds to send another set of k probes, but this time only keeping a fixed IP destination address while varying the port numbers. If different next-hops are revealed, then this time the MDA concludes that r_h belongs to a per-flow load balancer, otherwise to a per-destination load balancer. Relying on hypothesis (ii), and requiring a 95% confidence level of not missing a next-hop when $N = 2$ and $\hat{N} = 1$ (similar as before, this represents a worst case scenario), the MDA sets $k = 6$, which coincides with n_1 when $\alpha = 0.05$.

Chapter 3

Success and Failure of IXPs in LatAm

Contents

In this chapter we study the deployment of IXPs in LatAm. Indeed, this study sheds light on whether IXPs, that had a major sucess in the 2000s, are nowadays also benefiting the development of the Internet in regions other than Europe. We are interested in the public policies that lead to the creation of Latin American IXPs, their growth and development over time, and the role each of them plays in their national AS ecosystem to determine whether they are *failed IXPs*, i.e., the IXP has failed to attract members or there is even no IXP at all in the country, or have succeeded to proliferate. To provide a broader view, we compare IXPs deployed across multiple continents. In short, our contributions are:

1. We determine countries in LatAm with IXPs and construct the so far most comprehensive dataset gathering information about IXPs in LatAm in Sec. 3.1. In particular, we gather BGP data from multiple collectors of Packet Clearing House (PCH) located in the region. To the best of our knowledge, we are the first to explore this dataset. In addition, we extended the BGP view in Brazil leveraging a collector of Routeviews and several looking glasses distributed across multiple Brazilian IXPs. Moreover, we combine multiple additional data sources to derive metrics that help quantify the growth of IXPs and to better understand the role of transit providers at IXPs.

2. We provide insights in Sec. 3.2 about how countries' public policies have encouraged the development of IXPs in Latin America. Interestingly, local governments of each country were involved in the creation of more than 55% of the Latin American IXPs. Similar to the European IXP model, in Latin America a large number of non-profit organizations run IXPs (and also created them in some cases).

3. We study in Sec. 3.3 the topology, peering policy and infrastructure of the largest Latin American IXPs and compare them with the domestic IXPs of DE-CIX, i.e., a renown German IXP. Our findings show that IXPs are diverse, with heterogeneous deployments: they may have either one or multiple peering facility per IXP, with or without an associated ASN, and range from those where no peering policy to a mandatory peering requirement among all connected networks are enforced.

4. We analyze how IXPs have been increasingly gaining importance since their creation, and the members that compose them in Sec. 3.4 . While IXPs in LatAm and in developing regions in general have been able to attract domestic and regional members, European IXPs have also managed to gather members from different regions, which allows to speculate that there is room for these IXPs to continue growing in the future.

5. We focus on how traffic is carried from/to Latin American IXPs in 3.5. We find that transit providers peering at the IXPs in Latin America are mainly regional, but also find large international transit providers providing local service to domestic ASes in LatAm. In addition, we look for non-transit members of IXPs, i.e., ASes that only announce prefixes owned by themselves in the IXP. Interestingly, we find transit ASes actively choosing not to announce prefixes of their customers.

6. We analyze the success and failure of IXPs in LatAm in Sec. 3.6 attempting to relate this phenomenon with the presence of a balanced AS ecosystem, i.e., where IP addresses are more fairly distributed among ASes of the country. We find a negative correlation between the absence of monopolistic transit/access ASes owning most IP addresses assigned to a country and the success of national IXPs.

7. We release the code that allows both to fetch the publicly available data we

used and to replicate our results[1]. In addition, we make publicly available the dumps we manually collected at the Brazilian looking glasses[2].

In addition, we derive the main conclusions of this chapter in Sec. 3.7. The research presented in this chapter lead to the following pieces of work:

- Esteban Carisimo, **Julián M. Del Fiore**, Diego Dujovne, Cristel Pelsser, and J. Ignacio Alvarez-Hamelin. 2020. *A first look at the Latin American IXPs*, in SIGCOMM Comput. Commun. Rev. 50, 1 (January 2020), 18–24.
- Esteban Carisimo, **Julián M. Del Fiore**, Diego Dujovne, Cristel Pelsser, J. Ignacio Alvarez-Hamelin, *Country-level influence of IXPs in Latin America*, in Latin American Student Workshop on Data Communication Networks (LANCOMM) 2019.

3.1 Dataset

In this section we present the dataset we gathered to carry out our analysis. In particular, Sec. 3.1.1 shows preliminary findings concerning the countries in LatAm with IXPs, Sec. 3.1.2 describes the different sources of data we collected to analyze these IXPs, and Sec. 3.1.3 focuses on how we pre-processed BGP dumps.

3.1.1 Searching for IXPs in LatAm

To create a preliminary dataset listing IXPs operating in LatAm, we start by exploring PCH's Internet Exchange Directory [**IXPdir**], Internet eXchange Federation's IXP Database (IXPDB) [**ixpdb**], LACNIC's website reporting regional IXPs [**ixp_lacnic**] and ICANN LAC's IXP list [**icannlac**]. We then sanitize this dataset by validating each entry with either PeeringDB or the website reported in the original data source for the supposed IXP. This allows us to filter out duplicate entries with similar names (IX.br Paraná and IX.br Curitiba), IXPs not yet released (CABASE Corrientes) or peering facilities mistakenly reported as IXPs (Diveo NAP). To the best of our knowledge, our list is the most comprehensive gathered for the region.

Fig. 3.1 shows the number of IXPs per each country in LatAm in June 2020, excluding European overseas territories.[3] We found the existence of at least one IXP in 18 out of 24 countries. Remarkably, Brazil and Argentina count with 36 and 28 IXPs, respectively, and are followed by Chile, with 7 IXPs. The widespread success of IXPs in these countries shares a point in common: **the majority of the regional IXPs in Brazil (31) and Chile (5), and all in Argentina, are administrated by the same entities.** Even more, in both Argentina and Chile, their regional IXPs are interconnected. The network of IXPs in Brazil is operated by IX.br, a network partially supported by the Brazilian state; in Argentina, by the trade organization CABASE; and in Chile by PIT Chile (PIT-CL), a non-profit

[1] `https://github.com/CoNexDat/latam-ixp-obs`

[2] `https://cnet.fi.uba.ar/latam-ixp-obs/lg-ribs/`

[3] European overseas territories in Latin America: Aruba, French Guyana, Bonaire, Curacao, Saint Martin.

organization. We refer to these three as *networks of IXPs*. The remaining countries with IXPs usually own no more than two, except for Ecuador, that has 5 IXPs. Finally, Uruguay, Venezuela, El Salvador, Nicaragua, Guyana and Suriname are the only countries in the region that do not have an IXP yet.

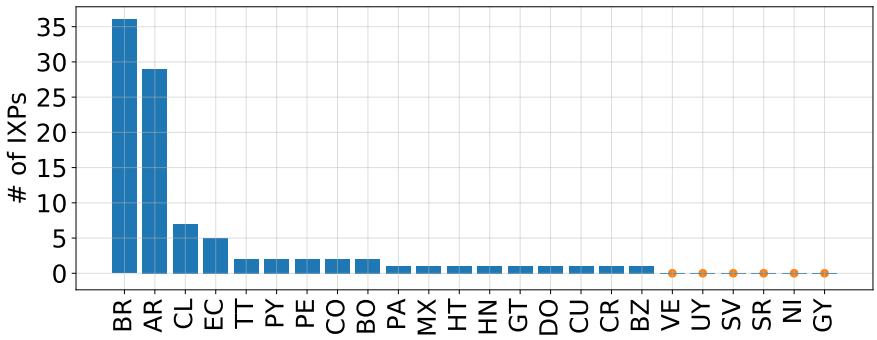


Figure 3.1: Number of IXPs per country in LatAm. Brazil, Argentina and Chile have multiple IXPs belonging to networks of IXPs. The remaining countries only have 0, 1 or 2 IXPs, except for Ecuador, that has 5.

3.1.2 Collecting data sources

Our study mainly relies on IPv4 **BGP table dumps** gathered by PCH with monitors co-located at IXPs in LatAm since 2010. In particular, we obtain BGP data from the following countries: Argentina, Belize, Chile, Costa Rica, Ecuador, Haiti, Honduras, Mexico, Paraguay, and Trinidad and Tobago (TT). In addition, PCH has presence in a Bolivian IXP, however this IXP counts with no members [**megalink**]. On the other hand, PCH has no presence in Brazil. Despite this, we were still able to gather BGP data in IX.br leveraging (i) a Routeviews collector that peers in an IXP of IX.br in Sao Paulo, and; (ii) the 31 looking glasses publicly accessible in IX.br [**carisimo2019**, **brito2016dissecting**]. Our dataset spans from 2010 up to date, except for the data collected at the looking glasses, which comprises monthly snapshots since June 2019. Indeed, IX.br does not keep historical track of them, however, we are able to obtain an additional snapshot previously collected by Brito *et al.* in 2015 [**brito2016dissecting**]. All the BGP snapshots we analyzed were collected the first day of each month.

To put our results in context, we also downloaded tables from PCH collectors in other regions: France-IX (Paris), DE-CIX (Frankfurt, Germany), JINX (Johannesburg, South Africa) and BKNIX (Bangkok, Thailand). We chose these IXPs because either themselves, or the countries where they are deployed, share properties with those deployed in LatAm: largest populations in their region (e.g. France, Germany and Brazil), similar age (e.g. BKNIX and the Chilean IXP are recently created IXPs, while DE-CIX and the Argentinian IXP have been both operating for more than two decades) and comparable current values of GDP per capita (e.g. South-east Asia, South Africa and Latin America) [**gdp_worldbank**].

To complement the BGP data we gathered, we use multiple resources. We queried **RIR delegation files** to determine the set of ASNs delegated to each

country[4]. On the other hand, we used **AS relationship files** to classify ASes into either stub or transit, **AS classification lists** to further differentiate between content providers, enterprises, etc., and **prefix-to-AS files** to compute the address space originated by each AS. All these files were obtained from CAIDA.[5] Moreover, we queried CAIDA's **AS-rank** API to quantify the relevance in the transit ecosystem of each AS in our dataset.[6] We use a snapshot of the APNIC **eyeballs dataset [apnic_eyeballs, apnic_eyeballs_2]** collected on June 1st 2020 to identify the domestic eyeballs of each country. We used **PeeringDB** to determine regional IXPs names and Route Server ASNs of IX.br, CABASE and PIT Chile.[7] In addition, we gathered **digitalized documents**, e.g. legal documents, newspapers, websites and presentations, concerning Internet's public policies applied by LatAm's governments.

3.1.3 Pre-processing BGP data

After gathering the BGP dumps, we pre-processed them. We observed that some ASes share full tables, and we argue that this not what actually gets advertised in the IXPs, i.e., following Gao-Rexford principles [6], no AS would offer cost-free transit via its upstream providers. Consequently, when analyzing each IXP, we relied only on entries provided by their route server: in these cases, the revealed routes are usually from ASes advertising their customers, at least partially. Finally, all tables were sanitized removing AS-path prepending and dropping entries with AS sets (less than 1%).

After sanitizing the BGP tables dumps, we extracted different sets of ASes from them. In particular, **IXP members or connected networks** were inferred as the first AS found in each AS path after the IXP's ASNs (e.g., Route Server, regional IXPs). Besides the IXP members, we also look at ASes connected via members, or **visible ASes**, that correspond to the set of ASes seen in BGP dumps, i.e. that appear in the AS paths of prefixes announced at the IXP.

We complement the previous data looking for insights about ASes in the additional data sources we used. From the list of all delegated ASNs we obtained from the RIR delegation files, we compute the set of **local ASes** or **domestic ASes** of each country. For this, we consider that ASes using ASNs that were delegated to a given country have the country nationality. Finally, the AS relationship files allow us to identify **active ASes or ASNs** at each month, i.e., those with at least one inferred AS relationship. Indeed, an ASN might be delegated, but not used in practice.

Finally, note that, we compute the aforementioned sets of ASes month after month for each IXP and country.

[4] `ftp://ftp.lacnic.net/pub/stats/lacnic/`
[5] `data.caida.org/datasets`
[6] `https://asrank.caida.org/doc`
[7] `https://www.peeringdb.com`

3.2 Public policies and IXPs

In this section we investigate the public policies behind the creation of IXPs in Latin America. For this, we rely on the set of digitalized documents we gathered. Table 3.2 shows the organizations that currently run these IXPs and that fostered their creation.

Notably, **out of 16 countries in LatAm, governments were involved in the creation of their national IXPs in more than 55% of the cases.** The president of Costa Rica signed an Executive Order [**CR2**, **CR3**] while parliament in Bolivia passed a law [**galperin2016localizing**]. Also, federal agencies such as Senatics in Paraguay [**PY1**], PUC in Belize [**BZ1**] and SENACYT in Panama [**PA1**] fostered IXP's creation. Regulators were involved in Mexico (IFT) [**MX1**], Honduras (CONATEL-HN) [**HN1**] and Paraguay (CONATEL-PY) [**PY2**]. In Brazil, the Internet Steering Committee (CGI), a multi-stakeholder board with several state representatives, was responsible for creating IX.br, the Brazilian IXP [**BR1**]. Further, Belize, Honduras and Paraguay have delegated IXP operations to universities.

On the other hand, Table 3.2 also indicates that, **similar to the European IXP model [chatzis2013there], in Latin America a large number of non-profit organizations created and run IXPs.** In particular, CABASE (AR) and CCIT (CO) are operated by organizations related to local ISPs associations as it happens in IXPs outside the region, e.g. DE-CIX (DE) [**decix_h**] and JINX (ZA) [**jinx_h**]. In addition, presence of state regulations also influenced the development of peering facilities in Chile. Undersecretary of telecommunications singed Resolution 1483 [**CL1**] in 1999 which forced traffic between Chilean ISPs to be carried by their local infrastructure. To fulfill this requirement, ISPs rapidly joined NAP Chile, a Chilean IXP. More recently, in 2016, PIT Chile was established on top of the dense interconnected infrastructure of NAP Chile, though bringing significant changes to the Chilean peering ecosystem: whereas NAP Chile was strictly limited to domestic ASes, PIT Chile was envisioned as a neutral IXP also allowing the presence of non-national ASes.

Finally, note that most countries that host a BGP monitor have small IXPs (e.g. with less than 30 connected networks that announce less than 2M unique IPs). Since this limits the conclusions that can be drawn in them, **our analysis focuses on CABASE, IX.br and PIT CL, the networks of IXPs in LatAm.**

3.3 IXP networks topology

In this section we study the topology, peering policy and infrastructure of networks of IXPs. We investigate those operated by CABASE, IX.br and PIT-CL in Argentina, Brazil and Chile, respectively. Furthermore, to provide some context, we compare them with the domestic IXPs of DE-CIX, i.e., only with those located in Germany. More precisely, we look at the number of IXPs per network, the number of peering facilities per IXP, and whether regional IXPs have their own ASNs. In addition, we determine if there exists interconnection among regional IXPs, and if IXP operators enforce any peering policy on their members. For all these analysis, we queried PeeringDB to obtain the list of the regional IXPs in CABASE, PIT-CL and IX.br, the codes that operators use to tag them (e.g., IX.br Fortaleza has been tagged as

Country	AR	BO	BR	BZ	CL	CO	CR	CU	EC	HT	HN	MX	PA	PY	PE	TT
Sponsored by	CABASE	Law	CGI	PUC	PIT CL	CCIT	Ex.Ord.	State	IXP.EC	AHTIC	CONATEL	IFT	SENACYT	SENATICS	NAP.PE	TTIX
Operated by	CABASE	State	NIC.br	UoBZ	PIT CL	CCIT	NIC.cr	NAP.CU	IXP.EC	AHTIC	UNAH	CITI	InteRED	NIC.py	NAP.PE	TTIX
BGP TDs Monitor	PCH	×	RVs/LGs	PCH	PCH	×	PCH	×	PCH	PCH	PCH	PCH	×	PCH	×	PCH
BGP TDs #Memb	127		1156	6	72		28		5	4	4	6		15		5
BGP TDs #AggIPs	7.9M		26M	67K	19.4M		401K		28K	102K	131K	795K		1.5M		196K

Figure 3.2: IXPs in Latin America excluding European overseas territories in June 2019. Countries are abbreviated by their ISO-standard code. Colors blue, yellow and magenta represent state agencies, non-profit organizations and universities, respectively. #AggIPs is computed on the address space announced by IXP members (excluding their customer cone and repeated prefixes due to MOASes).

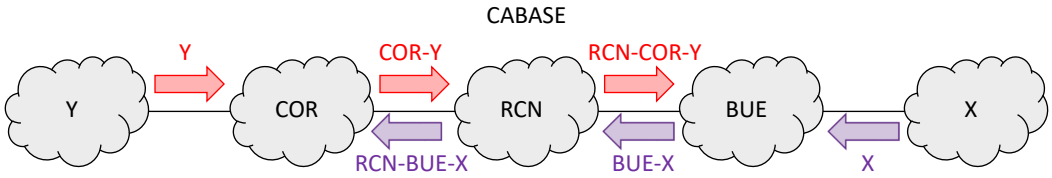


Figure 3.3: Mandatory multilateral peering policy in CABASE. Arrows indicate the direction of BGP announcements and their respective AS path. RCN is a central node that interconnects regional IXPs (e.g. BUE, COR) and forwards all announcements to all regional IXPs.

ce) and AS numbers. We found that, as of July 2019, IX.br, CABASE and PIT Chile run 31, 28 and 6 regional IXPs respectively in Brazil, Argentina and Chile. Table 3.1 summarizes the main characteristics of the IXP networks of CABASE, PIT-CL, IX.br and DE-CIX (only in Germany), which we now detail.

3.3.1 CABASE

From the topological point of view, this network has a central IXP, CABASE-RCN (AS52376). This IXP is located in the area of Buenos Aires, the province including the capital of Argentina. CABASE-RCN hosts PoPs and caches of content providers such as Netflix (AS2906), Google (AS15169) and Amazon (AS16509) [**cabase_rcn**].

Aware of the presence of content providers, network operators outside Buenos Aires city are looking forward to peer in CABASE-RCN. Hence, operators from a similar far region usually (i) build a regional IXP where they peer locally (sometimes actually encouraged by CABASE), and; (ii) hire a L2 carrier that links the regional IXP with CABASE-RCN. This not only reduces the wiring costs that each of these operators would have needed to pay to reach CABASE-RCN, but also allows them to negotiate better bidding rates [**galperin2016localizing**, **cabase_rcn**]. All in all, the network is composed of 28 IXPs, with 9 in the province of Buenos Aires, and the remaining ones scattered across the country. CABASE sponsors all the regional IXPs, and gives each of them an AS number (e.g, BUE-AS11058, COR-AS52374).

At the same time, CABASE applies a mandatory multilateral peering policy (MMPP) such that prefixes advertised in an IXP are announced to all members in the country-wide network via the central node [**pch_cabase_bue**]. This is illustrated in Fig. 3.3 for CABASE-BUE and CABASE-COR. The MMPP has two effects. First, all members benefit from the presence of the content providers peering in CABASE-RCN. Second, the BGP data gathered in only one IXP of CABASE allows to obtain a complete view of CABASE's members, i.e., where they are connected and their customer cones. We further verified this contrasting PCH's BGP snapshots collected in multiple of regional IXPs of CABASE. In particular, we rely on PCH's BGP collector in CABASE-BUE (eze) to study this network of IXPs.

3.3.2 PIT-CL

This network is composed of 5 IXPs. As so does CABASE, PIT-CL identifies each of its IXPs with an AS number (e.g, SCL-AS61522, ARI-AS61527). The larger IXP of PIT-CL is the one in Satiago de Chile, the capital. In particular, after

	CABASE	PIT-CL	IX.br	DE-CIX
CC	AR	CL	BR	DE
#IXPs in CC	28	5	31	5
ASN per IXP	✓	✓	✗	✓
IXP facilities	1/IXP	1/IXP	PIXes	Sites
IXPs Linked	✓	✓	✗	✓
Enforced Policy	MMPP	✗	✗	✗

Table 3.1: Topology and management characteristics of IXP networks across countries (CC). To the best of our knowledge, CABASE is currently the only IXP in the World that imposes a mandatory multilateral peering policy (MMPP) among its members.

its creation, PIT-CL-SCL incorporated four ISP NAPs CenturyLink/Level3, Claro, Entel and Telefonica. These NAPs dated from the late 90s, time when ISPs had to create domestic peering infrastructure to fulfill Chile's SubTel Resolution 1483 obliging Chilean traffic to be exchanged locally [**CL1**]. By inspecting the BGP data in PCH's BGP collector Santiago de Chile (SCL), we discovered that all the remaining IXPs appear as members in PIT-CL-SCL. Hence, as with CABASE, just looking at the BGP tables of PIT-CL-SCL, we were able to analyze all regional IXPs of PIT-CL. Finally, in contrast with CABASE, PIT-CL does not promote any specific peering policy [**pch_pit_cl**].

3.3.3 IX.br

The regional IXPs of IX.br are, in general, composed of multiple interconnected peering facilities called PIXes [**ixbr_cix_2016**, **caf**, **nicbr_pix_cix**]. In general, each IXP has a central PIX to which other PIXes are linked. The PIXes may be scattered around the metropolitan area where a regional IXP is placed, e.g., IX.br Sao Paulo (SP) comprises 35 of them. This contrasts with CABASE and PIT-CL, that operate a single peering facility per IXP. In addition, there are port resellers at IX.br, called CIXes, that offer L2 transit towards PIXes [**ixbr_cix_2016**]. In total, IX.br consists of 31 IXPs distributed across 18 states of Brazil. In particular, the state of Paraná, with 5 IXPs, is the one that holds more. Interestingly, as opposed to both CABASE and PIT-CL, the IXPs of IX.br are not linked, nor do they have different AS numbers that identify them. IX.br does not impose any peering policy to their members [**ixbr-peering-policies**]. Finally, note that we usually use Routeviews data to analyze IX.br-SP, but when we compare the regional IXPs of IX.br, we rely in the BGP data dumped in IX.br's looking glasses.

3.3.4 DE-CIX

This network counts with 22 IXPs distributed in different countries, out of which 5 are located in Germany [**DECIX-locations-webpage**]. Each regional IXP has its own AS number. In general, DE-CIX has multiple peering facilities per IXP, called sites. These are sub-divided in enabled sites, data centers where DE-CIX owns the hardware, and access sites that belong to carrier partners [**DECIX-connecting**].

ASes willing to join the IXPs may directly peer in these locations, or reach them remotely via L2 carriers, either DE-CIX resellers, DE-CIX transport partners or companies that establish Ethernet long-haul links. The IXPs in Frankfurt, Hamburg, Munich, Dusseldorf, New York, Marseille, Madrid, Lisbon, Palermo, and Istanbul are interconnected, such that via a service called "GlobePEER Remote", members in any of these IXPs can peer remotely at all these locations via VLANs. In addition, members in the IXP of Berlin have access to the announcements in Frankfurt, and vice versa. Finally, DE-CIX may filter invalid announcements, but does not impose any peering policy for their members [**DECIX-PP**, **DECIX-Filtering**].

3.3.5 Takeaways

In conclusion, the fact that no row in Table 3.1 has a unique constant value across all columns highlights that networks of IXPs are diverse. Indeed, the characteristics, in general, depend on the particular network analyzed. An important point to notice is that IX.br and DE-CIX, tend to have multiple peering facilities per IXP, but CABASE and PIT-CL do not. As we will see, the first two are networks of IXPs with numerous members, in particular many more than the latter two. Finally, we believe that, in general, IXPs do not impose any peering policy on their members. Therefore, in this sense, CABASE represents an interesting case of study.

3.4 IXPs: domestic, regional or worldwide?

Many of the IXPs in Latin America have already been running for years. Consequently, we aim to understand whether these IXPs have been able to consolidate in their region, as so have others in different geographical areas. We are also interested in how many foreign networks are attracted to Latin American IXPs. For this, we look at IXP members and visible ASes of each IXP in Sec. 3.4.1 and Sec. 3.4.2, respectively. Indeed, despite some ASes might not be members of the IXP, they might still indirectly benefit from it via their providers. Moreover, we want to understand if IXPs in other regions show similar behaviors.

3.4.1 IXP members

Fig. 3.4 displays the number of members and the fraction that are dual-stack adopters across regional IXPs of IX.br, CABASE, PIT-CL and DE-CIX. **We observe that the size of IX.br San Paulo (sp) is remarkable: with 1294 members, it holds more than DE-CIX-Frankfurt (824) and is also around an order of magnitude larger than CABASE-BUE (138) and PIT-CL-SCL (91).** Besides IX.br-SP, we note that IX.br operates other IXPs with more than 100 members. These are Rio de Janeiro - rj: 304, Fortaleza - ce: 219, Porto Alegre - rs: 175, Curitiba - pr: 113). For the remaining networks of IXPs in LatAm, this only holds for CABASE-BUE. Notably, IX.br's largest regional IXPs are of comparable size, or even hold more participants than DE-CIX Dusseldorf (141), Hamburg (134), Munich (130) and Berlin (95). On the other hand, concerning the fraction of members that are dual-stack adopters, while CABASE and PIT-CL at most reach 0.5 in Fig. 3.4, the values for all regional IXPs of IX.br except Brasilia and San Jose

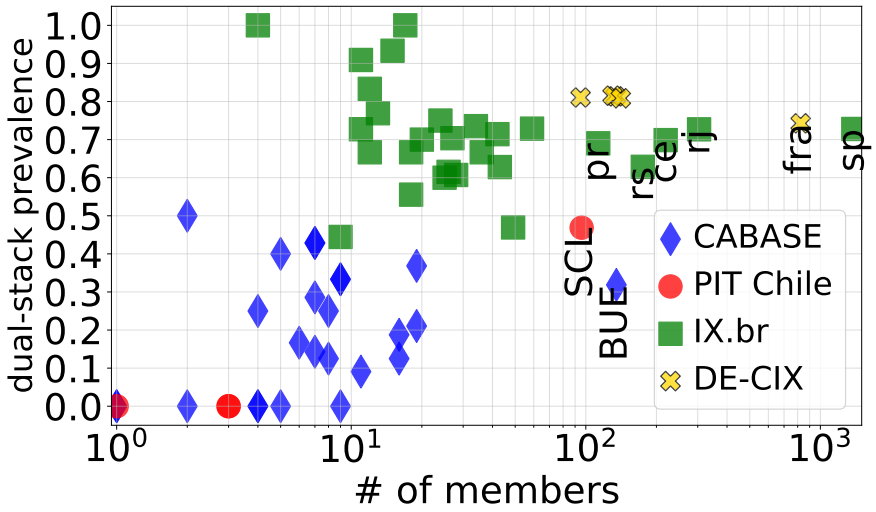


Figure 3.4: Number of connected networks and dual-stack adoption across regional IXPs in IX.br, CABASE, PIT Chile and DE-CIX in June 2020.

dos Campos are over this threshold. This evidences that IPv6 adoption in Brazil is a nation-wide process not just limited to largest IXPs. Lastly, we note that IX.br-SP and IX.br-rj and DE-CIX-fra have a similar dual-stack prevalence, around 0.75. This finding evidences that IPv6 adoption in Brazil is a nation-wide process and it is not just limited to the largest IXPs. Lastly, we note that the IXPs of IX.br in San Paulo and Rio de Janeiro, and that of DE-CIX in Frankfurt have a similar dual-stack prevalence, around 0.75.

We are interested in the common properties across the larger regional IXPs in LatAm. **We note that CABASE-BUE, PIT-CL-SCL, IX.br-SP and IX.br-rj are located in large cities (21.3, 6.3, 15.3 and 5.6 million inhabitants respectively in Sao Paulo, Rio de Janeiro, Buenos Aires and Santiago de Chile), economically central in their respective countries.** In addition, we note that these IXPs usually host CDNs and include renown transit providers among their members. For example, in CABASE-BUE and PIT-CL-SCL we find AS13335-Cloudflare (CDN) and AS3549-CenturyLink/Level3 (Tier-1 transit provider). Note that CABASE-BUE used to host more CDNs, but after CABASE-RCN was deployed, most CDNs peer now in the latter IXP. On the other hand, concerning the IXPs of IX.br, in Sao Paulo and Rio de Janeiro, we note the presence of several CDNs such as AS15169-Google, AS16509-Amazon, AS54113-Fastly (also present in Curitiba) and AS13335-CloudFlare (also present in Curitiba, Porto Alegre and Fortaleza). We also find several large international transit providers such as AS2906-NTT and AS3303-Swisscom peering at the IXP in Fortaleza. In addition, we see AS37468-Angola Cables at Fortaleza (until May 2020) and Sao Paulo, and AS4809-China-Telecom at Sao Paulo, Rio de Janeiro and Curitiba. Angola Cables deployed a transatlantic cable between Angola and Brazil, and only then irrupted in the Brazilian AS ecosystem [**fanou2020unintended**]. Similarly, China Telecom's map also displays submarine connectivity in the Brazilian shores [**chinatelecom**].

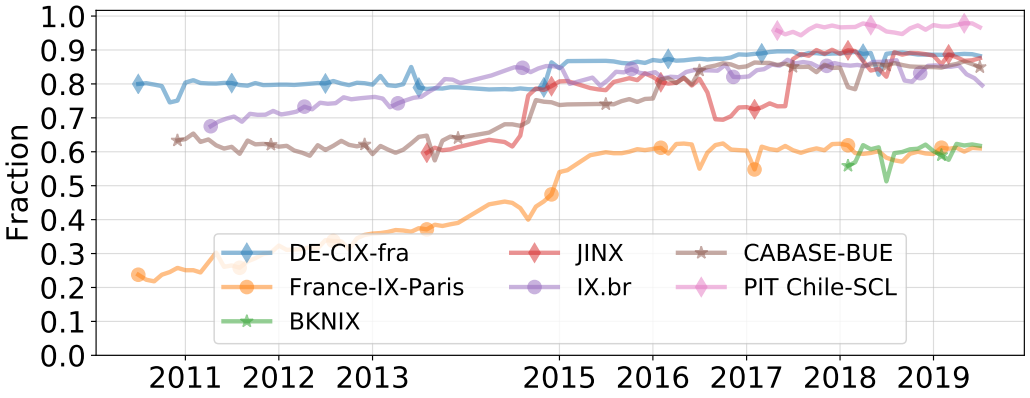


Figure 3.5: Fraction country's delegated-and-active ASNs visible at the IXPs.

3.4.2 Visible ASes: domestic impact and foreign attraction

Fig. 3.5 displays the ratio of local visible ASes to *all* delegated-and-active ASNs for IX.br-SP, CABASE-BUE and PIT CL-SCL in Latin America, and France-IX, DE-CIX, JINX and BKNIX from other regions. To compute this graphic, first we determine *all* delegated-and-active ASNs for each country, and thus for each IXP. For this, we simply filter out delegated but inactive ASNs, i.e., ASNs with no inferred AS relationship. Then we look for ASes that: i) are visible in each IXP and; ii) are local, i.e., own an ASN delegated to the country where the IXP is deployed. Finally, we compute the ratio between both values month after month.

Fig. 3.5 reveals that 80% of the Brazilian and Argentinian country delegated-and-active ASNs are visible at IX.br-SP and CABASE-BUE, respectively. This fraction is similar to the one observed in DE-CIX (Frankfurt) and by far larger than in France-IX (Paris), despite the large wealth gap (i.e. GDP per capita) between the European Union and Latin America [**gdp_worldbank**]. Indeed, **even though LatAm spans a larger geographical extension than Europe, IXPs of the region have still managed to deploy an infrastructure that allows them to host a large fraction of their local ASes.** In addition, while DE-CIX has been stuck in this fraction value since 2011, CABASE-BUE and IX.br-SP have been steadily growing since the beginning of the decade when they just had around 60%. The Brazilian IXP network growth in the past decade was driven by the investments in telecommunications to host the 2014 FIFA World Cup as well as the 2016 Summer Olympics [**BR0**, **BR2**]. On the other hand, CABASE's fraction of visible ASes, as well as number of regional IXPs, has increased since Google joined the IXP in late 2011.

In addition, Fig. 3.5 also shows that PIT Chile-SCL, that started operating in 2016, has a striking fraction of 90% even from the first snapshot we got from the PCH collector in 2017. This is the highest historical value in Latin America, and indeed high for an infant IXP: for example, BKNIX, which was launched in 2015, covers just 60% of the current delegated-and-active ASNs in Thailand. To grow rapidly, PIT Chile leveraged Chilean public policies (see Sec. 3.2).

Finally, note that JINX, the IXP in South Africa, has also been increasing the fraction of visible country delegated-and-active ASNs over time. The similarities

with the IXPs in Brazil and Argentina in terms of the same 20% of increase and the fact that the three IXPs have reached a value comparable to a big IXP such as DE-CIX, allows to speculate on a matureness process that replicates across continents: **regions where the Internet is rather underrepresented seem to, after many years, have been able to attract as many local ASes as some well-established IXPs in Europe.**

On the other hand, Fig. 3.6 shows[8] the prevalence of AS nationalities at each IXP, i.e., out of all visible ASes in an IXP, how many come from each country. As can be seen, the three bigger Latin American IXPs mainly provide local support: the largest fraction of visible ASes, around 75% in all cases, are from the countries where the IXPs are deployed. However, these IXPs are also able to extend to other countries in the region, which usually add up most of the remaining fraction in Fig. 3.6. These results are similar to the ones seen in BKNIK and JINX. Indeed, all these IXPs are not so internationally widespread, i.e., the ASes they host come from less than 50 different countries in all cases. All this is in clear contrast with what happens in European IXPs that rather act as international hubs: not only the number of visible nationalities is greater than 100 for France-IX and over 200 for DE-CIX, but also most of their visible ASes are actually not local regarding to where the IXPs are deployed. Despite these differences, it is remarkable that the US is always within the five most prevalent AS nationalities[9] for all IXPs: this is likely due to the advertisement of prefixes of relevant US-based companies (e.g., Google, Facebook, Netflix, CloudFlare, Fastly). Indeed, the fact that CDNs find in IXPs a way to remain close to their customers and to offer them a better service is particularly also true in Latin America, Asia and Africa.

3.5 Reaching IXPs: from stubs to large transit providers

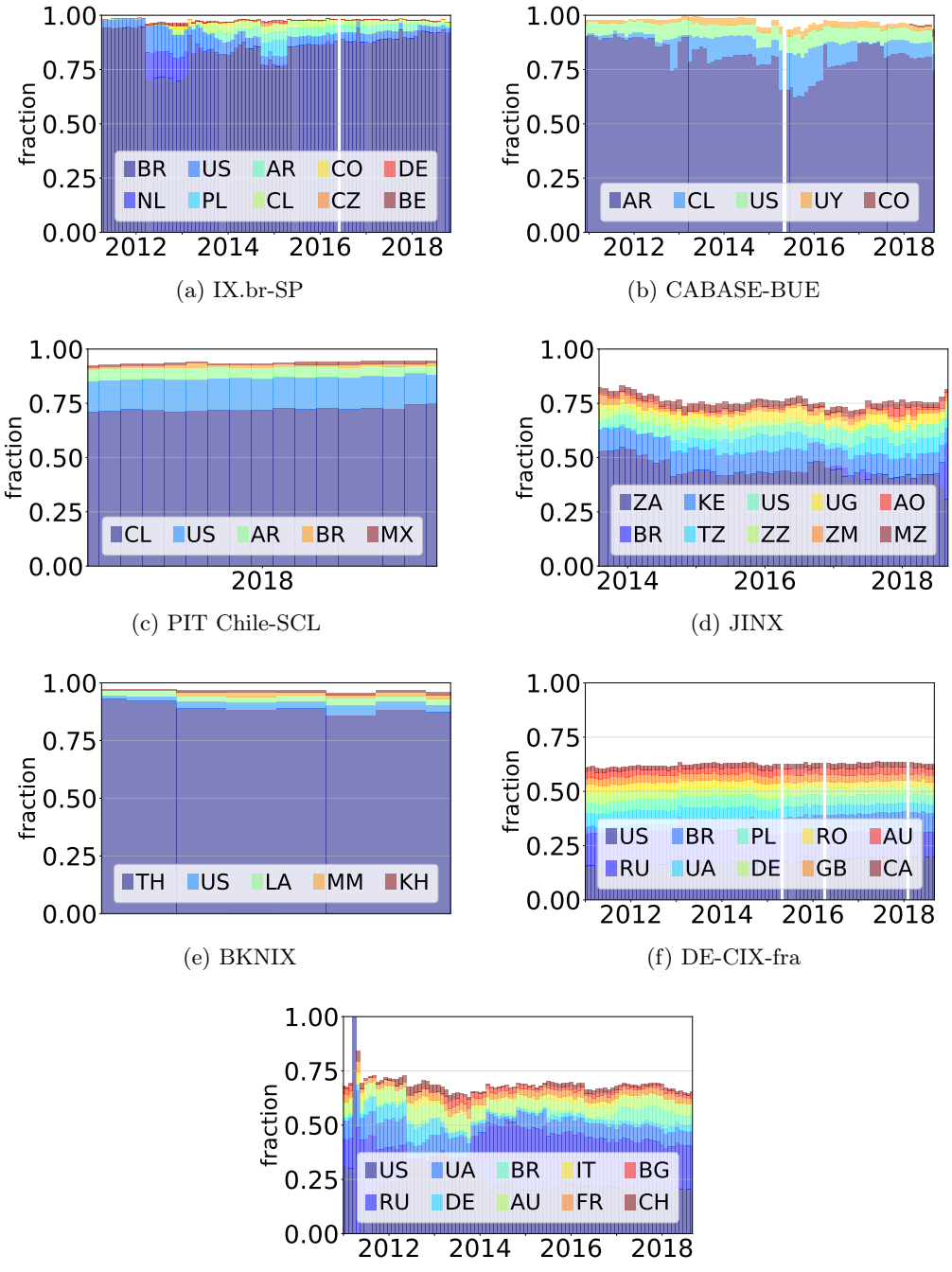
We are interested in how traffic is carried from/to Latin American IXPs. More precisely, since ASes in LatAm could be potentially scattered throughout vast geographic extensions, we would like to identify transit providers that have contributed to the consolidation of IXPs in their local country. We argue that finding large transit providers in an IXP might encourage other ASes to join it. On the other hand, ASes usually advertise their customer cone at IXPs. However, we wonder whether there exist non-transit members, i.e., ASes that only announce prefixes owned by themselves in the IXP. Interestingly, we find that there exist, and that this set of members is not only composed by stub ASes: there are transit ASes actively choosing not to announce prefixes of their customers.

3.5.1 Transit members

Fig. 3.7 shows a heatmap of the number of transit members with AS-rank within each reported interval for each IXP. We argue that this metric can capture the interest of large transit providers in participating at the IXPs.

[8] For this analysis, we filtered out the large number of prefixes announced by Hurricane Electric (AS6939), probably just on account of its open peering policy **[giotsas2015ipv6]**, in IX.br, JINX, DE-CIX and France-IX.

[9] By nationality we mean an AS that have been delegated to the US



(a) IX.br-SP

(b) CABASE-BUE

(c) PIT Chile-SCL

(d) JINX

(e) BKNIX

(f) DE-CIX-fra

(g) France-IX-Paris

Figure 3.6: Prevalent AS nationalities at IXPs in Latin America, Africa, Asia and Europe.

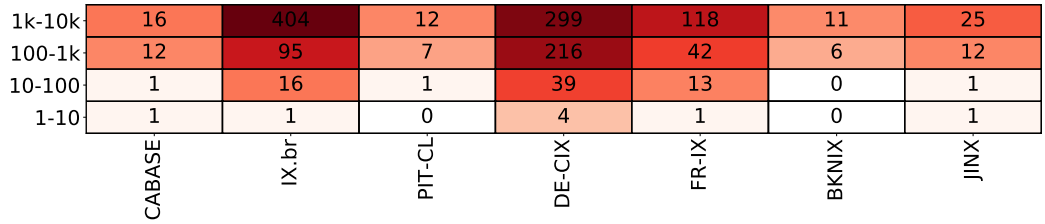


Figure 3.7: Relevance of IXP members in the global transit system. Bins represent an interval for CAIDA's AS-RANK while number on each cell counts the number of members within that interval.

Large European IXPs, are notably attractive for transit providers at the top of AS-rank since the number of transit members in the TOP10 peering at DE-CIX, AMS-IX and LINX is 4, 3 and 3, respectively. In addition, more than a third of transit providers in the TOP100 are members of these European IXPs. On the other hand, IX.br only hosts one transit member in the TOP10, however, when we extend the limit to ASes in TOP100, IX.br hosts 17 ASes within that range. We manually examine the 17 TOP100 transit providers peering at IX.br-SP and we found that 13 out of 17 are ASes that have been delegated to Brazilian organizations.[10] Furthermore, we counted the number of Brazilian ASes in the AS-rank TOP100 finding 18 transits providers, only 5 of which do not peer at IX.br-SP. Lastly, CABASE and PIT Chile, countries with no domestic transits in the AS-RANK TOP100, have been able to attract to 2 (Telecom Italia-6267 and Level3-AS3549) and 1 (Internexa-262589) foreign transits in TOP100, respectively.

Despite an AS might have a high AS-rank, this does not imply that it announces a large number of downstream ASes in the IXP. To shed light on this, Table 3.2 displays the five IXP members that announce the largest visible customer cones in IX.br-SP, CABASE-BUE and PIT Chile-SCL. Results show a richer AS ecosystem in Brazil: Algar (AS16375) alone announces more downstream ASes in IX.br-SP than all the visible ASes seen in CABASE-BUE as well as in PIT Chile-SCL. On the other hand, looking at the nationality of the TOP5 upstream ASes in each IXP, we see mainly domestic transit providers. Yet, there are exceptions: Internexa (AS262589, Colombia) and Silica (AS7049, Argentina) in IX.br, Level3 (AS3549, US) in CABASE-BUE and Internexa (AS52880, Colombia) in PIT Chile-SCL.

In addition, Table 3.2 shows that Level3 is the largest upstream AS in CABASE-BUE (AS3549) and, though not displayed in Table 3.2, also ranked sixth in PIT Chile-SCL (AS21838, legacy number of an acquired network [**impsat**]). We further investigated Level3's role in both IXPs and determined that this ISP actually acts as a domestic transit provider in LatAm: 204 out of 219 and 37 out of 43 downstream ASes announced by Level3 in CABASE-BUE and PIT Chile-SCL were delegated by LACNIC to Argentina and Chile, respectively.

Finally, Table 3.2 also unveils the presence of state-owned ISPs among the largest upstream ASes: Internexa (AS262589, AS262195) and ARSAT (AS52361). Internexa is a partially state-owned Colombian AS in which the Ministry of Finance and Public Credit holds 51% of the shares while Medellin county (Colombia) holds

[10]The other four ASes are: HE-6939, ACS-37468, Seabras-13786, China Telecom-4809.

IX.br-SP	ASN	16735	262589	7049	61832	28329
	#	903	381	218	209	207
CABASE-BUE	ASN	3549	52361	7049	19037	11664
	#	219	113	100	82	81
PIT Chile-SCL	ASN	7004	22661	52280	19228	14259
	#	88	87	70	57	57

Table 3.2: Largest sizes (#) of visible AS sets *per upstream AS* in IX.br-SP, CABASE-BUE and PIT Chile-SCL.

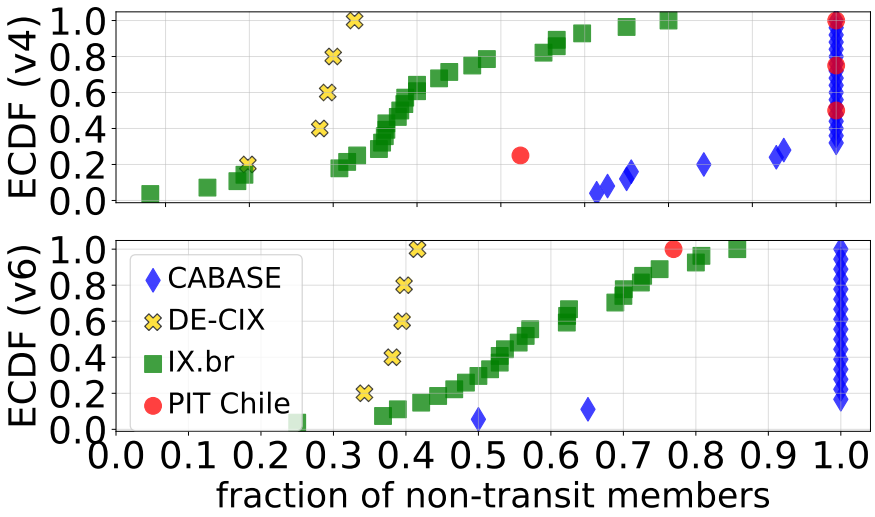


Figure 3.8: Fraction of non-transit members, i.e., that only announce prefixes owned by themselves at the IXP.

another 10% [**isa**]. On the other hand, ARSAT (AS52361) is a fully state-owned Argentinian transit provider [**arsat_jefatura**]. Note that, while ARSAT's transit service focuses in Argentina, Internexa's transit footprint comprises foreign countries, such as Argentina and Brazil.

3.5.2 Non-transit members

Fig. 3.8 presents the ECDF of the fraction of non-transit members across the IXP neworks of IX.br, CABASE, PIT-CL and DE-CIX, for IPv4 (top) and IPv6 (down). For IPv4, the median values across IX.br and DECIX are 0.48 and 0.39, respectively. Remarkably, the values are exactly 1 for CABASE and PIT-CL, which seems to confirm that these IXPs are mainly populated by small operators. In fact, 18/28 and 3/4 of CABASE's and PIT-CL's regional IXPs, respectively, only host non-transit members. On the other hand, we observe that the prevalence of non-transit members tends to be higher in IPv6. It is worth noting that CABASE has less dots in IPv6 when compared to IPv4 since in some regional IXPs, e.g. CABASE-SFE, we did not find any member announcing IPv6 prefixes. Similarly, in PIT-CL, we only note IPv6 operations in PIT-CL-SCL.

To gather more insights, we refine the classification in Fig. 3.8 by differentiating

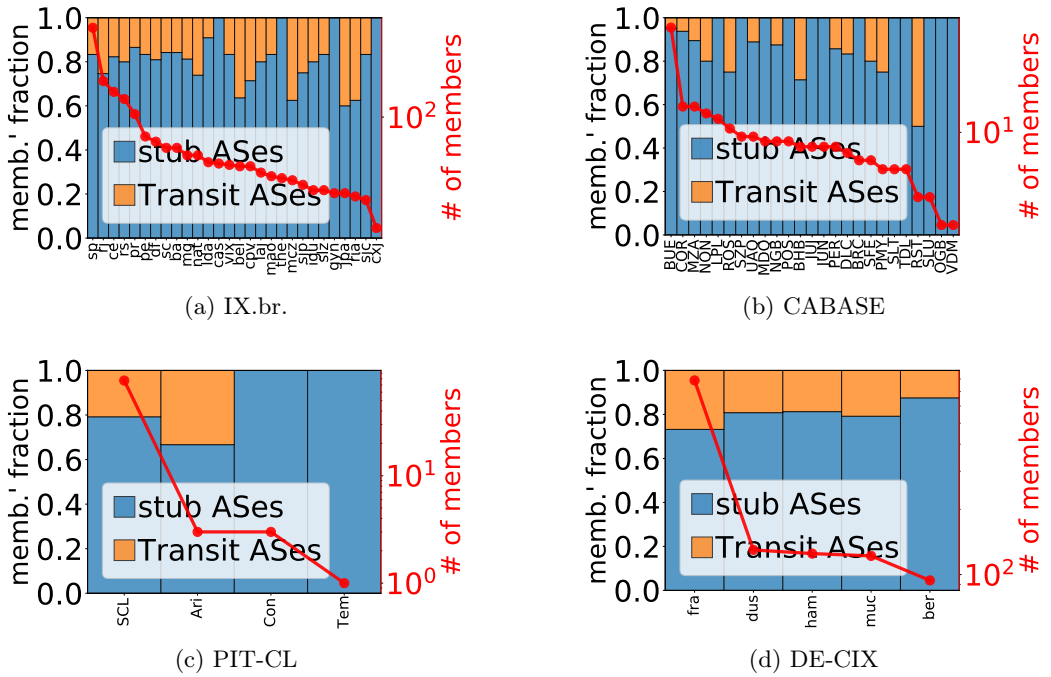


Figure 3.9: Classification of non-transit members across the networks of IXPs into stub or transit ASes for IPv4.

among non-transit members that are either transit or stub ASes, i.e., ASes with or without customers, respectively. Note that transit ASes that are non-transit members are ASes that deliberately choose not to announce their customers at the IXPs. Recall that we use AS-relationship files to identify stub ASes, and since these do not take into account IPv6, we are only able to spot them for IPv4.

Fig. 3.9 shows the prevalence of stub and transit ASes among non-transit members in IX.br, CABASE, PIT-CL and DE-CIX networks. In particular, the regional IXPs are sorted by number of members. We see that despite non-transit members are mostly stub ASes, there are some that are actually transit ASes not announcing their customer cone. In other words, transit ASes seem to benefit themselves from the announcements in the IXP, but do not seem to offer this service to their customers. We argue this case is worth of study in future work, as it may result from complex AS relationships among ASes. However, for regular customer-to-provider relationships, i.e., if transit ASes advertise the paths they learn in the IXP to their respective customer cones[11], this practice may likely lead to asymmetrical forward and return paths for customers of these non-transit members. The asymmetry arises if: (i) the non-transit member chooses an announcement done by another member in the IXP as having the best path towards a prefix; (ii) the non-transit member announces this path to its customers, some of which choose it as best path too. This way, traffic originated in customers flows via the IXP, but on the opposite direction this does not hold since the non-transit member does not announce its customer cone at the IXP. Finally, we verify that in the Latin American networks of

[11] An AS could filter such announcements, e.g., relying on BGP communities.

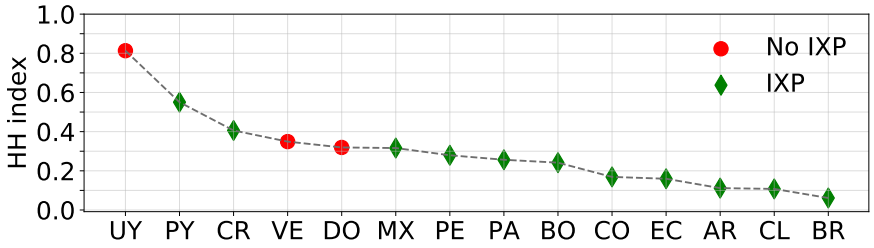


Figure 3.10: Herfindahl-Hirschman Index to determine originated address space concentration in countries that have been delegated more than 1M IP addresses.

IXPs, most IXP members peer in only one IXP, therefore not announcing customers does not seem to relate to ASes applying traffic engineering techniques in between regional IXPs.

3.6 IXPs and concentration

In this section we provide a plausible explanation of why some IXPs in LatAm are succesfull while others have failed. We argue that the presence of monopolistic ASes may discourage the deployment and growth of IXPs. Hence, we look whether the IPv4 address space delegated to Latin American countries is fairly distributed, i.e., if no AS owns most IP addresses assigned to a country.

We compute the address space announced by each AS according to what is reported in the prefix-to-AS files, and the total of each country aggregating that of their domestic ASes. To measure the fairness of the market, we use the Herfindahl-Hirschman Index (HHI), a statistical measure of concentration that ranges from 1 (single monopolistic origin) to 0. This metric is used by the US Department of Justice to apply antitrust regulations [**rhoades1993herfindahl**] and in ecology, known as *Simpson's Diversity Index*, to measure diversity.

Fig. 3.10 displays the HHI for Latin American countries delegated more than 1M IP addresses. The right end shows countries with low concentration ratio, such as Brazil, Chile and Argentina. Indeed, these countries host the largest IXP networks. On the contrary, the left side includes countries such as Uruguay, Dominican Republic and Venezuela, that do not have any IXP, and Paraguay, Costa Rica and Mexico, all possessing an HHI of more than 0.3.

We take Uruguay, Venezuela, Costa Rica and Mexico as cases of study and display in Table 3.3 the first and second dominant ASes that concentrate most of the IPs delegated to these countries. In all cases, the first dominant AS not only originates between 55% to 90% of its respective national address space, but also owns at least 47% more than the second. In particular, countries dominated by large state-owed providers such as Venezuela (CANTV) and Uruguay (ANTEL) are not even planning to release an IXP [**FreedomHouseVE2018**, **UY1**]. Costa Rica is the opposite example: while the state owns ICE, the main ISP that originates 63% of the national address space, the first national IXP was created by an executive order in 2014. Remarkably, ICE has never joined the IXP [**CR4**]. Mexico is another country with high HHI whose IXP just has 6 members. We suspect that, despite

	UY		VE		CR		MX	
ASN	6057*	19422	8048*	6306	11830*	52228	8151	13999
ip-cnt$_{cc}$	2.38M		5.15M		2.42M		24.9M	
ip-cnt	2.15M	90.1k	2.84M	629k	1.52M	197k	13.7M	2.05M
ip-frac	0.90	0.04	0.55	0.14	0.63	0.08	0.55	0.08

Table 3.3: The two largest origin ASes per country. $*$ indicates state-owned ASes.

the fact that the creation of the IXP in 2014 was sponsored by the Mexican government as a recommendation of the OECD [**OECDMX2**], the absence of Telmex (AS8151) [**Telmex2**], by far the first dominant AS in the country, discouraged the IXP growth.

In general, therefore, **we see that the countries where no IXP at all exist or they have failed to attract members have a higher HHI than those where IXPs have proliferated.**

3.7 Conclusions

This study contributes multiple findings regarding to Internet topology research:

- We are the first to study in depth the deployment of IXPs in Latin American, and the AS ecosystem in the region. To carry out this analysis, we construct the most comprehensive available BGP dataset of Latin America, which we complement with additional data sources. We release the code to replicate the analysis and to download the files we use.

- We find that Latin American states have been involved in the creation of national IXPs in several ways: legislation, regulation, sponsoring, funding, operations and serving traffic from/to IXPs. In many cases, similar to European IXPs, IXPs in LatAm are managed by non-profit organizations.

- We discover three consolidated IXPs, IX.br-SP, CABASE-BUE and PIT Chile-SCL, that gather mainly local but also regional ASes. These IXPs belong to networks of IXPs, similar to that of DE-CIX in Germany. We compare the characteristics of these networks and see heterogeneous deployments. In particular, we find that CABASE, to the best of our knowledge as no other IXP, establishes a multi-lateral mandatory peering policy forcing its IXP members to announce prefixes to all the remaining members in the IXP.

- We compare the Latin American IXPs of Argentina, Brazil and Chile with others deployed in other continents and find that some IXPs in developing regions not only have had a similar growth in the last years, but also seem to have reached matureness, i.e., have been able to attract as many local ASes as so do some well-established IXPs in Europe. However, European IXPs have also managed to gather members from different regions, a market that could be exploited in the future by the less renown, and rather local, IXPs in Latin America, Asia and Africa.

- We find that transit providers peering at the IXPs in Latin America are mainly regional, but also find large international transit providers providing local service to domestic ASes in LatAm. Besides this, we also find non-transit members, many of which are not stub ASes, but rather transit ASes that actively decide not to announce the prefixes of their customers at the IXPs, presumably representing a previously unreported peering policy.

- We study the correlation between the existence of ASes concentrating address space and the development and consolidation of IXPs. We notice that, in several Latin American countries, the existence of monopolistic ASes, some state-owned, seem to have prevented the proliferation of IXPs, and lead them to be failed IXPs.

We believe there are several studies that could enlarge our understanding on the status of the Internet in Latin America. In that sense, we want to study the deployment of CDNs in LatAm, and their co-location at IXPs. Moreover, we would like to compare peering policies across IXPs in all countries of LatAm. In addition, we would like to study the existence of multi-location members, i.e. members that peer at multiple IXPs of the same network of IXPs. Finally, we specially believe that it is worth to take closer look at Brazilian AS ecosystem, which represents 75% of LACNIC active ASes and has the largest IXP in the world.

In a longer term, despite the vantage points in the region are still scarce, we would like to design active-measurement campaigns to investigate Latin America's access to content and the role of IXPs in such phenomenon. At the same time, we want to investigate the IPv6 rollout in LatAm. Lastly, we are interested in extending our analysis to a larger number of countries, to have a wider perspective of how IXPs contribute to their domestic AS ecosystem.

Chapter 4

Filtering the Noise to Reveal BGP Lies

Contents

In this chapter we discuss the methodology we propose to detect highly-potential BGP lies by eliminating all sources of errors interfering with the collected data, i.e., by filtering noise affecting the comparison of CPs and DPs. While most of the related work essentially blames the IP-to-AS mapping for the observed discrepancies between CPs and DPs, our work relies on conservative heuristics that remove the noise in the measurements and the mapping errors. The mismatches we find after applying our filters show that the IP-to-AS mapping is not the only culprit for them. In short, our contributions are:

1. We study multiple cases of studies showing different causes that may lead to BGP lies in Sec. 4.1. In particular, these examples exemplify why pinpointing the root cause of BGP lies, a problem beyond the scope of this thesis, is challenging.

2. We model in Sec. 4.2 the different practical challenges that need to be addressed in order to be able to detect BGP lies. This ranges from the need of time synchronizing measurements, being able to measure in a platform where

both control paths and data paths can be collected in the same network, up to describing the multiple sources of noise that may interfere in the comparison of CPs and DPs, i.e. AS siblings, TPAs or IXPs and missing hops.

3. We develop a methodology allowing to compute the rate of BGP lies for a given vantage point in Sec. 4.3. Our modular framework has three steps:

   - a preparation stage that synchronizes CPs and DPs in time and semantic level, i.e., converts DPs from IP-paths to AS-paths;
   - a mapping relaxation stage that looks at DPs and CPs separately and tries to infer the noise affecting each of them. In this task, we uniformize AS siblings assigning them a unique mapping for both CPs and DPs, and convert possible TPAs affecting DPs into wildcards that may be mapped to any AS. The filtering rules and the order in which they are applied in this step can be modified, thus allowing to implement different noise-filtering models;
   - a wildcards correction stage that, while comparing CPs and DPs, attempts to infer values of the wildcards (including missing hops) present in the DPs (if any), and determines whether CPs and DPs match or mismatch.

4. We deploy 8 co-located vantage points, 6 in the PEERING Testbed and two in private networks, and carry out a long-term search of BGP lies in Sec. 4.4. To the best of our knowledge, our analysis is the first to extend over time and to deploy such large number of vantage points for a comparison of DPs and CPs.

5. We sanitize the dataset with different noise-filtering models we construct with our framework and compute the rate of BGP lies we find in the wild for each of them in Sec. 4.5. Our most conservative model, i.e., the one that filters more aggressively the noise and allows to obtain a lower bound of BGP lies, reveals a non-negligible amount of highly-potential lies. In addition, we find that in the vantage points where our framework is very effective to reduce the number of mismatches between CPs and DPs, the results usually remain quite stable over time. On the other hand, when our framework reveals a large number of potential BGP lies, the results have a larger per-day variation.

6. We release the dataset we collected and our code to foster replicability and reproducibility[1].

In addition, Sec. 4.6 draws final remarks of our study. The work presented in this chapter lead to the following publications:

- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *Filtering the Noise to Reveal Inter-Domain Lies*, in 2019 Network Traffic Measurement and Analysis Conference (TMA), pages 17–24, 2019, IEEE.

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

- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *A BGP-lying Tale: Stop Blamming the Mapping*, poster presented in TMA 2018.

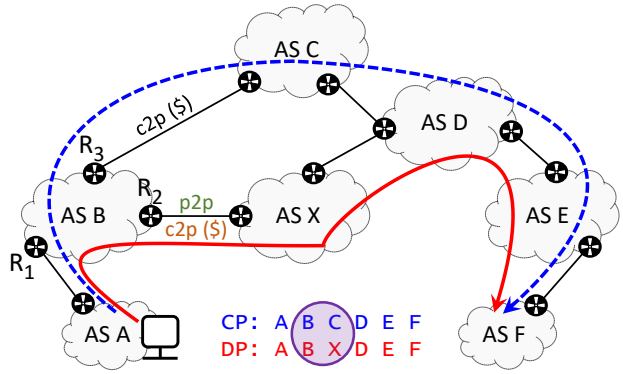
4.1 Modeling BGP lies

BGP lies may be rooted in different causes, and alter either DPs or CPs. The first include lies from an interested AS willing to save money avoiding to use a customer-to-provider link by using a peer-to-peer link, technical limitations, etc. On the other hand, the latter stems from AS poisoning and AS deletions, among others. Fig. 4.1 illustrates these type of lies assuming that Gao-Rexford policies are verified. In all cases, we consider that traffic is flowing from A towards a prefix owned by F. Although the traversed DP is *always* equal to $ABXDEF$, the CP obtained via BGP varies, as well as the reasons why it mismatches the DP.

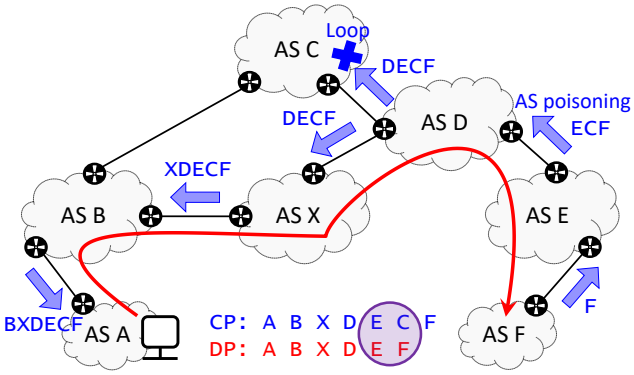
Focusing on those lies manipulating DPs, Fig. 4.1a shows a case where an interested lie occurs (green link) or a technical limitation (orange link) leads CPs and DPs to differ. In both cases we assume X learns the path $XDEF$ towards the origin F. As a first example, if B and X are engaged in a peer-to-peer relationship (link indicated in green), X does not export this path to B. Hence, B and A can only reach F via C: the CP for A is $ABCDEF$. However, in an attempt to avoid paying for transit, B assumes that X knows a path to reach F, and forwards it the traffic, e.g. using a static route. If X does not filter any traffic it receives from B, then DP equals $ABXDEF$ and differs from CP. In this scenario, B carries out a deliberate lie against A and X. On the other hand, if instead B is a customer of X (link displayed in orange), then B learns two paths to reach F: via X or, as before, via C. Since both paths have the same length, then R_3 and R_2 opt for the paths via C and X, respectively. Assuming R_1 has a shorter internal path to R_3 than to R_2, then A learns the same CP as before: the traffic should flow from R_1 to R_3 in B, and from there to C. However, because R_1 has a partial-FIB and uses R_2 as default gateway, the traffic finishes exiting B via X, through R_2. In this second example, the same mismatch as before between the CP and DP is now caused by a technical limitation in R_1.

On the other hand, lies affecting CPs, such as AS poisoning and AS deletions, are displayed in Fig. 4.1b and Fig. 4.1c, respectively. With AS poisoning, an AS can interfere with a competitor AS by poisoning it, i.e by prepending the ASN of the latter to the path. In such cases, the competitor AS finds itself already in the path and rejects the BGP update, possibly incurring in a loss of revenue. For example, in Fig. 4.1b, as E poisons C, then only X accepts the path advertised by D. Consequently, A finishes learning $ABXDECF$ as CP. Since C had been artificially added to the CP, the traffic does not actually traverse it, naturally leading to a mismatch between CP and DP. Other manipulations, such as AS deletions in CPs, can be used. As an example, in Fig 4.1c, B advertises to A a path where X previously deleted D and E, then CP equals $ABXF$. However, in practice, the DP crosses extra inter-domain links that do not appear in the CP.

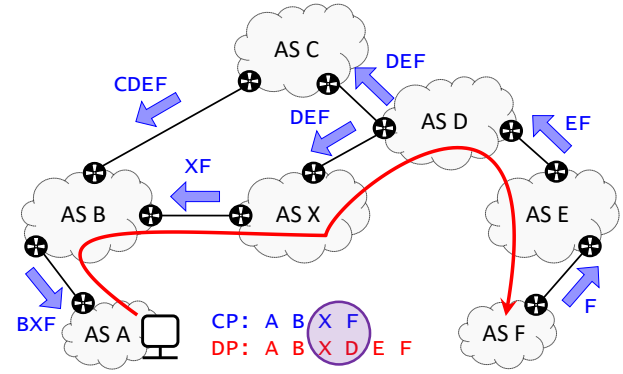
Finally, note that beyond detecting lies, pinpointing their type could be consider itself a different research topic. This is highlighted by the examples in Fig. 4.1, where the same mismatch between CPs and DPs may actually result from different root



(a) Interested lies and technical limitations



(b) AS poisoning



(c) AS deletions

Figure 4.1: BGP lies rooted in manipulations of either DPs (Fig. 4.1a) or CPs (Fig. 4.1b and 4.1c). In Fig. 4.1a, the green peer-to-peer (p2p) link highlights the case in which AS B carries an interested lie, whereas the orange customer-to-provider link (c2p) shows no monetary incentive to lie, and is produced due to technical limitations. On the other hand, while Fig. 4.1b shows a case concerning AS poisoning, adding ASNs that were actually not crossed to CPs Fig. 4.1c focuses on the contrary effect produced by AS deletions, in which ASNs are removed.

causes.

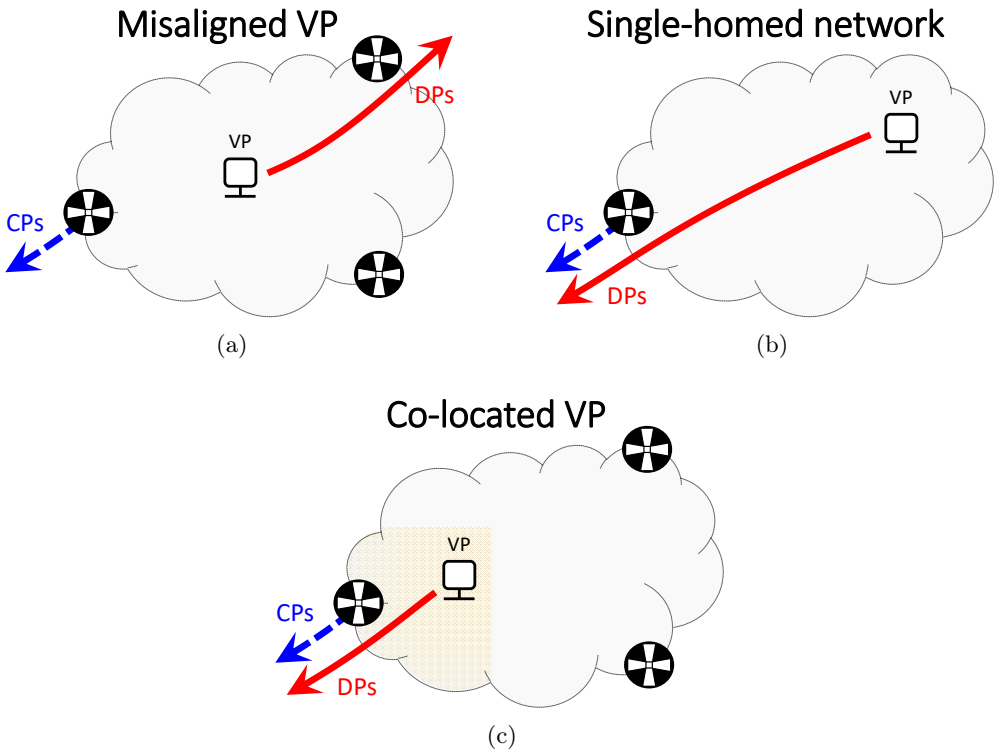


Figure 4.2: Illustration of different vantage points. To compare CPs and DPs, misaligned VPs (Fig. 4.2a) could generate false mismatches if DPs exit the AS through an ASBR different than the one that shares the CPs. A solution would be to only rely on single-homed networks (Fig. 4.2b), where since an AS has a unique ingress/exit point, the location of the VP is not critical. Generalizing the concept of singled-homed networks, co-located VPs (Fig. 4.2c) are those in which, CPs and DPs are ensured to be collected in the same place, and thus the comparison of CPs and DPs is valid. Notice however that, in the latter case, depending on the position of the VP inside the network, the co-located VP could potentially turn into misaligned VP, which highlights the difficulty in obtaining such type of VPs.

4.2 Problem Statement

In practice, to be able to compare CPs and DPs, synchronizing both paths in space and time is mandatory. While achieving time synchronization is simple, and can be done just relying on timestamps extracted from measurements, space-synchronization actually depends on the measuring platform. Both CPs and DPs, obtained from BGP peers and traceroute vantage points (VPs), respectively, need to be measured in the same location, i.e. collected within the same local network. In these cases, we refer to the VP as a co-located VP. For co-located VPs, in theory, DPs should match CPs for all destinations. To better illustrate this concept, Fig. 4.2 shows three cases: a misaligned VP, a VP in a single-homed network and a co-located VP. In the first one, DPs exit through an ASBR that is not the one from which CPs are gathered, potentially leading to a false inference of BGP lies.

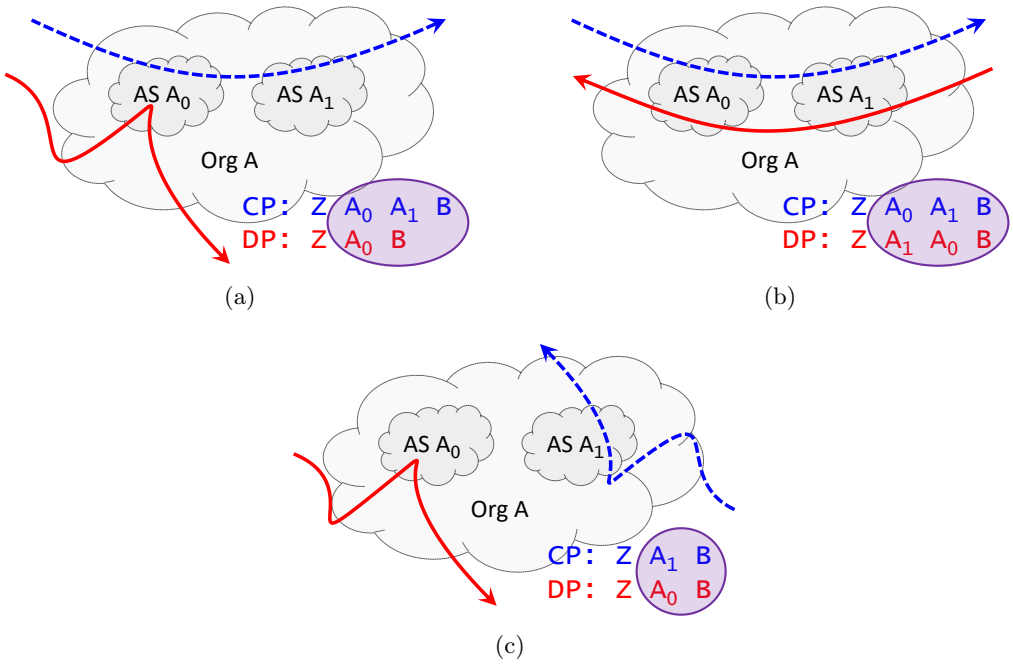


Figure 4.3: Mismatches between CPs and DPs due to noise generated by AS siblings.

On the other hand, single-homed networks ensure this property, but are not easily found in practice. Lastly, co-located VPs are those which, due to the location of the VP, CPs and DPs should theoretically match since packets exit the AS through the AS that shares the BGP data.

In addition to all this, DPs and CPs do not natively come at the same semantic level: the former are collected at the ground IP-level, while the latter are provided by BGP at the AS-level. Therefore, the collected traceroute data has to be converted, with the use of an IP-to-AS mapping function, into AS-level paths. However, this process is noisy and error-prone, and may lead CPs and DPs to mismatch even in the absence of real BGP lies. In general, the noise may either affect the IP-to-AS mapping tool in use, or the output of traceroute itself.

Concerning the IP-to-AS mapping process, IP addresses may either fail to be mapped due to an undefined mapping, i.e., the mapping is not defined or the IP address is mapped to multiple ASes, e.g. due to organizations that use interchangeably the ASNs of the AS siblings they own [70]. The case of AS siblings can generate mismatches between CPs and DPs in different ways, as shown in Fig. 4.3, While CPs may indicate that two AS siblings will be traversed, DPs may only reveal one (Fig. 4.3a), or both but in the inverse order (Fig. 4.3b). In addition, CPs may traverse one AS sibling, but DPs another one (Fig. 4.3c).

Besides the mapping errors, traceroute may provide both unreliable data including third-party addresses (TPAs) [77, 75] possibly due to IXPs [72, 88] or AS boundary allocation policies [73] and incomplete traces including missing hops [70, 86, 87]. Fig. 4.4 illustrates both what TPAs are (Fig. 4.4a) and how they introduce noise affecting the search of BGP lies (Fig. 4.4b). In general, TPAs may occur

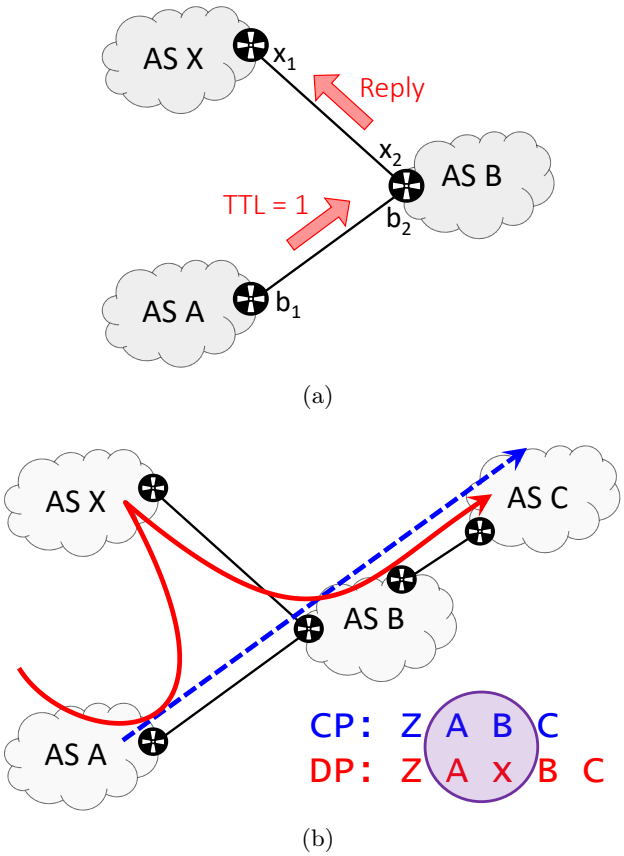


Figure 4.4: Noise generated by TPAs introduced due to IXPs or AS boundary allocation policies. While Fig. 4.4a shows what a TPA conceptually is, Fig. 4.4b illustrates how this becomes a source of mismatches when comparing DPs and CPs.

when a router replies with an interface other than the incoming one, as shown in Fig. 4.4a, where the router in question replies to traceroute using $x2$ as source IP address instead of $b2$. When this IP address belongs to an off-path AS, as is the case in Fig. 4.4b, the DP obtained after IP-to-AS mapping the outcome of traceroute introduces false links, between A and X, and X and B in the example. On the other hand, Fig. 4.1c shows an example where missing hops in traceroute (indicated as "*") do not allow to retrieve complete DPs. However, as we see in the figure, DPs do not match CPs. This particular example highlights that ASes, like AS A in Fig. 4.1c, may not only carry interested lies, but also be malicious ASes and try to hide the evidence dropping traceroute packets.

4.3 A Modular framework to detect BGP lies

The framework we propose to detect BGP lies is illustrated in Fig. 4.6. The three violet boxes are the blocks in which CPs and DPs are modified, and the additional blocks provide the logic and loops needed to carry both the filtering of noise and the comparison of the paths.

First, the **preparation stage** synchronizes CPs and DPs and translates the

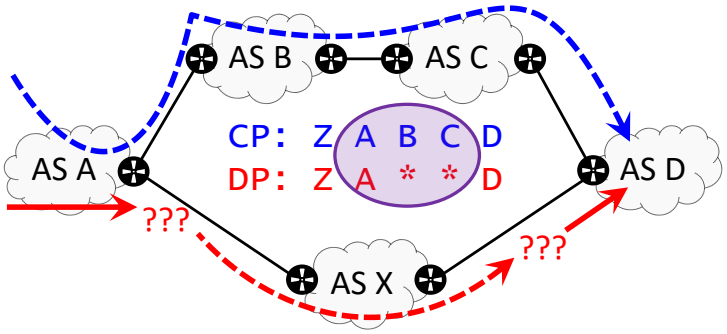


Figure 4.5: Noise generated by missing hops.

Stage	Rule	Issue addressed
Mapping Relaxation	`SIB`	AS siblings
	`looseTPA`	TPAs
	`strictTPA`	TPAs
Wildcards Correction	`match*`	Wildcards
	`nomatch*`	Wildcards

Table 4.1: Summary of the noise-filtering rules that may be applied at each stage.

latter from the IP to the AS level. On the other hand, in the remaining two violet blocks, heuristics are implemented as noise-filtering or path-rewriting rules. These filters or rules aim to mitigate the effects of AS siblings, TPAs and IXPs, and missing hops. In particular, Table 4.1 summarizes the rules applied in each of the blocks after the preparation stage, reporting the addressed issues as well as the actions taken to overcome them. While the preparation stage and the wildcards correction stage are mandatory, the mapping relaxation stage is optional: if the original IP-to-AS mapping is assumed to be immune to AS siblings and TPAs, no complementary rules relaxing, and thus correcting it, are required.

The **mapping relaxation stage** analyzes DPs and CPs separately, and tries to infer the noise resulting from AS siblings and TPAs affecting them. For this, two distinct rules, namely `SIB` and `TPA`, may be applied. The former relies on an AS-to-organization mapping function, while the latter replaces inferred TPAs with wildcards. Two variants exist for the `TPA` rule, either `strictTPA` or `looseTPA`, the latter being more permissive to infer IP addresses as possible TPAs. This step receives its name from the fact that the mapping is relaxed, i.e., we replace arguably inaccuracies of the original IP-to-AS mapping with more general representations, either an ASN used as representative of an organization or with wildcards.

After the mapping relaxation stage and while comparing CPs and DPs, the **wildcards correction stage** takes care of the wildcards in DPs (if any), that result from either missing hops or artificially from the correction introduced in the previous stage. The incomplete sequences of wildcards in DPs are either substituted with their respective missing series in CPs with the `match*` rule , or ignored, i.e. wildcards are deleted when no substitution is possible, according to the `nomatch*` rule. This correction stage embeds the comparison at index j to iteratively use ASNs in the CPs to complete DPs.

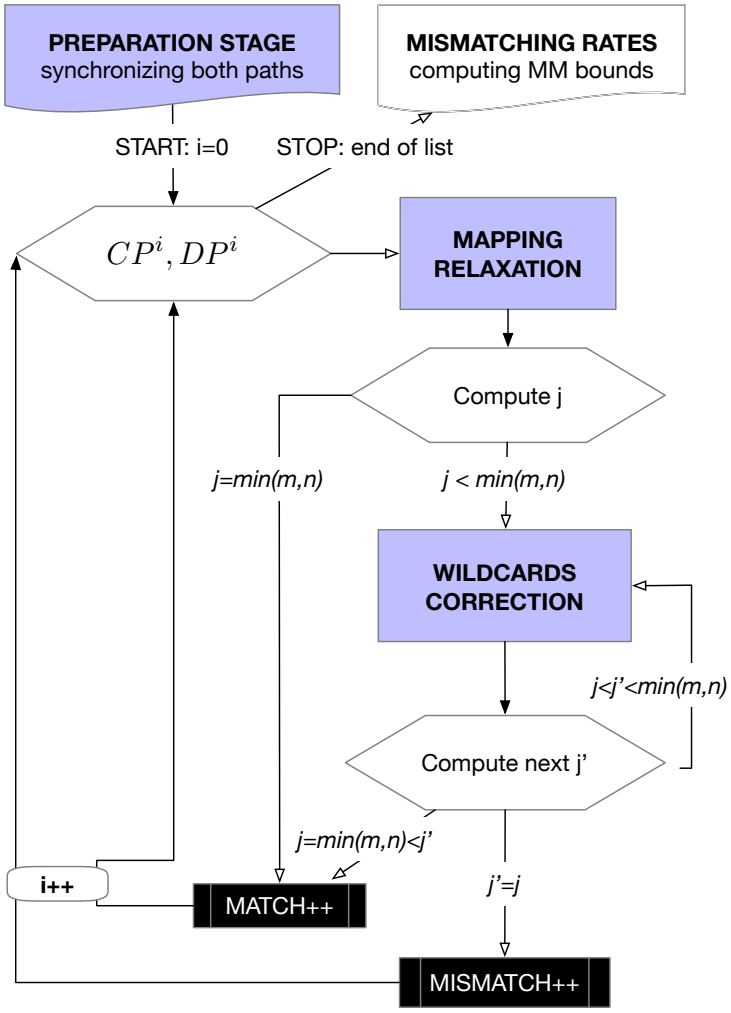


Figure 4.6: Our modular framework. The preparation stage synchronizes CPs and DPs in time and semantic AS-level. On the other hand, the mapping relaxation stage relaxes the original IP-to-AS mapping by gathering AS siblings into a unique representation and replacing inferred TPAs with wildcards. Finally, the wildcards correction stage infers values for wildcards resulting from either missing hops or artificially introduced in the previous step.

The rules applied inside each block *are not commutative*. Indeed, for example, the `TPA` rule may convert inferred TPAs into wildcards that could have been otherwise grouped among AS siblings with the `SIB` rule. This example highlights that applying any of the `TPA` rules before the `SIB` results into a more conservative noise-filtering model. On the other hand, the `nomatch*` rule would eliminate wildcards that the `match*` rule could have leveraged to complete missing pieces of DPs. Therefore, to obtain a more conservative model, the `match*` rule should precede the `nomatch*` rule.

In conclusion, depending on whether the mapping relaxation stage is used or not, which rules are applied and in which order they are implemented inside the

	Mapping Relaxation			Wildcards Correction	
Model/Rules	`SIB`	`looseTPA`	`strictTPA`	`match*`	`nomatch*`
Raw	✗	✗	✗	✗	(i)
Upper	✗	✗	✗	(i)	(ii)
Restricted	(i)	✗	(ii)	(iii)	(iv)
Lower	(ii)	(i)	✗	(iii)	(iv)

Table 4.2: Path-rewriting rules (columns) applied for different noise-filtering quantification models (rows). The Roman numbers report the order in which the rules are applied. In addition, ✗ denotes rules that are not applied in the given model.

blocks, several **noise-filtering models** can be designed in the seek of BGP lies. In particular, Table 4.2 shows the different models we implement. From green to gray, the mismatching rate of CPs and DPs, i.e, the resulting bound of plausible BGP lies, is expected to increase. The Raw and Upper models do not use the mapping relaxation stage, thus are expected to perform the worse, the latter better since it at least implements the `match*` rule. On the other hand, comparing the Lower and Restricted models, the first not only applies rules in a more conservative ordering in the mapping relaxation stage, but also implements the `looseTPA` rule rather than the `strictTPA`.

4.3.1 Preparation stage

The preparation stage is fed with a set of raw CPs and DPs, and outputs a pre-processed AS-formatted list of (CP^i, DP^i) couples by:

- synchronizing CPs and DPs, i.e. coupling each DP to a specific CP;
- IP-to-AS mapping each IP address appearing in the IP-level raw DPs;
- pre-processing each couple to purge them from minor mapping limitations.

Concerning the **synchronizing of CPs and DPs**, each DP obtained running `traceroute` is associated to the CP of the longest matching prefix that covers the target IP in the last RIB dumped before the `traceroute` was run. This overall process results in a list of synchronized couples, where DP is still at the IP level.

DPs are then converted into AS-level paths with an **initial IP-to-AS mapping function**. We map each IP address in the DPs to the OAS of the longest matching prefix covering the IP.[2] This process in general is not perfect: `traceroute` traces may include private IP addresses and missing hops. Moreover, some IP addresses may not necessarily be mapped to an unique and/or valid AS. OASes may include private, or more generally, prohibited ASNs (pASNs) that should not be advertised[3],

[2]The OASes of all prefixes were assumed to remain constant in the course of a day, and extracted from the first RIBs dumped every day.

[3]`https://www.iana.org/assignments/as-numbers`

and also AS sets. For all these cases, the mapping is undefined. Consequently, we decide to conservatively map each of these as wildcards ('*'), that can be eventually replaced by any ASN in the wildcards correction stage.

On the IP-to-AS mapping is performed, **pre-processing the couples** is required. Indeed, CPs may still include, and be affected by, AS Sets, pASNs, or AS prepending. Hence, a list of actions to purge them is required: (i) pASNs are removed when appearing at the end of a path; (ii) path couples with CPs containing AS sets or pASNs are discarded (less than 0.1% of the cases); (iii) repeated consecutive ASNs in CPs (AS prepending) are eliminated. Finally, (iv) in case of prematurely ending CPs (e.g. due to coarse grained prefixes) where DPs reveal extra path after the OAS, the remaining part of the path (if any) is trimmed.

After this stage, CPs and DPs are still subject to limitations introduced by AS siblings, TPAs or IXPs, and wildcard sequences. These sources of noise are filtered in the mapping relaxation (if enabled) and wildcards correction stages.

4.3.2 Mapping relaxation

Once the preparation stage is over, CPs and DPs may still suffer from AS siblings and TPAs from DPs. Both are accounted for in this block, relying on two rules, namely `SIB` and `TPA`. The former links AS siblings to unique representatives via an AS-to-organization mapping function, while the latter modifies DPs by replacing inferred TPAs with wildcards. Although their respective mode of operation is based on distinct conditions, both rules relax the mapping, i.e. they re-map ASes by either grouping them by organizations, or turning them into wildcards. Note that this stage does not compare DPs and CPs, but rather simplifies both independently.

4.3.2.1 `SIB` rule

To filter the noise resulting from AS siblings, we propose describing CPs and DPs at an organization-level rather than at AS-level. We thus rely on a **AS-to-ORG mapping function** denoted $CH(\cdot)$. Similar to an IP-to-AS mapping, the AS-to-ORG mapping consists in condensing paths: the former groups IPs into ASes whereas the latter gathers ASes into organizations. Note that the AS-to-ORG mapping has to be applied to both DPs and CPs to guarantee consistency. Algorithm 1 shows how we implement the `SIB` rule.

Algorithm 1: `SIB` rule. CPs and DPs are AS-to-ORG mapped.

Input: Control or Data path P
AS-to-ORG mapping $CH(\cdot)$

```
for all i ∈ [[0, n[[ do                                  // n = len(P)
    P[i] ← CH(P[i])
    if P[i] = P[i − 1] and P[i] ≠ ∗ then                 // n − −
        Delete P[i]
return P
```

To construct $CH(\cdot)$, we rely based on the OrgID field of CAIDA's AS Or-

ganizations Dataset.[4] We consider that each organization *Org* owns an AS sibling set $\mathbb{S}_{Org} = \{S_1, S_2, ..., S_N\}$ with $N \geq 1$. For each AS sibling set $\mathbb{S}_{Org}$, we arbitrarily define one of its AS siblings as the cluster head, denoted S. Then, $\forall S_i \in \mathbb{S}_{Org}$, $CH(S_i) = S$. Applying this additional mapping to the ASes in a path ensures that they are all mapped to the cluster head of the organization they belong to while leaving wildcards unchanged, i.e., $CH(*) = *$.

Additionally, possibly redundant ASNs resulting from multiple ASes being mapped to the same organization are purged. For example, let us consider a path P (either a DP or a CP) where each AS natively depicts the cluster head of its organization, except for $\mathtt{B}_1, \mathtt{B}_2 \in \mathbb{B}_{Org}$ such that $CH(B_i) = B$:

$$\mathtt{P = A\ B_1\ B_2\ *\ *\ C\ D} \xrightarrow{\mathtt{SIB}} \mathtt{P = A\ B\ *\ *\ C\ D}$$

Finally, note that the SIB rule keeps track of the number of IPs in each organization such that rules that are applied after it can take this parameter into account.

4.3.2.2 TPA rules

Traffic flows through paths that are usually represented by the IP addresses of the incoming interfaces of the routers that are traversed towards the destinations. Although routers most likely respond to traceroute with the IP of their incoming interface, they can be configured differently: the reply may report the IP address of another interface and specific inter-domain addressing allocation policies applied in IXPs or between incongruent remote BGP sessions may favor the occurrence of TPAs, that result in the insertion of off-path ASNs in AS-level DPs. The TPA rules aims to filter the noise resulting from TPAs. According to Algorithm 2, the TPA rules first locate candidate TPAs, and then depending on which between the strictTPA or looseTPA rules are applied, further tests are performed to validate the candidates. In all cases, rather than blindly and arbitrarily assigned to either the preceding or following AS, the inferred TPAs are conservatively replaced with wildcards.

Algorithm 2: TPA rules. The rules strictTPA and looseTPA differ only in that the triggering conditions, the latter being more permissive.

Input: $P \leftarrow DP, NH(\cdot), p$
for *all* $i \in [\![0, n[\![$ **do** // $n = len(P)$
  if $NH(i) < p$ **then**
    if *TPA rule* = *looseTPA* **then**
      $P[i] \leftarrow *$
    else if *TPA rule* = *strictTPA* **then**
      if $NH(i \pm 1) \geq p$ *and* $P[i \pm 1] \neq *$ **then**
        $P[i] \leftarrow *$
return P

[4] http://www.caida.org/data/as-organizations/20180703.as-org2info.txt

To identify candidate TPAs, the `TPA` rules check the number of consecutive IP addresses that are mapped to the same AS. When less than a threshold p, the AS is said to be *weakly represented* and, as such, becomes a suspected as appearing the the DP due to a TPA. To implement this checking, let $NH(\cdot)$ denote a function that takes an AS-level rank i as input, and returns the number of IP-level hops mapped to DP_i, i.e. the i^{th} AS-hop in the DP. We detect candidate TPAs when $NH(i) < p$. The $NH(\cdot)$ function is leveraged by our `TPA` rules, hereinafter assuming $p = 1$.

While the `looseTPA` rule assumes *all* candidate TPAs are, in fact, real TPAs and replaces them with wildcards, the `strictTPA` rule only performs the replacement if: (i) both adjacent hops are *not* wildcards; (ii) both adjacent ASes are *not* candidate TPAs. To better understand the difference between both implementations, consider a list $\bar{\mathbb{P}}$ of DPs where `x` and `y` are weakly represented ASes:

$$\bar{\mathbb{P}} = \begin{cases} \texttt{DP}_0 = \texttt{A B x C D E} \\ \texttt{DP}_1 = \texttt{A B x y D E} \\ \texttt{DP}_2 = \texttt{A B x * y D E} \\ \texttt{DP}_3 = \texttt{A B x * x C D E} \end{cases}$$

The `looseTPA` and the `strictTPA` rules act as follows:

$$\bar{\mathbb{P}} \xrightarrow{\texttt{looseTPA}} \begin{cases} \texttt{DP}_0 = \texttt{A B * C D E} \\ \texttt{DP}_1 = \texttt{A B * * D E} \\ \texttt{DP}_2 = \texttt{A B * * * D E} \\ \texttt{DP}_3 = \texttt{A B * * * C D E} \end{cases}$$

$$\bar{\mathbb{P}} \xrightarrow{\texttt{strictTPA}} \begin{cases} \texttt{DP}_0 = \texttt{A B * C D E} \\ \texttt{DP}_1\texttt{, DP}_2\texttt{, and DP}_3 \text{ remain unchanged} \end{cases}$$

As shown in the examples above, the noise-filtering capability of the `looseTPA` rule is more aggressive: *all* candidate TPAs are actually inferred to be, and turned into wildcards. Moreover, even when separated by unresponsive hops, the different appearances of the same AS are considered independent (see $\texttt{DP}_3$). On the other hand, `strictTPA` is less permissive: candidate TPAs and/or wildcards are considered logically exclusive. As a consequence, only $\texttt{DP}_0$ finishes being modified in the previous examples.

4.3.3 Wildcards correction stage

The last step required before evaluating if DPs and CPs match is dealing with wildcards in DPs. Indeed, DPs may be incomplete, i.e., include sequences of wildcards resulting from either missing hops in `traceroute`, undefined IP-to-AS mapping or due to the previous application of the `TPA` rules. We refer to the appearance of one or multiple wildcards in a row as a *wildcard sequence*. In general, wildcard sequences are bounded by two ASes: a *diverging AS* on the left and a *converging AS* on the right[5]. The objective of the wildcard correction block is to *correct* DPs by

[5]Trailing wildcards that constitute an exception for the presence of a converging AS, are silently discarded as carrying no additional information.

replacing wildcard sequences with their respective CP sequence (rule `match*`). However, when substitutions cannot be performed, or simply systematically by the Raw model, wildcards are deleted (rule `nomatch*`). Both rules require knowing the AS-hop j where the first wildcard appears (right after the diverging AS) to be applied in each wildcard sequence, as implemented in Algorithm 3.

To apply the `match*` rule, both the diverging and the converging ASes are required to appear in the CP. If so, and considering there is at least one intermediary AS in the CP sub-sequence between these two ASes, two possibilities may arise: (i) the number of ASes in the CP sub-sequence is smaller or equal to the length of the wildcard sequence in the DP or; (ii) the opposite. If (i) holds, the `match*` rule is able to correct the DP: the complete sequence of wildcards is substituted with the CP sub-sequence (extra wildcards, if any, being discarded); otherwise, the `match*` rule cannot rewrite the DP. In such cases, the DP may be further corrected with the `nomatch*` rule, that simply deletes the remaining wildcards and also the diverging AS when it matches the converging AS.

Algorithm 3: `match*/nomatch*` rules. While in rule `match*` wildcards are substituted (matched) with sequences of ASes in CPs, in rule `nomatch*` they are deleted.

```
match* rule ⟹ Input: P ← DP, R ← CP, j
    if (∃ k > j | P[l] = * ∀ l ∈ [[j, k[[) then
        if (∃ i ∈[[j, k]] | P[k] = R[i]) then
            Substitute P[j], ..., P[k − 1] with R[j], ..., R[i − 1]
    return P
nomatch* rule ⟹ Input: P ← DP, j
    if (∃ k > j | P[l] = * ∀ l ∈ [[j, k[[) then
        Delete P[j], ..., P[k − 1]
        if P[j] = P[j − 1] then
            Delete P[j]
    return P
```

In the following examples, let $\bar{\mathbb{P}}$ represent a list of DPs, each of which includes one or more sequences of wildcards, and that should be compared with the control path `CP`:

$$\texttt{CP} = \texttt{A B C D E F G H}$$

$$\bar{\mathbb{P}} = \begin{cases} \texttt{DP}_1 = \texttt{A B C D * * * G H} \\ \texttt{DP}_2 = \texttt{A B C D * G H} \\ \texttt{DP}_3 = \texttt{A B * B C D E * G H} \end{cases}$$

If rules `match*` and `nomatch*` are consecutively applied (e.g., in an if/else condition), as for all models except the Raw model, then:

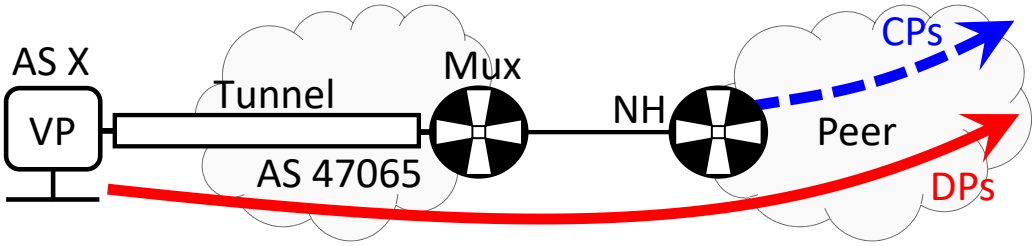


Figure 4.7: Simplified representation of a co-located VP provided by the Peering Testbed.

$$\bar{\mathbb{P}} \xrightarrow{\texttt{match}^*/\texttt{nomatch}^*} \begin{cases} \texttt{CP} = \texttt{DP}_1 = \texttt{A B C D E F G H} \\ \texttt{CP} \neq \texttt{DP}_2 = \texttt{A B C D G H} \\ \texttt{CP} = \texttt{DP}_3 = \texttt{A B C D E F G H} \end{cases}$$

Whereas in $\texttt{DP}_1$, rule $\texttt{match}^*$ replaces the three wildcards with the sub-sequence `E F` ($j = 5, k = 8, i = 7$), the substitution cannot be applied in $\texttt{DP}_2$ since the available wildcards are not enough. Moreover, even the subsequent use of rule $\texttt{nomatch}^*$ does not solve the difference between CP and $\texttt{DP}_2$. On the contrary, $\texttt{DP}_3$ matches CP after the first and second wildcard sequences are solved with rules $\texttt{nomatch}^*$ and $\texttt{match}^*$, respectively.

The rules applied in this block allow to conservatively bound the rate of BGP lies since for them each single wildcard can represent, despite unlikely, up to one entire AS (see $\texttt{DP}_3$ on the previous example).

4.4 The measurement platform and our campaign

We run our measurement campaign using co-located VPs: 6 are obtained from the **Peering testbed** [89], and 2 additional ones are homemade, i.e., manually deployed by us. With 8 co-located VPs, our analysis reaches a scale never achieved before for studies comparing CPs and DPs.

According to Fig. 4.7, the Peering testbed ensures that DPs and CPs are gathered in the same place by constructing tunnels to an ASBR called *Mux* (in practice there exist multiple Muxes, and we can choose which one to connect with). By setting a default route towards the next-hop NH, packets flow to Mux, that then relays traffic towards NH. This allows to select the peer from which DPs are obtained. In addition, relying on the configurations of the Peering Testbed, CPs can be obtained from the ASBR that has the IP address NH assigned to one of its interfaces. We focus on the peers reported in Table 4.3, which provide full-RIBs, i.e., transit for all prefixes usually announced on the Internet [90]. On the other hand, the homemade VPs are setup with virtual machines that use as default gateway an ASBR from which we can collect BGP data.

We collect CPs from BGP speakers both at the Peering testbed and the homemade VPs (*hmX*) every 2 hours. On the other hand, we gather DPs with Scamper [91], running ICMP Paris-`traceroute` from VPs placed next to the gateway

Peer	Organization	ASN	CP-DP match [%]
isi	Los Nettos	226	77.92
uw	University of Washington	101	77.93
neu	Northeastern University	156	76.84
uth	University of Utah	210	69.51
grt	GRNet	5408	77.93
cle	Clemson University	12148	77.93
hm1	University of Strasbourg	2259	77.94
hm2	RGnet, LLC	3130	77.90

Table 4.3: Peers that provide transit and full RIBs that were used as VPs. The ones on top are obtained from the Peering Testbed, and the ones below manually deployed by us.

for the homemade VPs, or tunneling through the Peering testbed up to the routers that provide the RIBs. The measurement campaign was designed to run daily with 80k traces per day. We chose the destinations we trace by uniformly sampling /24 prefixes in blocks allocated by RIRs [92]. We pick one IP from each of these prefixes. However, for a fraction of the traces, despite the prefixes are allocated, they are not advertised in BGP (even in full RIBs). Table 4.3 shows that more than 20% of the selected IPs disclosed the absence of a CP, with no matching BGP prefix. This effect is even worse in *uth*, that exhibits RIBs with slightly less entries than the other VPs.

4.5 Rate of BGP lies in the wild

In this section we present the bounds of BGP lies we find in the wild. The results are computed based on the daily measurements we carried between 05.10.2018 and 17.11.2018 for the VPs belonging to the Peering testbed and during approximately 8 months (from 18.04.2018 to 19.12.2018) in the case the homemade VPs. In Sec. 4.5.1 we provide an overall view of the mismatch rate for the four noise-filtering models we propose. Then, our study focuses in the Lower model, that has the most conservative design and presents the lowest bound of BGP lies. First, we analyze the impact of the set of rules that compose its mapping relaxation stage, in Sec. 4.5.2. Finally, in Sec. 4.5.3 we gather the VPs where the Lower model outputs the highest rates of BGP lies and try to identify the type of BGP lies that cause these results.

4.5.1 Performance of the different noise-filtering models

The rate of presumable BGP lies ($\mu \pm \sigma$) found in the measured peers with each of the noise-filtering models (Table 4.3) is shown in Fig. 4.8. Results are consistent across VPs, i.e. the bounds of mismatches from the Lower to Raw model always report an increasing number of BGP lies. Yet, distinct patterns among the different peers can be quantitatively observed, specially for *cle* and *hm1*.

Analyzing Fig. 4.8 in more detail, we note **the results for the Restricted and Lower models differ in less than 5% for all VPs**. Moreover, their values

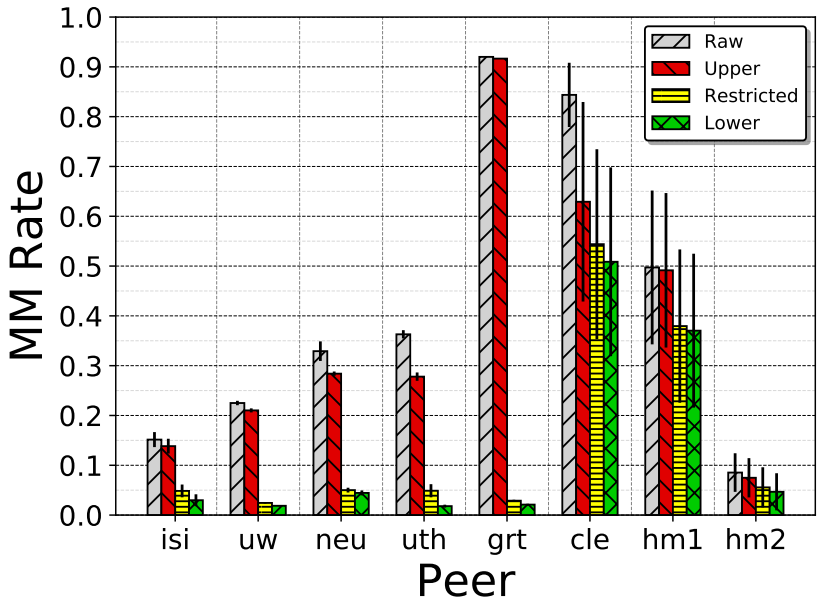


Figure 4.8: Mismatch (MM) rate according to the model in use. The bounds obtained for the Restricted and Lower model differ in less than 5% for all VPs. The rate of BGP lies for the lower bound is more than 35% for *cle* and *hm1*, more than 7 times compared to what is seen in the remaining VPs.

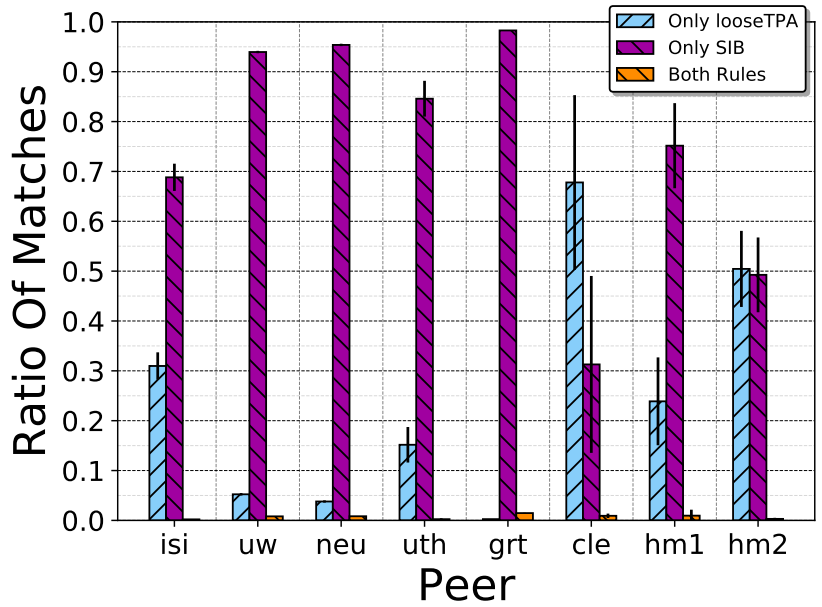


Figure 4.9: Ratio of matches in the Lower model that result from of extending the Upper model by including the mapping relaxation stage. In general, the `SIB` rather than the `looseTPA` rule proves to be more useful.

are lower than 5% in most cases. This small difference suggests that **TPAs and wildcards resulting from missing hops and/or undefined mapping are either not frequently found in sequence or, when they are, the DP still matches the CP.** Recalling Sec. 4.3, this result shows that the Lower model does not gain much from implementing `looseTPA` rather than `strictTPA`, neither from the more conservative ordering of the rules applied in the mapping relaxation stage (`SIB` rule after `TPA` rule).

On the other hand, the Raw and Upper model also perform similarly, though the latter generally shows a mismatch rate just a bit lower than the former. Therefore, in most cases, wildcards resulting from unresponsive hops and/or undefined mapping can be silently discarded. The only exception is *cle*, where the difference amounts to 23% due to missing hops that occur at the beginning of many DPs. Also, note that in the Lower and Restricted models, `TPA` rules in the mapping relaxation stage exchange inferred TPAs for wildcards, thus increasing the need of the wildcard correction step.

According to the design of the proposed framework, the *real* rate of BGP lies observed by VP is expected to be between the bounds that the Lower and Upper models respectively produce. In other words, the mismatches observed only through the Upper model are potential *false negatives* for the Lower model, i.e. potential lies wrongly filtered as noise. Consequently, the rate of lies may be as significant as the *Upper* bound, at worst. On the other hand, its value could be closer to the fully-conservative Lower bound, usually less than 5%. While this value is low, **it is not negligible: according to its conservative design, the *Lower model* is expected to filter most of the noise and to capture many actual lies.**

4.5.2 Effect of `SIB` and `TPA` rules on the mismatch rate

Our models can be grouped both in terms of design and performance: Raw and Upper on one side, and Restricted and Lower on the other. As illustrated in Fig 4.8, there exists a large gap in terms of the plausible BGP lies seen for these two groups, except in *hm2* where all bounds are surprisingly close to each other. While the Raw and Upper models just use the wildcards correction stage, the Restricted and Lower ones make also extensive use of the mapping relaxation stage. We now analyze if any of the rules in this latter block is more effective to decrease the number of mismatches, or if it is rather their combination that is required. Since each rule was designed to treat a specific limitation affecting DPs and CPs, this would also reveal if there is an outstanding kind of noise biasing severely the ground data. In particular, since the Restricted and Lower bounds perform similarly, we focus only on the Lower bound, and thus `SIB` and `looseTPA` rules.

The difference between red and green bars for each peer in Fig. 4.8 represents the amount of mismatches observed via the Upper model and not via the Lower one. In other words, it is the share of cases that benefit from the mapping relaxation stage. We analyze which of these cases actually profit from applying (i) only the `SIB` rule, (ii) only the `looseTPA` rule, and (iii) both. As shown in Fig. 4.9, less than 3% of the total cases across all VPs are filtered concurrently using both the `SIB` and `looseTPA` rules. Indeed, this small proportion indicates that, in general, **paths do not include simultaneously AS siblings and TPAs.** In addition, between

68% and 97% of the cases across all VPs (*cle* and *hm2* being the exception with less than 32% and 50%, respectively) require only using the `SIB` rule.

4.5.3 Looking closer at high mismatch rates

Although pinpointing and understanding the root causes of observed MMs —and defining whether they are deliberate or not— is challenging (see Sec. 4.1), the high mismatch rate observed for the Lower model in *cle* and *hm1* (together with their higher variability over time) encourage us to further investigate these cases. We see that *cle*'s provider sends traffic directly to the AS that is expected to be two AS-hops away, according to advertised CPs. While the presence of an *unintended lie* is a likely cause—also in line with the high variability observed—neither an *interested lie* nor *AS poisoning* can be discarded. On the other hand, a privileged view in *hm1* allows us to access the ground truth and to determine that most mismatches seen in this peer originate from *technical limitations* in the infrastructure of its provider AS. Indeed, the ASBR connecting to *hm1* had a partial-FIB with a persistent default route, and this lead traffic to exit the provider AS through a peering AS that was not necessarily the one included in CPs.

4.6 Conclusion

BGP lies are not straightforward to detect: noise in the ground data can generate mismatches between CPs and DPs. Since this noise can be confused as lies—and vice versa—filtering it is imperative. We propose a framework based on multiple path-rewriting rules that allows to produce four noise-filtering models to overcome the shortcomings produced by the noise, and to estimate bounds of BGP lies in the wild. We leveraged the Peering testbed that provides full-RIBs from multiple peers as well as co-located CPs and DPs, and carried out a longitudinal analysis as never done before, that spanned 8 VPs and up to 8 months of measurements. While the noise from TPAs was more prevalent for a limited number of VPs, the noise due to AS siblings generated most mismatches. We believe that this effect is largely VP-dependent.

Finally, we quantified the lower bound of the mismatch rate seen in the wild as being less than 5%. This value is small, but not negligible: since our approach is conservative, we expect to have filtered most of the noise and have captured many actual BGP lies. Moreover, this also means that there might be many false negatives, i.e. many lies that finished being filtered as if they were noise. At the same time, we further analyzed the nature of mismatches persisting after applying the most conservative filter in a VP where we have a privileged view, concluding that *technical limitations* related to a partial-FIB router relying on a default route in the infrastructure of the provider AS were causing the BGP lies.

In future work we plan on extending the measurement infrastructure and the coverage of our analysis. A difficulty to sort out in this aspect is that the study in this chapter requires using co-located VPs. The Peering Testbed allowed us to fulfill this requirement, but only for a reduced number of ASes that shared full-RIBs. As a first solution, we plan on relaxing the need of full-RIBs, and to craft the IP list to be measured on a per-peer basis, according to the prefixes that each AS

advertises. This will allow us to increase the number of VPs and the success rate of our measurements. In particular, considering that Routeviews and CAIDA are planning to implement Scamper at BGP collectors co-located at IXPs, this creates a good opportunity to extend our work.[6]

[6] https://www.caida.org/publications/presentations/2020/scamper_routeviews_kismet/

Chapter 5

The Art of Modeling and Detecting Forwarding Detours

Contents

In this chapter we take a close look at the phenomenon of forwarding detours. To the best of our knowledge, we are the first to tackle the problem of detecting forwarding detours, indistinctly of the underlying causes generating them, while filtering load balancing and traffic engineering techniques. In a nutshell, we make the following contributions:

- We develop a formalism around forwarding detours in Sec. 5.1. In particular, we construct a forwarding model and show that routing inconsistencies may evolve into forwarding alterations which then may originate forwarding detours.

- We study in Sec. 5.2 the forwarding pattern that FDs originate inside ASes, i.e., whether traffic between two fixed endpoints flows through different routes depending on the considered prefix, and compare it with that generated by load balancing and traffic engineering techniques. In particular, FDs, per-prefix LB and TE generate a similar forwarding pattern, which we call prefix-based.

- We design an algorithm able to detect prefix-based forwarding patterns in Sec. 5.3. Our framework relies only on IP-to-AS mapping data and data-plane information collected with `traceroute`. We present a novel strategy to seek for multi-path routing patterns, in multiple steps. Our technique groups prefixes for which the same internal routes of ASes are revealed, an idea that may be incorporated in topology discovery studies to reduce their associated probing cost.

- We propose an FD-detector in Sec. 5.4. Our solution adds a last phase to the previous algorithm: it applies an FD-verdict allowing to discriminate FDs from the other mechanisms that also generate prefix-based forwarding patterns. For this, we focus on extreme-FDs cases, i.e., scenarios where FDs affect numerous prefixes. We build a detector of forwarding detours as a tool ready to be run in the wild.

- We analyze the FD-phenomenon in the wild in Sec. 5.5, running our FD-detector from 100 nodes of the NLNOG RING monitoring infrastructure, and find FDs in 25 out of 54 ASes. We find a remarkable binary pattern in which transit traffic traversing between two border routers of an AS either never detours, or always does. 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].

In addition, we discuss the robustness of the FD-detector we implemented in Sec.5.6, and draw final remarks in Sec. 5.7. The work presented in this chapter lead to the writing of the following article and poster:

- **Julián M. Del Fiore**, Valerio Persico, Pascal Merindol, Cristel Pelsser and Antonio Pescapè. *The Art of Detecting Forwarding Detours*, to appear in IEEE Transactions on Network and Service Management (IEEE TNSM).
- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *Routing Inconsistencies at the FIB level*, poster presented in TMA 2019.

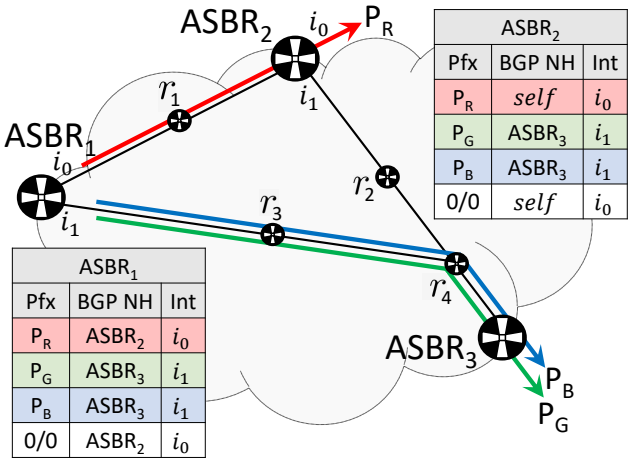
5.1 The origin of FDs: routing inconsistencies and forwarding alterations

This section formally defines the concepts of routing inconsistencies (RIes), forwarding alterations (FAs) and forwarding detours (FDs). First, Sec. 5.1.1 illustrates the intuition behind these concepts and how they relate. Then, in Sec. 5.1.2, we propose a simple, yet realistic, forwarding model that describes how routers take forwarding decisions. Then, in Sec. 5.1.3 we explain how the combination of these distributed decisions result into forwarding routes and internal routes of ASes. Second, we study when the forwarding is consistent in Sec. 5.1.4 and define the conditions leading to the existence of RIes in Sec. 5.1.5. Third, in Sec. 5.1.6 we describe how RIes may, in turn, evolve into FAs, and further show how these may generate FDs in Sec. 5.1.7. Finally, in Sec. 5.2.2 we analyze how FDs, LB and TE have similarities and differences in the forwarding patterns they generate, i.e., in how traffic subject to each of them flows inside ASes.

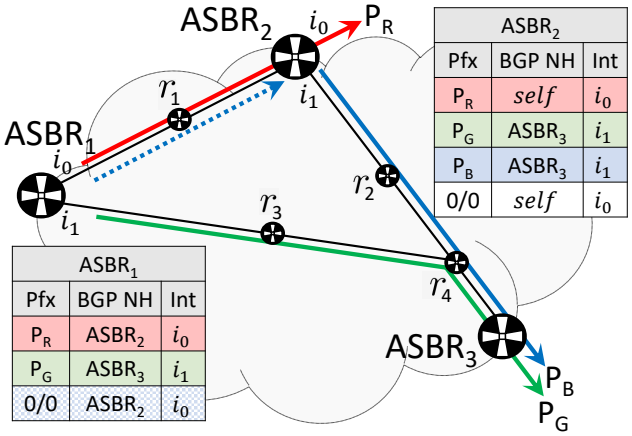
5.1.1 RIes, FAs and FDs in a practical example

To illustrate how FDs originate, Fig. 5.1 compares the scenarios when either a network with a fixed topology is composed of only full-FIB routers (5.1a), or includes an ASBR with a partial-FIB and a default route (5.1b). In Fig. 5.1a, $ASBR_1$ forwards transit traffic through optimal paths, according to the IGP metric in use, to reach all possible exit points, i.e., the other ASBRs. On the other hand, in Fig. 5.1b, $ASBR_1$ has a partial-FIB and forwards traffic destined to prefix P_B via its default route, relying on $ASBR_2$ (blue dotted line). Since $ASBR_2$ considers $ASBR_3$ the best ASBR for P_B, $ASBR_2$ redirects traffic targeting P_B towards $ASBR_3$. In this way, $ASBR_2$ introduces a forwarding alteration, i.e., changes the expected forwarding route. 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,

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



(a) All ASBRs have a full-FIB



(b) $ASBR_1$ has a partial-FIB

Figure 5.1: Routing consistence vs. forwarding detours. In the left case, transit traffic entering the AS through $ASBR_1$ always flows in the AS towards the remaining ASBRs through the best IGP paths. In contrast, in the right figure, $ASBR_1$ has a partial-FIB lacking the entry for the blue prefix P_B, leading to FDs for this prefix, and for multi-path routing patterns between $ASBR_1$ and $ASBR_3$, as traffic concerning prefix P_G does not detour.

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. between $ASBR_1$ and $ASBR_2$, once traffic concerning P_B exited the tunnel, $ASBR_2$ would still redirect it to $ASBR_3$.

5.1.2 Lookup functions: prefixes, gateways and next-hops

In our forwarding model, each router r inside any AS X determines next-hops relying on three lookup functions, namely $\mathcal{N}_r(G)$, $\mathcal{G}_r(P)$ and $\mathcal{P}_r(d)$. Each of these functions conveys a different objective:

The function $\mathcal{P}_r(d)$ receives a destination IP address d, and returns the most

specific prefix covering d. Therefore, $\mathcal{P}_r(d)$ can be either an internal or external prefix, i.e.,

$$\mathcal{P}_r(d) = \begin{cases} \text{Internal prefix}, & d \in X \\ \text{External prefix}, & \text{otherwise} \end{cases}$$

where d is advertised in the IGP of X in the first case, and learned via BGP, and thus associated to transit traffic, in the latter.

The function $\mathcal{G}_r(P)$ takes a prefix P as argument and outputs the IP address inside X to be reached in order to eventually access prefix P. We refer to $\mathcal{G}_r(P)$ as the *gateway* for P. The interpretation of the gateway $\mathcal{G}_r(P)$ varies depending on whether P is internal or external, i.e.

$$\mathcal{G}_r(P) = \begin{cases} d, & P \text{ is an internal prefix} \\ BGP(P), & \text{otherwise} \end{cases}$$

where $BGP(P)$ returns the iBGP next-hop for P.[2] Hence, the gateway $\mathcal{G}_r(P)$ should be, theoretically, the egress-ASBR for transit traffic, i.e., the last hop in X that forwards transit traffic to the eBGP next-hop that advertised P.

The function $\mathcal{N}_r(G)$ computes the next-hop towards an IP address G, usually a gateway, inside X, i.e. the router linked via an outgoing interface of r to which a packet has to be sent in order to ultimately reach G. The value of $\mathcal{N}_r(G)$ depends on that of G, i.e.,

$$\mathcal{N}_r(G) = \begin{cases} \delta(d), & G = r \\ IGP(G), & \text{otherwise} \end{cases}$$

where $IGP(G)$ provides the IGP next-hop towards G and $\delta(d)$ is a function of d that takes different values depending on whether $d = r$ or $d \neq r$. In the first case, the forwarding stops and the packet is handled by the higher layers of the protocol stack, i.e., $\mathcal{N}_r(G) = \delta(r) = \emptyset$. On the other hand, if r is not the destination, then r acts as the egress-ASBR used to reach the external destination d, and thus $\mathcal{N}_r(G) = \delta(d)$ is the eBGP next-hop for P.

To forward packets towards a destination IP address d, routers apply the lookup functions we have defined in sequence. Like this, packets are forwarded to the next-hop $\mathcal{N}_r \circ \mathcal{G}_r \circ \mathcal{P}_r(d)$ leading to the gateway $\mathcal{G}_r \circ \mathcal{P}_r(d)$ defined for the longest matching prefix $\mathcal{P}_r(d)$ of d.

5.1.3 What is an internal route of an AS?

As routers at each hop receive and forward packets, the chaining of these events results into a forwarding route.

Forwarding route: *a forwarding route of an AS X towards a destination d, denoted $R_X(d)$, is a sequence of routers $R_X(d) \triangleq [r_0, r_1, ..., r_j, ..., r_n]$, where, we consider implicit that $r_j \triangleq r_j(d)$ to simplify notation, such that:*

$$\forall j \in \{0, ..., n-1\},\ r_{j+1} = \mathcal{N}_{r_j} \circ \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d)$$

[2] When the BGP next-hop-self option is enabled, this is exactly the case. We design our model relying on this feature for convenience, to simplify the illustrations in this section that would otherwise require showing a neighboring AS, but without loss of generality.

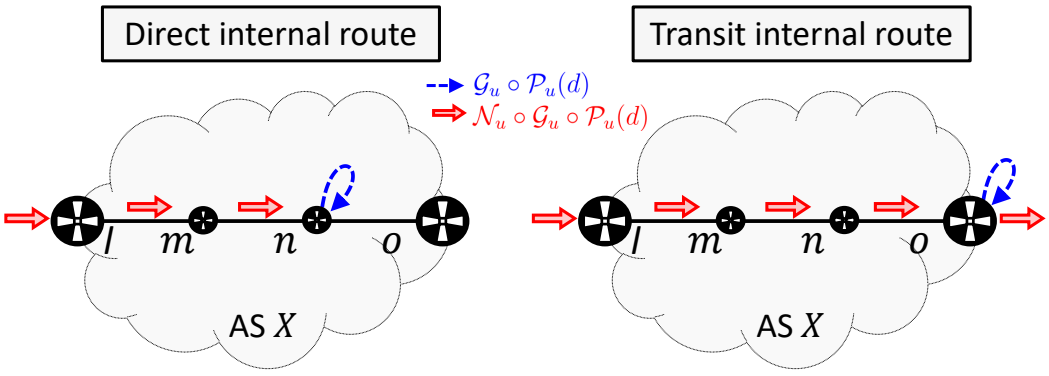


Figure 5.2: Direct internal routes and transit internal routes inside an AS X. The first are those internal routes for which the destination, n in this case, belongs to the same AS that owns the routers, that is, X. On the other hand, for the latter, traffic is only transiting through AS X and exits through an egress-ASBR, namely o. For both types of internal routes, the first router l is an ingress-ASBR of X.

The condition only defines how routers should choose next-hops to create a forwarding route, however, it does not apply on router r_n that could either equal d or not. This is not critical since, from forwarding routes, we will usually be interested in extracting the internal routes of the ASes that are traversed.

Internal route: *an internal route of an AS X towards a destination d, is a forwarding route $R_X(d)$ such that:*
i) $\forall j \in \{0, ..., n\}$, $r_j \in X$
ii) r_0 *is an ingress-ASBR of* X
iii) $\mathcal{G}_{r_n} \circ \mathcal{P}_{r_n}(d) = r_n$

A priori, the strict notion of what an internal route is only requires condition i, but we narrow their scope additionally considering conditions ii and iii to align our model to the FD-detector we implement later. While condition ii is self-explanatory, according to condition iii, router r_n chooses itself as gateway, hence $\mathcal{N}_{r_n} \circ \mathcal{G}_{r_n} \circ \mathcal{P}_{r_n}(d) = \mathcal{N}_{r_n}(r_n)$. Given this holds, r_n is the last hop in X, and there exist two options:

- r_n is directly the destination d and $\mathcal{N}_{r_n}(r_n) = \mathcal{N}_{r_n}(d) = \emptyset$. We refer to these routes as **direct internal routes (DIRs)**.

- $r_n \neq d$ and acts as egress-ASBR of X, consequently $\mathcal{N}_{r_n}(r_n) \notin X$. The packets being forwarded represent transit traffic in these cases, hence we say the routes are **transit internal routes (TIRs)**.

To better illustrate the difference between TIRs and DIRs, Fig. 5.2 presents an example. In both scenarios, packets enter AS X through the ingress-ASBR l, however, while on the left case traffic stops in router n, that is the destination, router o acts as egress-ASBR for the transiting traffic on the right side.

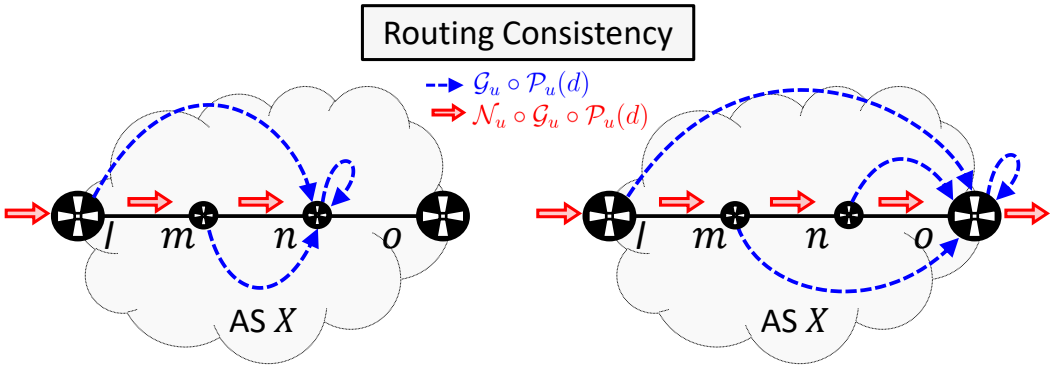


Figure 5.3: Routing consistency. In both cases, all routers along the internal routes choose the same best covering prefix for the destination (not explicitly shown) and the same gateway, n and o on the left and right side, respectively.

5.1.4 When is the routing consistent?

According to our definition of internal routes, indistinctly of being TIRs or DIRs, the actual gateway and best covering prefix that each router along the route chooses is not defined. The only exception is the gateway of r_n, that selects itself, as emphasized in Fig. 5.2 with the blue (looping) arrows in routers n and o, respectively. In particular, when all routers along the route choose the same best covering prefix P and gateway G, we say that the resulting internal route $R_X(d)$ is consistent.

Routing consistency: *an internal route $R_X(d)$ is consistent when all routers along the route choose the same best covering prefix P and gateway G for d, i.e., when*

$$\forall\, r_j \in R_X(d),\, \mathcal{P}_{r_j}(d) = P \;\wedge\; \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d) = \mathcal{G}_{r_j}(P) = G$$

.

In particular, note that r_0, the first router in $R_X(d)$, imposes conditions, i.e., it must hold that $P \triangleq \mathcal{P}_{r_0}(d)$ and $G \triangleq \mathcal{G}_{r_0} \circ \mathcal{P}_{r_0}(d) = r_n = \mathcal{G}_{r_n} \circ \mathcal{P}_{r_n}(d)$.

An example is illustrated in Fig. 5.3, where compared to Fig. 5.2, we explicitly show the gateway chosen by the routers along the routes. Since all blue arrows point to n and o, and we assume that all routers select the same best covering prefix for d, then both the TIR and DIR are consistent.

5.1.5 What produces routing inconsistencies?

An internal route $R_X(d)$ may be inconsistent at two different levels depending on the best covering prefixes and gateways that routers along the route choose for d.

Routing inconsistency at the prefix level: *there exists a routing inconsistency at the prefix level in an internal route $R_X(d)$ when an upstream router r_k of $R_X(d)$ chooses a different best covering prefix than a downstream router r_j of $R_X(d)$, i.e.,*

$$\exists\, j < k \leq n \mid \mathcal{P}_{r_j}(d) \neq \mathcal{P}_{r_k}(d)$$

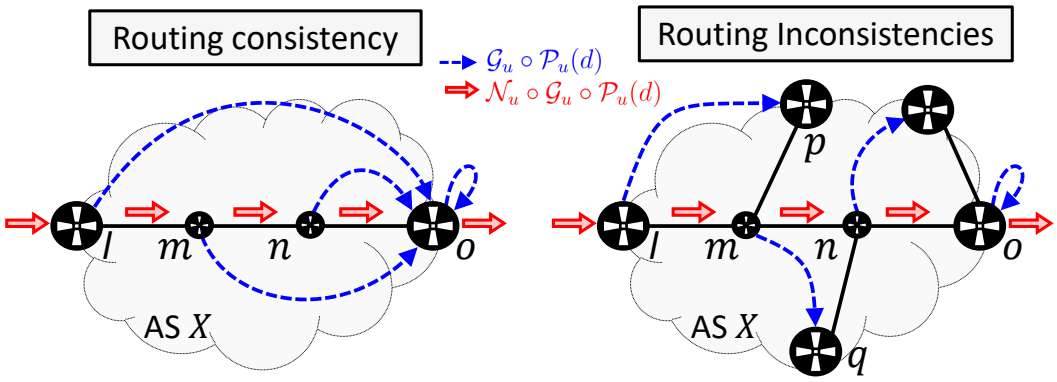


Figure 5.4: Routing consistency vs inconsistency. While all routers choose o as gateway on the left side, all blue arrows point to different routers on the right side. In this particular example, the resulting forwarding route is the same in both cases, though as we study next, for some RIes, this is not always true.

Routing inconsistency at the gateway level: *there exists a routing inconsistency at the prefix gateway level in an internal route $R_X(d)$ when an upstream router r_k of $R_X(d)$ chooses a different gateway than a downstream router r_j of $R_X(d)$, i.e.,*

$$\exists\, j < k \leq n \mid \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d) \neq \mathcal{G}_{r_j} \circ \mathcal{P}_{r_k}(d)$$

Note that both types of RIes **are not exclusive**: the one at the prefix level may lead to another at the gateway level. As an example, if r_j and r_k are routers in $R_X(d)$, such that r_j is a partial-FIB router with a default route and r_k is full-FIB router, this will likely generate RIes at both levels. As r_j has a partial-FIB, multiple originally disaggregated prefixes present in r_k are aggregated in the FIB of r_j into a default route, which originates RIes at the prefix level. While r_k may associate different gateways to each of these prefixes, a unique arbitrarily chose, a default one, is used by r_j. Consequently, this may likely potentially leading to additional RIes at the gateway level.

Another important detail is that RIes at the prefix level that do not generate others at the gateway level are not detectable. Indeed, despite the best covering prefix selected for the same destinations may vary from router to router, as long as they continue choosing the same gateway, the forwarding is not impacted. More generally, in practice only RIes at the gateway level can be detected, though it is not possible to determine at which level these were actually originated without privileged knowledge, e.g., being able to check the routing table of all traversed routers. Therefore, to simplify our notation and to focus on detectable RIes, **we will consider implicit that when we say that there exists a routing inconsistency in an internal route, we refer to at the gateway level**, and it may also be at the prefix level.

Fig. 5.4 illustrates the difference between cases where the routing is consistent (left) or there exist RIes (right). While all blue arrows point to o on the left side, the routers along the route choose all different gateways on the right side. In this

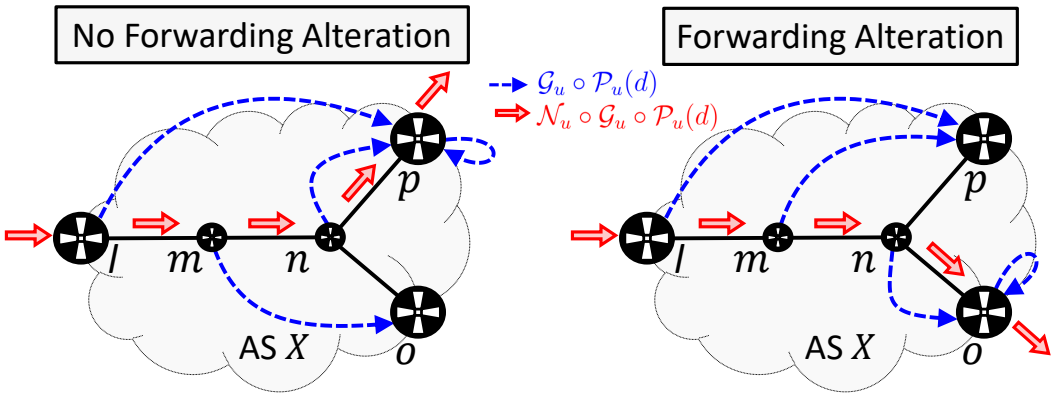


Figure 5.5: Routing inconsistencies and forwarding alterations. In both cases there exist RIes, however, while on the left case the action of router m does not produce FAs, in the right case router n deviates traffic towards router o, generating FAs.

particular example, the resulting forwarding route is the same in both cases, though as we study next, this is not always the case.

5.1.6 What leads to forwarding alterations?

When the resulting internal route $R_X(d)$ is different to the forwarding route that would have been used if all routers had chosen the same gateway, we say that the difference was generated by a FA in $R_X(d)$.

Forwarding alteration: *there exists a forwarding alteration in an internal route $R_X(d)$ if an upstream router r_k uses different next-hops for the gateway it chooses and the gateway a downstream router r_j selects, i.e.,*

$$\exists\, j < k \leq n \mid \mathcal{N}_{r_k} \circ \mathcal{G}_{r_k} \circ \mathcal{P}_{r_k}(d) \neq \mathcal{N}_{r_k} \circ \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d)$$

Theorem – Forwarding alterations imply routing inconsistencies: *if there exists a forwarding alteration in an internal route $R_X(d)$, then $R_X(d)$ is inconsistent.*

Proof. Assume there exists a forwarding alteration in $R_X(d)$, but no routing inconsistency occurs. If $R_X(d)$ is not subject to RIes, then $\forall j$, $\mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d) = G$. As a consequence $\mathcal{N}_{r_k} \circ \mathcal{G}_{r_k} \circ \mathcal{P}_{r_k}(d) = \mathcal{N}_{r_k}(G)$ and $\mathcal{N}_{r_k} \circ \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d) = \mathcal{N}_{r_k}(G)$, i.e., if the gateways match, so does the next-hop r_k uses. This contradicts our original hypothesis stating that FAs occur, proving that FAs imply RIes. ■

Note that FAs imply RIes, but the converse may not hold, i.e., the relationship is **FAs** $\Rightarrow$ **RIes**. Indeed, for $j < k$, even though routers r_j and r_k may choose different gateways, G_j and G_k respectively, when $\mathcal{N}_{r_k}(G_k) = \mathcal{N}_{r_k}(G_j)$, then no FA occurs. In these cases, we say that the existing RIes are **not visible**. This may particularly happen in networks that **lack path diversity**, e.g. if the disagreeing router r_k only has one possible next-hop, then it can never introduce a FA.

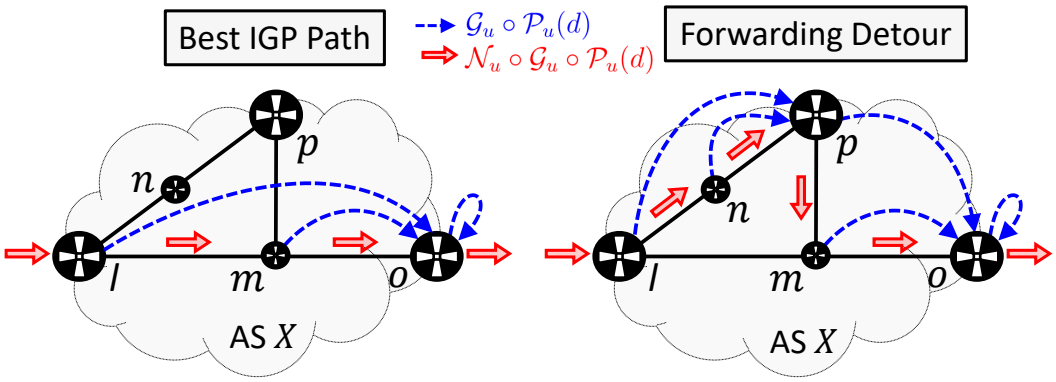


Figure 5.6: Best IGP path vs Forwarding Detour. On the left, all routers choose o as gateway, hence no RIes occur and traffic flows through the best IGP path from l to o. On the contrary, on the right, router p introduces RIes choosing o instead of itself as gateway. As p should act as egress-ASBR, but instead sends traffic to m, then $p<$ introduces a FA. Since l would have straightforwardly sent traffic to m, had it selected o as gateway, then the resulting internal route is subject to FDs.

An example is illustrated in Fig. 5.5 where RIes either do not introduce FAs (left) or do so (right). In the left case, there exists a routing inconsistency as router m chooses router o as gateway while router l chooses p. However, the forwarding route still ends up in p, being the same it would have been if m had chosen p as gateway, therefore no FA occurs. On the other hand, on the right figure there exist RIes as router n chooses different gateways than routers l and m, and a FA occurs: router n pushes traffic towards o instead of p. This affects the resulting route, which would have been the same as in the left side figure if n had chosen p as gateway.

5.1.7 When do forwarding detours occur?

A forwarding detour occurs when the internal route in use inside an AS does not match the best IGP path between the endpoints of the route, i.e., r_0 and r_n. The illustrations in Fig. 5.6 show the difference between best IGP paths and FDs. In the left case, the routing is consistent since all routers choose the same gateway o, and thus the forwarding route coincides with the best IGP path available between l and o. On the contrary, on the right case, router p introduces RIes, and a FA such that packets exit the AS via o. Since l would have used the direct link with m to forward packets towards o, instead of that via n, the FA translates into a FD.

Forwarding detour: *an internal route $R_X(d)$ detours, i.e., a forwarding detour occurs for destination d, when an upstream router r_j in $R_X(d)$ would not have used r_{j+1} in $R_X(d)$ as next-hop to reach a downstream router r_z of $R_X(d)$, i.e.,*

$$\exists\, j < z \leq n \mid \mathcal{N}_{r_j}(r_z) \neq r_{j+1}$$

Theorem – Forwarding detours imply forwarding alterations: *if an internal route $R_X(d)$ detours, then $R_X(d)$ is subject to forwarding alterations.*

Proof. Assume $R_X(d)$ is subject to forwarding detours, but no forwarding alteration occurs in $R_X(d)$. If $R_X(d)$ is not subject to FAs, then it must hold that $\forall j < k$, $\mathcal{N}_{r_k} \circ \mathcal{G}_{r_k} \circ \mathcal{P}_{r_k}(d) = \mathcal{N}_{r_k} \circ \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d)$. This means that, even though gateways $\mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d)$ and $\mathcal{G}_{r_k} \circ \mathcal{P}_{r_k}(d)$ may differ for different routers, the next-hop that all routers choose, besides matching the IGP next-hop for their respective gateway, also matches the IGP next-hop for the gateways chosen by the other routers. By definition, this implies that $R_X(d)$ is the best IGP path leading to all these gateways and, therefore, no detours can occur. This contradicts our original hypothesis stating that $R_X(d)$ was subject to forwarding detours, proving that FDs imply FAs. However, for the sake of completeness, we complete the proof. For this, recall that a property of IGPs is that sub-paths of best IGP paths are also best paths. This means that if any router $r_z \in R_X(d)$ would have been the destination d, then the sub-path leading to it would still have been a best IGP path. Since when $d = r_z$, the destination would be internal, therefore $\forall j < z$, $\mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d) = r_z$ would hold. Consequently $\mathcal{N}_{r_j} \circ \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d) \triangleq r_{j+1} = \mathcal{N}_{r_j}(r_z)$, which contradicts the formal definition of forwarding detours. Again, this proves that FDs imply FAs. ■

Note that FDs imply FAs, but the converse may not hold, i.e., the relationship is **FDs** $\Rightarrow$ **FAs**. This is highlighted by the right side of Fig. 5.5, where router n deviates traffic towards router o, but the resulting route is still the best IGP path between routers l and o. In general, this occurs when the the best IGP path between r_0 and r_n either includes the sub-path from r_0 to r_k, the router that introduces the FA, or is a sub-path of the path between r_0 and $\mathcal{G}_{r_0} \circ \mathcal{P}_{r_0}(d)$.

By combining our two theorems, we thus have that **FDs** $\Rightarrow$ **FAs** $\Rightarrow$ **RIes**. When FDs occur, there is a router r_k between r_j and r_z that introduces a RI choosing a different gateway than r_j. This router r_k uses different next-hops to reach both gateways, i.e., $\mathcal{N}_{r_k} \circ \mathcal{G}_{r_k} \circ \mathcal{P}_{r_k}(d) \neq \mathcal{N}_{r_k} \circ \mathcal{G}_{r_j} \circ \mathcal{P}_{r_j}(d)$, and introduces a FA. Since r_k (re)directs traffic towards $\mathcal{G}_{r_k} \circ \mathcal{P}_{r_k}(d)$, when r_j would not have chosen r_{j+1} as next-hop for this gateway, a FD results. This is exactly the case in the right side of Fig. 5.6 where $r_j = l$, $r_k = p$ and r_z could be either m or o.

Finally, note that when FDs occur, they create **multi-path routing patterns** inside ASes. This results from the fact that the best IGP path between the endpoints is used to forward traffic of prefixes either non-subject to RIes, or subject to FAs but not FDs (left side of Fig, 5.6), but for those destinations and prefixes subject to FDs, the resulting internal routes differs (e.g. right side of Fig. 5.6).

In general, between two routers i and e of AS X, different root causes may lead to distinct FDs between these endpoints, forming a set $\mathcal{R}_X^{FD}(i, e)$ of detouring routes. Each detouring route of $\mathcal{R}_X^{FD}(i, e)$ will be associated to a specific sets of prefixes, all subject to the same FAs. This is illustrated in Fig 5.7 where $\mathcal{R}_X^{FD}(l, o) = \{R_1, R_2, R_3\}$ such that $R_1 = [l, n, p, m, o]$, $R_2 = [l, m, q, o]$ and $R_1 = [l, n, p, m, q, o]$, each respectively shown in Fig 5.7a, 5.7b and 5.7c, while the best IGP path between l and o actually is $[l, m, o]$.

5.2 Similarities and differences between FDs, LB and TE

In this section we show why differentiating between FDs, LB and TE is not trivial. While Sec. 5.2.1 provides examples of simple but inaccurate FD-detectors, Sec. 5.2.2

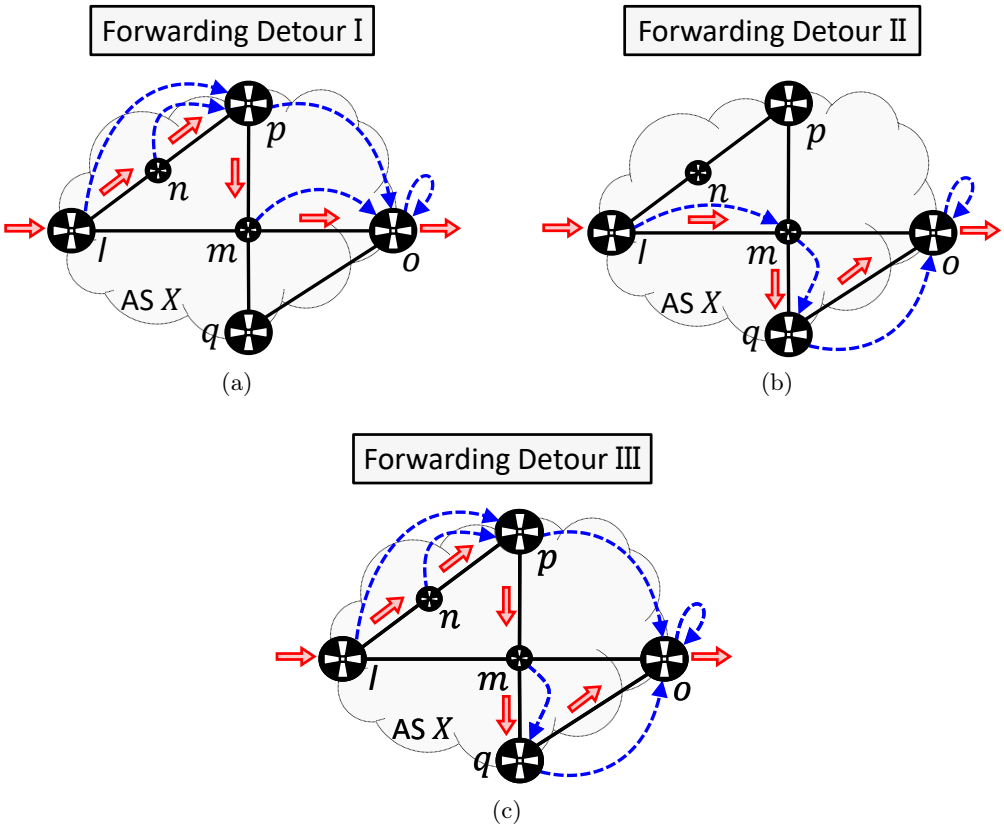


Figure 5.7: Multiple distinct forwarding detours between the same endpoints. In Fig. 5.7a, router p introduces a FA, and this leads to a FD. The cases of Fig. 5.7b and 5.7c represent more complex scenarios where multiple FAs occur. In particular, the FAs are introduced by m and q on the first case, and additionally by p on the latter. The three scenarios produce forwarding routes subject to distinct FDs that may co-exist when they affect different sets of prefixes.

explains why looking at which routes are revealed when different prefixes are targeted is the first step to develop a more effective approach to detect FDs.

5.2.1 Simple but naive methods to detect FDs

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. Considering Fig. 5.8, 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?

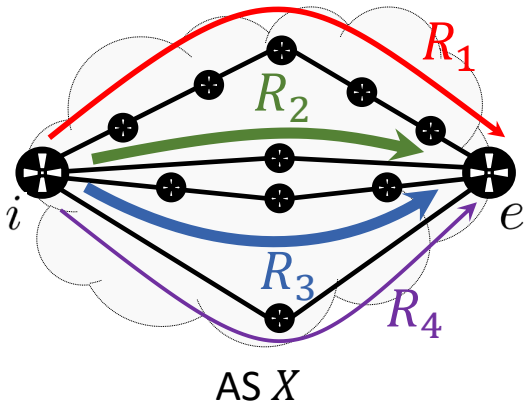


Figure 5.8: Forwarding pattern when $\mathcal{R}_X^{FD}(i,e) = \{R_1\}$, $\mathcal{R}_X^{LB}(i,e) = \{R_2, R_3\}$ and TE $\mathcal{R}_X^{TE}(i,e) = \{R_4\}$. The size of every arrow is proportional to number of prefixes for which each route is used.

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. 5.8, and vice-versa. On the other hand, TE routes are not restricted to be shortest paths between two endpoints. 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 transit traffic traverses exactly two ASBRs inside an AS. Under this assumption, we could first learn the ASBRs from all traces, and then pinpoint FDs looking if three or more ASBRs of the same AS were traversed in any trace. For example, in Fig. 5.1b, 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. 5.1a 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, as we detail next.

5.2.2 Forwarding patterns for LB, TE and FDs

To better understand the similarities and differences between FDs, LB and TE, we propose to analyze the **forwarding pattern** between i and e, i.e., whether the routes of X leading from i to e which traffic traverses vary depending on the considered prefixes. In other words, we propose to closely study which routes of X, leading from i to e, are used depending on the targeted prefixes. In particular, LB may be deployed with different LB flavors that produce specific forwarding patterns. In particular, we focus on hash-based LB flavors, recalling that per-packet LB is

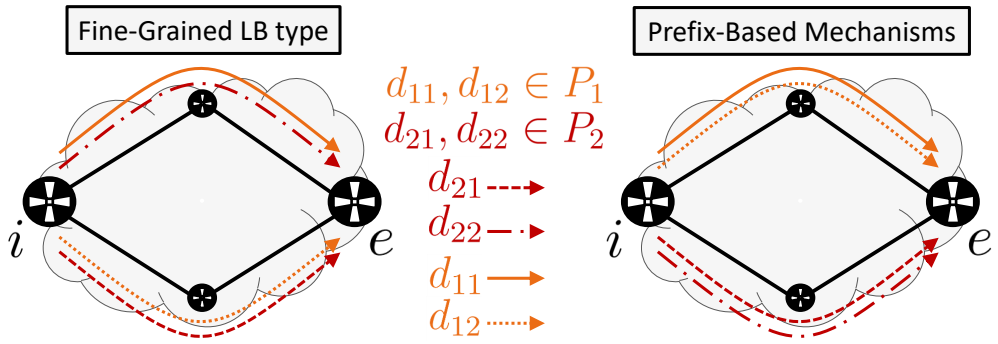


Figure 5.9: Forwarding patterns for F-LB and prefix-based mechanisms. For F-LB, all internal routes of $\mathcal{R}_X^{LB}(i,e)$ are used for both P_1 and P_2. On the other hand, for prefix-based mechanisms, e.g. C-LB, traffic targeting P_1 and P_2 flows through different interal routes. This also holds for FDs and TE, where prefixes subject to them flow across the routes in $\mathcal{R}_X^{FD}(i,e)$ and $\mathcal{R}_X^{TE}(i,e)$ respectively, while the remaining prefixes are associated to best IGP paths. Note, that for prefix-based mechanisms, different routes may be revealed tracing different prefixes, but each internal routes is usually used to forward traffic of multiple distinct prefixes.

rarely found in practice (see Sec. 2.2.2). As we study next, per-prefix LB, FDs and TE produce a similar forwarding pattern that we call **prefix-based forwarding pattern**, that differs from that produced by per-dest LB and per-flow LB.

The set $\mathcal{R}_X^{LB}(i,e)$ results from the presence of load balancers, i.e., routers that may use different next-hops towards the same gateway, and thus generate FAs but not FDs.[3] To forward a packet, a load balancer first needs to determine the gateway G to be reached, e.g., G is usually the iBGP next-hop for the longest matching prefix of an external destination IP address d. In particular, b may know a set possible next-hops that it may use to reach G. As an example, for ECMP, this set comprises all next-hops for which the IGP cost towards the gateway is the same. Every time b has to forward a packet towards G, one out of all the available next-hops needs to be chosen. The way this choice is made depends on the LB flavor deployed.

When b implements a **fine grained LB type (F-LB)**, i.e., either per-dest LB or per-flow LB, the next-hop lookup function that b uses is $\mathcal{N}_b(G,d,h)$. In addition to gateway G, b uses the destination IP address d and another parameter h as argument. In particular, h may represent the transport ports, the source IP address, or a combination of them. For a fixed gateway, keeping one parameter constant while varying the remaining one, e.g., h and d respectively, allows to explore the different next-hops. Consequently, running traceroute, the internal route each trace reveals may vary as the destination changes. This change of route also applies even when the destinations traced belong to the same prefix. This property is illustrated on the left side of Fig. 5.9, 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, exploring one prefix is enough to reveal all routes of $\mathcal{R}_X^{LB}(i,e)$ for F-LB.

[3]Strictly speaking, the FAs that load balancers produce are not of the same type as the ones we analyzed before. Indeed, rather than based on underlying RIes, these FAs result load balancers b that implement more sophisticated next-hop lookup functions $\mathcal{N}_r(\cdot)$, which we study next.

On the other hand, for **coarse-grained LB types (C-LB)**, namely per-prefix LB, b relies on the outcome of a function $\mathcal{N}_b(G, \mathcal{P}_b(d))$ to determine the next-hop used, where $\mathcal{P}_b(d)$ represents the most specific prefix covering d according to b. In contrast with F-LB, besides gateway G, only $\mathcal{P}_b(d)$ is used as argument. As a consequence, packets are discriminated 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. For this reason, we say that C-LB is a **prefix-based mechanism**. As an example, on the right side of Fig. 5.9, per-prefix load balancer i chooses different next-hops for prefixes P_1 and P_2, but always the same and unique for traces belonging to the same prefix. Indeed, with C-LB, there is no route variation for destinations belonging to the same prefix.

From this analysis, we can derive a critical concept: **the forwarding pattern of C-LB, besides being different to that produced by F-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 a **prefix-based forwarding pattern**. As we will see, by actively probing destinations across multiple prefixes, and taking note of which routes are used for each prefix, it is possible to differentiate F-LB and from prefix-based mechanisms. While for the first, all prefixes tend to group into a unique set associated to a common set of routes, for the latter disjoint sets of prefixes, one per route, appear instead. As per-prefix LB, FDs and TE are alike, differentiating among them is more challenging than when just considering per-dest/flow LB.

The proposal of studying forwarding patterns, though more promising than the heuristics in Sec. 5.2.1, still does not explain how to actually identify the existence of $\mathcal{R}_X^{FD}(i, e)$, i.e., how to differentiate the underlying cause generating a prefix-based mechanism. We first focus on being able to detect prefix-based mechanisms in Sec. 5.3 and then explain how this can be turned into an FD-detector in Sec. 5.4.

5.3 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. 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

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

of the prefixes for which an ASBR-couple is observed. Finally, the merging phase discriminates between F-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. 5.4.

5.3.1 Exploration phase

This step collects ASBR-couples and internal routes across ASes. For this, we perform a lightweight traceroute campaign, launching one trace per prefix (e.g. a /24 subnet). An IP-to-AS mapping tool is used to determine ASBR-couples, and the internal routes inside each AS. Since the internal routes of each AS are collected for transit traffic, they are actually TIRs. Since the destinations that are used are randomly chosen, it could happen that few TIRs are gathered for some couples. To enlarge the set of internal routes for each of them, we also collect their DIRs. Each DIR is obtained by tracing the egress-ASBR of each couple, and has a key role in the detection of FDs, as we will detail in Sec. 5.4. We discard those couples for which the DIR cannot be determined (see Sec. 5.4.2).

As a last step, we annotate the prefixes for which each internal route was revealed. For TIRs, this is the /24 subnet (usual longest BGP prefix [**VisibilityIPv4**]) covering the destination IP of the trace from which the internal route was extracted. On the other hand, we consider /32 prefixes for DIRs, e.g. for a couple (i, e), then $e/32$. In the left table of Fig. 5.10 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.

5.3.2 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. 5.10, for both prefix-based mechanisms and F-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 shown on the figure, the prefix-grouping phase returns the same result for F-LB and prefix-based mechanisms. Thus, further analysis is required to be able to differentiate between them.

Finally, 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 $\mathbb{P}_X(i, e)$. Note that, at this stage, $\mathbb{P}_X(i, e) = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_r\}$ is a set of sets of prefixes, whereas $\mathcal{R}_X(i, e) = \{R_1, R_2, \dots, R_r\}$ 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. 5.10, where $r = 4$.

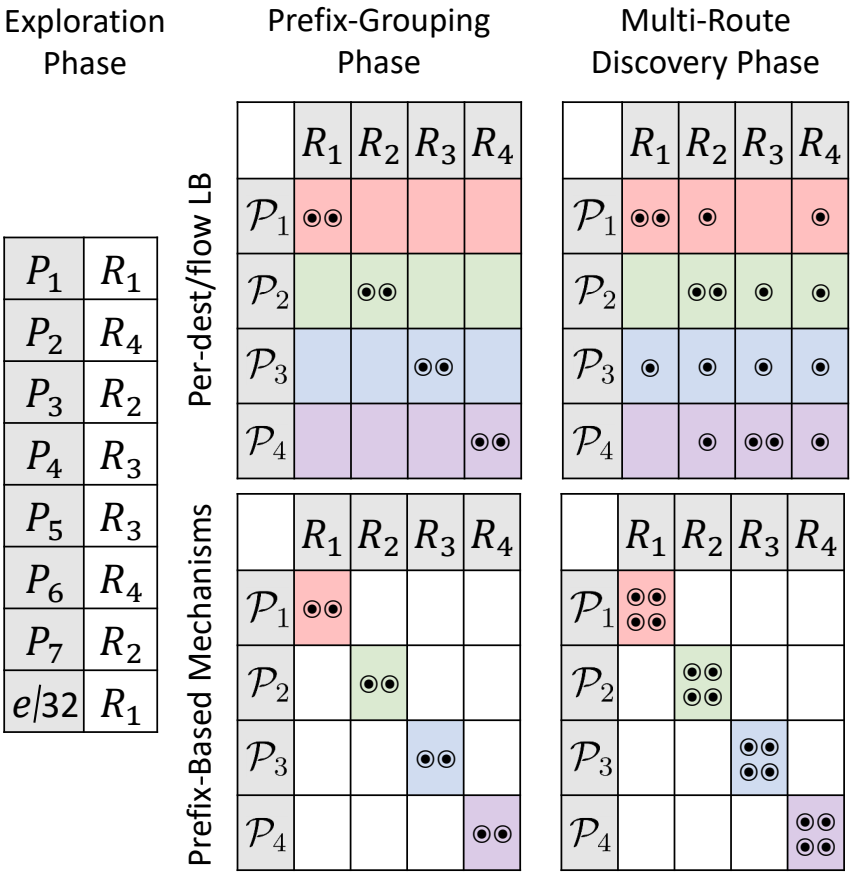


Figure 5.10: Detecting the type of forwarding pattern for an ASBR-couple (i, e). While the colored cells represent the routes associated with each set of prefixes, the dots show those revealed while tracing. The exploration phase runs `traceroute` and reveals one internal route per measured prefix. The prefix-grouping phase then groups those prefixes for which the same route was revealed. At this stage, the result is the same for F-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 F-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 F-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.

5.3.3 Multi-route discovery phase

This block extends the probing and collection of TIRs 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 the (possibly enlarged) set of routes $\mathbb{R}_X(i,e)$ are then passed to the merging phase.

The right matrices of Fig. 5.10 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. 5.2.2, the outcome of the multi-route discovery phase is different for prefix-based mechanisms and F-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 F-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 for F-LB, i.e. either per-dest LB or per-flow LB, the destination IP address is directly used as argument in the implementation of function $\mathcal{N}_b(\cdot)$. 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 F-LB, 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. 5.10.

5.3.4 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 F-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 F-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 F-LB and prefix-based mechanisms, and thus allows to determine if the forwarding pattern for an ASBR-couple (i,e) inside AS X is prefix-based.[5] For F-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. 5.10, 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. 5.10, the composition of $\mathbb{P}_X(i,e)$ and $\mathbb{R}_X(i,e)$ does not change, thus it holds that $s = r > 1$, being $s = 4$ in this particular example. [7]

[5] Note that F-LB and prefix-based mechanisms may interfere with each other, generating more complex forwarding patterns. As we discuss in Sec. 5.6, 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 $r > 1$.

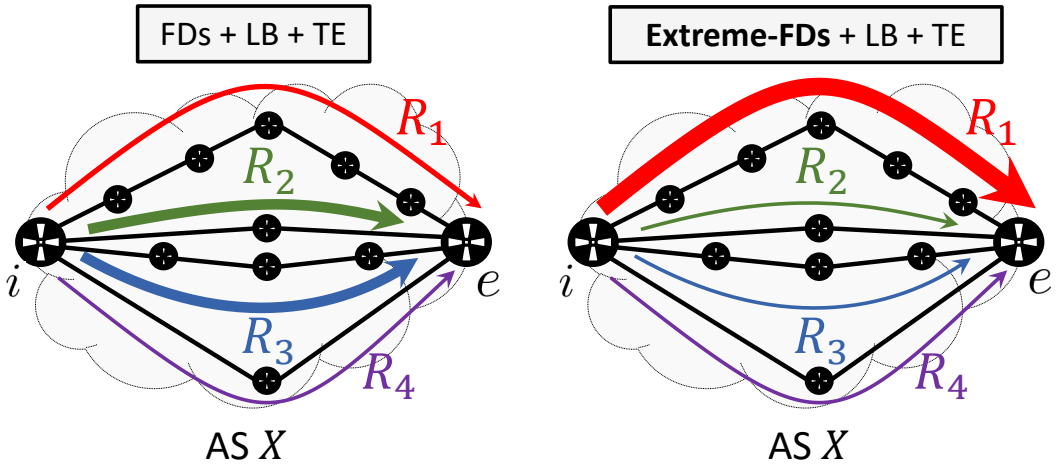


Figure 5.11: Impact of extreme-FDs on forwarding patterns. For both cases, $\mathcal{R}_X^{FD}(i,e) = \{R_1\}$, $\mathcal{R}_X^{LB}(i,e) = \{R_2, R_3\}$ and TE $\mathcal{R}_X^{TE}(i,e) = \{R_4\}$. The size of every arrow is proportional to number of prefixes for which each route is used. While the forwarding pattern inside AS X on the left case undergoes no major change due to FDs, on the right case it is largely modified by the occurrence of extreme-FDs, i.e., FDs for most prefixes.

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.

5.4 An FD-detector

In this section we present the FD-detector we designed. To tackle the challenging task of differentiating among prefix-based mechanisms, i.e., to correctly detect FDs while avoiding to misclassify per-prefix LB and TE, we propose to focus on cases involving **extreme-FDs**. In the scenarios, since 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 illustrated in Fig. 5.11, where $\mathcal{R}_X^{FD}(i,e) = \{R_1\}$, $\mathcal{R}_X^{LB}(i,e) = \{R_2, R_3\}$ and $\mathcal{R}_X^{TE}(i,e) = \{R_4\}$. On the left case of Fig 5.11, the forwarding pattern indicates that 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 $\mathcal{R}_X^{FD}(i,e)$ and $\mathcal{R}_X^{TE}(i,e)$. On the contrary, on the right side of Fig. 5.11, extreme-FDs occur and the forwarding pattern notoriously contrasts with that described before.

In particular, Sec. 5.4.1 explains how the detector of prefix-based forwarding patterns can be turned into an FD-detector by adding a last phase: an FD-verdict looking for a lonely DIR, i.e., a DIR in a set of routes associated with few prefixes. Finally, Sec. 5.4.2 describes how we implemented our FD-detector based on current

probing tools.

5.4.1 The FD-verdict: looking for 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 of the couple. 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 $\mathcal{R}_X^{LB}(i, e)$.

When we conclude for a prefix-based forwarding pattern relying on the detector of Sec. 5.3, 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), in general $\sum_{k=1}^{s} |\mathcal{P}_k|$, the total number of prefixes for which the couple is revealed, and s, the number of sets conforming the partitions $\mathbb{P}_X(i, e)$ and $\mathbb{R}_X(i, e)$ 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 modifies the usual distribution of prefixes across routes: fewer prefixes are associated with LB routes, as seen in the left side of Fig. 5.11. 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, as seen on the right side of Fig. 5.11. 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.

[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.

To illustrate the behavior of the FD-verdict, let us recall the example of Fig. 5.10, 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. 5.10, 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 F-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.

5.4.2 The FD-detector: a tool to be run in the wild

In this section we describe how we turned the algorithm of Sec. 5.3, incorporating the FD-verdict, into a tool able to detect FDs in the wild.

Measurement Infrastructure We run our FD-detector leveraging 100 vantage points (VPs) of the NLNOG RING monitoring infrastructure on May 26th 2020.[9] 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, we are able to tune the probing rate and number of concurrent measurements.[10] 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.

Collecting Traces We used `scamper` [96] to run ICMP-Paris `traceroute` [97] at 200 pps towards a list of IP addresses extracted from the Internet Address Hitlist provided by the USC/ISI ANT project [98], 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 TIRs, i.e., internal routes of ASes traversed by transit traffic, that thus do not own the traced IP addresses.

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

[9] `https://ring.nlnog.net`
[10] `https://atlas.ripe.net/docs/udm/#rate-limits`

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. '$*$', a missing hop). In other words, we only keep unambiguous ASBR-couples. To map from IP-to-AS, we use `bdrmapIT` [99], configured on top of CAIDA's IP-to-AS mapping dataset [**pfx2as**]. Internal routes $R_X(d_j)$ including loops or hops mapping to an AS distinct from X are discarded. Moreover, by discarding internal routes where i and e are directly connected, we filter invisible MPLS tunnels [**TNT**]. 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., $\sum_j |\mathcal{P}_j| < 100$.

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 another ingress-ASBR are discarded. 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 in both traces we encounter a third ingress-ASBR i'', then the 4 couples are removed. On the other hand, 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 also discard (i', e), thus only keeping (i, e) at the end of the process.

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. 5.2.2, and illustrated in Fig. 5.9 and Fig. 5.10, 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, varying the traced destination would allow to reveal all LB paths even for a unique prefix in the case of F-LB. 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. 5.6. In any case, the prefix-grouping phase greatly reduces the number of prefixes to be probed, thus allowing for this concession.

Dealing with missing hops The internal routes collected may include missing hops, that appear as "∗". 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.

5.5 Capturing forwarding detours in the wild

In this section we discuss the results we obtained running our FD-detector in the wild. First, Sec. 5.5.1 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. 5.5.2 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. 5.5.3, 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. 5.5.4, 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 in Sec. 5.5.5, i.e., with the observed binary characteristics, and present the efforts we invested in validating our results in Sec. 5.5.6.

5.5.1 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.5.4.2). On the other hand, when tracing the egress-ASBRs to collect DIRs, we had a success rate usually between 50% and 60%.

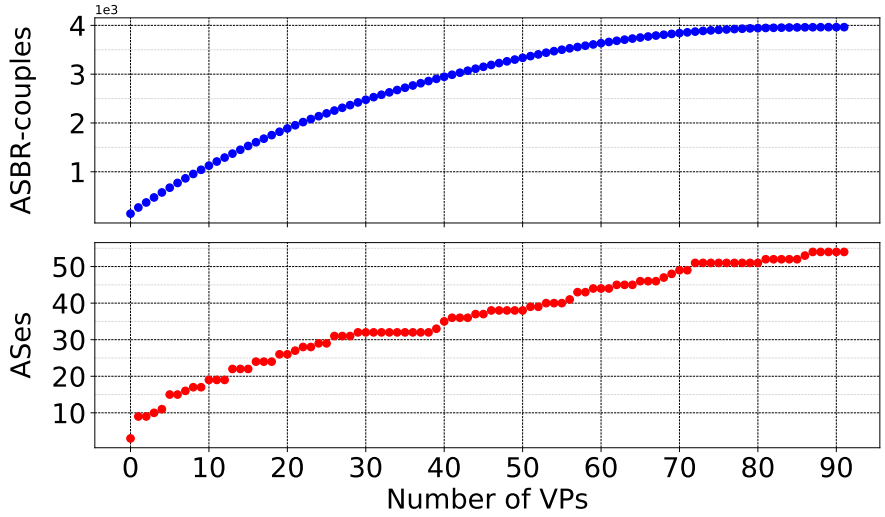


Figure 5.12: Marginal utility of adding NLNOG RING's VPs in terms of distinct ASBR-couples (top) and unique ASes (bottom). For more than 70 VPs, the gain in negligible.

Our FD-detector was able to analyze 3963 ASBR-couples spanning 54 ASes. Fig. 5.12 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.

5.5.2 Fowarding 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. 5.13 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 F-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 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.

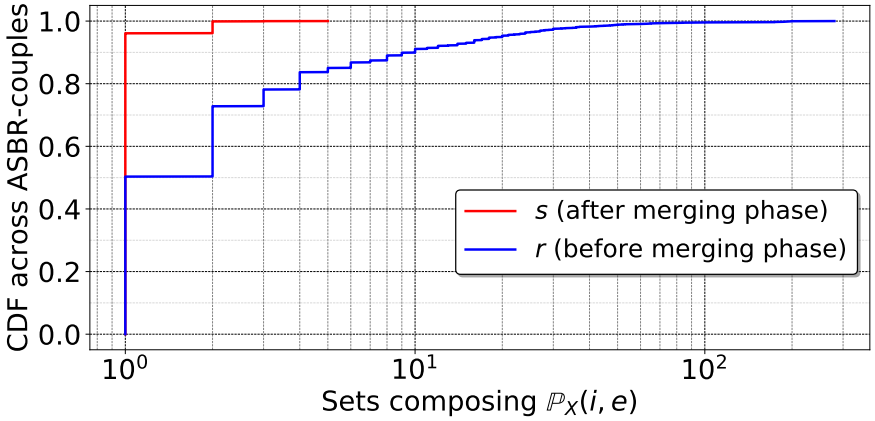


Figure 5.13: Cumulative number of sets composing $\mathbb{P}_X(i,e)$ across ASBR-couples before (r) and after (s) the merging phase. When $r = 1$, no multi-path routing pattern was observed. The difference with $s = 1$ relates to cases where we find a forwarding pattern that corresponds to that of per-dest/flow LB. Finally, when $s \geq 2$ a prefix-based forwarding pattern is observed. In these cases, in general, $s = 2$, and they are FDs.

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. 5.14 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. 5.4.1). The curve in Fig. 5.14 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. 5.14 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. 5.14 are those ASBR-couples for which $s = 2$ in Fig. 5.13. 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). In other words, no TIR follows the same path as the one seen for the DIR. 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.

5.5.3 Distribution of FDs per AS and ASBR-couples

Fig. 5.15 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

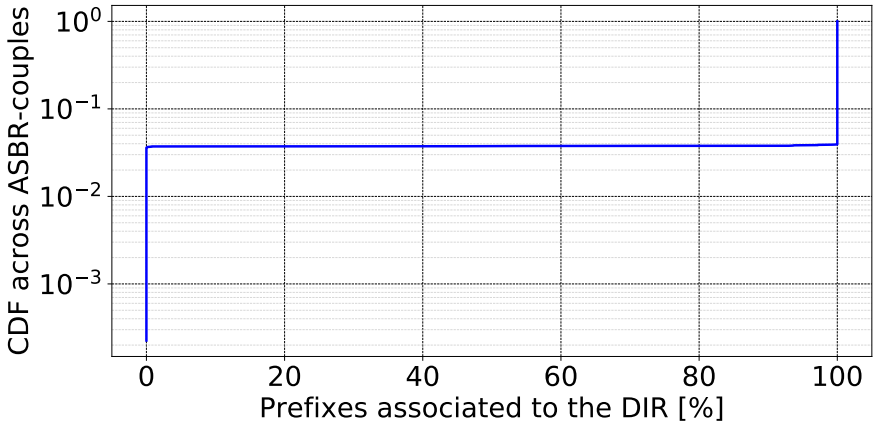


Figure 5.14: Cumulative number of prefixes associated to the DIR across ASBR-couples. We observe a clear binary pattern: for any couple, either all traffic detours (left side, ∼4%), or none does (right side, ∼96% of the cases). Hence, our FD-detector is not sensible to the value of the threshold $t(Z, \mathbb{P}_X(i, e))$.

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. 5.15 seems to be populated with ASes with a high AS Rank [12], 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. 5.16 complements Fig. 5.15 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.

5.5.4 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. 5.16, there exist multiple ASBR-couples (i, e) subject to FDs for which the same ingress-

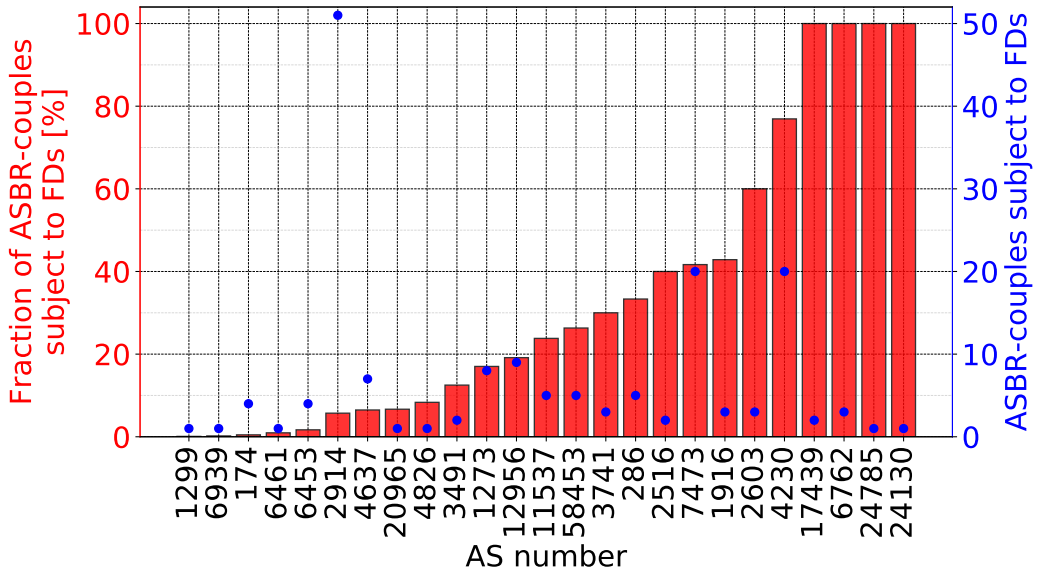


Figure 5.15: Quantification of ASBR-couples subject to FDs per AS. While most ASes have less than 10 couples subject to FDs (blue dots), the fraction they represent out of the total in their AS (red bars) largely varies. This indicates that the problem of FDs is AS-dependent.

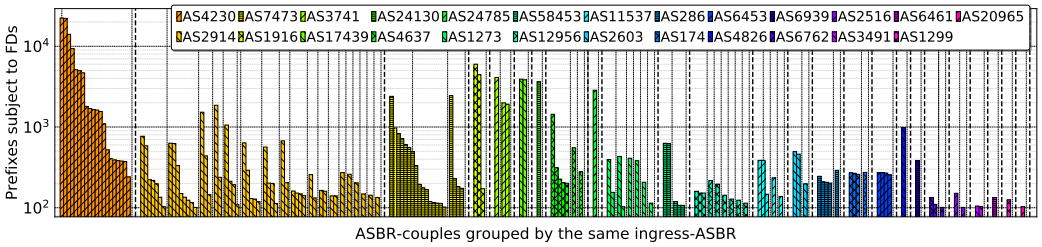


Figure 5.16: Number of prefixes subject to FDs per ASBR-couple. The bars are separated by dashed lines to emphasize a distinct ingress-ASBRs. The number of ingress-ASBRs, ASBR-couples and prefixes subject to FDs strongly depends on the AS studied.

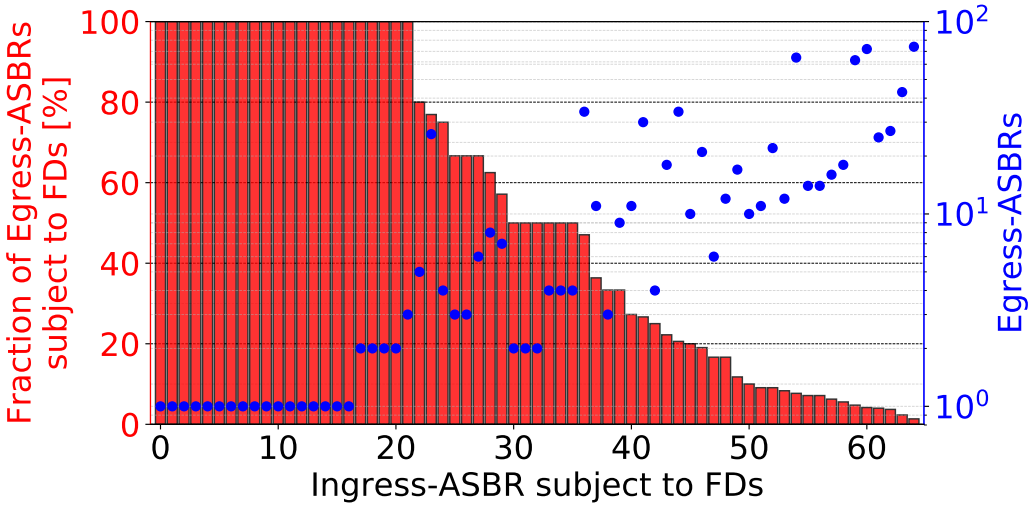


Figure 5.17: Fraction of egress-ASBRs that are subject to extreme-FDs (red bars) out of the total (blue dots) for each ingress-ASBR. The tendency shows that the more egress-ASBRs per ingress-ASBR, the less the fraction subject to FDs. However, for 17 ingress-ASBRs we cannot conclude anything since they only appear in one ASBR-couple.

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. 5.17 shows the fraction of egress-ASBRs subject to FDs associated to each 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.

5.5.5 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. 1.3), 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. 5.14) 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. 5.1 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. 5.17 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. 5.15, 5.16 and 5.17, 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.

5.5.6 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. 5.1 accurately describes the network hosting such router. While for some prefixes the router was generating BGP lies [43], 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.

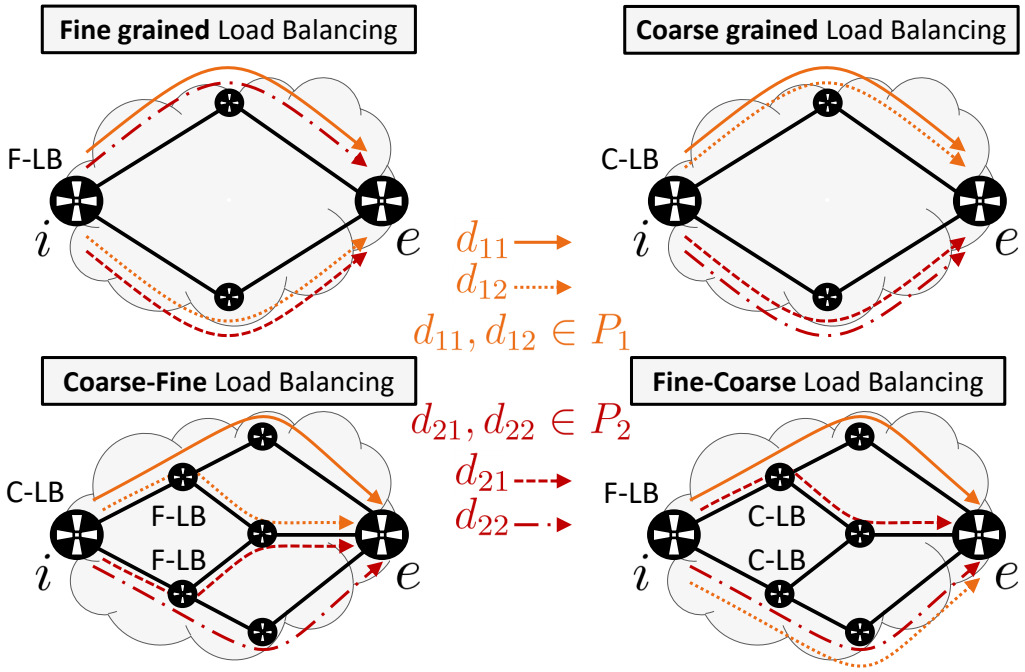


Figure 5.18: Complex LB flavors. In particular, coarse-fine LB implies that C-LB is followed by F-LB. This LB flavor generalizes C-LB: a set of prefixes is now associated to more than one route, and these routes are reserved only for those prefixes. As with C-LB, for CF-LB it would hold that $s > 1$. On the other hand, fine-coarse LB results from applying F-LB upstream of C-LB. Different to F-LB, with FC-LB not all routes are used for all prefixes. However, this LB flavor preserves the property of F-LB where routes are not reserved for a unique specific set of prefixes, but rather they may be used for different ones. As a consequence, the merging phase would likely output $s = 1$ for FC-LB.

5.6 Discussion: robustness of the FD-detector

In this section we analyze how our FD-detector performs face to complex forwarding patterns in Sec. 5.6.1, explain why routing changes and IP-to-AS mapping errors do not induce the results we obtained in Sec. 5.6.2 and 5.6.3, illustrate why the probing cost of the multi-route discovery phase was sufficient in Sec. 5.6.4 and discuss why our analysis does not require alias resolution techniques in Sec. 5.6.5.

5.6.1 An FD-verdict handling all interactions of FDs and LB

The LB types studied in Chapter 2, Sec 2.2.2, and Chapter 5, Sec. 5.2.2, F-LB and C-LB, may be mixed to produce **hybrid LB flavors**. For example, in the bottom panels of Fig. 5.18 we consider configurations where for a given gateway e, a load balancer i has n next-hops that are also load balancers with m possible next-hops. The top panels show the **basic LB flavors** we analyzed before, i.e. F-LB and C-LB, to emphasize how the hybrid LB flavors generalize them. As we detail next, our FD-detector is not affected by the hybrid LB flavors since the merging phase would output a value of s equal to the one obtained with the simpler LB flavors

they generalize.

When C-LB is applied upstream of F-LB, this combination results in a generalization of C-LB, that we name **coarse-fine grained LB (CF-LB)**. For traces concerning a fixed set of prefixes $\mathcal{P}_j$, instead of a unique route R_j, a set of m 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 $\mathcal{P}_j$. Similar to C-LB, for CF-LB the property $s > 1$ still holds (in particular it would be $s = n$). This is shown on the bottom-left topology of Fig. 5.18, where $s = 2$ and the two top routes and bottoms one are used for P_1 and P_2, respectively. On the other hand, when load balancers are applied in the reverse order, that is, with F-LB followed by C-LB, we call the resulting LB flavor **fine-coarse grained LB (FC-LB).** With FC-LB each prefix is not anymore forwarded thorough all routes of $\mathcal{R}_X^{LB}(i, e)$ as with F-LB. Indeed, in these cases, tracing a set of prefixes $\mathcal{P}_j$, the same set of routes $\mathcal{R}_j$ is consistently found. It can be shown that each set $\mathcal{R}_j$ contains n routes, and that there are m^n sets such that, different to C-LB and CF-LB, $\forall j, k, \mathcal{R}_j \cap \mathcal{R}_k \neq \emptyset$. These intersections usually contain multiple routes, and thus the merging phase would likely output the same as for F-LB, i.e., $s = 1$. The effect of FC-LB is shown in the bottom-right topology of Fig. 5.18 where we see that multiple routes are used to forward traffic concerning P_1 and P_2 respectively, some in common and others not.

Finally, note that detouring traffic may traverse load balancers, thus FDs may be subject to LB. In particular, if either F-LB or FC-LB was used, then no major changes would occur since the FD-detector would be able to group the detouring routes into a unique set of routes during the merging phase. On the other hand, if either C-LB or CF-LB was applied, then the prefixes subject to FDs would be evenly distributed across multiple non-overlapping routes or sets of routes, respectively. Hence, our FD-detector would not be able to merge these load balanced FDs into a unique set of routes. A priori, this would generate that the extreme-FDs pattern would not be clear, since traffic would distribute across different sets of routes. However, we actually designed the FD-verdict to take this case into account: rather than searching for a set that presumably is the one resulting from FDs (that when C-LB or CF-LB are applied would not be easy to spot), 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.

5.6.2 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.

5.6.3 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` [99] failed to work properly. We will assume that (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.

5.6.4 Measurement stopping points

The MDA uses adaptive measurement stopping points (see Sec. 2.3.3) while we 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 [57]. 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 [65]. Indeed, 2 and 16 represent the largely most common and the maximum number of next-hops usually found in practice, respectively [**TracerouteConfuses**, 58, 67]. As discussed in Sec. 5.5.2, the patterns we observe in our results highlight the effectiveness of the merging and multi-route discovery phases.

5.6.5 Alias Resolution: a nice, but dangerous additional feature

Similar to the LB studies presented in Sec. 1.3, our methodology performs its analysis at the IP-level. However, alias resolutions techniques (e.g. MIDAR [**MIDAR**]) 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.

5.7 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 chapter, first we model how RIes and FAs produce FDs, and then 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 our study, 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.

Chapter 6

Conclusion and Research Directions

Contents

In this section we conclude the thesis, highlighting our main scientific contributions in Sec. 6.1 and pointing out future research directions in Sec. 6.2.

6.1 Takeaways

In this thesis, the objective is to detect hidden broken pieces of the Internet, i.e., malfunctioning components, networks facing limitations and even selfish networks prioritizing their own revenue rather than the better performance of the Internet.

First, we investigate whether IXPs have been successful in Latin America, a region that has previously received little attention in Internet studies. This study has four main contributions.

- We construct the most comprehensive available BGP dataset of Latin America.
- We find that Latin American states have been involved in the creation of national IXPs in several ways: legislation, regulation, sponsoring, funding, operations and serving traffic from/to IXPs. In many cases IXPs in LatAm, similar to others in Europe, are managed by non-profit organizations.
- We see that IXPs in developing regions such as Africa and Latin America not only have had a similar growth in the last years, but also seem to have reached maturity, i.e., have been able to attract as many local ASes as so do some well-established IXPs in Europe. However, European IXPs are rather

international hubs, meaning that they have also managed to gather members from different regions. This internationalized market could be exploited in the future by the less renown, and rather local, IXPs in Latin America, Asia and Africa.

- We notice that, in several Latin American countries, the existence of monopolistic ASes, some state-owned, seem to have prevented the proliferation of IXPs, and lead them to be failed IXPs, i.e., countries with no IXP at all, or with IXPs that have not managed to attract a large number of members.

Second, we study whether BGP lies occur on the Internet, i.e., if packets flow on the Internet through AS-paths other than the ones advertised in BGP messages. In a nutshell, our contributions are:

- We explain different ways in which an AS may generate BGP lies, either affecting the control paths that are advertised or the data path that packets follow. We show that detecting BGP lies is challenging since the detection of lies may be wrongly triggered by noise affecting the measurements, e.g. AS siblings, third-party addresses and missing hops.

- We propose a modular framework allowing to detect highly-potential BGP lies. For this, we apply path-rewriting rules that filter the noise affecting the comparison of control paths and data paths. Our framework is modular, allowing to implement multiple noise-filtering models, each of which tackles differently the correction of errors interfering with the collected data.

- We run measurements from 8 vantage points and over time, a coverage never achieved before for this kind of analysis. We find that, even relying on the most conservative noise-filtering model, some mismatches between control and data paths remain, likely representing BGP lies. In the vantage points where few lies are found, the results are stable in time, otherwise we see that the number of discrepant paths per day have a larger variation.

Lastly we study forwarding detours, i.e., if traffic inside ASes eventually flows through forwarding routes that divert or diverge from the expected best IGP paths. We make the following contributions:

- We investigate the root causes that produce forwarding detours, showing that they result from routing inconsistencies that generate forwarding alterations. However, not all routing inconsistencies and forwarding alterations generate forwarding detours.

- We explain why detecting forwarding detours is challenging: they generate multi-path routing patterns similar to those introduced by load balancing and traffic engineering techniques. Moreover, we take into account per-prefix LB, an LB flavor never previously studied in the literature, and propose a new taxonomy differentiating between fine-grained and coarse-grained LB types, which vary the granularity at which flows are defined with respect to the destination IP address of packets.

- We design a methodology that, without requiring privileged knowledge from the networks being analyzed, e.g., knowing the IGP metric, is able to detect whether forwarding detours occur inside them. Our methodology consists in studying the forwarding patterns inside ASes, i.e., the specific sets of forwarding routes that are revealed for different prefixes when tracing multiple IP addresses contained in each of them.
- We propose an FD-detector, to the best of our knowledge the first of its kind, tuned to detect extreme-FDs, i.e., FDs that affect numerous external prefixes. We validate the behavior of the FD-detector with emulations and on a network where we have ground truth.
- We analyze the FD-phenomenon in the wild running our FD-detector from 100 nodes of the NLNOG RING monitoring infrastructure, and find FDs in 25 out of 54 ASes. We find forwarding detours in multiple ASes, with a remarkable binary pattern in which transit traffic traversing between two border routers of an AS either never detours, or always does.

All in all, the studies we carry on in this thesis expose that the Internet is not an infallible system, and that its hidden broken pieces can be exposed by crafting refined methodologies as the ones we develop. To foster replicability and reproducibility, we release the dataset we collected and the code we developed for each of the analysis presented in this thesis.

6.2 Future Work

In this section, we discuss research directions to continue the work in this thesis concerning BGP lies in Sec. 6.2.1, forwarding detours in Sec. 6.2.2 and a combination of both in Sec. 6.2.3. In addition, we present future work we envision in the field of load balancing and topology discovery in Sec. 6.2.4.

6.2.1 BGP lies: more VPs, anomaly detection and malicious ASes

Recalling the analysis in Chapter 4, our results show that the number of mismatches between control paths and data paths are usually stable, except in the cases where the plausible number of BGP lies is high. This leads us to the following question.

Research Question

By continuously tracking the rate of BGP lies over time, can we turn our framework into an anomaly detector?

We believe this is a plausible option for the mid-term. For this, first we would have to identify VPs for which the number of observed discrepancies between control paths and data paths are stable. Notice that determining this does not require

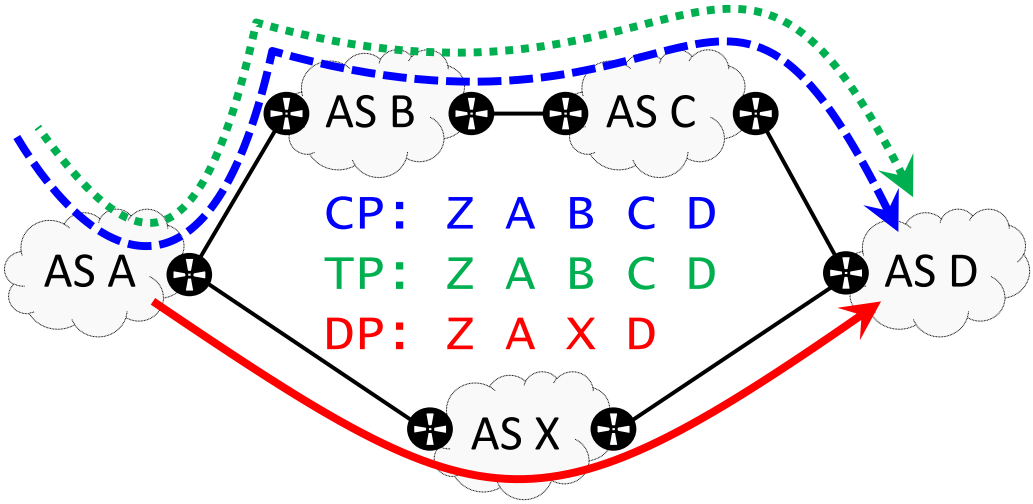


Figure 6.1: BGP lies that are hid by the malicious behavior of AS A. In this example, A is able to discriminate probes issued by traceroute from regular traffic. As a consequence, A sends measuring probes thought B, i.e., the traceroute path (TP) matches the CP. In addition, A diverts regular traffic from the path advertised in BGP, forwarding "regular" packets to X.

running our framework to filter noise, nor does it necessarily require finding co-located VPs, as long as the results are consistent over time. Finally, relying on machine learning or deep learning techniques, we can build a system that keeps track of the variation in time of the mismatch rate, and raise an alarm when anomalous events occur.

In the long term we aim at shedding light on the root cause of the mismatches between CPs and DPs, i.e. we will focus in the even more challenging task of detecting not only BGP lies, but also pinpointing their types and incentives. In that sense, a current limitation of the work in Chapter 4 is that we assume that BGP lies may occur at either the control or data plane, but we do not consider that an AS may be malicious and simultaneously affect both CPs and DPs, forcing these paths to match only when measurements are being carried. In other words, the main problem is that our framework trusts traceroute blindly, and does not question whether the path it reports may differ from the one that any other type of traffic might follow, as shown in Fig. 6.1. Indeed, `traceroute` may be handled differently from regular traffic [28, 29]. For example, we tested configurations on routers and found that it is actually quite simple to enforce different routing policies based on the TTL value in packets. Hence, a malicious AS may divert packets with a TTL above certain threshold, and respect the paths for the ones below it, that are traceroute-like. This brings us to a new research question.

Research Question

Can we extend our methodology to detect malicious ASes trying to hide their lies?

In this case, our task will be to play the role of detectives: the more questions (tests) are performed, the more chances the (malicious) liar may commit an error and the BGP lies be exposed. In particular, for the moment we envision a complementary test comparing the distribution of the RTT for packets with high and low TTL values, respectively. When the difference is high, we argue that this could represent a lie, and trigger an alarm. Instead, when the values are similar, further analysis would be required. For example, other options include re-running traceroute with different transport protocols, or using the IP record route option.

6.2.2 Forwarding detours: finding the forwarding alteration and an FD-detector-lite

Our FD-detector of Chapter 5 detects FDs between ASBRs of an AS. Looking to complement the analysis, we propose the following research question.

Research Question

Can we extend our FD-detector to also pinpoint the routers that introduce the RIes and FAs that generate the FDs we find?

We argue that this can be achieved by generalizing the concept of the DIR. Rather than only targeting the egress-ASBR, we can target all interfaces seen in the detouring paths. In the process of tracing the intermediate interfaces, from the egress-ASBR up to the ingress-ASBR, we expect that, eventually, there will be a DIR that will match a sub-path of the detouring route. This is the triggering event that allows us to identify the router that introduces the forwarding alteration that leads to the FD. The complete procedure is illustrated in Fig. 6.2. First, the forwarding detour needs to be identified running the FD-detector of Chapter 5. The revealed FD is shown in Fig. 6.2a. From o to l, each of the interfaces seen in the detouring path are traced. The first step is to target m, as shown in Fig. 6.2b. The violet route that is revealed does not match the red sub-path extracted from the original detouring route in Fig. 6.2a, hence the process continues. In the following step, displayed in Fig. 6.2c, p is traced. In this case, the red and violet paths match. When this happens, the tracing stops and p is declared to be the router that introduces forwarding alterations.

Since there exists no previous study tackling the same problem of Chapter 5, our solution to detect FDs is as general as possible, considering almost all corner cases. Not only we address the difficulty of filtering per-prefix LB, but also we take

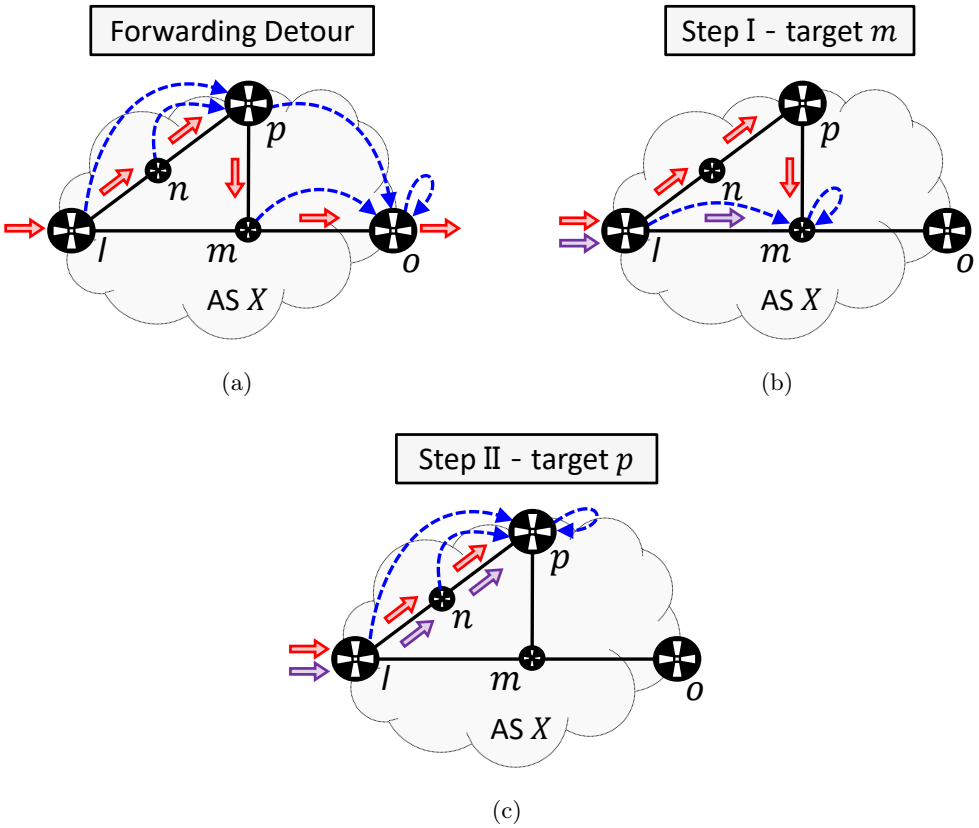


Figure 6.2: Methodology to detect the router that introduces the forwarding alteration leading to a forwarding detour. The red arrows indicate sub-paths of the detouring route, whereas the violet ones are the paths revealed running traceroute towards intermediate interfaces found in the aforementioned path. Fig. 6.2a shows the original detouring route found with the FD-detector of Chapter 5. In the first step after detecting the FDs, shown in Fig. 6.2b, the violet path obtained targeting m and the red sub-path extracted from the path in Fig. 6.2a differ, and thus the measuring process continues. Finally, when tracing p in Fig. 6.2c, the violet and red paths match. Consequently, p is identified as the router that introduces the forwarding alteration that leads to a forwarding detour between l and o.

into consideration how LB may interfere with FDs and design an adaptive threshold to declare the occurrence of FDs. Our results, suggesting that all FDs are extreme-FDs, and that there exists a binary pattern where either all traffic detours or none does suggest we could potentially envision a simpler FD-detector. This brings us to the following question.

Research Question

Can we leverage our better understanding of the FD-phenomenon to generate an FD-detector-lite, with a simpler design?

For this, first we would need to further characterize FDs, i.e., in future work we aim to study whether the RTT, the number of hops and the number of MPLS tunnels observed for detouring paths are greater than the ones for best IGP paths. Moreover, we want to study the number of paths related to the set of detouring routes and to the best IGP paths, respectively. We believe that leveraging this knowledge, we would be able to build an FD-detector-lite. This new detector, different from the one we present in this manuscript, could rely on assumptions based on this FD-characterization.

6.2.3 Where BGP lies and FDs meet: a partial-FIB detector

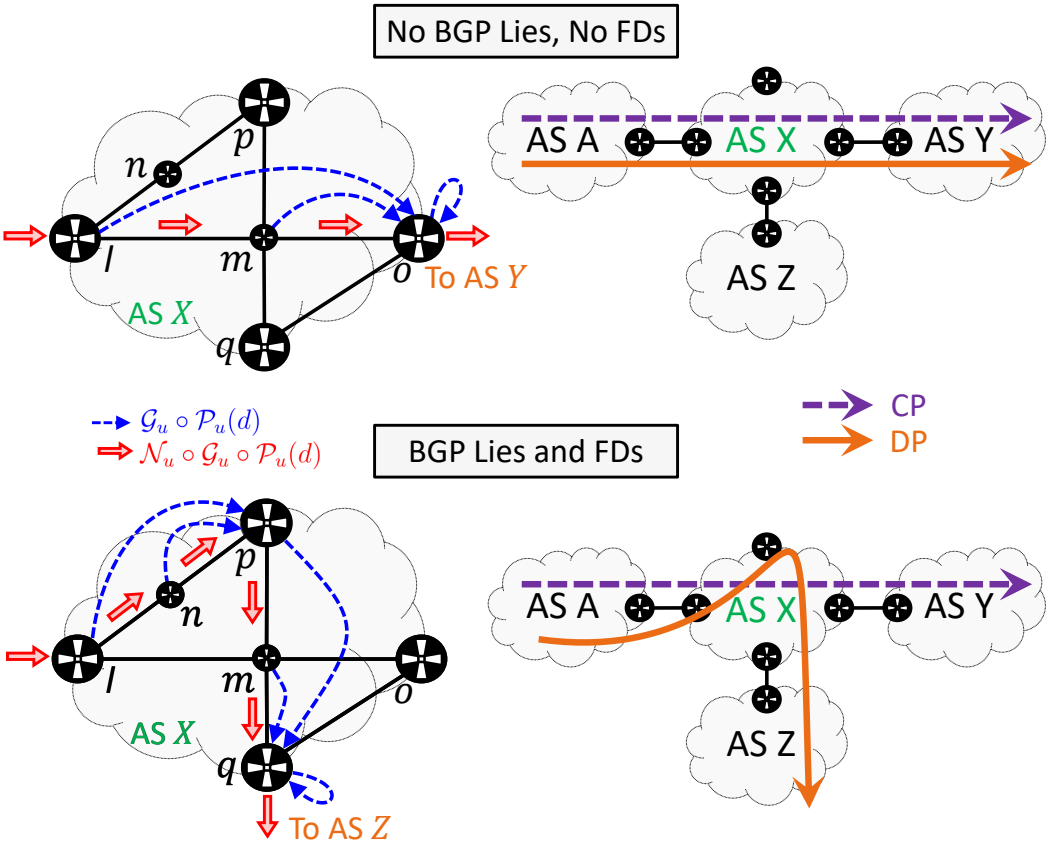


Figure 6.3: Simultaneous BGP lies and forwarding detours. The top figure shows a full-FIB scenario. In this case, packets flow through best IGP paths inside AS X, in particular from l to o. Once o is reached, packets are forwarded towards AS Y, the AS announced in CPs. Consequently, in this case no BGP lies nor FDs occur. On the other hand, the right side figure shows a new scenario where l has a partial FIB. As l uses p as default gateway, and p redirects traffic towards q, an FD occurs. Moreover, since q acts as egress-ASBR and sends packets to AS Z, besides the FD, a BGP lie also occurs. Indeed, instead of forwarding traffic to Y, X is pushing it towards Z. This highlights the potential of combining the framework of Chapter 4 and the FD-detector of Chapter 5.

The studies of Chapter 4 and 5 are closely related: besides the fact that both

allow to detect different hidden broken pieces of the Internet, some of the root causes that generate them may be the same. We previously saw that ASes with technical limitations may use partial-FIB routers, but this may lead to BGP lies. In addition, scalability workarounds using partial-FIB routers with a default route may lead to FDs. Consequently, this brings us to the next question.

Research Question

Can we combine the FD-detector and the noise-filtering framework to generate a partial-FIB detector?

In general, even though a router may have a partial-FIB, it is likely that it is still a full-RIB router. This happens because the memory routers use to maintain RIBs is not a constrained resource as is the one used for FIBs. As a consequence, in terms of BGP data, a router usually has complete routing knowledge: it knows to which ASBR packets should be sent for each prefix, and the AS-path that will be correspondingly followed. Moreover, the router announces the related best paths to ASBRs in neighbouring ASes. However, for those destinations that are not installed in the FIB, the router actually uses a default route pointing towards a default gateway. When the default gateway re-directs traffic towards an ASBR different than the iBGP next-hop the partial-FIB aimed to reach, it is likely that FDs occur. On the other hand, if the default gateway acts as egress-ASBR, it may introduce a BGP lie if it pushes traffic to an AS other than the one appearing on the AS-path. Even more interestingly, when the default gateway pushes traffic to a third ASBR both FDs and BGP lies may occur. The example of Fig. 6.3 illustrates this latter case, and compares it with the case where no FDs nor BGP lies occur to emphasize the contrast. We would like to investigate the potential of combining the methodologies we derive in Chapter 4 and 5 to generate a partial-FIB detector in future work.

6.2.4 A better model of LB, a more efficient MDA

The study we carry on in Chapter 5 allowed us to acquire a deep understanding of LB practices. Besides considering per-prefix LB, an LB flavor omitted in the literature, we proposed a new taxonomy differentiating between coarse-grained and fine-grained LB types. We also modeled the function $\mathcal{N}_b(\cdot)$ that load balancers of different LB flavors apply to choose next-hops. In future work, we would like to continue contributing to this field, looking to answer the following research questions.

Research Question

- **Can we incorporate networking knowledge to the MDA, whose functioning rather relies only on a mathematical model applicable to any generic problem?**
- **Can we plug the prefix-grouping phase of the FD-detector into the MDA to reduce the amount of probing?**
- **Can we use results extracted from previous LB surveys to recalculate the stopping points of the MDA?**

As a first step, a large set of studies concerning related work have suggested that load balancers apply different LB flavors for ICMP and UDP packets[1], but we would like to make a deep study of this phenomenon, not only to validate these results, but also to show the consequences in performance of such characteristic. An interesting point is that, if routers really mainly apply per-dest LB for ICMP packets, the studies putting efforts and describing methodologies to fix the return path of probes are simply overkill and produce no real effect. Indeed, the return path would only depend on the source and destination IP addresses, and trying to fix the ICMP checksum would not produce any change. Therefore, we would like to investigate the configuration of routers and check if there is actually a default option in router's software that triggers this particular behavior. In future work, we would like to study this effect in detail. The method for the moment is not well defined, but we envision studying whether the RTT we obtain varies when targeting different destination IP addresses while using different source IP addresses attached to the same VP. In addition, we will contrast the obtained with ICMP and UDP probes.

As a second step, complementing the modeling of function $\mathcal{N}_b(\cdot)$, we plan on using our forwarding model to shed light on how LB works, since fixing what I believe is a misconception in all the related work on LB I have presented in this manuscript, the probing cost of LB studies may be reduced. I argue that, in general, LB studies adopt a mathematical perspective and wrongly assume that next-hops that load balancers use only depend on the destination IP address. Indeed, they miss the networking perspective, and therefore neglect the effect of the gateway that routers select as the best one for each destination IP address. The conceptual difference is critical: the current model adopted in LB and topology discovery studies implicitly assumes that the Internet is flat, i.e., a unique network. However, the Internet is an interconnection of ASes, each of which may decide whether to apply LB or not independently. In other words, end-to-end paths are a concatenation of the internal routes that traverse each AS (plus the links between ASes). Therefore, LB or topology discovery studies should not be run on a per-destination basis, but rather at the per-ASBR-couple scale. This does not mean that destination IP addresses

[1]This is clearly seen in the results in [58], which suggest that load balancers rather use per-dest LB for ICMP. This result has been also suggested in previous studies, but no further analyzed.

are not important: tracing them provides a means to reveal the ASBRs of each AS, and allows to seek for multi-path routing patterns across ASBR-couples (and links between ASBRs of different ASes, assuming multi-path BGP eventually becomes more popular). In addition, to be fair, running the MDA on a per-destination basis actually makes sense towards the end of paths. Indeed, at the edges, the last AS-hops approach the destination IP address and are not repeatedly seen. Since when comparing paths gathered for multiple destination IP addresses, the last hops are usually disjoint, then exploring the ending hops of each trace, one by one, is mandatory. However, the idea that targeting different destination IP addresses may help to continuously reveal new routes closer to where the VP is located seems misplaced, without hesitation. As an example, if the MDA is run towards 1000 IP addresses, the network of the provider AS is measured an equal number of times, which largely seems unnecessary and a waste of probing if the same ASBR-couples of the AS are repeatedly revealed and analyzed.

As a third step, we would like to build a new version of the MDA to reduce the probing cost, namely the Topology Feedback MDA, or TF-MDA in short. By leveraging our understanding that LB applies based on gateways and not on destination IP addresses, we would like to incorporate the exploration, prefix-grouping and multi-route discovery phases of our FD-detector (see Chapter 5) to the MDA. The name topology feedback results from the fact that we aim to split the measuring process that the standard MDA does at once, and blindly, in two. Indeed, recall that the MDA analyzes one /24 prefix at a time, each independently. We propose to analyze multiple prefixes concurrently, and to do it in two steps. The exploration and prefix-grouping phases are used to identify those prefixes for which traces traverse the same ASBR-couples of each AS. The topology knowledge extracted from these steps would be fed back to our algorithm. Taking advantage of the information obtained in the previous step, the additional probing would be wisely crafted, e.g., rather than measuring all prefixes as the MDA does, the TF-MDA could for example just pick one prefix per-ASBR couple. To achieve more robust results, we could envision measuring more prefixes, up to 4, as we do in the FD-detector. Beyond the idea of picking one prefix per ASBR-couple, there is room to think of this as an optimization problem. The objective may be to study all ASBR-couples while minimizing the associated cost or fixing the probing budget while maximizing the number of analyzed ASBR-couples.

Fourth, we plan on refining the stopping points that the MDA uses, and generate a Bayesian-MDA that reduces the required probing cost. The authors in [56] show with their own results that the model they propose loosely bounds the probability of failing to discover all links when running the MDA. As a result, in practice, using the end-to-end path level version of the stopping points results in a waste of probing. As I argued before in Sec. 2.3.3 of Chapter 2, this seems to be produced by an unrealistic tuning of the parameter Q_I, the theoretical maximum number of interfaces expected in a trace. Although the authors did not do it to defend their proposal, I am looking forward to show in emulations or simulations that their model is actually correct despite the inaccurate choice they make for the value of Q_I. The explanation to the results they obtain is that most diamonds found on the Internet usually have a width of 16 at most, but the authors propose $Q_I = 30$. Hence, either emulating a topology with load balancers with multiple next-hops or tuning Q_I differently

would output different discovery success rates when changing the parameter β, the failure probability of discovering all next-hops of all interfaces across the path being analyzed. In any case, besides this subtle issue, the more concerning problem is that 13 years after the MDA was first proposed, the community has not yet been able to generate any alternative version. Indeed, all the subsequent tools, such as Diamond-Miner, MCA, MDA-lite, etc., still rely on the loosely defined stopping points of the MDA. After such a long time of research, nowadays we have multi-path routing surveys with results that can, and should, be used to re-think the mechanics of the MDA. In short, there is an opportunity similar to the one expressed when proposing the TF-MDA: the MDA sins by not taking into consideration any feedback from measurements or a priori knowledge. To produce an overcoming version of the MDA, we propose to use Bayesian inference, and to create a Bayesian-MDA. Rather than assuming that the number of next-hops that a load balancer may use are all equiprobable, we propose to study the empirical distribution obtained in the available surveys. This knowdledge can be plugged into the MDA using Bayes theorem, and be used to guide the probing required, i.e., to generate a new formula to calculate stopping points. These new stopping points will have lower values that the one the MDA proposes, and allow to save probing cost.

Finally, we envision to combine TF-MDA and Bayesian-MDA, generating the Ultimate-MDA, or U-MDA in short. The combination of both approaches, we believe, would result in a significant saving of probing cost associated to multi-path discovery campaigns, and allow to explore faster the topology of the Internet.

Bibliography

[1] République Française. *Smic (Salaire minimum de croissance)*. URL: https://www.service-public.fr/particuliers/vosdroits/F2300.

[2] L'Institut national de la statistique et des études économiques (Insee). *Taux d'inflation – Données annuelles de 1991 à 2019*. URL: https://www.insee.fr/fr/statistiques/2122401.

[3] Yakov Rekhter, Tony Li, Susan Hares, et al. *A border gateway protocol 4 (BGP-4)*. 1994.

[4] Lixin Gao. "On Inferring Autonomous System Relationships in the Internet". In: *IEEE/ACM Trans. Netw.* 9.6 (Dec. 2001), 733–745. ISSN: 1063-6692. DOI: 10.1109/90.974527. URL: https://doi.org/10.1109/90.974527.

[5] Lixin Gao and Jennifer Rexford. "Stable Internet Routing Without Global Coordination". In: *IEEE/ACM Trans. Netw. 2001* ().

[6] Lixin Gao and Jennifer Rexford. "Stable Internet Routing Without Global Coordination". In: *IEEE/ACM Trans. Netw.* 9.6 (Dec. 2001), pp. 681–692. ISSN: 1063-6692. DOI: 10.1109/90.974523. URL: http://dx.doi.org/10.1109/90.974523.

[7] Phillipa Gill, Michael Schapira, and Sharon Goldberg. "A survey of interdomain routing policies". In: *ACM SIGCOMM Computer Communication Review* 44.1 (2013), pp. 28–34.

[8] Vasileios Giotsas, Matthew Luckie, Bradley Huffaker, and kc claffy. "Inferring Complex AS Relationships". In: *Proceedings of the 2014 Conference on Internet Measurement Conference*. IMC '14. Vancouver, BC, Canada: ACM, 2014, pp. 23–30. ISBN: 978-1-4503-3213-2. DOI: 10.1145/2663716.2663743. URL: http://doi.acm.org/10.1145/2663716.2663743.

[9] Ruwaifa Anwar, Haseeb Niaz, David Choffnes, Ítalo Cunha, Phillipa Gill, and Ethan Katz-Bassett. "Investigating Interdomain Routing Policies in the Wild". In: *Proceedings of the 2015 Internet Measurement Conference*. IMC '15. Tokyo, Japan: ACM, 2015, pp. 71–77. ISBN: 978-1-4503-3848-6. DOI: 10.1145/2815675.2815712. URL: http://doi.acm.org/10.1145/2815675.2815712.

[10] Amogh Dhamdhere and Constantine Dovrolis. "Ten years in the evolution of the internet ecosystem". In: *IMC 2008*. 2008, pp. 183–196.

[11] Amogh Dhamdhere and Constantine Dovrolis. "The Internet is flat: modeling the transition from a transit hierarchy to a peering mesh". In: *CoNEXT 2010*. 2010, p. 21.

[12] *AS Rank*. https://asrank.caida.org.

[13] Peyman Faratin. "Economics of overlay networks: An industrial organization perspective on network economics". In: *Proceedings of the NetEcon+ IBC workshop*. 2007.

[14] John Moy. *OSPF Version 2*. RFC 2328. Apr. 1998. DOI: 10.17487/RFC2328. URL: https://rfc-editor.org/rfc/rfc2328.txt.

[15] *Use of OSI IS-IS for routing in TCP/IP and dual environments*. RFC 1195. Dec. 1990. DOI: 10.17487/RFC1195. URL: https://rfc-editor.org/rfc/rfc1195.txt.

[16] Edsger W Dijkstra et al. "A note on two problems in connexion with graphs". In: *Numerische mathematik* 1.1 (1959), pp. 269–271.

[17] V. Jacobsen. *traceroute, Feb. 1989*. URL: ftp://ftp.ee.lbl.gov/traceroute.tar.gz.

[18] *Internet Protocol*. RFC 791. Sept. 1981. DOI: 10.17487/RFC0791. URL: https://rfc-editor.org/rfc/rfc791.txt.

[19] *Internet Control Message Protocol*. RFC 792. Sept. 1981. DOI: 10.17487/RFC0792. URL: https://rfc-editor.org/rfc/rfc792.txt.

[20] Fred Baker. *Requirements for IP Version 4 Routers*. RFC 1812. June 1995. DOI: 10.17487/RFC1812. URL: https://rfc-editor.org/rfc/rfc1812.txt.

[21] Alia Atlas, JR. Rivers, Naiming Shen, Ron Bonica, and Carlos Pignataro. *Extending ICMP for Interface and Next-Hop Identification*. RFC 5837. Apr. 2010. DOI: 10.17487/RFC5837. URL: https://rfc-editor.org/rfc/rfc5837.txt.

[22] Brice Augustin et al. "Avoiding traceroute anomalies with Paris traceroute". In: *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*. 2006, pp. 153–158.

[23] Ubuntu Manpage Repository. *traceroute*. URL: http://manpages.ubuntu.com/manpages/trusty/man1/traceroute.db.1.html.

[24] Riccardo Ravaioli, Guillaume Urvoy-Keller, and Chadi Barakat. "Characterizing ICMP rate limitation on routers". In: *2015 IEEE International Conference on Communications (ICC)*. IEEE. 2015, pp. 6043–6049.

[25] Hang Guo and John Heidemann. "Detecting ICMP rate limiting in the Internet". In: *International Conference on Passive and Active Network Measurement*. Springer. 2018, pp. 3–17.

[26] Joao L Sobrinho. "An algebraic theory of dynamic network routing". In: *IEEE/ACM Transactions on Networking* 13.5 (2005), pp. 1160–1173.

[27] S. Murphy. *BGP Security Vulnerabilities Analysis*. RFC 4272. '06.

[28] Roland Meier, Petar Tsankov, Vincent Lenders, Laurent Vanbever, and Martin Vechev. "NetHide: Secure and Practical Network Topology Obfuscation". In: *USENIX Security 2018*.

[29] Samuel Trassare, Robert Beverly, and David Alderson. "A Technique for Network Topology Deception". In: *Proc of the MILCOM 2013*.

[30] CIDR-REPORT *Status summary*. URL: https://www.cidr-report.org.

[31] Ahmed Elmokashfi and Amogh Dhamdhere. "Revisiting BGP churn growth". In: *ACM SIGCOMM Computer Communication Review* 44.1 (2013), pp. 5–12.

[32] *BGP in 2018 — BGP Churn*. https://blog.apnic.net/2019/01/22/bgp-in-2018-bgp-churn/.

[33] David Hauweele, Bruno Quoitin, Cristel Pelsser, and Randy Bush. "What do parrots and BGP routers have in common?" In: *ACM SIGCOMM Computer Communication Review* 46.3 (2018), p. 2.

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

[35] *Internet Touches Half Million Routes: Outages Possible Next Week*. https://dyn.com/blog/internet-512k-global-routes/.

[36] *768k Day. Will it Happen? Did it Happen?* https://labs.ripe.net/Members/emileaben/768k-day-will-it-happen-did-it-happen.

[37] Florin Coras, Damien Saucez, Loránd Jakab, Albert Cabellos-Aparicio, and Jordi Domingo-Pascual. "Implementing a BGP-free ISP core with LISP". In: 2012.

[38] S. Vissicchio, L. Cittadini, and G. Di Battista. "On iBGP Routing Policies". In: *IEEE/ACM Transactions on Networking* 23.1 (2015), pp. 227–240. ISSN: 1063-6692. DOI: 10.1109/TNET.2013.2296330.

[39] Stefano Vissicchio, Luca Cittadini, Laurent Vanbever, and Olivier Bonaventure. "iBGP deceptions: More sessions, fewer routes". In: Mar. 2012, pp. 2122–2130. DOI: 10.1109/INFCOM.2012.6195595.

[40] Joao Luis Sobrinho, Laurent Vanbever, Franck Le, Andre Sousa, and Jennifer Rexford. "Scaling the Internet Routing System Through Distributed Route Aggregation". In: *IEEE/ACM Trans. Netw.* 24.6 (Dec. 2016), pp. 3462–3476. ISSN: 1063-6692.

[41] Hitesh Ballani, Paul Francis, Tuan Cao, and Jia Wang. "Making Routers Last Longer with ViAggre". In: *ACM NSDI 2009*. Boston, MA, USA, 2009.

[42] Slimming down the Internet routing table *by Tore Anderson*. URL: https://www.redpill-linpro.com/sysadvent/2016/12/09/slimming-routing-table.html.

[43] 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)*. 2019, pp. 17–24. DOI: 10.23919/TMA.2019.8784618.

[44] Yves Vanaubel, Pascal Mérindol, Jean-Jacques Pansiot, and Benoit Donnet. "MPLS Under the Microscope". In: *the 2015 ACM Conference*. New York, New York, USA: ACM Press, 2015, pp. 49–62.

[45] Bob Thomas, Loa Andersson, and Ina Minei. *LDP Specification*. RFC 5036. Oct. 2007. DOI: 10.17487/RFC5036. URL: https://rfc-editor.org/rfc/rfc5036.

[46] Ahmed Bashandy, Clarence Filsfils, Stefano Previdi, Bruno Decraene, Stephane Litkowski, and Rob Shakir. *Segment Routing with the MPLS Data Plane*. RFC 8660. Dec. 2019. DOI: 10.17487/RFC8660. URL: https://rfc-editor.org/rfc/rfc8660.txt.

[47] *MPLS: Layer 3 VPNs Configuration Guide, Cisco IOS XE Fuji 16.7.x (Cisco ASR 920 Series)*. https://www.cisco.com/c/en/us/td/docs/routers/asr920/configuration/guide/mpls/16-7-1/b-mp-l3-vpns-xe-16-7-1-asr920/ecmp-load-balancing.html.

[48] *BGP Best Path Selection Algorithm*. https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/13753-25.html.

[49] Sandeep Kumar Singh, Tamal Das, and Admela Jukan. "A survey on internet multipath routing and provisioning". In: *IEEE Communications Surveys & Tutorials* 17.4 (2015), pp. 2157–2175.

[50] Fabien Viger, Brice Augustin, Xavier Cuvellier, Clémence Magnien, Matthieu Latapy, Timur Friedman, and Renata Teixeira. "Detection, understanding, and prevention of traceroute measurement artifacts". In: *Computer Networks* 52.5 (2008), pp. 998 –1018.

[51] *How Does Load Balancing Work?* https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html.

[52] *Per-Flow and Per-Packet Load Balancing*. https://support.huawei.com/enterprise/en/doc/EDOC1100055041/ebc8ad42/per-flow-and-per-packet-load-balancing.

[53] Ka-Cheong Leung, Victor OK Li, and Daiqin Yang. "An overview of packet reordering in transmission control protocol (TCP): problems, solutions, and challenges". In: *IEEE transactions on parallel and distributed systems* 18.4 (2007), pp. 522–535.

[54] Sumet Prabhavat, Hiroki Nishiyama, Nirwan Ansari, and Nei Kato. "On load distribution over multipath networks". In: *IEEE Communications Surveys & Tutorials* 14.3 (2011), pp. 662–680.

[55] John Bellardo and Stefan Savage. "Measuring packet reordering". In: *Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. 2002, pp. 97–105.

[56] D. Veitch, B. Augustin, R. Teixeira, and T. Friedman. "Failure Control in Multipath Route Tracing". In: *IEEE INFOCOM 2009*. 2009, pp. 1395–1403.

[57] Kevin Vermeulen, Justin P Rohrer, Robert Beverly, Olivier Fourmaux, and Timur 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.

[58] Christophe Diot, Darryl Veitch, Italo Cunha, Rafael Almeida, and Renata Cruz Teixeira. "Classification of load balancing in the Internet". In: *Proceedings of IEEE INFOCOM*. Beijing, China, 2020.

[59] *How Does Load Balancing Work?* `https://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html`.

[60] *Configuration Guide for Cisco ASR 9000 Series Routers*. `https://www.cisco.com/c/en/us/td/docs/routers/asr9000/software/asr9k-r6-5/interfaces/configuration/guide/b-interfaces-hardware-component-cg-asr9000-65x/b-interfaces-hardware-component-cg-asr9000-65x_chapter_01000.html`.

[61] *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`.

[62] *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`.

[63] *CEF Polarization*. `https://www.cisco.com/c/en/us/support/docs/ip/express-forwarding-cef/116376-technote-cef-00.html`.

[64] *ECMP load balancing with masquerade*. `https://wiki.mikrotik.com/wiki/ECMP_load_balancing_with_masquerade`.

[65] Brice Augustin, Timur Friedman, and Renata Teixeira. "Multipath tracing with Paris traceroute". In: *2007 Workshop on End-to-End Monitoring Techniques and Services*. IEEE. 2007, pp. 1–8.

[66] Brice Augustin, Timur Friedman, and Renata Teixeira. "Measuring load-balanced paths in the Internet". In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 2007, pp. 149–160.

[67] Brice Augustin, Timur Friedman, and Renata Teixeira. "Measuring multipath routing in the Internet". In: *IEEE/ACM Transactions on Networking* 19.3 (2010), pp. 830–840.

[68] Kevin Vermeulen, Stephen D Strowes, Olivier Fourmaux, and Timur Friedman. "Multilevel MDA-Lite Paris Traceroute". In: *Proceedings of the Internet Measurement Conference 2018*. 2018, pp. 29–42.

[69] Robert Beverly. "Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery". In: *Proceedings of the 2016 Internet Measurement Conference*. IMC '16. Santa Monica, California, USA: Association for Computing Machinery, 2016, 413–420. ISBN: 9781450345262. DOI: `10.1145/2987443.2987479`. URL: `https://doi.org/10.1145/2987443.2987479`.

[70] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H. Katz. "Towards an Accurate AS-level Traceroute Tool". In: *SIGCOMM 2003*.

[71] Z. M. Mao, D. Johnson, J. Rexford, Jia Wang, and R. Katz. "Scalable and accurate identification of AS-level forwarding paths". In: *IEEE INFOCOM 2004*.

[72] Y. Hyun, A. Broido, and k. claffy. *Traceroute and BGP AS Path Incongruities*. Tech. rep. CAIDA, 2003.

[73] Yu Zhang et al. "A Framework to Quantify the Pitfalls of Using Traceroute in AS-Level Topology Measurement". In: *IEEE JSAC 2011* ().

[74] Y. Hyun, A. Broido, and k. claffy. "On Third-party Addresses in Traceroute Paths". In: *PAM 2013*.

[75] Pietro Marchetta, Walter de Donato, and Antonio Pescapè. "Detecting Third-Party Addresses in Traceroute Traces with IP Timestamp Option". In: *PAM 2013*.

[76] Matthew Luckie and kc claffy. "A Second Look at Detecting Third-Party Addresses in Traceroute Traces with the IP Timestamp Option". In: *PAM 2014*.

[77] N. Ahmed and K. Sarac. "An experimental study on inter-domain routing dynamics using IP-level path traces". In: *IEEE 40th Conference on Local Computer Networks (LCN) 2015*.

[78] M. Luckie, A. Dhamdhere, B. Huffaker, D. Clark, and k. claffy. "bdrmap: Inference of Borders Between IP Networks". In: *IMC 2016*.

[79] Alexander Marder and Jonathan M. Smith. "MAP-IT: Multipass Accurate Passive Inferences from Traceroute". In: *IMC 2016*.

[80] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, kc claffy, and Jonathan M. Smith. "Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale". In: *IMC 2018*.

[81] Tian Bu, Lixin Gao, and Donald F Towsley. "On characterizing BGP routing table growth." In: *Computer Networks ()* (2004).

[82] 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.

[83] Stefano Secci, Jean-Louis Rougier, Achille Pattavina, Mauro Marinoni, Guido Maier, and Estrellita M T Elena. "Detection of BGP route deflections across top-tier interconnections". In: 2009.

[84] Sharad Agarwal, Chen-Nee Chuah, Supratik Bhattacharyya, and Christophe Diot. "The impact of BGP dynamics on intra-domain traffic." In: *SIGMETRICS* (2004).

[85] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. "Internet Optometry: Assessing the Broken Glasses in Internet Reachability". In: *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement*. IMC '09. Chicago, Illinois, USA: ACM, 2009.

[86] X. Zhang and C. Phillips. "A Survey on Selective Routing Topology Inference Through Active Probing". In: *IEEE Commun. Surveys Tuts. 2012* ().

[87] Pietro Marchetta and Antonio Pescapè. "DRAGO: Detecting, quantifying and locating hidden routers in Traceroute IP paths". In: *INFOCOM Workshops 2013*.

[88] George Nomikos and Xenofontas Dimitropoulos. "traIXroute: Detecting IXPs in traceroute paths". In: *PAM 2016*.

[89] Brandon Schlinker, Kyriakos Zarifis, Italo Cunha, Nick Feamster, and Ethan Katz-Bassett. "PEERING: An AS for Us". In: *HotNets-XIII 2014*.

[90] PEERING *testbed sessions*. `https://peering.usc.edu/peers/`.

[91] Matthew Luckie. "Scamper: A Scalable and Extensible Packet Prober for Active Measurement of the Internet". In: *IMC 2010*.

[92] *RIRs prefixes*. `https://labs.apnic.net/delegated-nro-extended`, Apr. '18.

[93] *Configuring Per-Prefix Load Balancing*. `https://www.juniper.net/documentation/en_US/junos/topics/usage-guidelines/policy-configuring-per-prefix-load-balancing.html`.

[94] Kevin Vermeulen, Justin P. Rohrer, Robert Beverly, Olivier Fourmaux, and Timur Friedman. "Diamond-Miner: Comprehensive Discovery of the Internet's Topology Diamonds". In: *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. Santa Clara, CA: USENIX Association, Feb. 2020, pp. 479–493. ISBN: 978-1-939133-13-7. URL: `https://www.usenix.org/conference/nsdi20/presentation/vermeulen`.

[95] NLNOG RING *monitoring infrastructure*. `https://ring.nlnog.net`.

[96] *Scamper*. `https://www.caida.org/tools/measurement/scamper/`. [Online; accessed October 2018].

[97] *Paris Traceroute*. `https://paris-traceroute.net/`. [Online; accessed October 2018].

[98] Xun Fan and John 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.

[99] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, Jonathan 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.

[100] Brice Augustin, Timur Friedman, and Renata Teixeira. "Measuring load-balanced paths in the Internet". In: *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. ACM. 2007, pp. 149–160.

Glossary

AS Autonomous system

ASBR Autonomous system border router

ASN Autonomous system number

BGP Border gateway protocol

CDN Content distribution network

CF-LB Coarse-fine grained load balancing

C-LB Coarse grained load balancing

CP Control path

DIR Direct internal route

DP Data path

eBGP External border gateway protocol

ECMP Equal-cost multi-path

FA Forwarding alteration

FC-LB Fine-coarse grained LB

FD Forwarding detour

FIB Forwarding information base

F-LB Fine-grained LB

Flow-ID Flow identifier

iBGP Internal border gateway protocol

ICMP Internet control message protocol

IGP Internal gateway protocol

IP Internet protocol

ISP Internal service provider

IXP Internet exchange point

LatAm Latin America

LB Load balancing

MDA Multi-path detection algorithm

OAS Originating autonomous system

Org Organization

UDP User datagram protocol

RIB Routing information base

RTT Round-trip time

TCP Transmission control protocol

TE Traffic engineering

TIR Transit internal route

TPA Third-party address

VP Vantage point

List of Figures

List of Tables

Extended Summary in French

Résumé

L'internet est une interconnexion de réseaux indépendants, appelés systèmes autonomes (AS). Étant donné que les AS sont construits sur du matériel et des logiciels, et que les opérateurs réseau, c'est-à-dire des administrateurs humains, gèrent les AS, l'internet est limité et génère certaines défaillances. Par exemple, les humains sont sujets à des erreurs et peuvent prendre des décisions arbitraires, les entreprises sont généralement avides de revenus et le matériel peut tomber en panne et nécessiter une maintenance ou un remplacement. Tous ces facteurs peuvent conduire à des défaillances de l'Internet, c'est-à-dire à des composants défectueux, à des réseaux confrontés à des limitations ainsi qu'à des réseaux égoïstes qui donnent la priorité à leurs propres revenus plutôt qu'aux meilleures performances de l'Internet.

L'objectif de cette thèse est de détecter des éléments défaillants de l'Internet. Tout d'abord, nous étudions le déploiement des points d'échange Internet (IXP) en Amérique latine, une région qui n'avait jusqu'à présent reçu que peu d'attention dans les études sur Internet. Nous construisons l'ensemble de données le plus complet sur l'état de l'Internet en Amérique latine et caractérisons l'écosystème des IXP dans la région. Nous constatons que si certains IXP en Amérique latine ont réussi à proliférer, certains pays sont en **échec IXP**, c'est-à-dire aucun IXP du tout, ou bien l'IXP n'a pas réussi à attirer des membres. Deuxièmement, nous étudions si les AS génèrent, intentionnellement ou non, des **mensonges BGP**, c'est-à-dire si les routes d'acheminement par lesquelles les paquets circulent réellement sur Internet divergent des chemins que les AS annoncent sur BGP, le protocole de routage utilisé sur Internet. En pratique, cette comparaison est complexe car, outre les multiples niveaux auxquels les données doivent être synchronisées, les sauts manquants, les adresses tierces et les AS jumeaux peuvent introduire des erreurs en déclenchant à tort la détection des mensonges BGP. Nous développons une méthodologie permettant de filtrer ce bruit, et d'effectuer des mesures de terrain. Nous trouvons des cas où, après avoir assaini l'ensemble des données avec notre cadre modulaire de filtrage, les chemins ne correspondent toujours pas. Enfin, nous étudions comment le trafic circule à l'intérieur des AS et nous nous concentrons sur la détection des **détours d'acheminement**, c'est-à-dire les cas où les itinéraires d'acheminement ne correspondent pas aux meilleurs itinéraires disponibles, selon le protocole de routage utilisé. Nous développons un formalisme expliquant quand les détours de réachem-

inement se produisent, et mettons en œuvre un détecteur permettant de différencier les détours de réacheminement des techniques d'équilibrage de charge et d'ingénierie du trafic. Nous effectuons des mesures avec notre détecteur et trouvons des détours dans plusieurs AS avec un motif binaire remarquable, de sorte que le trafic de transit passant entre deux routeurs de bordure d'un AS ne fait jamais de détour, ou en fait systématiquement.

Introduction

L'Internet peut apparaître comme un système infaillible qui ne tombe jamais en panne, mais ce n'est pas le cas. Internet est simplement une interconnexion de réseaux indépendants, appelés systèmes autonomes (AS). Les systèmes construits par l'homme ne sont généralement pas parfaits : ils ont tendance à comporter des modules qui peuvent tomber en panne et nécessiter une maintenance, ou qui, après un certain temps, peuvent devenir obsolètes et nécessiter un remplacement. Tous ces facteurs peuvent conduire à des défaillances de l'internet, c'est-à-dire à des composants défectueux, à des réseaux confrontés à des limitations et même à des réseaux égoïstes qui privilégient leurs propres revenus plutôt que les meilleures performances de l'internet. L'objectif de cette thèse est de détecter de tels problèmes. Cette tâche est difficile car les phénomènes que nous voulons découvrir peuvent être cachés n'importe où sur l'Internet, qui compte environ 70K ASes en novembre 2020.

Points d'échange Internet en Amérique latine La première composante de l'Internet que nous étudions dans le cadre de cette thèse porte sur les points d'échange Internet (IXP), les installations d'interconnexion communément utilisées par les AS. La structure de l'Internet, c'est-à-dire la manière dont les AS établissent des connexions entre eux, a été largement modifiée par l'irruption des IXP dans les années 2000 [11]. Les IXP permettent aux AS d'établir des connexions à une plus grande échelle et de réaliser des économies. Toutefois, la popularité des IXP varie selon les régions. Dans certains cas, des pays peuvent avoir des IXP **en échec**, c'est-à-dire aucun IXP du tout, ou un IXP qui n'a pas réussi à attirer suffisamment de membres. Alors qu'il existe de grands IXP en Europe, tels que DE-CIX, LINX et AMS-IX qui ont fait l'objet d'une étude [**ager2012anatomy**], il n'existe aucun rapport faisant état d'une réussite similaire en Amérique du Sud. Des études récentes se sont concentrées sur le rôle des IXP dans l'écosystème africain des SA [**fanou2017investigating**, **fanou2015diversity**, **fanou2017reshaping**]. Dans d'autres régions, en revanche, on sait peu de choses sur les IXP et même sur l'internet dans son ensemble.

En particulier, le cas de l'Amérique latine (LatAm), correspondant au *Registre Internet régional* (RIR) nommé LACNIC, est un cas d'étude intéressant. L'Amérique latine couvre 20 millions de km^2 [**worldbank1**] et comprend 20 pays : juste après l'Amérique du Nord, elle a le taux de population urbaine le plus élevé (80%) [**worldbank2**]. En outre, l'Amérique latine (LatAm) compte 652 millions d'habitants [**un1**] et possède trois des quatre plus grandes zones métropolitaines des Amériques (Sao Paulo, Mexico et Buenos Aires avec des populations respectives de 21,3M, 21,2M et 15,3M d'habitants) [**un2**]. LatAm a également des chiffres intéressants en ce qui concerne l'internet, étant donné qu'il contribue à l'écosystème mondial des sociétés anonymes

avec 14,5 En outre, 6458 ASN ont été délégués par NIC.br (Brazilian NIR) à des organisations basées au Brésil. Entre 2005 et 2015, cette région a connu une progression significative des taux de pénétration de la téléphonie fixe et mobile, atteignant respectivement 40,57% et 57,41% de la population [**katz2018accelerating**]. En outre, plusieurs pays de la région ont récemment bénéficié de la création d'IXP nationaux [**galperin2016localizing**].

Malgré le développement progressif de l'Internet en Amérique latine, la forme du réseau latino-américain reste relativement inexplorée. Cela nous encourage à explorer son interconnexion et sa structure. Étant donné que les IXP ont contribué à aplatir l'Internet dans les années 2000, il est naturel de se demander si, 20 ans plus tard, ces infrastructures de peering profitent également aux pays en développement, dont beaucoup ont une surface beaucoup plus grande. Cela nous amène à notre première question de recherche.

Défi Scientifique

Tous les IXP de LatAm ont-ils réussi à proliférer ou y a-t-il des IXP qui ont échoué ? Si certains ont échoué, pourquoi ?

Les réseaux LatAm sont peu explorés, probablement en raison d'une rareté historique de données Internet représentatives. Par exemple, l'empreinte du RIPE Atlas et de l'Ark CAIDA dans le LatAm est composée de 285 sondes sur 11 142 (2,56 %) et de 12 sondes sur 190 (6,32 %), respectivement. Ces chiffres diminuent si l'on considère l'IPv6 : seulement 2,22 % (101/4 556) des sondes compatibles IPv6 du RIPE sont situées en Amérique latine. D'autre part, on sait que le manque de données BGP permet de dessiner une représentation assez incomplète des écosystèmes AS [**lakhina2003sampling**]. En ce sens, Routeviews [2] et RIPE RIS [3] n'ont déployé respectivement que deux et un collecteurs de données BGP dans la région, deux étant redondants puisqu'ils sont placés au même IXP brésilien à Sao Paulo.

Malgré les limites susmentionnées, de rares études Internet se sont concentrées sur cette région. Berenguer *et al.* [**berenguer2016hidden**] a appliqué des métriques de la théorie des graphes pour évaluer l'augmentation de l'ensemble des données lorsque les itinéraires collectés à partir de lunettes locales sont ajoutés aux décharges RIPE RIS et RouteViews BGP. Brito *et al.* [**brito2016dissecting**] a recueilli des données BGP par looking glass co-localisé dans chaque IXP régional de IX.br, le réseau IXP brésilien, et les a ensuite comparés avec les IXP d'autres régions en termes de réseaux connectés et de prévalence des politiques de peering. Une étude complémentaire des mêmes auteurs comprenait une analyse du déploiement de l'IPv6 [**brito2016analysis**], toutefois, elle est limitée à la taille du préfixe IPv6 et au nombre d'entrées IPv6 dans les tables de routage. Muller *et al.* [**muller2019challenges**] s'est appuyé sur les données sFlow recueillies à un IXP régional de IX.br pour déduire le trafic spoofé traversant l'IXP. Formoso *et*

[2] `http://www.routeviews.org/`

[3] `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`

al. [**formoso2016looking**] s'est appuyé sur les sondes de l'Atlas RIPE déployées en LatAm pour mesurer la latence entre les pays, en déduisant les trajets asymétriques et les pays mal interconnectés.

En particulier, le Chapitre 3 étudie le déploiement des IXP dans le LatAm, et exhibe le premier morceau d'Internet défaillant que nous analysons. En effet, les IXP sont une histoire de succès ou d'échec selon le pays considéré en Amérique latine. Alors que l'Argentine, le Brésil et le Chili comptent des IXP qui ont réussi à proliférer, d'autres pays ont échoué. Nous nous plongeons dans les raisons pour lesquelles certains IXP sont capables de rassembler un grand nombre de membres qui annoncent plusieurs adresses IP alors que d'autres non. Nous constatons une corrélation négative entre le succès des IXP et la présence d'AS monopolistiques concentrant l'espace d'adresses IPv4 déléguées aux pays. En outre, comme cette région n'a jamais été analysée de près, nous saisissons l'occasion et caractérisons également l'écosystème des AS dans la région. Nous constatons que les points d'échange Internet dans la région LatAm, comme dans d'autres régions en développement, sont principalement peuplés d'acteurs nationaux ou régionaux, alors que ceux qui sont de renommée internationale en Europe se comportent plutôt comme des points d'échange internationaux.

À la recherche de mensonges BGP Le deuxième élément de l'Internet que nous examinons dans cette thèse est le protocole Border Gateway Protocol (BGP), utilisé sur l'Internet pour le routage entre AS. BGP dicte la manière dont les AS échangent des informations d'accessibilité concernant les préfixes IP que chacun d'entre-eux possède. En bref, chaque AS annonce à ses AS voisins les préfixes qu'il possède en BGP, et ceux-ci relaient à leur tour le message aux autres AS. Dans ce processus, les messages de routage qui sont envoyés conservent la trace du chemin suivi par l'AS, c'est-à-dire une liste des AS, du premier au dernier, qui ont annoncé le préfixe. Dans cette thèse, nous appelons le chemin AS le chemin de contrôle (CP), puisqu'il est construit sur le plan de contrôle de BGP. D'autre part, nous faisons référence à l'ensemble des AS que les paquets parcourent effectivement vers leur destination comme des chemins de données (DP), puisque cela se produit au niveau du plan de données de BGP. Une analogie d'un AS annonçant un préfixe avec un CP donné à un AS voisin, est la signature d'un contrat. Plus précisément, l'annonce de BGP joue le rôle du contrat, et le service qui est offert est que, pour atteindre un préfixe donné, le DP répliquera l'ensemble ordonné d'AS exprimé dans le CP. Par conséquent, les DP sont censés correspondre aux CP annoncés pour tous les préfixes.

L'hypothèse sous-jacente selon laquelle les CP correspondent aux DP pour tous les préfixes annoncés dans BGP n'est pas triviale à vérifier : les outils de dépannage actuels, par exemple traceroute, permettent généralement de récupérer les chemins IP, mais pas directement la route de transfert au niveau des AS qui a été suivie. La confiance implicite qui présume que les AS annoncent les chemins qu'ils utilisent pour le transfert de paquets peut être naive. Les opérateurs de réseau peuvent manipuler les CP [27] et les DP [28, 29], ce qui peut les conduire à une non-concordance. Chaque fois que le CP et le DP d'un préfixe ne correspondent pas, on dira qu'un mensonge BGP s'est produit. Notez que ce terme s'applique indépendamment du fait que les CP et/ou les DP soient modifiés, ou si cela résulte d'un comportement

délibéré ou involontaire.

L'objectif qui se cache derrière les mensonges de BGP peut être multiple. Un AS peut tenter de rediriger et d'intercepter le trafic, ou entraver son suivi avec des conséquences sur la capacité à résoudre les problèmes de connectivité. De plus, les mensonges de BGP peuvent conduire à la violation d'accords entre des AS adjacents, avec des représailles juridiques potentielles. Ces mensonges peuvent être délibérés pour brouiller les interceptions de trafic ou être motivés par des intérêts économiques, par exemple attirer du trafic en promettant des itinéraires intéressants mais en utilisant des alternatives moins coûteuses. D'autre part, ils peuvent résulter de topologies logiques et physiques incongrues, en particulier lorsque les sessions BGP ne sont pas établies sur de simples liaisons inter-domaines point à point. D'autres peuvent être dues à des limitations techniques, telles qu'une mémoire limitée sur les routeurs empêchant le stockage de la table de routage complète, c'est-à-dire résultant d'une FIB partiel. En conclusion, la facilité avec laquelle les mensonges BGP peuvent se produire constitue la seconde pièce cassée d'Internet que nous allons étudier et quantifier.

Cette tâche de détection des mensonges BGP nécessite de relever un défi pratique considérable : les données recueillies à partir des mesures et des outils nécessaires à cette analyse sont bruitées, de sorte que les mensonges BGP peuvent être mal interprétés (comme du bruit, ou vice versa). Par conséquent, pour tirer des conclusions représentatives, le cadre élaboré doit filtrer le bruit affectant les mesures. C'est ce qui motive notre deuxième question de recherche.

Défi Scientifique

Pouvons-nous développer un cadre qui filtre le bruit affectant la comparaison des CP et des DP et qui permette de quantifier le taux quotidien de mensonges BGP qui se produisent sur Internet ?

Il existe de nombreux documents qui se sont concentrés sur la comparaison des CP et des DP, et sur la caractérisation des différentes sources de bruit affectant cette tâche. Mao *et al.* [70] constate que les IXP, les AS jumeaux et ASes annonçant des préfixes IP pour lesquels ils ne sont pas les véritables origines sont les causes prédominantes des décalages entre CP et DP. Dans un travail de suivi [71], ils développent une approche systématique pour corriger les correspondances IP-AS inexactes en réaffectant l'origine des préfixes. Hyun et al. [72] analysent également les divergences entre les CP et les DP et concluent que les insertions d'IXP dans les DP et d'AS sous la même propriété sont la principale cause qui conduit à des discordances. Toutefois, dans leur étude, les traces incomplètes sont écartées et la comparaison ne repose pas sur les dernières mises à jour du BGP, c'est-à-dire que les CP et les DP ne sont pas synchronisés. Zhang et al. [73] extraient les fragments de discordance des CP et des DP et montrent que le principal écueil de l'utilisation de traceroute dans les mesures de topologie au niveau des AS provient de l'apparition

d'adresses IP attribuées par les voisins des AS. Cependant, leur plateforme de mesure souffre de l'incapacité à garantir que les VP du plan de données et de contrôle sont co-localisés. D'autre part, Hyun et al. [74] ont introduit le concept d'adresses IP tierces, ou TPAs. Selon les auteurs, il est peu probable, bien que possible, de trouver plusieurs TPA d'affilée cartographiés sur le même AS. Leur étude conclut que les TPA ne sont pas courants et qu'ils ne déforment pas les cartes des AD de manière significative. Une analyse ultérieure de Marchetta et al. [75] utilisant les options d'horodatage IP affirme le contraire. Ils constatent que les TPA consécutifs sont courants, et peuvent même cacher entièrement un AS d'un chemin de niveau AS. Cependant, une étude ultérieure de Luckie et al. [76] rapporte que la plupart des IPs observés dans les traces `traceroute` proviennent d'interfaces en liaison, donc sur le chemin. Ils affirment que les techniques utilisant les horodatages IP ne sont pas fiables pour détecter les TPA. Ahmed et. al [77] proposent une méthode hors ligne qui marque jusqu'à deux IP qui apparaissent dans une rangée comme des TPA possibles s'ils introduisent un AS qui soit viole des chemins sans vallée, soit se traduit par de nouvelles relations AS. D'autres travaux concernant la détection des TPA pour déterminer correctement les limites des AS comprennent bdrmap [78], MAP-IT [79] et bdrmapIT [80]. Bdrmap déduit les adresses d'interface de liaison inter-AS entre un réseau avec un VP traceroute et les réseaux directement connectés qui s'appuient sur des techniques de sondage de résolution d'alias, et les déductions de relations AS. D'autre part, MAP-IT tente d'obtenir la même chose pour tous les AS connectés en se basant sur les résultats du traceroute collectés à partir de plusieurs VP distribués dans différents AS. Enfin, bdrmapIT combine les deux précédentes, améliorant l'inférence de la propriété des routeurs et l'identification des liens entre les AS.

Dans le Chapitre 4 nous modélisons comment les AS peuvent introduire des mensonges BGP et nous proposons une méthodologie pour les détecter. En particulier, nous présentons un cadre permettant de filtrer le bruit, c'est-à-dire les inexactitudes de cartographie introduites par les AS jumeaux, les TPA et les sauts manquants, qui peuvent affecter la comparaison entre les chemins de contrôle annoncés dans BGP et les chemins de données que suivent les paquets sur l'internet. En particulier, notre méthodologie repose sur une heuristique estimant si le bruit peut affecter les chemins que nous collectons, puis tente de les corriger. Notre cadre est modulaire, c'est-à-dire que l'utilisateur peut choisir parmi plusieurs filtres qui varient la manière dont il estime ce qui résulte du bruit ou non, et permettent ainsi de mettre en œuvre différents modèles de filtrage du bruit. Nous effectuons des mesures à long terme sur l'internet et assainissons l'ensemble des données grâce à notre outil. Nos résultats montrent que, même en se basant sur le modèle de filtrage du bruit le plus conservateur, il subsiste encore quelques décalages entre les itinéraires de transmission et de contrôle que nous recueillons sur l'internet, ce qui représente probablement des mensonges BGP. Dans les points de collecte où l'on trouve peu de mensonges, les résultats sont stables dans le temps, sinon nous constatons que le nombre de chemins divergents par jour présente une plus grande variation. Par rapport à la littérature, notre étude déploie non seulement plus de points de mesure, mais va également au-delà de ce qui avait été fait auparavant en fournissant des résultats basés sur une analyse quotidienne. Alors que la plupart des travaux connexes blâment essentiellement la cartographie IP-AS pour les écarts observés entre les CP et

les DP, notre travail repose sur des heuristiques conservatrices qui éliminent le bruit dans les mesures et les erreurs de cartographie, en essayant de minimiser les faux positifs dans la détection des mensonges BGP. En d'autres termes, à la différence des études précédentes, notre objectif est de détecter les "vrais" mensonges BGP, d'où notre cadre de travail conçu pour fournir une borne inférieure. Les erreurs que nous constatons après avoir appliqué nos filtres montrent que la cartographie IP-to-AS n'est pas la seule coupable.

Modélisation et détection des détours d'acheminement La dernière composante de l'Internet que nous analysons sont les protocoles de routage interne (IGP), les protocoles de routage que les AS utilisent à l'intérieur de leurs réseaux. Les IGP se caractérisent par le fait qu'ils aboutissent à un schéma d'acheminement tel que le trafic traversant les AS passe par les meilleurs chemins qui minimisent la distance ou le coût selon une métrique laissée au choix de l'opérateur réseau de chaque AS.

Le flux Internet complet, qui atteignait 854 000 de préfixes en novembre 2020, a augmenté d'environ 50 000 préfixes par an au cours des dix dernières années. L'augmentation soutenue du nombre de préfixes annoncés sur le BGP a conduit les ASes à échanger davantage de messages de mise à jour [31, 32, 33], et à souffrir de problèmes d'évolutivité. En effet, compte tenu de la tendance actuelle, le maintien d'un FIB complet peut s'avérer difficile, en particulier pour les AS incapables de mettre à jour régulièrement leurs équipements réseau [**zhao2010routing**, 34, 35, 36].

Dans ce contexte, les opérateurs de réseaux se repose sur des alternatives peu onéreuses pour supporter d'anciens routeurs incapables de maintenir une FIB complet en mémoire. Par exemple, dans un coeur d'infrastructure sans BGP, les techniques de tunneling réduisent la taille de la FIB sur ces routeurs [37]. En outre, la diffusion partielle d'iBGP reposant sur des hiérarchies de réflecteurs de route peut également accroître l'extensibilité [38]. Cette technique permet aux routeurs de conserver moins de pairs BGP et, dans certains cas rares, peut même empêcher la redistribution complète des préfixes BGP dans l'AS [39]. En outre, les routeurs à mémoire limitée peuvent regrouper les routes pour limiter le nombre d'entrées FIB [40]. D'autres types de contournement consistent à stocker une FIB partielle [41], et à rediriger le trafic via des routes par défaut vers des routeurs plus performants (par exemple, ayant une FIB complète). Certains opérateurs réseau appliquent même cette technique sur les commutateurs dotés de capacités IP [42].

Si les techniques susmentionnées peuvent paraître efficaces à première vue, les AS qui s'y fient peuvent souffrir de *détournements d'acheminement*, c'est-à-dire que le trafic peut passer par des itinéraires d'acheminement qui dévient ou s'écartent des meilleures chemins IGP attendus. Cela peut se produire lorsque les routeurs le long d'une route choisissent différentes passerelles de sortie, ou ASBR, pour le même préfixe. Pour cette raison, nous disons que ces préfixes *subissent des FD*. En général, l'existence simultanée de préfixes soumis ou non aux FD génère des modèles *multi-path routing*. Cependant, contrairement au routage "hot-potato", les FDs augmentent la distance IGP nécessaire pour traverser un AS et entraînent sans doute un gaspillage de ressources à l'intérieur du réseau. Pour tenter de supprimer les FD, les opérateurs de réseau peuvent mettre en œuvre des techniques de tunneling.

Cependant, ces mécanismes ne permettent de contourner les FD qu'à l'intérieur de chaque tunnel/segment (pour les routeurs de coeur non ASBR BGP en particulier) mais peuvent ne pas le faire entre les points d'extrémité d'un AS. De plus, outre les effets secondaires des solutions de contournement de l'extensibilité, des bogues dans le logiciel du routeur, tels que les zombies BGP [**fontugne2019bgp**], peuvent également créer des FD. Par conséquent, les opérateurs de réseau peuvent ignorer les FD qui se produisent sur leur AS, et fournir des performances dégradées aux AS des clients. Cela nous amène à une troisième question de recherche.

Défi Scientifique

Pouvons-nous modéliser formellement les causes profondes qui produisent les détours d'acheminement, et concevoir une méthodologie efficace pour les détecter ?

Concernant les travaux connexes, en 2004, lorsque les FIB completes ne comptaient que 100 000 entrées, contre plus de 800 000 aujourd'hui, Bu et al. [81] ont étudié l'augmentation des tables BGP causée par ce qu'ils ont appelé une croissance exponentielle de l'internet. Alors que leur étude s'est concentrée sur les raisons de cette augmentation, nous nous intéressons aux conséquences; plus précisément, à leur impact sur l'acheminement des données à l'intérieur des AS. Plusieurs propositions visent à réduire la taille des tables de routage en agrégeant les routes [40] et en redirigeant parfois le trafic vers des routeurs plus performants [41]. La croissance des FIB favorise en effet l'utilisation de solutions de contournement comme les FIB partiels et les routes par défaut, qui peuvent à leur tour conduire à des FD.

D'autre part, les déviations de route sont un phénomène connu qui a été étudié sous différents angles, cependant, aucun n'est exécuté à la même échelle, ni avec le même objectif que le nôtre. Elena et al. [82] mettent en évidence les déviations à l'échelle d'un AS, bien que leur objectif soit de détecter la diversité des chemins sur Internet. Ils concluent que l'équilibrage de charge (LB ou multi-path routing) intra-domaine n'était pas bien déployée à l'époque. Secci et al. [83] étudient les déviations de bout en bout créées par BGP. S'ils étudient également les déviations intra-domaine, ils se concentrent sur la dynamique et les oscillations dues à l'attribut MED. Agarwal et al. [84] analysent les changements de routage de BGP en tant que déviations. Ils essaient de détecter les déviations intra-domaine pour construire des matrices de trafic précises. Bush et al. [85] étudient l'utilisation de routes par défaut, ou filet de sécurité, assurant l'accessibilité lors des événements de routage. Pour cela, ils empoisonnent les routes et vérifient ensuite si les préfixes associés sont toujours accessibles.

Enfin, de nombreuses études ont été menées sur les schémas de routage à trajets multiples (LB). Augustin et al. [22] présentent Paris-traceroute, une version de traceroute prenant en compte l'équilibrage des charges par flux, permettant d'éviter l'inférence erronée de liens, de boucles et de cycles vus dans le traceroute standard, comme l'a étudié plus en détail Viger et al. [50]. Sur la base des principes

de Paris-traceroute, Augustin et al. [65] développent l'algorithme MDA, permettant de détecter les équilibreurs de charge par flux et par paquet. Dans des études ultérieures, ils étendent MDA pour détecter également les équilibres de charge par destination [66, 67]. Veitch et al. [56] affinent la génération de la liste des points d'arrêt de MDA pour limiter la probabilité d'échec de la découverte complète par plusieurs chemins. Vermeulen et al. [68] proposent MDA-Lite, une version allégée de la MDA qui nécessite moins de sondes, mais qui peut échouer à découvrir tous les nœuds et liens. Plus récemment, ils proposent Diamond Miner [57], un système capable de produire des cartes topologiques multi-trajets sur Internet en moins de 3 jours de mesure [57]. DiamondMiner met en œuvre MDA avec un mode de sondage sans état s'appuyant sur Yarrp [**yarrp-imc16**], un outil de sondage à grande vitesse randomisé. Almeida et al. [58] généralisent MDA et proposent l'Algorithme de classification multi-chemins (MCA). En général, tous ces travaux montrent que les LB par flux et par destination sont les options LB les plus répandues. À l'exception de Diamond Miner, ils mènent tous des campagnes de mesure de l'ordre de 10K et pas plus de 70K d'adresses IP de destination à partir de 32 points de mesure au maximum.

Le chapitre 5 décrit pourquoi la détection des FD est complexe, et montre l'outil que nous développons pour atteindre cet objectif. Alors que les travaux connexes se sont concentrés sur l'effet d'une cause particulière qui pourrait potentiellement créer des FD, notre solution permet une analyse systématique qui peut être appliquée pour détecter les FD à l'intérieur des AS de toute sorte. La méthodologie que nous proposons analyse de près la manière dont le trafic circule à l'intérieur des AS et ne nécessite pas de connaissances privilégiées concernant les réseaux analysés, par exemple la connaissance de la métrique IGP utilisée, pour conclure à l'absence de FD. Il est à noter que la détection des FD est une tâche particulièrement difficile dans ces circonstances, c'est-à-dire lorsqu'aucune hypothèse n'est faite sur la métrique IGP utilisée par les AS, puisque les détours d'acheminement et les meilleurs chemins IGP sont en fin de compte simplement des itinéraires d'acheminement. En outre, comme pour les FD, des techniques telles que l'équilibrage de la charge et l'ingénierie du trafic génèrent également des modèles de routage à trajets multiples. Contrairement aux détours de réacheminement, ceux-ci sont toujours considérés comme optimaux en termes de coût IGP ou pour les besoins spécifiques de l'AS qui les déploie, respectivement. Notre analyse peut compléter les études axées sur l'équilibrage de charge puisque nous envisageons également le LB par préfixe, une option de déploiement qui n'est pas abordée dans la littérature. Notre détecteur ne se contente pas de relever tous ces défis, mais utilise également une nouvelle étape de regroupement des préfixes, qui pourrait permettre de réduire le coût de sondage des campagnes de découverte des chemins LB, et de découvrir d'autres possibilités d'équilibrage de charge par destination. Nous testons notre détecteur FD dans l'Internet et découvrons des détours d'acheminement dans plusieurs AS, avec un remarquable motif binaire dans lequel le trafic de transit passant entre deux routeurs de bordure d'un AS ne fait jamais de détour, ou en fait pour chaque préfixe. Enfin, le concept des détours d'acheminement se concentre sur les conséquences, c'est-à-dire sur les chemins qui diffèrent des meilleurs chemins IGP, mais ne dit rien sur la manière dont ils sont générés, c'est-à-dire sur leurs causes profondes. Pour éclairer ce sujet encore inexploré, nous développons un formalisme autour des détours d'acheminement per-

mettant de décrire formellement les phénomènes qui conduisent à l'apparition de ces détours.

Résumé Chapitre 3

Dans ce chapitre, nous étudions si les IXP, qui ont connu un grand succès dans les années 2000, profitent aujourd'hui également au développement de l'Internet dans d'autres régions que l'Europe. Nous nous concentrons en particulier sur l'Amérique latine, une région qui est restée assez inexplorée malgré ses caractéristiques attrayantes. Nous nous intéressons aux politiques publiques qui ont conduit à la création des IXP d'Amérique latine, à leur croissance et à leur développement dans le temps, et au rôle que chacun d'entre eux joue dans l'écosystème national des AS, c'est-à-dire s'ils sont *échec des IXP* ou ont réussi à proliférer. Pour avoir une vision plus large, nous comparons les IXP déployés sur plusieurs continents. En bref, nos contributions sont les suivantes :

1. Nous déterminons les pays en LatAm avec des IXP et construisons l'ensemble de données le plus complet à ce jour en rassemblant des informations sur les IXP en LatAm dans Sec. 3.1. En particulier, nous recueillons des données BGP auprès de plusieurs collecteurs situés dans la région. À notre connaissance, nous sommes les premiers à explorer cet ensemble de données. En outre, nous avons étendu la vue BGP au Brésil en utilisant un collecteur de vues de routes et plusieurs lunettes de visée réparties sur plusieurs IXP brésiliens. En outre, nous combinons plusieurs sources de données supplémentaires pour obtenir des mesures qui aident à quantifier la croissance des IXP et à mieux comprendre le rôle de leurs membres fournisseurs de transit.

2. Nous donnons un aperçu dans Sec. 3.2 de la manière dont les politiques publiques des pays ont encouragé le développement des IXP en Amérique latine. Il est intéressant de noter que les gouvernements locaux de chaque pays ont été impliqués dans la création de plus de 55% des IXP d'Amérique latine. Comme pour le modèle européen des IXP, en Amérique latine, un grand nombre d'organisations à but non lucratif gèrent des IXP (et les ont également créés dans certains cas).

3. Nous étudions comment les IXP ont gagné en importance depuis leur création, et les membres qui les composent. Tandis que les IXP en Amérique latine et dans les régions en développement en général ont été capables d'attirer des membres nationaux et régionaux, les IXP européens ont également réussi à rassembler des membres de différentes régions, ce qui permet de spéculer sur la possibilité pour ces IXP de continuer à se développer à l'avenir.

4. Nous analysons le succès et l'échec des IXP en LatAm dans Sec. 3.4 en essayant de relier ce phénomène avec la présence d'un écosystème AS équilibré, c'est-à-dire où les adresses IP sont plus équitablement réparties entre les AS du pays. Nous constatons une corrélation négative entre l'absence de monopole des AS de transit/accès possédant la plupart des adresses IP attribuées à un pays et le succès des IXP nationaux.

5. Nous publions le code qui permet à la fois de récupérer les données publiques que nous avons utilisées et de reproduire nos résultats[4]. En outre, nous mettons à la disposition du public les données que nous avons recueillies manuellement dans les looking glass brésiliens[5].

De plus, nous tirons les principales conclusions de ce chapitre dans la Sec. 3.7, tandis que les annexes de la Sec. 3.3 et de la Sec. 3.5 fournissent des résultats complémentaires à notre étude. Les recherches présentées dans ce chapitre ont conduit aux travaux suivants :

- Esteban Carisimo, **Julián M. Del Fiore**, Diego Dujovne, Cristel Pelsser, et J. Ignacio Alvarez-Hamelin. 2020. *A first look at the Latin American IXPs*, dans SIGCOMM Comput. Commun. Rev. 50, 1 (janvier 2020), 18-24.

- Esteban Carisimo, **Julián M. Del Fiore**, Diego Dujovne, Cristel Pelsser, J. Ignacio Alvarez-Hamelin, *Country-level influence of IXPs in Latin America*, dans l'Atelier des étudiants d'Amérique latine sur les réseaux de communication de données (LANCOMM) 2019.

Résumé Chapitre 4

Dans ce chapitre, nous présentons la méthodologie que nous proposons pour détecter les discordances entre les trajets AS-transfert et les trajets AS BGP, que nous appelons les mensonges BGP. Cette tâche nécessite notamment de relever un défi pratique considérable : tant les techniques de mesure active de pointe, par exemple `traceroute`, que les outils de cartographie IP-to-AS utilisés de nos jours sont bruitées. Par conséquent, puisque les mensonges peuvent être mal interprétés comme du bruit, ou vice versa, il est impératif de filtrer ce bruit. Pour résoudre ce problème, nous proposons un cadre modulaire permettant de détecter les mensonges BGP et leurs bornes en éliminant toutes les sources d'erreurs interférant avec les données collectées, c'est-à-dire en filtrant le bruit affectant la comparaison des CP et des DP. En bref, nos contributions sont les suivantes :

1. Nous étudions de multiples cas d'études montrant différentes causes pouvant conduire à des mensonges BGP dans Sec. 4.1. En particulier, ces exemples illustrent pourquoi il est difficile de déterminer la cause première des mensonges BGP, un problème qui dépasse le cadre de cette thèse.

2. Nous modélisons dans Sec. 4.2 les différents défis pratiques qui doivent être relevés afin de pouvoir détecter les mensonges BGP. Cela va de la nécessité de synchroniser les mesures dans le temps, de pouvoir mesurer dans une plateforme où les chemins de contrôle et les chemins de données peuvent être collectés dans le même réseau, jusqu'à la description des multiples sources de bruit qui peuvent interférer dans la comparaison des CP et des DP, c'est-à-dire les AS jumeaux, les TPA ou les IXP et les sauts manquants.

[4] `https://github.com/CoNexDat/latam-ixp-obs`
[5] `https://cnet.fi.uba.ar/latam-ixp-obs/lg-ribs/`

3. Nous développons une méthodologie permettant de calculer le taux de mensonges BGP pour un point de mesure donné dans le cadre de la Sec. 4.3. Notre cadre comporte trois étapes :

   - a étape de préparation qui synchronise les CP et les DP dans le temps et au niveau sémantique, c'est-à-dire qui convertit les DP des chemins IP en chemins AS ;
   - un étape de relaxation cartographique qui examine les DP et les CP séparément et tente de déduire le bruit affectant chacun d'eux. Dans cette tâche, nous uniformisons les AS jumeaux en leur attribuant une cartographie unique pour les CP et les DP, et nous convertissons les TPA possibles affectant les DP en éléments neutres qui peuvent être mis en correspondance avec n'importe quel AS;
   - a étape de correction des éléments neutres qui, en comparant les CP et les DP, tente de déduire les valeurs des éléments neutres (y compris les sauts manquants) présents dans les DP (le cas échéant), et détermine si les CP et les DP correspondent ou non.

   Notre cadre est modulaire : les règles de filtrage sur la relaxation de la cartographie, et l'ordre dans lequel elles sont appliquées, peuvent être sélectionnées, ce qui permet de mettre en œuvre différents modèles de filtrage du bruit.

4. Nous déployons 8 points d'observation co-localisés, 6 dans les PEERING Testbed et deux dans les réseaux privés, et la travail effectué sur cette recherche longitudinale se trouve dans Sec. 4.4. À notre connaissance, notre analyse est la première à s'étendre dans le temps et à déployer un si grand nombre de points d'observation pour une comparaison des DP et des CP.

5. Nous assainissons l'ensemble des données avec différents modèles de filtrage du bruit que nous construisons avec notre cadre et calculons le taux de mensonges BGP que nous mesurons pour chacun d'eux dans Sec. 4.5. Notre modèle le plus conservateur, c'est-à-dire celui qui filtre plus agressivement le bruit et permet d'obtenir une limite inférieure de mensonges BGP, révèle une quantité non négligeable de mensonges. En outre, nous constatons que dans les points de mesure où notre cadre est très efficace pour réduire le nombre de discordances entre les PC et les PD, les résultats restent généralement assez stables dans le temps. D'autre part, lorsque notre cadre révèle un grand nombre de mensonges BGP potentiels, les résultats présentent une plus grande variation par jour.

La Sec. 4.6 tire les dernières remarques de notre étude. Les travaux présentés dans ce chapitre ont donné lieu à la publication suivante :

- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser et Antonio Pescapè. *Filtering the Noise to Reveal Inter-Domain Lies*, dans 2019 Network Traffic Measurement and Analysis Conference (TMA), pages 17–24, 2019, IEEE.

- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *A BGP-lying Tale: Stop Blamming the Mapping*, poster dans TMA 2018.

Résumé Chapitre 5

Dans ce chapitre, nous examinons de près le phénomène des détours d'acheminement. À notre connaissance, nous sommes les premiers à nous attaquer au problème de la détection des détours d'acheminement, sans distinction des causes sous-jacentes qui les génèrent, tout en filtrant les techniques d'équilibrage des charges et d'ingénierie du trafic. En bref, nous apportons les contributions suivantes :

- Nous développons un formalisme autour des détours d'acheminement dans le Sec. 5.1. En particulier, nous construisons un modèle d'acheminement et montrons que les incohérences d'acheminement peuvent évoluer en modifications d'acheminement qui peuvent alors être à l'origine de détours d'acheminement. En outre, nous étudions le modèle de réacheminement : les détours de réacheminement prennent naissance à l'intérieur des AS, c'est-à-dire si le trafic entre deux points d'extrémité fixes emprunte des itinéraires différents selon le préfixe considéré, et nous le comparons à celui généré par les techniques d'équilibrage de charge et d'ingénierie du trafic. En particulier, les FD, les LB et les TE par préfixe génèrent un schéma d'acheminement similaire, que nous appelons "basé sur le préfixe".

- Nous concevons un algorithme capable de détecter les modèles de transfert basés sur les préfixes dans les Sec. 5.3. Notre cadre repose uniquement sur les données de cartographie IP-AS et les informations de plan de données collectées avec `traceroute`. Nous présentons une nouvelle stratégie pour rechercher des modèles de routage multi-chemins, en plusieurs étapes. Notre technique regroupe des préfixes pour lesquels les mêmes routes internes des AS sont révélées, une idée qui peut être incorporée dans les études de découverte de topologie pour réduire leur coût de sondage associé.

- Nous proposons un détecteur FD dans Sec. 5.4. Notre solution ajoute une dernière phase à l'algorithme précédent : elle applique un verdict permettant de discriminer les FD des autres mécanismes qui génèrent également des schémas d'acheminement basés sur des préfixes. Pour cela, nous nous concentrons sur les cas extremes, c'est-à-dire les scénarios où les FDs affectent de nombreux préfixes. Nous construisons un détecteur de détours de réacheminement comme un outil prêt à être utilisé sur le terrain.

- Nous analysons le phénomène des FD sur le terrain dans Sec. 5.5, en faisant fonctionner notre détecteur de FD à partir de 100 nœuds de l'infrastructure de surpervision NLNOG RING, et nous trouvons des FD dans 25 des 54 AS sondés. Nous trouvons un motif binaire remarquable dans lequel le trafic de transit passant entre deux routeurs de bordure d'un AS ne fait jamais de détour, ou en fait toujours. Nous avons validé le comportement du détecteur de FD avec des émulations et sur un réseau où nous avons le contrôle.

- Nous publions l'ensemble des données que nous avons collectées, les configurations d'émulations et notre code pour favoriser la reproductibilité.

Nous discutons de la robustesse du détecteur FD que nous avons mis en œuvre dans la Sec.5.6, et nous tirons des remarques finales dans la Sec. 5.7. Les travaux présentés dans ce chapitre ont conduit à la rédaction de l'article suivant :

- **Julián M. Del Fiore**, Valerio Persico, Pascal Merindol, Cristel Pelsser et Antonio Pescapè. *The Art of Detecting Forwarding Detours*, accepté dans l'IEEE Transactions on Network and Service Management (IEEE TNSM).
- **Julián M. Del Fiore**, Pascal Merindol, Valerio Persico, Cristel Pelsser and Antonio Pescapè. *Routing Inconsistencies at the FIB level*, poster dans TMA 2019.

Dans l'ensemble, les études que nous menons dans le cadre de cette thèse révèlent qu'Internet n'est pas un système infaillible et que ses pièces défaillantes peuvent être mises en lumière en élaborant des méthodologies comme celles que nous développons.