UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

A system for the Detection of Limited Visibility in BGP

Autor: Andra Lutu
Director: Dr. Marcelo Bagnulo Braun, PhD

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA

Leganés, Septiembre de 2014

TESIS DOCTORAL

A System for the Detection of Limited Visibility in BGP

Autor: Andra Lutu
Director: Dr. Marcelo Bagnulo Braun, PhD

Firma del tribunal calificador:

Firma:

Presidente:

Vocal:

Secretario:

Calificación:

Leganés, de de

Acknowledgements

I would firstly like to acknowledge the support of my advisor, Marcelo Bagnulo, to whom I am forever grateful for sharing his knowledge, for his guidance, patience and trust, from which I am truly fortunate to benefit. In addition, his assistance helped me obtain an internship at IIJ Innovation Institute and another at the Center for Applied Internet Data Analysis (CAIDA), at the Supercomputer Center, UCSD, which fueled my desire to continue my work on real-world problems.

I would like to acknowledge the support of Cristel Pelsser from IIJ, whose pragmatic approach to interdomain routing made me challenge and strengthen the research ideas developed in this thesis. The two-months internship at IIJ and subsequent ongoing collaboration resulted in in an important part of this thesis.

I am grateful to Olaf Maennel for his in-depth knowledge of interdomain routing and refreshing perspective on the research problems that we tackled throughout our lengthy collaboration.

I would also like to acknowledge the support of my co-authors, Rade Stanojevic and Jesus Cid-Sueiro. I am sincerely privileged and greatly appreciate their help and useful comments.

The staff and students at IMDEA Networks Institute, the Telematics Department of University Carlos III of Madrid and, in particular, the NETCOM research group provided me with an excellent research environment. I am especially grateful to Alberto Garcia-Martinez, who devoted a significant amount of time to listen, guide and support me. I would like to also thank Francisco Valera Pintor and Pierre Francois for frequent useful discussions.

I thank Alberto Dainotti and Amogh Dhamdhere for guiding me during my internship at CAIDA, for their time, energy and invaluable feedback.

Lastly, I thank my parents, to whom I owe everything. I thank the rest of my family and all my loved ones for being always there for me in the best and, more importantly, in the worst of times. I thank all who stand by me, all whose names I do not mention here, but keep forever in my thoughts.

Abstract

The performance of the global routing system is vital to thousands of entities operating the Autonomous Systems (ASes) which make up the Internet. The Border Gateway Protocol (BGP) is currently responsible for the exchange of reachability information and the selection of paths according to their specified routing policies. BGP thus enables traffic to flow from any point to another connected to the Internet. The manner traffic flows is often influenced by entities in the Internet according to their preferences. The latter are implemented in the form of routing policies by tweaking BGP configurations. Routing policies are usually complex and aim to achieve a myriad goals, including technical, economic and political purposes. Additionally, individual network managers need to permanently adapt to the interdomain routing changes and, by engineering the Internet traffic, optimize the use of their network.

Despite the flexibility offered, the implementation of routing policies is a complicated process in itself, involving fine-tuning operations. Thus, it is an error-prone task and operators might end up with faulty configurations that impact the efficacy of their strategies or, more importantly, their revenues. Withal, even when correctly defining legitimate routing policies, unforeseen interactions between ASes have been observed to cause important disruptions that affect the global routing system. The main reason behind this resides in the fact that the actual inter-domain routing is the result of the interplay of many routing policies from ASes across the Internet, possibly bringing about a different outcome than the one expected.

In this thesis, we perform an extensive analysis of the intricacies emerging from the complex netting of routing policies at the interdomain level, in the context of the current operational status of the Internet. Abundant implications on the way traffic flows in the Internet arise from the convolution of routing policies at a global scale, at times resulting in ASes using suboptimal ill-favored paths or in the undetected propagation of configuration errors in the routing system. We argue here that monitoring *prefix visibility* at the interdomain level can be used to detect cases of faulty configurations or backfired routing policies, which disrupt the functionality of the routing system. We show that the lack of global prefix visibility can offer early warning signs for anomalous events which, despite their impact, often remain hidden from state of the art tools. Additionally, we

show that such unintended Internet behavior not only degrades the efficacy of the routing policies implemented by operators, causing their traffic to follow ill-favored paths, but can also point out problems in the global connectivity of prefixes.

We further observe that majority of prefixes suffering from limited visibility at the interdomain level is a set of more-specific prefixes, often used by network operators to fulfill binding traffic engineering needs. One important task achieved through the use of routing policies for traffic engineering is the control and optimization of the routing function in order to allow the ASes to engineer the incoming traffic. The advertisement of more-specific prefixes, also known as prefix deaggregation, provides network operators with a fine-grained method to control the interdomain ingress traffic, given that the *longest-prefix match rule* over-rides any other routing policy applied to the covering less-specific prefixes.

Nevertheless, however efficient, this traffic engineering tool comes with a cost, which is usually externalized to the entire Internet community. Prefix deaggregation is a known reason for the artificial inflation of the BGP routing table, which can further affect the scalability of the global routing system. Looking past the main motivation for deploying deaggregation in the first place, we identify and analyze here the economic impact of this type of strategy. We propose a general Internet model to analyze the effect that advertising more-specific prefixes has on the incoming transit traffic burstiness. We show that deaggregation combined with selective advertisements (further defined as strategic deaggregation) has a traffic stabilization side-effect, which translates into a decrease of the transit traffic bill. Next, we develop a methodology for Internet Service Providers (ISPs) to monitor general occurrences of deaggregation within their customer base. Furthermore, the ISPs can detect selective advertisements of deaggregated prefixes, and thus identify customers which may impact the business of their providers. We apply the proposed methodology on a complete set of data including routing, traffic, topological and billing information provided by an operational ISP and we discuss the obtained results.

Contents

List of Figures

Chapter 1

Introduction

The performance of the global routing system is vital to thousands of entities operating the Autonomous Systems (ASes) which make up the Internet. The Border Gateway Protocol (BGP) is currently responsible for the exchange of reachability information between ASes and for the selection of the best paths to be used in order to enable the flow of traffic between two nodes at the interdomain level. The manner in which traffic flows is often influenced by network entities according to their needs, thus making BGP a policy-driven protocol. By tweaking configurations to influence the BGP decision process, the network operators are able to implement their routing preferences, designed to accommodate myriad economic, technical and political goals. Furthermore, network operators can adapt to the developing Internet ecosystem and, by engineering the Internet traffic, strive to optimize the use of their network.

Despite the flexibility offered, the added complexity of routing policies comes with consequences, e.g., unforeseen side-effects impacting other entities active in the routing system, widespread undetected misconfiguration and even conflicting configurations leading to bogus routing policies to activate. In the context of tangled interconnections between ASes and their routing policies, unexpected developments have been observed to cause important disruptions that affect the global routing system [1, 2]. The main reason behind this resides in the fact that the actual interdomain routing is the result of the interplay of many routing policies from ASes across the Internet, possibly bringing about a different outcome than the one expected. Consequently, in order to ensure the efficiency of their routing policies, ASes periodically control how their preferences resonate in the routing system or how others' routing policies are affecting them.

Not to mention, the mere implementation of routing policies is a complicated process in itself, involving fine-tuning operations. Thus, it is an error-prone task and operators might end up with faulty configurations that impact the efficacy of their strategies or, more importantly, their revenues. Flawed routing policies cause for anomalies to emerge in the Internet, including interdomain prefix leaks, e.g., the case of Dodo leaking its full

BGP routing table to provider Telstra in February 2012 [3], prefix hijacks, e.g., the well-known case of Pakistan Telecom hijack of YouTube [4] or prefixes not being distributed everywhere, e.g., the case where some multi-homed networks could not see the prefix of the DNS K root server [5]. Over the last years, a lot of effort has gone in the direction of identifying, classifying and eliminating some of these anomalies [6].

In this thesis, we perform an extensive analysis of the intricacies emerging from the complex netting of routing policies at the interdomain level, in the context of the current operational status of the Internet. Abundant implications on the way traffic flows in the Internet arise from the convolution of routing policies at a global scale, at times resulting in ASes using suboptimal ill-favored paths or in the undetected propagation of configuration errors in the routing system. We argue here that monitoring *prefix visibility* at the interdomain level can be used to detect cases of faulty configurations or backfired routing policies, which disrupt the functionality of the routing system. We show that the lack of global prefix visibility can offer early warning signs for anomalous events which, despite their impact, often remain hidden from state of the art tools, e.g., [7, 8]. Additionally, we show that such unintended Internet behavior not only degrades the efficacy of the routing policies implemented by operators, causing their traffic to follow ill-favored paths, but can also point out problems in the global connectivity of prefixes.

We further observe that majority of prefixes suffering from limited visibility at the interdomain level is a set of more-specific prefixes. We learn that network operators intentionally limit the visibility of distinct fragments of their address space by selectively announcing the more-specifics to different upstream providers. This strategy is adopted by ASes to fulfill binding traffic engineering needs. Interdomain traffic engineering requirements usually depend on the connectivity of the AS with others and on the type of business handled by the network [9]. One important task achieved through the use of routing policies for traffic engineering is the control and optimization of the routing function in order to allow the ASes to engineer the incoming traffic. The advertisement of more-specific prefixes, also known as prefix deaggregation, provides network operators with a fine-grained method to control the interdomain ingress traffic, given that the *longest-prefix match rule* over-rides any other routing policy applied to the covering less-specific prefixes.

Nevertheless, however efficient, this traffic engineering tool comes with a cost, which is usually externalized to the entire Internet community. Prefix deaggregation is a known reason for the artificial inflation of the BGP routing table [10], which can further affect the scalability of the global routing system. Recognizing the stable usage of prefix deaggregation as an efficient traffic engineering method [11], we look past the initial motivations behind deploying this type of strategy, and instead focus on its *aftermath*. More specifically, we investigate here its potential economic impact on the transit traffic bill, independently on the main reasons driving the network operators to use prefix deaggre-

gation in the first place. We show that by limiting the visibility of more-specific prefixes with the usage of selective advertisements of deaggreagated prefixes, network operators reduce the route diversity towards each prefix announced. Consequently, this also reduces the incoming traffic fluctuations on the corresponding transit link, thus possibly decreasing the monthly traffic bill paid to the transit providers. This can further impact the routing policies of its transit providers and, potentially, the providers' revenues.

We provide next an overview of the contributions included in this thesis, which we organize in five different chapters, as follows.

1.1 The BGP Visibility Scanner

In Chapter 3 of this thesis, we begin by presenting the ***BGP Visibility Scanner***, the tool we have developed to enable network operators to validate the correct implementation of their routing policies by monitoring prefix visibility. The BGP Visibility Scanner is publicly available at **visibility.it.uc3m.es**. The scanner corroborates BGP routing information from more than 150 independent observation points in the interdomain to create a multi-angled view on the efficiency of routing policies. We have publicly released the visibility scanner in November 2012. Ever since, the tool has been well received by the operational community. The BGP Visibility Scanner [12] has been presented in different network operators group meetings, including NANOG, LACNOG, UKNOF, EsNOG, RIPE, and has also been announced on RIPE Labs [8]. It continues to evolve and to attract a large amount of attention and feedback [13], thus validating its usefulness for the operational community.

The implementation of routing policies is a complicated process, involving subtle tuning operations serving all the origin's goals. For example, ill-defined outbound filters may lead to an AS unknowingly leaking internal routes to the Internet and impacting the effectiveness of its own active routing policies [6]. Moreover, because of the increasing density of AS-level interconnections [14], the manner a routing policy is implemented may not achieve the expected outcome. In order to avoid the distortions of their routing policies due to accidental mis-configurations or adverse effects within the complex external web of routing policies, ASes need to monitor the manner in which their preferences resonate in the global routing system. To this end, operators need to complement their internal perspective on routing with the information retrieved from multiple external sources, e.g. publicly available looking-glasses. However useful, these tools have obvious limitations [15], e.g. allowing only for single per-route queries and not storing any historical information.

There are numerous efforts towards detecting security related routing conditions, such as prefix hijacking (e.g., PHAS [16]). Also, various tools exist to provide useful information for operators [17, 18]. Multiple operational misconfigurations have been reported [6], but

attempts go far beyond this. They include *RIPE Labs* [8], which has a whole section devoted to tools that assists operators or Renesys [19] and BGPmon [20], which operate this type of services to operators for a fee. Unlike tools which integrate a vast amount of operational problems [7], we do not focus on inferring and/or monitoring the AS-level topology of the Internet, but on monitoring the healthy deployment of routing policies through prefix visibility. In this sense, our work is very closely related to the work on BGP wedgies by Griffin et al [2, 21]. However, none of those theoretical work is able to detect problematic routing conditions based on raw BGP observations. All of this work requires access to configuration files, which are typically not shared. The latter are considered a company secret which BGP was designed to hide, making it hard to be inferred [15]. While we understand the limitations of BGP protocol monitoring, we noticed that still a great deal that can be inferred. In this sense, our work aims at reporting and aggregating the information to make it usable for operators.

Despite that many other similar tools [7, 8] leverage the massive amount of available routing data, the BGP visibility scanner is, to the best of our knowledge, the only tool offering specific information on global prefix visibility. The tool allows networks to check how their own routes are being propagated in the Internet, verify the results of the implemented routing policies and identify possible cases where these policies backfired. By merging all the available information from the ASes enabled as active monitors active in the RIPE RIS [22] and RouteView [23] Projects, we create a *visibility scanner* for all the IPv4 and IPv6 prefixes active in the interdomain. The tool is subject to the limitations of the available public looking-glasses, which we further address accordingly. It is important to note that the properties of BGP do not allow us to get a complete picture on all policies, but nevertheless those public observations points provide a multi-angle perspective on the interdomain routing.

Moreover, our tool has already proven its capability of triggering visibility alarms and helping networks deal with the problems caused by their own routing policies. Since the tool first became publicly available, it gathered, at the time of writing, over 5,000 queries performed for more than 1,500 different origin ASes. We have invited the users actively performing queries on the BGP Visibility Scanner to participate in a survey regarding the status of the retrieved LVPs for the corresponding origin AS. Leveraging the feedback received, we build a unique ground-truth dataset including 20,000 *LVPs*. For each of these prefixes, the network operators reported which was the expected visibility status of the prefixes after defining their interdomain routing policies. We match the origin's intention with the observed visibility status of the prefixes identified with the BGP Visibility Scanner, and separate the *LVPs* in two pre-determined classes: *intended* and *unintended*. As a results, we identify 1,150 prefixes of the class *intended* and a staggering 18,850 *LVPs* of the class *unintended*, which we further help eliminate.

1.2 Winnowing Unintended Limited Visibility Prefixes

In Chapter 4, we continue our analysis of prefix visibility and propose a machine learning model to automatically identify potential cases of bogus routing policies or misconfiguration. The problem we challenge here is **distinguishing the *unintended prefixes suffering from limited Internet visibility*, caused by configuration errors or unforeseen routing policies interactions, from the *intended prefixes with limited visibility at the interdomain level*, which are natural expressions of intentional routing policies in the Internet**. Though some legitimate routing policies of an AS intentionally constrain the visibility of its prefixes in the Internet, the limited visibility can, more than often, stem from human operator errors or unpredicted interplay with the external entanglement of otherwise correctly defined routing policies. In light of the perpetuity of the above-mentioned causes of anomalous interdomain events, there is an overall acute need for a simple warning system for faulty configurations and/or problematic external routing conditions to assist operators in optimizing the performance of their routing policies. In this thesis, we further argue that monitoring the *prefix visibility* at the interdomain level can be used to detect a subset of anomalous events that still remain hidden from state of the art tools [7, 8], despite their important impact on the routing system.

We design a machine learning *Winnowing Algorithm* able to predict with 95% accuracy if the limited visibility of a prefix is intended or unintended. We exploit the **unique ground-truth dataset of 20,000 prefixes**, for which the expected visibility status has been confirmed by the networks operators themselves, while actively using the BGP Visibility Scanner. We then rely on the robust machine learning concept of *boosted classification trees* [24] to train the system on this ground-truth operational set of prefixes with limited visibility and thus enable it to learn the patterns of misconfigurations and backfired routing policies which are normally hard to detect. The classification model uses only visibility-related per-prefix features in order to predict with 95% accuracy if the limited visibility of a prefix is intentional or not.

1.3 Reachability of Limited Visibility Prefixes

In Chapter 5, we continue our analysis of limited visibility prefixes and we further analyze the potential correlation between the limited visibility and the reachability of the prefix at the interdomain level. Routing policy anomalies and misconfiguration not only impact the operations and strategies of Internet entities by causing the traffic to follow an ill-favored path. Policies may, at times, affect the propagation of routes, making some paths unavailable at a global level, and sometimes preventing a prefix to be learned altogether. There are cases when the limited visibility prefixes become globally

unreachable, since there might not always be a less-specific covering prefix to ensure that the destinations attached are globally reachable.

We aim to establish if prefix visibility at the interdomain level can be further used to alert network operators about reachability issues their prefixes might be suffering. In other words, we analyze if the routing anomalies that render a prefixes as limited visibility can also deteriorate the reachability of the corresponding address space. To this end, we perform reachability measurements towards prefixes with various degrees of visibility and that may or may not be covered by a less-specific prefix, with the goal of establishing the existence of a correlation between visibility and reachability of prefixes in the Internet. We find that for the set of limited visibility prefixes which are not covered by a less-specific prefix with high visibility, i.e., the dark prefixes, reachability is also an issue, especially in IPv6. Contrariwise, the limited visibility prefixes covered by a high visibility prefixes do not show any reachability problems, despite the fact that the traffic may be following the ill-favored path of the less-specific prefix.

1.4 The Inadvertent Economic Impact of Strategic Deaggregation

We previously observed that the majority of prefixes with limited visibility identified with the BGP Visibility Scanner includes prefixes that are covered by less-specific prefixes which, in general, have high visibility at the interdomain level. Often, network operators intentionally limit the visibility of distinct fragments of their address block by selectively announcing the more-specific prefixes to different upstream providers, while still injecting the covering less-specific prefix to all the providers. This is usually driven by severe traffic engineering needs, e.g., load balancing the traffic by originating several more-specific prefixes and announcing different prefixes via different AS paths. In Chapter6 we focus on this particular traffic engineering strategy and study its potential economic implications.

The injection of *more-specific prefixes* through BGP offers a fine-grained method to control the interdomain ingress traffic, since it allows networks to divide their assigned address blocks in different-sized sinks of traffic. This type of phenomenon is commonly known as *prefix deaggreagation*. We further define the observed specific deaggregation phenomena as *strategic deaggregation*, i.e., the action of deaggregating the address block and selectively injecting each more-specific prefix to different transit providers or different disjoint subsets of transit providers. We use here the term *strategic* to accentuate the fact that the decision is based on optimizing behavior, since it might increase the benefits for the network deploying it. This relies on definitions provided in rational choice theory. Using this technique, geographically-spread networks can, for example, divert different amounts of traffic corresponding to different points of presence (PoP), thus attracting

the desired amount of traffic into their network through the PoP closest to the final destination.

Several adjacent phenomena associated with deaggregation have been identified and studied by the research community. The most important negative side-effect of the widespread adoption of this technique is the artificial inflation of the BGP routing table, which can affect the scalability of the global routing system. This issue has become an important concern of the entire Internet community over the past years [10, 25, 11]. Judging from this perspective, this type of behavior is considered to be harmful [10], as it heavily impacts the global routing table scalability and it acts counter to the goals of the Classless Inter Domain Routing (CIDR) architecture, which encourages aggressive address aggregation. However, it has been proved that, despite being frowned upon, deploying this type of strategy is a constant occurrence in the Internet [11]. Thus, recognizing as a reality the stable usage of strategic deaggregation as an efficient traffic engineering method in the Internet, we look past the motivations behind deploying deaggregation, and instead we focus on its economic *aftermath*.

In spite of the negative overtone surrounding prefix deaggregation, a series of collateral benefits emerge as by-products from the use of deaggregation, without, however, constituting the central drive for ASes to deploy such strategies in the first place. For example, one alleged benefit is the increased security of the network announcing more specifics in the interdomain. Some even claim that prefix deaggregation can be useful as a technique for reducing prefix-hijacking attacks [26]. In this thesis, we investigate the potential economic impact of strategic deaggregation, independently on the reasons driving the network operators to employ the technique in the first place [27, 28]. We find that, as a result of the unique interaction between the routing path changes in the current Internet [29], the skewed distribution of traffic demand on sources and the widely used 95^{th} percentile billing model [30, 31] for calculating the monthly transit cost, the deaggreagating ASes enjoy one particular by-product of strategically deaggregating their address space: *the decrease of the monthly transit bill*. We propose a general Internet model to analyze the impact of strategic deaggregation in terms of transit traffic cost. Based strictly on the information contained in public BGP routing tables, we show that strategic deaggregation may decrease the transit bill of a given customer by 5% in average. Clearly, in the operational Internet some of these ASes that were taken into account for this approximation are much more affected by routing changes than others, and thus may experience . However, this average gives us a somewhat concrete idea on the manner in which the combination of routing changes, the skewed distribution of traffic on sources (analyzed in [32]) and the popular 95% percentile billing scheme [30, 31] inflate ones transit bill.

1.5 Strategic Deaggregation Detection

We have previously verified from an analytical point of view that through combining selective advertisements with the deaggregation technique, an AS can inadvertently enjoy a reduction of the traffic fluctuations on the transit links. In other words, indirectly restricting the visibility of a certain destination prefix through selective advertisements of more-specific prefixes translates into monetary savings for the deaggregating network. This economic impact is enabled by the current operational status of the Internet, and in particular the widely used billing model based on the peak traffic usage [30, 31]. In this scenario, the customer network not only acts counter to the best recommended practices regarding deaggregation, but might also indirectly impact the business of its providers.

In Chapter 7, we take the point of view of a transit provider with customers that might be deploying strategic deaggregation and ask a two-staged question:

(1) How extensive is the use of prefix deaggregation among the customer networks? To answer this question, we further propose a methodology to identify cases of strategic deaggregation deployed by ASes in the customer base of an operational Internet Service Provider (ISP). In this way, we enable any operator with the necessary methodology aimed to detect the customers which are heavy deaggregators and further monitor their behavior in time.

(2) Can it be verified that deaggregation combined with selective advertisements decreases the transit bill of some customers? Customers which exhibit this behavior may be able to game the 95^{th} percentile billing rule and possibly have a negative impact on the routing strategies of their ISPs. We propose a passive measurement approach for the detection of strategic deaggregation events and to asses their economic consequences. Our approach requires obtaining and processing routing, topology, traffic and billing information and molding it in order to reach the correct level of understanding on the impact different customers might have on their providers.

The novelty of this methodology is the manner in which it merges different types of information characteristic to a transit provider, in order to have a complete picture on the operations of its customer networks. Any ISP interested in detecting the occurrence of this phenomena within its customer base can build the necessary dataset and apply the proposed detection methodology.

We demonstrate the use of this methodology on the real traffic and routing data from a major Japanese ISP to identify real occurrences of the interest phenomena. Overall, we do not observe much deaggregation generated from the customer networks of this operational ISP. And even more, despite the fact that the proposed analytical model does successfully support the observed phenomena, we do not identify many cases of strategic deaggregation performed by the ISP's customers. We do, however, distinguish and analyze a strategic deaggregation case that fulfills all the constraints imposed in the methodology.

Regardless of the main goal to be achieved through deaggregation, we observe that, in certain conditions, the deaggregating AS can indeed enjoy a decrease of its transit traffic bill as a by-product of the strategic deaggregation deployed within certain conditions. In this representative case of strategic deaggregation detected within the customer base of a major Japanese ISP, the deaggregating AS enjoyed a 20% decrease on its transit bill.

1.6 Thesis Overview

The remainder of this thesis is organized in six additional chapters. In Chapter 2 we provide an background to acquaint the reader with the details required for the specific issue being addressed in each of the following chapters. Next, in Chapter 3 we present the BGP Visibility Scanner, a novel tool we developed and released in order to help operators monitor the sanity of their routing policies by verifying their prefix visibility in the Internet. Using the feedback received from active users of the tool, who have benefited from the information provided, we build a large ground-truth dataset consisting of 20,000 prefixes diagnosed with limited visibility. We further label these LVPs as intended or unintended, depending on the match between the observed visibility status and the reported intention of the origin AS. Leveraging this unique dataset, we present in Chapter 4 a machine-learning algorithm which can accurately distinguish between the intended and unintended LVPs identified by the BGP Visibility Scanner. Throughout the remainder of this thesis, we further characterize and analyze the implications of the limited visibility prefixes in the Internet. In Chapter 5 we set out to determine if the limited visibility of prefixes also impacts the manner in which traffic flows in the Internet. We find that for the dark prefixes global reachability is an issue, since these prefixes with limited visibility do not have a covering less-specific HVP to ensure the global reachability. For the rest of the LVPs which do have a less-specific covering prefix with high visibility to ensure global reachability, the traffic destined to the more-specific LVP ends up following initially ill-favoured paths. In some cases, these latter LVPs are generated by ASes which adopt prefix deaggregation strategies to attain an optimized use of their network and engineer the incoming traffic to follow the preferred incoming path. In other words, some ASes intentionally limit the visibility of distinct fragments of their address block by selectively announcing the more-specifics to different upstream providers. In Chapter 6, we analyze the economic impact of strategic deaggregation in term of monthly transit bill reduction. We look for real-world occurrences of these strategies in Chapter 7. We propose a new methodology which we further exemplify on a dataset obtained from an operational ISP. Finally, in Chapter 8 we conclude the thesis and discuss directions for future work.

1.7 Statement of Research Contributions

The work outlined in this thesis is focused on developing techniques, tools and models to assist network operators, protocol designers and researchers in understanding the manner in which BGP routing policies take effect in the Internet and which may be their possible impacts on the community. To this end, we analyze two different phenomena characteristic to interdomain routing: *prefix visibility* at the interdomain level and *prefix deaggregation*, with a focus on the collateral benefits of strategic deaggregation.

We introduce the concept of *prefix visibility*, which we use as a simple warning to further detect cases of erroneous or bogus routing policies. In order to define prefix visibility, we compare the BGP information contained in different *Global Routing Tables (GRTs)* from ASes which voluntarily make their data public. We loosely define the GRT as the routing table provided by an Internet Service Provider (ISP) to its customers requesting a full routing feed. Each ISP maintains its own version of the GRT, which may vary from one network to another in terms of routes contained. We define **Limited-Visibility Prefixes (LVPs)**as stable long-lived Internet routes that are visible in the GRTs of at least two different ASes and in *at most* 95% of all the GRTs from the ASes analyzed. The choice of the 95% visibility threshold allows for a 5% error in the routing tables sampling process, also accommodating possible glitches that may appear in the data. Complementarily, we define the **High-Visibility Prefixes (HVPs)** as the set of prefixes that are propagated in *at least* 95% of all the available GRTs. We also identify the **Dark Prefixes (DPs)** [33], which represent the subset of *LVPs* that are not covered by any *HV* less-specific prefix.

We observe that majority of LVPs are deaggregated more-specific prefixes, often used by network operators to implement their routing policies and achieve stringent traffic engineering needs, e.g., load balancing. The traffic destined to these limited visibility less-specifics is globally routed using sometimes ill-favored paths towards the less-specific prefix with high visibility in the Internet, which basically acts like a traffic magnet. The impact of *prefix deaggregation* on the routing system has long been a reason of debate in the Internet community. Though usually frowned upon, this strategy is even more commonly used nowadays, especially in light of the IPv4 address space depletion.

The contributions of this thesis are summarized as follows. First, we propose the **BGP Visibility Scanner** [12, 34]s, publicly available at **visibility.it.uc3m.es**, a tool which allows network operators to check the visibility of their IPv4 and IPv6 prefixes and detect unintended policies. We have publicly released the visibility scanner in November 2012. Ever since, the tool has been well received by the operational community. The BGP Visibility Scanner [12] has been presented in different network operators group meetings, including NANOG, LACNOG, UKNOF, EsNOG, and has also been announced on RIPE Labs [8]. It continues to evolve and to attract a large amount of attention and

feedback [13], thus validating its usefulness for the operational community.

Second, we build **a unique ground-truth dataset of 20,000 LVPs** [35, 34], for which the expected visibility status has been confirmed by the network operators themselves, while actively using the BGP Visibility Scanner. After comparing the observed with the expected visibility status, we label each of these LVPs as *intended* to have limited visibility, or as *unintended*. Collecting feedback from operators regarding their intended routing policies has not been an easy task. We have invited the operators using our tool to fill in an on-line survey form after performing a query in the BGP visibility Scanner, to provide more information on the observed visibility status of their prefixes. Additionally, we have actively been in contact with various operators who asked for our support while debugging their routing policies. The dataset brings additional value in that it accurately documents distinct causes for the limited visibility of prefixes in the Internet and provides a deep understanding of the routing conditions which allows them to emerge. It records and explains multiple cases of misconfigurations, unforeseen interactions and intentional routing policies effects.

We propose the **Winnowing Algorithm** [35, 34], a machine learning classification algorithm able to automatically distinguishing the *unintended LVPs*, caused by misconfiguration or unforeseen routing policies interactions, from the *intended LVPs*, which emerge as a natural expression of intentional routing policies in the Internet. Using the BGP Visibility Scanner as a data mining tool, we corroborate the per-prefix visibility information with the ground-truth information from active users of the tool to generate a simple alarm simple for misconfigurations or bogus routing policies which emerge as unintended limited-visibility prefixes. The resulting Winnowing Algorithm has a 95% level of accuracy. This further proves that visibility features are generally powerful to detect anomalies which, despite their impact on the routing system, are hard to single out due to the limited and distributed nature of the data.

Third, we analyze the **correlation between global visibility and global reachability** of both IPv4 and IPv6 prefixes [36, 34]. The faulty configurations, complex inter-domain interactions or bogus routing policies not only impact the efficacy of the intended routing policies of ASes. Sometimes, the impacted prefixes are prevented to be learned altogether, making the attached host globally unreachable. It is expected that *limited visibility does not necessarily imply limited reachability*, since the less-specific high visibility covering prefix provides reachability. This is not true for the DPs, which lack a covering less-specific prefix with high visibility to ensure that the attached hosts are still reachable. We show that the lack of global visibility of a prefix does sometime signal the potential risk of limited global reachability, especially in the IPv6 Internet. While the IPv4 dark address space can be largely explained as configuration slips, route leaks or errors, this is not valid for the IPv6 DPs. We find that the subset of IPv6 dark prefixes are highly unreachable. We believe that this is a serious problem for the IPv6 Internet, as limited

reachability of a non-negligible set of prefixes undermines the global connectivity of the Internet.

Fourth, we observe that the majority of limited-visibility prefixes identified with the BGP Visibility Scanner are actually more-specific prefixes covered by a less-specific prefix with high visibility. Given that majority of prefixes suffering from limited visibility are the result of **strategic deaggregation**, we turn our attention to this controversial practice and analyze its **aftermath in terms of the impact on the transit traffic dynamics and on the subsequent transit bill paid by a deaggregating AS to its transit providers** [27, 28]. We propose an Internet model to analyze the cost for transit in the case when strategic deaggregation is being used, as opposed to the case when routing is not influenced by this phenomenon. We demonstrate that by using deaggreagation and scoped advertisements, the originating AS reduces the path diversity towards the injected prefixes. Thus, the amount of traffic destined to a particular sub-block of addresses is bound to the incoming link on which it was injected. This eliminates the possibility for any traffic fluctuations due to routing changes towards that particular prefix. Since the transit bill depends on the peak traffic usage and not on the total traffic usage, avoiding traffic fluctuations implies a lower monthly bill.

Fifth, we propose **a passive measurement approach for the detection of strategic deaggregation events and to asses their economic consequences in the real world** [37, 28]. The novelty of the approach is the manner in which it merges different types of information characteristic to an real-world operational ISP in order to have a complete picture on the operations of its customer networks. This requires obtaining and processing routing, topology, traffic and billing information and molding it in order to reach the correct level of understanding on the impact different customers might have on their providers. Any ISP interested in detecting the occurrence of this phenomena within its customer base can build the dataset and apply the proposed methodology.

1.8 Publications Arising From This Thesis

Components of this thesis have previously been published:

- A. Lutu, M. Bagnulo, J. Cid-Sueiro, and O. Maennel, *Separating wheat from chaff: Winnowing unintended prefixes using machine learning.* In: Proceedings of 33rd IEEE International Conference on Computer Communications, ser. IEEE INFOCOM 2014, April 2014.

- A. Lutu, M. Bagnulo, C. Pelsser, and O. Maennel, *Understanding the Reachability of IPv6 Limited Visibility Prefixes.* In: Passive and Active Measurement, ser. Lecture Notes in Computer Science, 2014, vol. 8362.

- A. Lutu, M. Bagnulo, and O. Maennel, *The BGP Visibility Scanner*. In: 16th IEEE International Global Internet Symposium (GI 2013), collocated with INFOCOM 2013, April 2013, Turin, Italy.

- A. Lutu, M. Bagnulo, and R. Stanojevic. *An Economic Side-Effect for Prefix Deaggregation*. In: The Seventh Workshop on the Economics of Networks, Systems and Computation (NetEcon 2012), collocated with INFOCOM 2012, March 2012, Orlando, Florida, USA.

- A. Lutu, C. Pelsser, M. Bagnulo and K. Cho, *The Aftermath of Prefix Deaggregation*. In: 25th International Teletraffic Conference (ITC25), 10-12 Sept. 2013, Shanghai, China.

- A Lutu, M Bagnulo, C. Pelsser, O. Maennel and J. Cid-Sueiro, *"The BGP Visibility Toolkit: Detecting Anomalous Internet Routing Behavior"*. IEEE/ACM Transactions on Networking. *Submitted, Major Revision due Nov. 2014*. Impact Factor: 1.986.

- A Lutu, M. Bagnulo, C. Pelsser, K. Cho and R. Stanojevic, *"An Analysis of the Economic Impact of Strategic Deaggregation"*. Computer Networks. *Major Revision Submitted, Aug. 2014*. Impact Factor[1]: 1.282.

[1]Thomson Reuters, Journal Citation Reports, 2014.

Chapter 2

Background and Related Work

The main task of the Internet is to provide connectivity to every node attached to it. In order to achieve this goal, organizations forming the Internet, identified by one or more unique Autonomous System Numbers (ASNs), must ensure that they have paths to reach all the rest of operational networks. These paths may traverse multiple ASes, making the Internet literally a network of networks. Modern interdomain routing with BGP is based on policies, rather than finding shortest paths, since it quickly became apparent that the latter would be insufficient to handle the myriad operational, economic, and political factors involved in routing [9]. BGP thus propagates routes across the Internet while respecting the policies of individual ASes [38]. Network operators express their routing preferences by influencing the path decision process independently of others. These policies are achieved by tuning a variety of parameters available in BGP. Consequently, network managers began to modify routing configurations to achieve countless goals held by the router's owner that controlled which routes were chosen and which routes were propagated to which neighbors.

Routing policies stem from a conglomerate of economic, political or performance considerations dictate how traffic flows in the Internet [39, 9, 40]. The relationship between a pair of ASes often falls into one of the following two broad categories [9, 40, 41]:

1. Customer-Provider: The customer AS financially compensates the provider AS for connectivity to the global Internet.

2. Peer-Peer: A mutually beneficial relationship between two ASes to provide connectivity to each others customers. No payment is required for traffic exchanged between the two peer ASes.

ASes with wide geographic coverage and substantial backbone infrastructure are considered large or Tier1 (e.g., AT&T, Level3 etc.). A natural Internet hierarchy exists where smaller ASes are customers of larger ASes and thus the prior depend on the latter to reach every other node connected to the Internet. In this context, multiple layers of

transit connectivity can be traversed until an end-user is reached. However, lately it has been argued that the Internet is slowly migrating from a hierarchical structure to a flat one, with an increasing density of peering links [42]. Furthermore, as the routing necessities of ASes grow more complex, also the relationship between networks fall out of the two categories defined above. In the last years, customized policies became more often than it was previously believed [43, 44].

2.1 Monitoring BGP Routing Policies Efficiency

The flexibility to set policies allowing any route to be preferred over any other has important ramifications, including unforeseen side-effects impacting other entities active in the routing system, widespread undetected misconfiguration and even conflicting configurations leading to bogus routing policies to activate. Given that ASes independently define their routing policies, it may happen that perplexing results of the interactions between ASes have been observed to cause important disruptions that affect the global routing system [1, 2]. The main reason behind this resides in the fact that the actual interdomain routing is the result of the interplay of many routing policies from ASes across the Internet, possibly bringing about a different outcome than the one expected. Consequently, in order to ensure the efficiency of their routing policies, ASes periodically control how their preferences resonate in the routing system or how others' routing policies are affecting them. To this end, most of the work related to our efforts described in Chapter 3 tackle the analysis of BGP raw data, which can be tricky and difficult. There are numerous efforts towards detecting security related routing conditions, such as prefix hijacking (e.g., PHAS [16]). Also, various tools exist to provide useful information for operators [17, 18]. Multiple operational misconfigurations have been reported [6], but attempts go far beyond this. They include *RIPE Labs* [8], which has a whole section devoted to tools that assists operators or Renesys [19] and BGPmon [20], which operate this type of services to operators for a fee.

Unlike tools which integrate a vast amount of operational problems [7], we do not focus on inferring and/or monitoring the AS-level topology of the Internet, but on monitoring the healthy deployment of routing policies through prefix visibility. In this sense our work is very closely related to the work on BGP wedgies by Griffin et al [2, 21]. However, none of those theoretical work is able to detect problematic routing conditions based on raw BGP observations. All of this work requires access to configuration files, which are typically not shared. The latter are considered a company secret which BGP was designed to hide, making it hard to be inferred [15]. While we understand the limitations of BGP protocol monitoring, we noticed that still a great deal that can be inferred. In this sense, our work aims at reporting and aggregating the information to make it usable for operators. Despite that many other similar tools [7, 8] leverage the massive amount

of available routing data, the BGP visibility scanner is, to the best of our knowledge, the only tool offering specific information on global prefix visibility.

2.2 The Machine Learning Approach for BGP

In Chapter 4 we propose a machine learning algorithm to automatically detect anomalous events which, despite their high impact on the routing policies, emerge as legitimate expressions of routing policies. We note that the proposed decision tree algorithm relies exclusively on prefix visibility features. Machine learning in the context of interdomain routing has been already proven to be a successful approach. Using traffic feature distributions, Lakhina et al. [45] show that the existence of some anomalies can be detected from traffic flows. Furthermore, a Bayesian framework has been previously proposed for detecting mistakes in the router configuration files using statistical anomalies [46]. Relying on network data, the usage of statistical algorithms has been advanced to detect deviations from the long-term profile of BGP routing updates [47]. Similarly, an instance-learning framework has been previously recommended for identifying deviations from the normal defined state of BGP routing dynamics [48]. Likewise, Li et. al advance a rule-based framework for the detection of abnormal routing behavior caused by a major worm or a blackout [49].

In Chapter 4, we design a tree-based classification model. We begin by establishing the base learning model, which we further boost for improved accuracy. Tree-based learning methods rely on iteratively partitioning the data into smaller groups of similar elements [50]. The splitting of the data is done using the features that best separate the outcomes. The key idea is to chose the splits which maximize the group homogeneity, i.e., how similar are the elements within the same group, or until the small groups are sufficiently "pure". Choosing the right number of splits is a challenge, since we can easily overfit the model by considering splits that are very specific to the training data, or, contrariwise, underfit it by considering shallow general splits. Finding the correct balance is conditioned by finding the optimal set of features used to partition the data.

We select the set of features which gives us the best classification model with the help of Receiver Operating Characteristic (ROC) curves [50]. ROC curves come from signal detection theory that was developed during World War II for the analysis of radar images. An ROC curve shows the trade-off between the true positive rate or sensitivity (proportion of positive tuples that are correctly identified) and the false-positive rate (proportion of negative tuples that are incorrectly identified as positive) for a given model. That is, given a two-class problem, it allows us to visualize the trade-off between the rate at which the model can accurately recognize yes cases versus the rate at which it mistakenly identifies no cases as yes for different portions of the test set. Any increase in the true positive rate occurs at the cost of an increase in the false-positive rate. The area under the ROC curve

(AUC) is a measure of the accuracy of the model.

Boosting [24] is one the most powerful learning mechanisms proposed in the last 20 years. The main idea behind this algorithm is to combine many weak classifiers to produce one robust classification algorithm. Algorithm 1 presents the popular AdaBoost algorithm, first proposed in [24]. Unlike other boosting algorithms, AdaBoost adjusts adaptively to the errors of the weak classifier considered. Algorithm 1 assumes the simple case of two possible outcome classes, i.e., $Y = 0, 1$ and a random sequence of training samples $(x_1, y_1), ..., (x_N, y_N)$. Boosting is then used to find a hypothesis h_f with low error rate after combining using a majority vote the output of all the weak hypotheses obtained from each of the T boosting iterations.

Algorithm 1 AdaBoost

Input:

- sequence of N labeled samples $(x_1, y_1), ..., (x_N, y_N)$
- distribution D over the N examples
- weak learning algorithm **WeakLearn**
- integer T specifying number of boosting iterations

Initialize the weight vector: $w_i^1 = D(i)$ for $i = 1, ..., N$.
Do for $t = 1, 2, ..., T$

1. Set
$$p^t = \frac{w^t}{\sum_{i=1}^{N} w_i^t}$$
2. Call **WeakLearn**, providing it with the distribution p^t; get back a hypothesis $h_t : X \to [0, 1]$
3. Calculate the error of $h_t : \epsilon_t = \sum_{i=1}^{N} p_i^t |h_t(x_i) - y_i|$
4. Set $\beta_t = \epsilon_t / (1 - \epsilon_t)$
5. Set the new weights vector to be
$$w_i^{t+1} = w_i^t \beta_t^{1-|h_t(x_i)-y_i|}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1, & \text{if } \sum_{t=1}^{T} \left(log(\frac{1}{\beta_t}) \right) h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} log(\frac{1}{\beta_t}) \\ 0, & \text{otherwise} \end{cases}$$

2.3 Measuring Prefix Reachability

In Chapter 5, we perform an extensive analysis of how BGP route propagation affects global reachability. We focus on characterizing the reachability status of the limited

visibility address space both for IPv4 and IPv6.

Reachability is the main service provided by the Internet, thus attracting much interest from the research community, which generated numerous studies on the matter. Many have tried to understand and characterize pathological behaviors at the interdomain level which endanger this basic service, e.g., bogon advertisements [51], hijacking [52] or misconfigurations [53]. Several others have been concerned with establishing the correlation between end-to-end Internet path performance degradation and routing dynamics [54,55]. There is also a large amount of studies concentrated on BGP and its intricacies which may impact prefix reachability, e.g., slow BGP convergence [56], routing instabilities [57], or complex policies interactions [2]. Few other works have been concerned with proposing tools to asses and identifying problems in the reachability of prefixes in the Internet [58]. In particular, the efforts of Mao et. al in [59] are somewhat on the same lines as the goal we try to achieve with our methodology.

It has been previously confirmed that, when it comes to establishing the "real" reachability in the Internet, performing data-plane measurements should take precedence over the control-plane measurements [60]. Few tools are available for testing Internet reachability today, out of which *ping* and *traceroute* are the most popular. Despite their wide usage, these tools suffer from limitations which have been long known within the research community [61, 62]. Ping can be filtered by firewalls and NATs. In addition, ping probes between any two hosts, cannot confirm that reachability is bug-free [62]. If some routers do not propagate routes for part of the address space, routing will try to find paths around the problematic regions. Dynamics in the routing paths or load-balancing are only partially visible by traceroute [61]. Additionally, it has been found that ICMP traceroute is the most efficient traceroute probing method [60, 63, 64], which is further consistent with our own results.

In this chapter, we found especially difficult to test the reachability of the IPv6 address space which, as opposed to the IPv4 address space, does not account with numerous connected hosts. More specifically, we turn our attention to the IPv6 dark prefixes and attempt to measure their reachability at the interdomain level, using as a benchmark the status of the IPv4 Internet. During the past years, IPv6 has received a lot of attention both from the operational and the research community. Various related works look into the transition of the IPv4 network infrastructure to IPv6 [65] and how the Internet topology, routing and performance across the two compares [66]. It had been proven that the routing dynamics in the IPv6 topology are largely similar to those know from IPv4, even if the degree of IPv6 deployment is still far behind the IPv4 expansion [66].

2.4 Prefix Deaggregation

In a more tangled Internet, both economically and technically, individual network managers need to permanently adapt to the interdomain routing changes and, by engineering the Internet traffic, optimize the use of their network. For operational reasons, the majority of the Internet entities has expressed the necessity to better control the incoming and/or outgoing flows of Internet data traffic. Internet traffic engineering [67,68,9] is the aspect of network engineering dealing with issues of performance optimization of the IP networks [68]. Interdomain traffic engineering requirements are diverse and depend on the connectivity of the AS with others and on the type of business handled by the network [9]. One important task achieved through the use of traffic engineering tools is the control and optimization of the routing function in order to allow the ASes to shift the traffic inside and outside their network in the most effective way.

Traffic engineering with BGP is performed through a series of methods which allows for a better control on the traffic flow both inside and outside a network [], providing operators with the means to optimize the use of their networks. Within the traffic engineering toolbox, the injection of artificially deaggregated prefixes through BGP offers a fine-grained method to control the interdomain ingress traffic. When combined with routing policies in the form of selective advertisements [69], prefix deaggregation enables operators to control how the traffic enters their network, which is one of the most challenging task to be achieved in traffic engineering with BGP. Prefix deaggregation is recognized as a steady long-lived phenomenon at the interdomain level, despite the negative overtone surrounding this approach [11,25]. The most important negative side-effect of the widespread adoption of this technique is the artificial inflation of the BGP routing table, which can affect the scalability of the global routing system. This issue has become an important concern of the entire Internet community over the past years [10]. From this perspective, this type of behaviour is considered to be harmful, as it heavily impacts the global routing table and it acts counter to the goals of the Classless Inter Domain Routing (CIDR) architecture, which encourages aggressive address aggregation.

Aside from the above-mentioned traffic-engineering benefits, there are also a number of collateral benefits resulting from the use of deaggregation in the Internet. For example, one alleged benefit is the increased security of the network announcing more specifics in the interdomain. Some claim that prefix deaggregation can be useful as a technique for reducing prefix-hijacking network attacks [26]. In Chapter 6, we study the impact strategic deaggregation has on the transit traffic bill of the networks originating the more-specific prefixes. We show that, given that fact that provider ASes have the incentives to respect the routing advertisements of their customers [38], the deaggregating AS may enjoy a decreased monthly transit bill. In Chapter 7, we further propose and test a methodology to detect such scenarios in the real-world Internet.

Chapter 3

The BGP Visibility Scanner

Problematic routing conditions and complex interactions between policies in the Internet have been predicted manifold [2]. However, to detect them it is required that ISPs make their configurations public, which appears to be unlikely in today's Internet. In this chapter, we investigate to what extent it is possible to discover the match between the *intended result* of applying routing policies and the *actual result* reflected in the global routing system. Just by using public routing data, we present a methodology that scans raw BGP routing tables, filters and analyzes them, so that we can extract potential problematic policy configurations or, contrariwise, verify that the intended ones took the expected effect.

We propose the ***BGP Visibility Scanner*** which allows network operators to validate the correct implementation of their routing policies, by corroborating the BGP routing information from approximatively 130 independent observation points publicly available at the moment of the study in the interdomain. The Visibility Scanner is public and can be accessed at **visibility.it.uc3m.es**. The tool allows networks to check how their own routes are being propagated in the Internet, verify the results of the implemented routing policies and identify possible cases where these policies backfired. By merging all the information retrieved from the ASes enabled as active monitors in the RIPE RIS [22] and RouteView [23] Projects, we create a *visibility scanner* for all the IPv4 prefixes advertised in the Internet. The tool is subject to the limitations of the available public looking-glasses, which we further address accordingly. It is important to note that the properties of BGP do not allow us to get a complete picture on all policies, but nevertheless those public observations points provide a multi-angle perspective on the interdomain routing. Moreover, our tool has already proven its capability of triggering visibility alarms and helping networks deal with the problems caused by their own routing policies.

We focus our analysis on a particular expression of routing policy interaction, namely the interdomain route propagation process and the manner it is reflected in the interdomain global routing tables. We define the **Limited-Visibility Prefixes (LVPs)** as being

stable long-lived Internet routes that are not present in all the global routing tables analyzed, but seen by at least two ASes. Contrariwise, we also define the **High-Visibility Prefixes (HVPs)** as the set of prefixes that are propagated within almost all the available full routing feeds. We note that *the limited visibility does not necessarily imply limited reachability.* There could be a *HV* less-specific prefix that provides reachability. In this sense, we also identify a set of so-called **Dark Prefixes (DPs)**, which represents a subset of the *LVPs* that are not covered by any *HV* less-specific prefix. These prefixes represent address space that, in the absence of a default route, may not be globally reachable. We throughly investigate this issues in Chapter 5.

There are several reasons behind the existence of *LVPs*, which can be classified as follows:

- *Intentional/Deliberate.* Some ASes create *LVPs* on purpose. There are several ways this can be done, including scoped advertisements (e.g. geographically scoped prefixes to offer connectivity only to networks located in a certain region) or advertisements only through (some) peering and not transit relationships.

- *Unintentional/Accidental.* In many cases, *LVPs* are the result of errors in the configuration of filters of the origin or other ASes that have received the prefix announcement. Additionally, some prefixes are announced by the origin ASes with the intention of being globally distributed, but some of the ASes receiving the prefix decide to filter them. A notable example of this is filtering by prefix length.

We perform a differential analysis of routing tables to retrieve *LVPs* on a daily basis and we make the results of our study available on-line, thus creating the possibility for a close-to-real-time verification of the effectiveness of eventual modifications in the implemented routing preferences. We integrate in our analysis previously "cleaned" routing information, after the elimination of routes that do not represents an expression of routing policies, but of other network-specific operations, e.g., internal routes visible in only one monitor, converging routes, MOAS prefixes, bogon prefixes etc.

We merge this information and we use it to craft the *BGP Visibility Scanner*. The resulting tool further allows the validation of individual routing policies efficacy in the interdomain by checking the external visibility of the injected routes. The tool has been active and available for the operational community since November 2012, allowing any network operator to check the visibility status of their prefixes. At the time of writing, the visibility scanner detects on a daily basis approximatively 95,000 IPv4 and IPv6 *LVPs*. The daily set of prefixes with limited visibility can further be queried using the BGP Visibility Scanner public web-page. Additionally, we collect feedback on the intended visibility status of the LVPs from the operators of the networks originating the prefixes and which are actively using our tool. This further enables us to verify if the intention

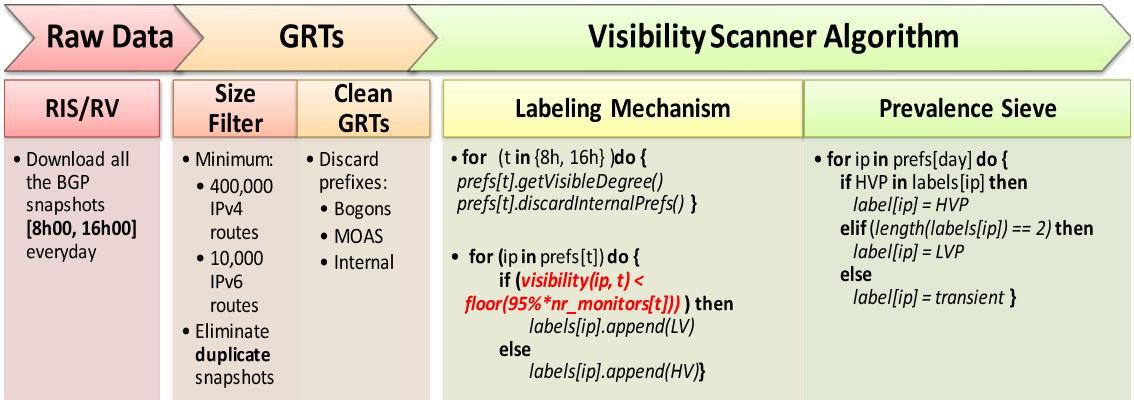

Figure 3.1: Methodology used for determining the LVPs and the HVPs.

of the origin network is reflected in the observed visibility status of its prefixes and to gather ground-truth on the various causes for LVPs.

3.1 Methodology

We collect the routing information from the two major publicly available repositories at RouteViews and RIPE RIS. The two repositories gather BGP data throughout the world, currently deploying 24 different collection points, which we further refer to as *collectors*. The collectors periodically receive BGP routing table *snapshots*, i.e. one time instance of a routing table, from approximatively 500 active monitors. A *monitor* represents a network peering with the public RIS/RouteViews repositories and periodically propagating its routing information.

We depict in Figure 3.1 all the required steps for processing the raw data into the *LV* and *HV* prefix sets, starting with the above-described data collection process until applying the visibility algorithm. We next expand on each of the different processing phases.

3.1.1 Refining the Raw Routing Data

We focus here on the second processing block of the flow chart in Figure 3.1, and we look at the steps we take to obtain the set of global routing tables (GRTs).

Conceptually, the so-called *Default Free Zone (DFZ)* represents the set of BGP-speaking routers that do not need a default route to forward packets towards any destination in the Internet. The routing table maintained in one of the DFZ routers is commonly known as the *global routing table*. Realistically speaking though, due to the current operational status of the Internet routing, such a GRT of the BGP routing is an idealized concept. However, Internet Service Providers (ISPs) do maintain their own version of the *global routing table*, which is propagated to customer networks upon request. In order to

provide the accurate multi-angled perspective on interdomain routing state, the data we are looking for is the long-term expression of external routing policies which is reflected in the GRTs.

For the purpose of this thesis, we loosely define the GRT as the entire routing table provided by a DFZ network to its customers requesting a full routing feed. This is not a formal definition, but it properly captures the main idea behind the type of data required for our study. Given that the GRTs are not accessible as such, we rely on the publicly available raw BGP routing tables, which we refine and mold in order to isolate just the information we require. We can identify the sets of HV and LV prefixes only by comparing the GRTs from the active monitors. However, the monitors have different policies with respect to the public routing repositories, thus providing different types of routing feeds. We are able to identify three different types of feeds injected to collectors, namely:

- *Partial Routing Tables:* this type of feed can be though as the result of establishing a peering-like business relationship between the monitor and the collector. This routing feeds are of no use for our study, since the implicit lack of information is the result of a deliberate propagation choice from the monitor, and not the reflection of the routing policies of the route originating AS. By definition, these feeds are not GRTs and we further discard them from our analysis.

- *Global Routing Tables:* full routing feeds from the monitors. This is the main raw information that we want to further process and use for our analysis.

- *Global Routing Table, including internal routes:* in some cases, it may happen that the monitor announces, aside from the complete routing table, other additional internal information. This additional information is again of no interest for our study, since we do not focus on the internal operations of a network. Consequently, we need to identify and filter out these particular routes.

3.1.1.1 Filter routing tables based on minimum size restriction

In order to identify the feeds which constitute a GRT, the primary characteristic of the routing feeds on which we focus is the actual size of the routing table snapshot. Based on the BGP Analysis Report [70], we consider that an IPv4 global routing table from a monitor should have no less than 400,000 routing entries [70]. Similarly, an IPv6 global routing table should not contain less than 10,000 paths. Consequently, we check over 500 routing feeds collected from the two repositories, and we discard all the BGP feeds that have less than the imposed lower-limit of prefixes.

In order to further verify the results of this heuristic, we verify the routing policy of the collector storing the routing information and the routing policy of the monitor offering its routing feed towards the collector. In particular, within the RIS project, for

each collector it is specified the number of so-called *full routing feeds*, which is consistent with the number of tables with more than *400,000* IPv4 entries or *10,000* IPv6 entries. Moreover, for collector rrc00 located in Amsterdam, it is explicitly mentioned that it gathers *default free routing updates from its active peers*. This hints the fact that the *complete routing feed* is the GRT, as defined for the purpose of this thesis. All this information comes to strengthen the assumption that the size of the *Global Routing Table* should always be above the considered threshold.

Furthermore, we check in the `whois` database the public routing policies for the ASes peering with the two public repositories. We are able to retrieve information for 34 monitors feeding a routing table with more than *400,000* entries. We see that for 18 of them the public routing policy is *ANNOUNCE ANY*. This further confirms the assumption that the full routing feeds received by the collectors are actually consistent with propagating all the available routes maintained by the monitor. The rest 16 monitors are advertising the policy *ANNOUNCE AS_name*, which is not clearly defined.

In order to address the limitations of using the `whois` database, we check the publicly available topology maps [71] to infer the business relationship of the monitors towards the repositories. Consequently, we are able to check if the relationship with the repository is a *provider-to-customer* (p2c) relationship, meaning that the monitor may be exporting its complete routing table. We are however not able to verify this for AS6447 of the RouteViews Project. For AS12654 RIPE RIS project, we were able to validate 21 such relationships.

After applying the size filter to all the raw feeds, we are left only with the Complete Routing Tables and the Complete Routing Tables with internal information. We later deal with the additional refinements in order to eliminate the surplus data consisting on internal paths possibly included in these first results.

3.1.1.2 Eliminate duplicate routing feeds

One unique AS may peer with several of the repositories from RIS/RouteViews in different geographic points, thus propagating more than one routing feed to different collectors. After checking the content of duplicate feeds from the same AS and comparing them, we find that these multiple routing table snapshots are very little difference or are even identiacal. Since analyzing the routing feeds including duplicates may trigger the generation of false positive *HVPs*, we only keep one unique instance of the routing table snapshots.

3.1.2 Sanitary Checks

After applying the previously described heuristics, we are ultimately left with the GRTs, which we further use in our analysis. We perform a couple of “sanitary“ checks

on the data contained in the GRTs, in order to further discard the information that is of no interest for our study, e.g., internal paths advertised by monitors to the collectors, bogons etc. Hence, we apply the *bogon filter* and the *MOAS filter* on all the GRTs, as depicted in the third step in Figure 3.1.

3.1.2.1 Eliminate the bogon and martian routes from GRTs

Although we are able to identify the interest routing tables by following the previous steps, not all the content within the routing feeds represents legitimate routes. Bogon prefixes are a class of routes that should never appear in the Internet. Bogons are defined as *Martians*, representing private and reserved address space or *Fullbogons*, which include the IP space that has been allocated to a Regional Internet Registry (RIR), but has not been assigned by that RIR to an actual Internet Service Provider (ISP) or other end-user. Internet Assigned Numbers Authority maintains a convenient IPv4 and IPv6 summary page listing allocated and reserved netblocks, and each RIR maintains a list of all prefixes that they have assigned to end-users. The bogon reference pages maintained by Team Cymru within the Bogon Reference project [72] include information and resources to assist those who wish to properly filter bogon prefixes within their networks. We use the periodically updated filters from The Bogon Reference [72] in order to make sure that we eliminate any possible bogon route included in the GRTs.

Not consider the MOAS prefixes. We continue the sanitary checks by focusing this time on Multiple-Originating AS (MOAS) prefixes [73]. The Multiple-Originating AS (MOAS) [73] prefixes cannot be qualified within our study, since for these prefixes we are not able to identify which origin AS might be suffering/generating the reduced visibility of its prefixes. We plan to address this issue in the future work. Therefore, at the time of the study, we identify approximatively 4,500 MOAS prefixes. Due to the relatively small number of prefixes in this category , we proceed to their elimination, ensuring a minimum impact on our results.

3.1.3 The Visibility Scanner Algorithm: the Labeling Mechanism

We say that a prefix is *visible* to an AS if the latter has a stable active route in its BGP *Global Routing Table (GRT)* for the prefix in question. A prefix is *globally visible* when almost all the ASes in the Internet have an active stable route for it. In this paper, we show that the lack of global prefix visibility can offer early warning signs for anomalous events that, despite their impact, often remain hidden from state of the art tools, e.g., [7, 8]. Additionally, we show that such unintended Internet behavior not only degrades the efficacy of the routing policies implemented by operators, but can also point out problems in the global connectivity of prefixes. In order to evaluate the global visibility of a prefix, we compare the content of the GRTs from the ASes that make their

routing data public.

Having obtained the "clean" version of the GRTs, we proceed to applying the **Visibility Scanner Algorithm** for identifying prefixes with stable limited visibility in the interdomain. At this point it is important to filter out the cases of limited interdomain visibility caused by other factors unrelated to routing policies, e.g. BGP convergence or leaking internal routes to the collector. In order to avoid the problem of internal paths leaking towards the collectors, we remove all the routes learned from only one monitor which is also the route originating AS.

In order to address the confusion caused by converging prefixes emerging as false positive limited visibility prefixes in our results, we analyze two 8-hours apart routing table snapshots and the corresponding per-prefix visibility information. We use the two different timescales for data processing in order to ensure the correct separation of the external long-term expressions of the routing policies implemented by an individual AS, and filtering out converging routes or routes with temporary limited visibility due to internal operational activities. We monitor the propagation of routes, evaluate the *visibility degree* at every sampling moment and assign *visibility labels* based on our results. We define the *visibility degree* as the number of GRTs within the sample that contain (i.e. see) a certain prefix, and the *visibility label* as the visibility status of each prefix, i.e. *LV* for Limited Visibility and *HV* for High Visibility. The visibility scanner algorithm is composed of the forth and fifth steps of the processing flow depicted in Figure 3.1, which we call *prevalence sieves*.

The Labeling Mechanism: assigning prefix visibility labels. Based on the visibility degree of the prefixes at each of the two sampling moments (i.e. 08h00 and 16h00), we assign a *visibility labels* at each sampling moment to every prefix discovered at each moment, as described in 3.1. Though we are now working with "*clean*" routing feeds, these routing tables may still contain some internal routes leaked by the originating AS. As mentioned before, these routes are not of interest for our study. Hence, we proceed to eliminate them after checking the per-prefix *AS-Path* information and the visibility degree (i.e. the number of monitors which see the prefix) in each of the two snapshot. We begin by checking in every snapshot during a day the per-prefix visibility degree and the AS-Path length. Thus, we discard all the prefixes for which we identify a visibility degree of 1 at any time (i.e. the prefix is seen only by one monitor) and an `AS-Path` length of 1.

After this step, me move on to assigning the *visibility labels* for each prefix, according with the *prevalence sieve* explained next. We use a 95% *minimum visibility rule* in order to assign the labels according with the observed visibility degrees. Consequently, *we define Limited Visibility Prefixes as prefixes present in less than* 95% *of the active monitors at a sampling time.* Otherwise, the prefixes complying with the 95% minimum visibility rule are defined as High Visibility Prefixes. Ideally, a *HVP* should be contained in absolutely all the routing tables contained in the sample. The choice of the 95% allows for a 5%

error in the sampling also accommodating possible glitches that may appear in the data. Moreover, according to our threshold sensitivity analysis, we find that the set of *LVPs* is not highly sensitive to the values of the prevalence sieve threshold. We expand on this in section 3.2.1.

Visibility Label Prevalence Sieve. Eliminating Converging Prefixes. The visibility label *prevalence sieve* accounts for the dynamics of a prefix in time, as presented in the last block from Figure 3.1. After applying the *prevalence rule* integrated in the sieve, we decide the per-day label for the prefix. The high visibility of a prefix in at least one monitor sample hints the fact that the route could reach all the observed ASes. Should this change during the analyzed time, it might be a cause of, for example, topology changes or failures. Therefore, we consider that *the HV label always prevails*, i.e. if a prefix is tagged as *HV* in one of the samples, it is tagged as *HV* in the final set.

Otherwise, when no *HV* label is tagged, we analyze the cases of *LV* prefixes emerging in our results. If a prefix is tagged as *LVP* only once in the two sampling times, it might be a symptom of a prefix being withdrawn or, contrariwise, in the process of converging after just being injected. Having a single *LV* label means that the prefix is not present in the other sample, i.e. the prefix is no longer present in any of the routing tables, and there can be several explanations for this, including the prefix being withdrawn. In any case, these particular routes cannot be qualified within our study, thus we filter out any prefix with only one label in a day (and that label being *LV*). This helps us to eliminate routes that are not an expression of the routing policies, but a second-effect of other Internet operations. The only case where we can say a prefix has limited visibility and mark it accordingly, is when both labels assigned at each sampling time are *LV*.

3.1.4 Identifying Dark Prefixes

Once we have identified the two main sets of prefixes, i.e. the *LVPs* and the *HVPs*, we move on to verifying the reachability of the LVPs in order to identify possible cases of reduced reachability. Consequently, for each of the prefix in the LVP category, we build the covering trie of less specific HV prefixes, from which we ultimately retrieve its root prefix (i.e. the smallest covering HV prefix). In the eventuality of not identifying any such globally visible less-specific prefix, we mark the LV prefix as *Dark* and continue our analysis.

3.2 The *LVPs* in Rough Numbers

The set of LV prefixes identified using the BGP Visibility Scanner methodology is made publicly available, so that each network can check the status of its prefixes. The results are refreshed on a daily basis such that the operators can have an updated view on the efficiency of their routing policies, both in IPv4 and IPv6. We present next a few

statistics regarding the number of both IPv4 and IPv6 LVPs, as observed during the first 15 months (i.e., November 2012 until January 2014) when the BGP Visibility Scanner has been active. This allows us to understand the magnitute of the limited visibility phenomenon in the Internet.

Every day we collect more than *500* routing feeds, for each of the two different sampling moments. After applying the *cleansing process*, we distinguish, in average, *150 IPv4 GRTs* injected to the public repositories by unique ASes. We then compare the content of the 150 GRTs in order to identify the IPv4 LVPs. In rough numbers, the daily total number of prefixes is around *550,000 prefixes*. Out of these, around *10,000 IPv4 prefixes* are singled out as internal routes and, consequently, discarded from our analysis. Furthermore, we remove the converging routes that may emerge as limited visibility in the visibility scanner. This incurs the elimination of about *8,000* additional IPv4 prefixes in average. For the remaining prefixes we continue our visibility analysis and assign LV/HV visibility tags. We subsequently identify about *90,000 prefixes* in average that are tagged *LVP* and around *420,000* prefixes marked *HVP*. When checking how the two sets of prefixes overlap, we find that there are more than *2,500* IPv4 LV prefixes without a covering high-visibility prefix, which we mark *DP*. We have observed more than 3,800 ASes which inject limited visibility prefixes, out of which less than 1,000 ASes originate DPs.

The size of the IPv6 GRT is much smaller than the one for IPv4. We adjust the size filter applied in the parsing phase of the raw data to be at least 10,000 routes. Also, we use the bogon filters corresponding to IPv6. Consequently, we identify and compare the content of 110 GRTs to determine the set of IPv6 limited visibility prefixes. The daily overall total number of prefixes is approximatively *16,500 prefixes*. Out of these, on average *150 IPv6 prefixes* are discarded as internal routes advertised to the collector. We further eliminate the identified converging routes, i.e., approximatively *10* additional prefixes in average. On average, *3,500 IPv6 prefixes* are tagged *LVP* and approximatively *12,500* IPv6 prefixes are marked *HVP*. Therefore, 20% of all the IPv6 prefixes identified from all the analyzed routing tables are LVPs. This is consistent with the result for the IPv4 LVPs, where out of all the prefixes learned, 20% have limited visibility [12]. This can further be observed in Figure 3.2, showing the empirical CDF of the prefix visibility within the sample of global routing tables, both for IPv4 and IPv6.

When checking how the LVPs and HVPs overlap, we find that, for IPv6, there are more than *500* LV prefixes without a covering HVP. We label the latter as *DPs*. This represents approximatively 14% of the whole set of v6LVPs and 3.75% of the v6HVP set. When comparing with the situation in IPv4, where in average only 3% of the LVPs (and 0.6% of the HVPs) are marked as *dark*, we find that we have almost 5 times more IPv6 dark address space. This is relevant because these prefixes may have limited reachability, in the lack of a default route. We further analyze the correlation between the limited visibility and the limited reachability of the LVPs in Chapter 5. We observe that more

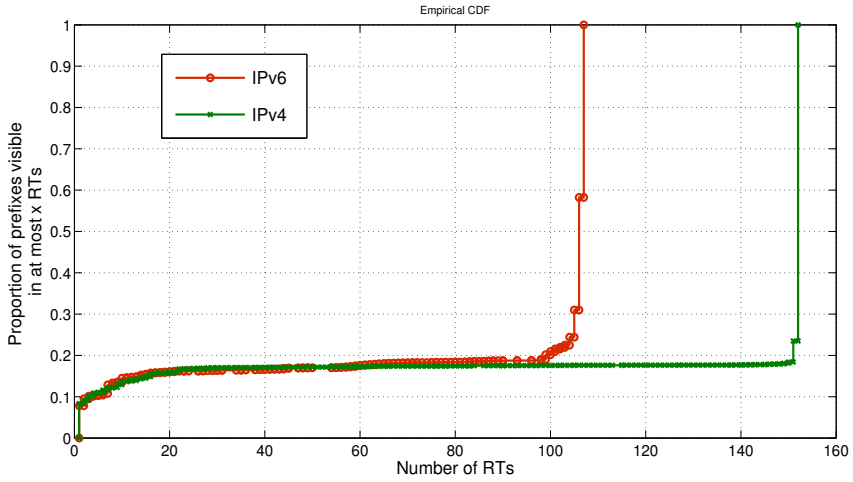

Figure 3.2: The Empirical CDF for prefix visibility within the sample of Global Routing Tables (RTs), both for IPv4 and IPv6.

than 13% of all IPv6 active ASes inject LVPs and approximatively 5% of all IPv6 active ASes originate DPs. In IPv4, we see that 9% of all ASes originate LVPs and only 2% are also injecting DPs. This result also further hints the early stages of development of the IPv6 architecture, previously established in [66]. These numbers may vary from day to day, given that neither the monitors providing their global routing tables, nor the actual content of the GRTs are constant over time.

3.2.1 Characteristics of the Prefix Visibility Categories

We have previously defined the LVP set using a 95% prevalence rule. It is important to understand which is the sensitivity of the threshold to the actual data conditions. We represent in Figure 3.3 the distribution of prefixes on the possible degrees of visibility for the sample of data from October 2012. We note that by varying the prevalence threshold value, the size of the two prefix sets does not suffer important changes (e.g. after changing the minimum threshold to 90%, only approximatively 800 prefixes are added to the HVP set). Also, the concentration of prefixes in the extremes values of the visibility degree hints the fact that by increasing the number of routing feeds in the sample of GRTs, the number of identified *LVPs* should also increase.

When comparing the three sets of data identified for IPv4, i.e. *LVP, HVP and DP*, we first observe that the limited visibility issue appears for prefixes of various lengths, from /5 to /32, as depicted in Figure 3.4.

However, we do note the lack of prefixes more-specific than /24 in the HVP set, which is consistent with the best recommended BGP practices. Thus, one cause for the restricted visibility of the prefixes with a prefix mask longer that /24 might be the failure

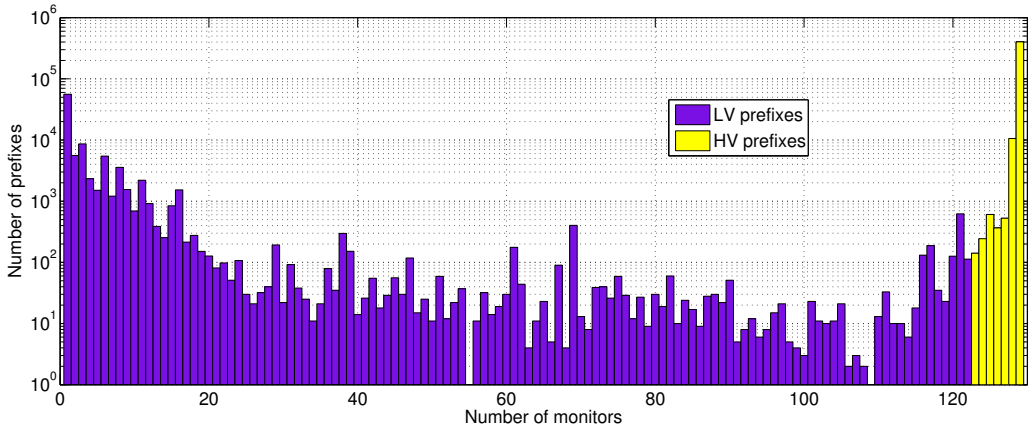

Figure 3.3: **Distribution of IPv4 prefixes on visibility degree.**

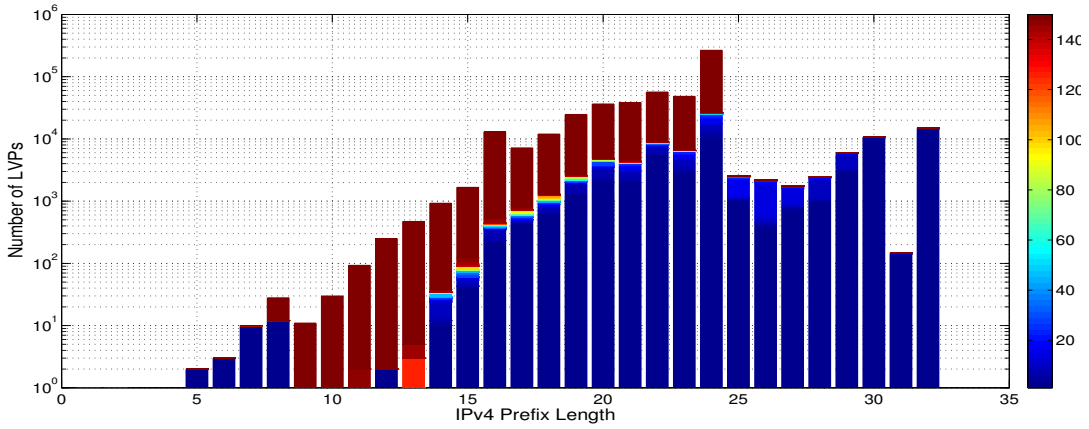

Figure 3.4: **IPv4 Prefix length visibility:** each bar shows the number of prefixes with a certain mask length. The color code represents the visibility distribution of the prefixes within each prefix-length category, according with the visibility degrees marked in the color legend in the right part of the plot.

of some networks to filter routes more-specific than /24. Also, due to the fact that prefix length filters may be asymmetric or even missing in some cases, this type of interaction might derive in a generator of *LVPs*. We also observe the presence of *LV* prefixes less specific than /8, which, due to their small degree of visibility, may be accidentally leaked in the Internet.

Similarly, Figure 3.5 depicts the distribution of IPv6 limited visibility prefixes (v6LVPs) per prefix length, color-coded to match the visibility degree of the prefixes in question. All the prefixes with a length longer than /48 are labeled as v6LVPs by the BGP Visibility Scanner i.e. /48's do not propagate globally in the IPv6 routing system. This is consistent with the status in IPv4, where every prefix more-specific than /24 is labeled LVP.

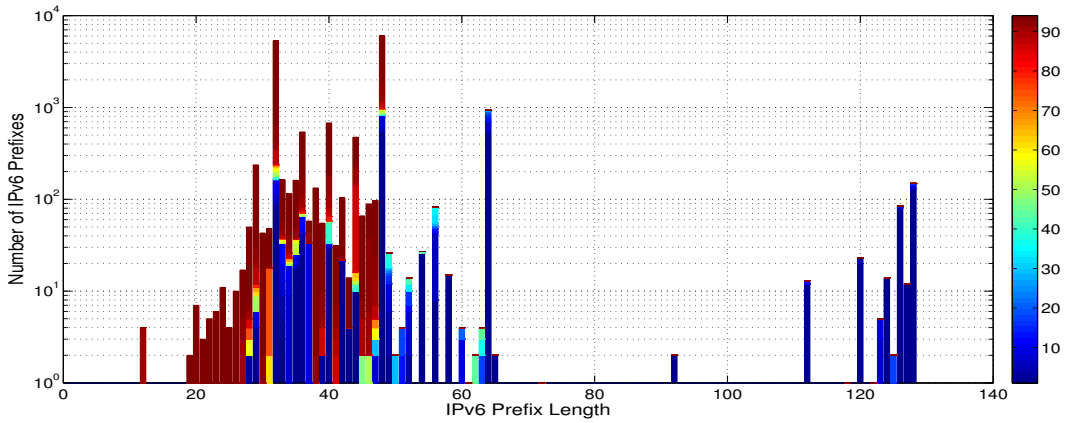

Figure 3.5: **IPv6 Prefix length visibility:** the bars are color-coded to show the visibility degree of the prefixes: from dark blue for LV, going to dark red for HV.

Moreover, when we check the average `AS-Path` length of the *LVPs*, we observe a straightforward difference between the mean `AS-Path` length for *HVPs* of 4 and the mean `AS-Path` length for *LVPs* of 3. This is easily observed from the probability distribution function (PDF) in Figure 3.6. This information shows a more limited realm of expansion for the *LVPs* than for the general *HVPs*, restraining it closer to the prefix originating network.

After applying the methodology every day during October 2012, we are then able to perform a stability analysis of the visibility label assigned to the *LVPs* identified using the BGP Visibility Scanner. In Figure 3.7 we can observe the evolution during the whole month of the number of *LVPs* and *DPs* resulting from detecting the *LVPs* only in one day and from detecting the *LVPs* that were stable during the last 7 days. For the latter, we merely compare the labels tagged on the prefixes discovered during the latest seven days prior to the moment of analysis. Just like in the prevalence sieve in Section 3.1.3, the *HV* label always prevails and we mark as *HV* any prefix with such a label. We discard any prefix with a number of labels lower than 5, i.e. which has been missing from the routing tables for more than 2 days. We assume a prefix is *stable-LV* only when it has at least 5 *LV* labels, no *DP* label and no *HV* label. Also, if a prefix happened to be labeled at some point during the 7 days as *DP*, it is sufficient to show the potential connectivity problems of that particular prefix and we further mark it *DP*. The number of *stable-LVPs* detected based on the latest seven days from the moment of analysis is not much smaller than the per-day number of *LVPs*, despite the implicit removal of possible false-positives and thus pointing to the fact that *LVPs* are a long lived phenomenon.

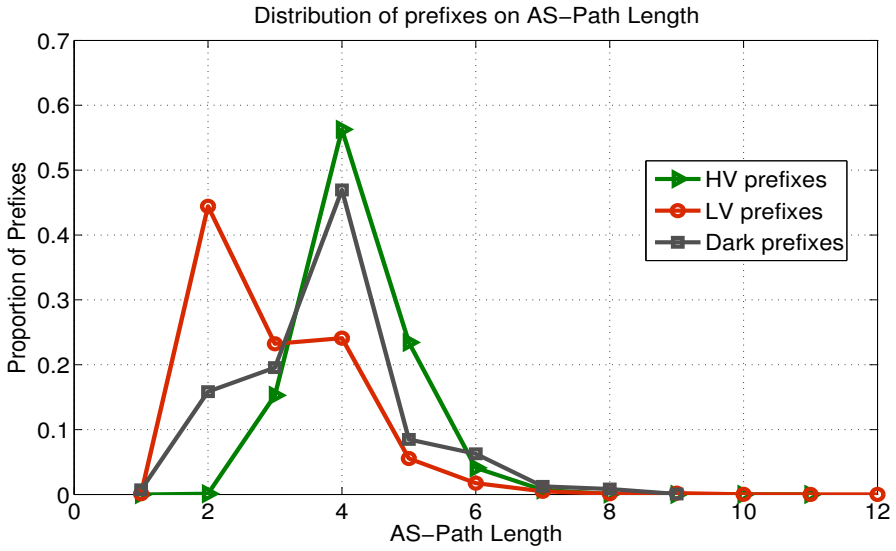

Figure 3.6: Empirical PDF for the distribution of the BGP `AS-Path` length for IPv4 prefixes with the three different degrees of visibility: HVPs, LVPs and DPs.

3.3 Ground-Truth: Understanding *LVPs* through Operational Use Cases

The daily set of prefix visibility data can be accessed by querying the BGP Visibility Scanner on a per-origin AS basis. Since the tool first became publicly available, it gathered over 5,000 queries performed for more than 1,200 different origin ASes. We have invited the users actively performing queries on the BGP Visibility Scanner to participate in a survey regarding the status of the retrieved LVPs for the corresponding origin AS. Leveraging the feedback received, we build a unique ground-truth dataset including 20,000 *LVPs*. For each of these prefixes, the network operators reported which was the expected visibility status of the prefixes after defining their interdomain routing policies. We match the origin's intention with the observed visibility status of the prefixes identified with the BGP Visibility Scanner, and separate the *LVPs* in two pre-determined classes: *intended* and *unintended*. As a results, we identify 1,150 prefixes of the class *intended* and a staggering 18,850 *LVPs* of the class *unintended*. We note that the ground-truth dataset exhibits an important disproportion between the two defined classes. The *class imbalance problem* is a well-know issue in the machine-learning domain and it is characteristic to many other real-life applications.

After analyzing the feedback received on 20,000 operational LVPs out of the 90,000 identified on a daily average, we learn about a significant variety of factors which lead to prefixes with limited visibility. We develop next on a few relevant operational examples.

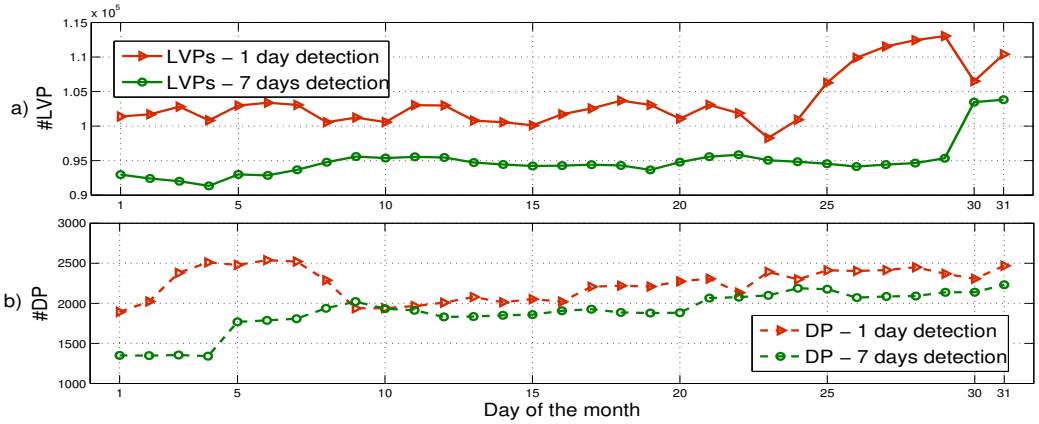

Figure 3.7: Comparing the visibility status determined for one-day and the visibility status calculated throughout a period of seven-days for IPv4 prefixes.

3.3.1 Intended LVPs

Some ASes create *LVPs* on purpose. There are several ways this can be done, including scoped advertisements of some prefixes (e.g. geographically scoped prefixes to offer connectivity only to networks located in a certain region) or advertisements only through (some) peering and not transit paths. We next provide real cases of ASes deliberately restricting the propagation of their prefixes. For example, using the BGP Visibility Scanner, we were able to verify and validate the routing policies of two of the Internet DNS root-servers. For each root-server we have identified the presence of one more-specific *LV* prefix, which is meant for providing connectivity only to direct peers and, consequently, is tagged with the well-known *NO-EXPORT* community. The limited visibility of the more-specific prefix correctly reflects the impact of the *NO-EXPORT* community on the connectivity of the prefix. However, the *LV* prefix has global reachability due to the presence of *HV* less-specific prefixes, which is used by the root-servers in order to avoid connectivity issues.

In another case, the tool also validated the routing policy of a large content provider which deliberately limits the visibility of one of its prefixes in order to ensure that the incoming traffic is fed only through a geographically-specific local path.

3.3.2 Unintended LVPs

The second type of use cases we present captures unintended results of routing policies, i.e., accidental misconfigurations or unforeseen interactions between external routing policies at the interdomain level.

1) Misconfigurations/Accidental Errors:

In many cases, *LVPs* are the result of errors in the configuration of filters of the origin

or other ASes that have received the prefix announcement. For example, a large and widely-spread ISP learned that a large set of prefixes with limited visibility were leaking through some of its direct peers. After further investigation, the ISP was able to identify the misconfiguration of its outbound prefix-filters, which should have otherwise ensured that those prefixes were not being advertised to other networks. After correcting these issues, the origin AS successfully eliminated *4,000 unintended LVPs* of whose existence it was previously unaware even if they were affecting the receiver's routing policies for these prefixes. We note that these misconfigurations remained undetected for a very long time, given that we were able to detect the LVPs in question with more than 6 months before the BGP Visibility Scanner became operational.

2) Inflicted by Third-parties:

The interactions with legitimate and correctly defined routing policies of third-party ASes can limit the visibility of some prefixes at the interdomain level. For example, in the case of two different networks with correctly defined routing policies, the operators reported that the limited visibility of their prefixes is due to the impact of the filtering policies deployed by third-party ASes. More exactly, since the *LVPs* detected did not have an object defined in the Regional Registry's database, they were discarded by the ASes filtering based on the information retrieved from the registry databases. This, consequently, caused the prefix to suffer from unintended limited visibility at the interdomain level. Thanks to the BGP Visibility Scanner, the origin networks have discovered and solved the issue.

A clear example of the serious impact that these type of undetected mistakes might have on the origin networks is the case of an ISP whose prefixes were labeled by the Visibility Scanner as long-lived *dark prefixes*. This not only means that the prefixes were not globally propagated, but might have been suffering from limited reachability in the Internet. After investigating this issue, the origin AS found that, due to a mistake in the configurations of its transit provider, the prefixes were not being correctly advertised.

We have previously stated that the anomalous events included in the unintended *LVPs* dataset remain undetected for a long time. For example, for the majority of the operational cases of unintended *LVPs* above-explained, we have observed that the prefixes were originated long time before the BGP Visibility Scanner tool was used to verify their existence. In order to support our statement on the average lifetime of a limited-visibility prefix, we use the BGP Visibility Scanner to generate the daily set of *LVPs* across a period of 11 months, from the beginning of June 2012 until the end of April 2013. This enables us to capture the dynamics of the *LVP* within this period. Figure 3.8 depicts the empirical CDF of the *LVPs* known to be unintended on the time they were active within this 11 months period.

This includes prefixes which were "born" before June 2012 and prefixes which remained "alive" after April 2013. For example, the number of LV prefixes already active on June

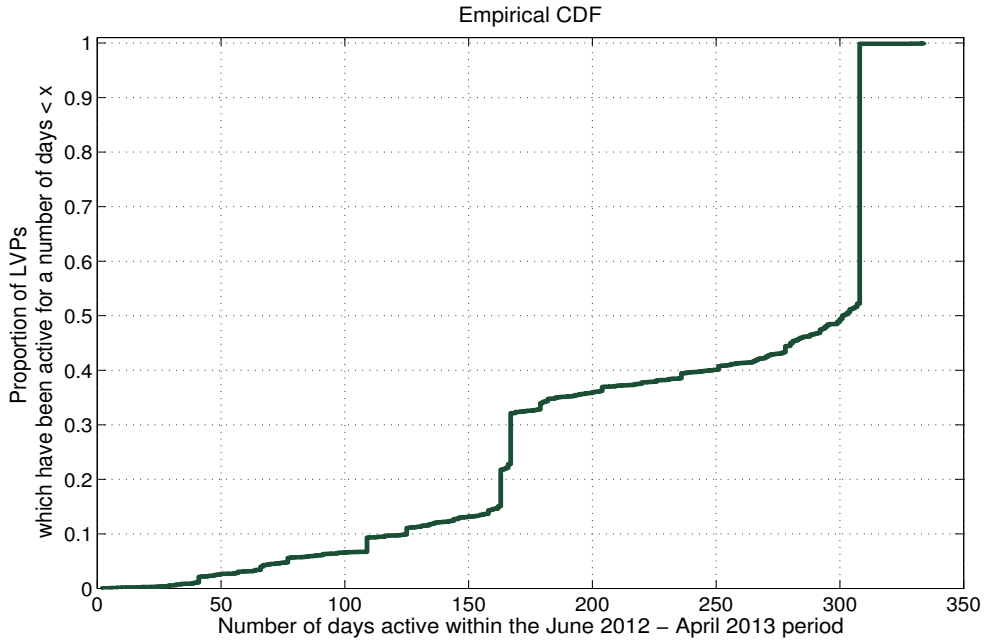

Figure 3.8: Empirical CDF of *LVPs* known to be *unintended* on the number of days they were active from June 2012 until the end of April 2013.

1st is 14,600 out the total 18,500 unintended *LVPs*. Another interesting observation is that in Figure 3.8 we can see that there is an important surge of *LVPs* active for 150 days within those 11 months. This is consistent with a few days after the moment when the BGP Visibility first became active, i.e., on the 1st of November 2012. The textitLVPs disappeared after the origin ASes learned that it was accidentally leaking 4,000 prefixes because of routing misconfigurations, as previously explained in Section 3.3.2. Additionally, we observe another important surge of *LVPs* at 310 day active. This is consistent with the case of the ISP leaking more than 13,000 *LVPs* to its customers. These prefixes were first activated in July 2012, and remained active after April 2013. The origin AS is currently addressing this issue.

3.4 Summary

In this chapter, we investigate to what extent it is possible to discover the match between the *intended result* of applying routing policies and the *actual result* reflected in the global routing system. Just by using publicly available data, we present an initial methodology that scans raw BGP data, filters and analyzes it, so that we can extract potential problematic policy configuration. We have defined the terms of *limited visibility* and *dark prefixes*, which can be considered early warning signs for routing policies backfiring and not achieving their desired outcome. Despite many years of research on BGP

data, such problems have not been sufficiently addressed [33]. We have presented our methodology to operators and received very promising feedback. For example, we found approximatively *90,000 stable LVPs* which, in a first phase after talking to operators, decreased with approximatively *3,000 LVPs*. The latter prefixes were proven to be actual symptoms of ill-configured routing policies. The Visibility Scanner allows per origin-AS queries for the *LVPs* generated and provides additional information about them. As future work, we intend to improve the quality of our heuristics by continuing to validate our methodology with operators. Also, since the methodology can be applied on any set of similar data, we would like to integrate into the tool the private views from operators.

Chapter 4

Winnowing Unintended Limited Visibility Prefixes

The BGP Visibility Scanner described in Chapter 3 allows network operators to verify on a daily basis if the prefixes advertised have limited visibility at the interdomain level. Though this allows operators to validate the efficacy of their routing policies or detect errors and misconfiguration, the system cannot immediately distinguish the LVPs that are intentionally restricted from being globally propagated, from the ones that are suffering from limited visibility in the Internet. In this chapter, we propose the Winnowing Algorithm, a machine learning tool which can automatically distinguish cases of *unintended LVPs*, generated by errors or by complex interaction between networks, from the *intended LVPs*, which emerge as expected expressions of routing policies.

We verify first if an *unsupervised learning* approach is suitable for distinguishing the unintended *LVPs* from the total set of limited-visibility prefixes. Generally speaking, an unsupervised learning or clustering algorithm can automatically find classification rules for the interest data, without the use of prior ground-truth data. To this end, we perform a series of exploratory analysis on the LVPs dataset, primarily using BGP-related information, e.g., the AS-Path length, visibility in the Tier1 group of ASes or the type of business relationship between the networks present in the AS-Path of the *LVPs*. However, none of these features are enough to properly define clear boundaries for the two predefined classes of *LVPs*. For example, we evaluating the LVPs average BGP AS-Path length in Section 3.2.1. Though we observe that the complete set of *LVPs* is characterized by a smaller average AS-Path length than the benchmark set of *HVPs* [12], we see no further clear trend within the set of prefixes with limited visibility which could predict their class. Additionally, we investigate the idea of an hierarchical clustering [74] analysis of the *LVPs*, in order to distinguish disjoint groups of similar observations based on features extracted from BGP-specific information. In particular, we consider the set of monitors "seeing" a particular *LVP* as a dissimilarity metric for applying the *hierarchical*

clustering algorithm. The latter, meant to find general rules to segment the set of limited-visibility in unintended and intended prefixes, has proven to be inconclusive as it did now show any clearly delimited clusters. Consequently, we recognize this situation as a so called *data rich but information poor* setting.

The widening gap between the existent *LVP* data and the information calls for a supervised learning approach, which can extract valuable knowledge embedded in the vast ground-truth dataset we have been able to build. We propose next the Winnowing Algorithm, an anomaly detection mechanism which builds on the ground-truth dataset collected with the BGP Visibility Scanner to automatically determine the status of the LVPs detected based on prior information. The machine learning tool is thus designed to tackle the limitations of the unsupervised learning approaches. We use the BGP Visibility Scanner as a *data mining tool* for identifying stable prefixes with limited visibility at the interdomain level and monitoring their status in time. We loosely use the term *data mining* as the process of collecting, searching through, and analyzing a large amount of data in order to discover patterns or relationships. The supervised learning approach advances a decision tree model that uses specific visibility features in order to classify the *LVPs* in the two above-mentioned classes. We further show that the per-prefix visibility features derived by monitoring the prefix visibility status reported by the visibility scanner over a period of two weeks are generally powerful to detect prefixes which are suffering from limited unintended visibility.

Figure 4.1 illustrates the process we follow to winnow unintended LVPs from the rest of legitimate LVPs. We first describe the visibility features taken into consideration for characterizing each prefix contained in the ground-truth dataset of 20,000 pre-classified LVPs. We present the proposed machine learning study design and talk about the error measures we try to optimize. We then advance a decision tree model using the optimal set of features, chosen according to the *information gain* metric. Leveraging the popular AdaBoost [24] algorithm, we boost the obtained basic model for achieving higher accuracy. We finally test the boosted tree-based model on the hold-out dataset, which is not used during the learning phase.

4.1 Data for Supervised Learning

We present here the way in which we pre-process the ground-truth dataset built using the BGP Visibility Scanner to further use it for supervised learning. This corresponds to the first step we observe in the workflow illustrated in Figure 4.1. The ground-truth consists of 20,000 *LVPs*, each pre-classified to indicate if the prefix has *unintended* limited visibility or if it is the consequence of *intended* interdomain behavior. We note that the dataset exhibits an important disproportion between the two defined classes, with 1,150 prefixes of the class *intended* and 18,850 *LVPs* of the class *unintended*. The *class imbalance*

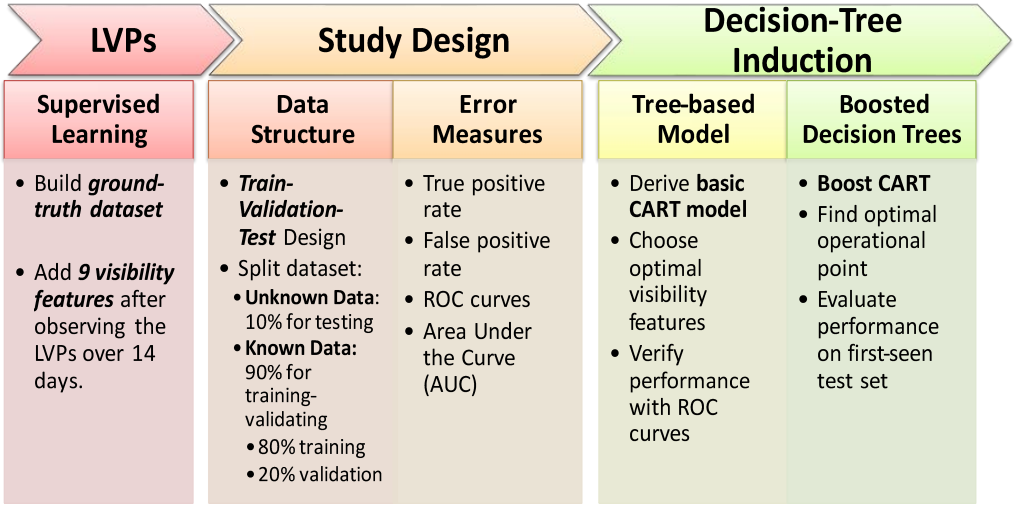

Figure 4.1: Winnowing Unintended LVPs: detailed methodology.

problem is a well-know issue in the machine-learning domain and it is characteristic to many other real-life applications.

This extensive set of data cannot be used as such for detecting the corresponding label for new occurrences of LVPs. In order to use it for training the machine learning Winnowing Algorithm, we first identify the full set of visibility features which we attach to each prefix in order to typify and further distinguish the two classes. For every *LVP*, the corresponding origin AS is observed over a period of *14 days* prior to the feedback moment, to characterize the visibility dynamics captured in the BGP Visibility Scanner. All the possible visibility parameters are listed and explained in Table 4.1.

Table 4.1: The list of per-LVP visibility features. All the values are calculated for an observation period of **14 days**. The features are ordered in decreasing order of their importance, according to the information gain metric.

Extracted per-LVP Feature	**Explanation**	**Information [weights]**	**Gain**
`mean_nrPrefs`	Average number of LVPs generated by the same origin AS	0.319	
`mean_MonitorsDetecting`	Average *proportion of active monitors* detecting the LVP	0.308	
`std_MonitorsDetecting`	Standard deviation of the *proportion of active monitors* detecting the LVP	0.3068	
`std_nrPrefs`	Standard deviation of the number of LVPs generated by the same origin AS	0.3060	
`mean_VisibilityDegree`	Average *absolute visibility degree* for the LVP	0.244	
`std_VisibilityDegree`	Standard deviation of the *absolute visibility degree* for the LVP	0.234	
`length`	Prefix length of the LVP detected by the BGP Visibility Scanner	0.183	
`TimeActive`	Proportion of time the LVP remained with limited visibility	0.153	
`VisibilityLabel`	Visibility label assigned by the BGP Visibility Scanner [*LVP*/*DP*]	8.61e-05	

4.2 Study Design

In this section, we explain the study design we follow for deriving the Winnowing Algorithm. This corresponds to the second processing block of the flowchart illustrated in Figure 4.1. We design the learning process in a training-validation-test format. In other words, we use cross-validation to estimate how the classification model behaves on an independent never-before-seen set of data. Also known as *rotation estimation*, this approach implies splitting the data into *known data*, which we use for training and validation, and *unknown data*, also known as *hold-out test data*, which we use for final testing. The idea of cross-validation is to repetitively split the known data into training and validation disjoint sub-sets, in order to estimate the accuracy of the model. We use the training dataset to first derive the classification algorithm. In order to avoid issues like overfitting and gain more insight on how the model could generalize to new independent data, we then perform an initial testing on the validation data. We further tune the decision model to achieve optimal performance on the validation dataset. We manually repeat the training and validation for various splits of the known data. In order to determine which is the algorithm with the best results across all possible data splits, we define a set of error measures which we further explain in detail. This final steps allows us to understand the performance of a prediction model when employed for independent cases.

4.2.1 Data Structure

We explain next which are the constraints we fulfill when splitting the pre-procesed ground-truth dataset in three different data subsets, namely training, validation and hold-out test sets. The three datasets considered in the study design must be perfectly disjoint (i.e., we should observe no prefixes nor origin ASes in common between any two datasets of the three defined). We thus split the ground-truth such that all the LVPs generated by the same AS are included in one unique dataset. We impose these restrictions in order to ensure a correct estimation of the algorithm performance when predicting the class of LVPs originated by **new** ASes, on which we have no prior ground-truth. This is a challenge, since predicting the class of LVPs originated by a network on which we have previously trained is significantly easier.

We first create the hold-out test dataset, by randomly choosing 10% of all the ASes which provided feedback on the visibility status of the LVPs. This hold-out test data is under no circumstances to be used in the training-validation phase of the learning process. Its main purpose is to estimate the performance of the optimal winnowing algorithm by using independent data on which the algorithm was not previously trained.

We split the remaining data in two different sets, namely the training and validation datasets. We perform the separation such that the training dataset has approximatively

80% of the remaining ground-truth dataset, and the validation set, the 20% left. We require that this constraint is respected for the total number of prefixes and also for the number of different ASes, i.e., 80% of ASes must be in the training dataset and the rest of 20% in the validation dataset. Additionally, we require that the 80-20 split for the training-validation datasets is also respected for each of the two classes of prefixes. In other words, we must have 80% of the intended LVPs in the training and the rest 20% in the validation dataset and the same for the unintended LVPs. We impose these rules to ensure a similar distribution of prefixes and ASes in the training and in the validation datasets. We identify exactly 989 different ways in which the training-validation split can be done such that all the imposed constraints are met. In rough numbers, this means training on about 15,000 *LVPs*, validating on about 4,500 *LVPs* and, finally, testing on approximatively 100 *LVPs* which were not used in the training-validation process.

4.2.2 Error Measures

We define here the error measures which we choose to describe the performance of the derived classification algorithm. The *accuracy* of a classifier is defined as the percentage of ground-truth tuples which are correctly classified when tested on a set of data the model was not previously trained on. However, even when we obtain a very high value for the accuracy of the classifier, it may be the case that the model does not recognize very well the tuples of one of the two classes, especially when dealing with *unbalanced* classes in the data, which is our case. To address these limitations and to evaluate the model performance, we define the following concepts:

- *True Positive tuples [TP]*: number of tuples classified as unintended, which really are of unintended class.
- *False Positive tuples [FP]*: number of tuples classified as unintended, which really are of intended class.
- *True Negative tuples [TN]*: number of tuples classified as intended, which really are of intended class.
- *False Negative tuples [FN]*: number of tuples classified as intended, which really are of unintended class.

We can further define the two error metrics which allow us to correctly evaluate the performance of the classification by capturing the per-class classification accuracy. Namely, we use **True Positive rate** [$\mathrm{TP_{rate}}$] and **False Positive rate** [$\mathrm{FP_{rate}}$], defined as follows:

$$TP_{rate} = P(unintended \mid unintended\ ) \sim \frac{TP}{TP + FN},$$
$$FP_{rate} = P(unintended \mid intended) \sim \frac{FP}{TN + FP}.$$

In other words, the TP_{rate} represents the probability of predicting a tuple as unintended, conditioned by the fact that the tuple is indeed unintended. Similarly, the FP_{rate} represents the probability of classifying a tuple as unintended, conditioned by the fact that the tuple is actually intended.

We use *Receiver Operating Characteristic (ROC) curves* to visualize the performance of a classifier. Given a binary classification problem, like the one we are currently addressing, the ROC curves allow us to visually analyze the trade-off between the true positive rate and the false positive rate. Many classification models, including decision trees, assign a probability to every tuple, expressing the degree to which the tuple is considered to belong to a certain class. By setting a decision threshold on these probabilities, we obtain a categorical classifier, i.e., the tuples are classified as *unintended* if their probability is higher than the fixed threshold, and "intended" otherwise. The performance of such a model is characterized by a single ($\mathrm{TP_{rate}}$, $\mathrm{FP_{rate}}$) pair of values which can be plotted in the ROC space. When considering different values of the decision threshold, we obtain a set of points capturing the $\mathrm{TP_{rate}}$ to $\mathrm{FP_{rate}}$ trade-off which can be plotted in the ROC space. Together, these points can be used to derive the ROC curve of the decision model. Generally, the ROC curve gives an aggregated view on the performance of the model, without reference to a specific threshold value.

To asses the general performance of various models using the ROC space, we can measure the Area Under the Curve (AUC). The area under the curve of a receiver operating characteristic (ROC) curve is a way to reduce ROC performance to a single value representing expected performance. The ROC space usually shows an ascending diagonal line, corresponding to the ROC curve of a non-informative classifier (i.e. one making stochastic decisions independent on data). As the ROC curve goes closer to this line, the AUC goes closer to 0.5 and the model becomes less and less accurate, up to the point of random or even worse than the random. Contrariwise, an AUC closer to 1 shows high performance for the model.

The ROC curve can be used to determine the operating point for the classification model. Note that, since each point in the curve corresponds to a decision threshold, selecting the operating point is equivalent to selecting a decision threshold. This selection may depend on the design considerations. For instance, if positive and negative examples are equally likely, the operating point maximizing the sum between the $\mathrm{TP_{rate}}$ and $1 - \mathrm{FP_{rate}}$ could be a good choice, because this is equivalent to maximizing the number of correctly detected tuples. However, the decision model operating with this threshold may not provide good results on new datasets with different distributions of tuples per class. To this end, a robust choice of the operating point is the *break even point*. The latter represents the value of the threshold where FP is equal to FN, and it can be shown to optimize the performance of the classifier under worst case conditions, i.e. under adversarial choices of the class distributions [50].

4.2.3 Decision Tree Induction

After previously defining the data structure, error measures and tools for assessing the performance of a classification model, we next explain the constructive process we follow in order to derive the tree-based Winnowing Algorithm. Following the flowchart depicted in Figure 4.1, we thus proceed to the last block, namely the decision tree induction.

In the model, we choose decision trees as base learners which are boosted to create a robust classification model. Tree-based learning methods rely on iteratively partitioning the data into smaller groups of similar elements [50]. The splitting of the data is done using the features that best separate the outcomes. The key idea is to chose the splits which maximize the group homogeneity, i.e., how similar are the elements within the same group, or until the small groups are sufficiently "pure". Choosing the right number of splits is a challenge, since we can easily overfit the model by considering splits that are very specific to the training data, or, contrariwise, underfit it by considering shallow general splits. Finding the correct balance is conditioned by finding the optimal set of features used to partition the data.

We use the extensively tested and popular machine learning method called Classification and Regression Trees (CART) [75] for deriving and fine-tuning the base tree model. We derive the decision trees using the standard library *tree* for R [76]. Using all the 989 different data splits, we determine the optimal decision tree. The latter is further used in the following step, where, by boosting, we combine multiple such base learners to form a robust classification model.

In Algorithm 2, we show the main phases traversed in the process of Decision-Tree Induction, corresponding to the last step of the methodology depicted in Figure 4.1. In order to derive the optimal decision model, we first perform the feature selection process. We use the training datasets to build the decision tree, whose performance we initially evaluate using the validation datasets. After we determine the set of features that maximizes the performance of the base learner, we boost the decision tree for increased accuracy. We determine the overall optimal decision threshold and we test the calibrated boosted decision tree on the hold-out test data. We further explain in more detail each of the considered steps.

4.2.4 Feature Selection

We expand next on the first phase of the Decision Tree Induction process, succinctly described in Algorithm 2. We perform the feature selection in accordance with the *information gain* for each of the visibility features. The information gain is a widely accepted measure for evaluating the capacity of a feature to distinguish between tuples of different classes. In the third column of Table 4.1 we show the weights associated to each of the 9 different visibility parameters after evaluating the information gain. In order to

Algorithm 2 Decision-Tree Induction

1) Feature Selection

- for $n = 1, ..., N$:
- **Learning from n features:**
  - append the n-th feature (ranked using the information gain) to the training set;
  - for each train/validation split:
    - train CART model;
    - for each threshold value:
      - compute TP_{train}, TN_{train}
  - compute average ROC;
  - compute AUC(n) from the average ROC;
- take subset of n* features maximizing AUC(n).

2) Boosting

- for each train/validation split (out of 989 possible) :
  - train AdaBoost using n* features
  - compute TP_{train}, TN_{train}
- compute threshold average ROC curve over all the splits;
- take threshold value from the break-even point in the average ROC.

3) Testing

- train AdaBoost using n* features and the whole dataset, excluding hold-out test data;
- compute TP, TN on the test set.

select the subset of features which ensures the optimal performance of the base learner for any training-validation data configuration, we adopt a progressive approach. We first verify the performance in every of the 989 training-validation splits of a tree model using only the highest weighted feature to classify the samples in the validation dataset. In a nutshell, we grow a different decision tree on each of the 989 training datasets, using as class-discriminating feature on the *mean_nrPrefs*. We then validate each of these 989 trees on the validation dataset of the considered data split and derive one ROC curve. Once having obtained 989 different ROC curves, we calculate the average performance of the decision tree over all these splits by evaluating the average true positive rate and false positive rate at different decision thresholds. For every value of the threshold, the averaging algorithm selects from each ROC curve the corresponding point. These points are then averaged and produce a point for every value of the threshold, thus generating the threshold average ROC curve.

Next, in decreasing order of the per-feature Information Gain weight values, we progressively add one new feature to the tree classification model and evaluate its performance, e.g., if we were initially classifying only with *mean_nrPrefs*, in the next model we add *mean_MonitorsDetecting*, and so on. In total, we derive 989 x 9 ROC curves corresponding to each of the [feature subset - data split] combination. We repeat the process explained above for deriving the threshold averaged ROC curve per feature subset. We depict in Figure 4.2 the threshold-averaged ROC curve in each of the 9 cases.

To further identify the optimal set of features, we compare the AUC for the 9 different ROC curves in Figure 4.2. Consequently, we observe that the classification tree using the first 7 most-important features has the highest performance, with an average AUC equal to 0.94. In the overall best operating point for all the 989 data splits, the decision tree has an average $\mathrm{TP}_{\mathrm{rate}}$ equal to 0.99 and an average $\mathrm{FP}_{\mathrm{rate}}$ equal to 0.1.

4.2.5 Boosting for Improved Accuracy

We previously determined that the optimal decision tree uses only the first 7 most-important visibility features (as explained in Table 4.1) to classify the *LVPs*. We now move on to the second phase of the decision tree induction present in Algorithm 2, namely boosting the base learning model.

Boosting is one of the most powerful learning mechanisms proposed in the last 20 years, used to improve the accuracy of a classification algorithm [74]. The main idea behind this algorithm is to combine many base classifiers (e.g., in our case, CART models built with 7 features) to produce one robust classification algorithm. Unlike other boosting algorithms, AdaBoost [24] adjusts adaptively to the errors of the base learners derived at each iteration. We use the AdaBoost.M1 algorithm implemented in the publicly available package *adabag* [77] for R.

In order to improve the classification performance across all the possible data splits,

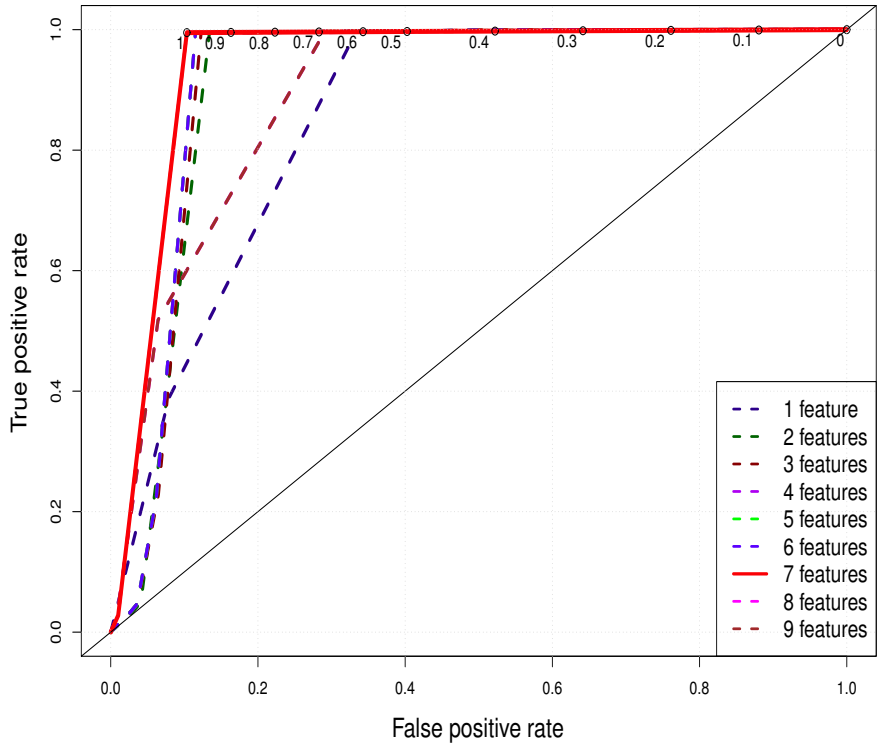

Figure 4.2: Threshold-average ROC curves for performance estimation of the decision tree built with the 9 feature-sets. The red continuous curve for the model using the 7 most important features has the highest AUC and, thus, is the optimal model.

we combine 50 such base learners using the boosting ensemble technique. We choose to run 50 boosting rounds to make sure that we do not over-fit the algorithm to the training data. After running experiments with variable numbers of boosting iterations, we find that a number of 50 boosting rounds improves the overall classification performance, without running the risk of over-fitting the classification tree.

To further assure a good general performance of the boosted tree-based model with 7 features, we determine next the overall optimal decision threshold over the 989 splits. For each of the 989 boosted decision trees obtained, we derive the associated ROC curve, to obtain an aggregated view of the performance of each classifier. Since we aim to decide which is the optimal operating point over *all* the 989 models, we first calculate the *threshold averaged ROC curve* for the 989 ROC curves.

In Figure 4.3 we depict the resulting averaged ROC curve, which we further use to calibrate the model. We first note that, independently of the threshold value, the classification model is generally very accurate for any of the training-validation splits, with an AUC equal to 0.997. Moreover, we observe that in the best operating point, the decision algorithm has an average true positive rate equal to 0.98, and an average false

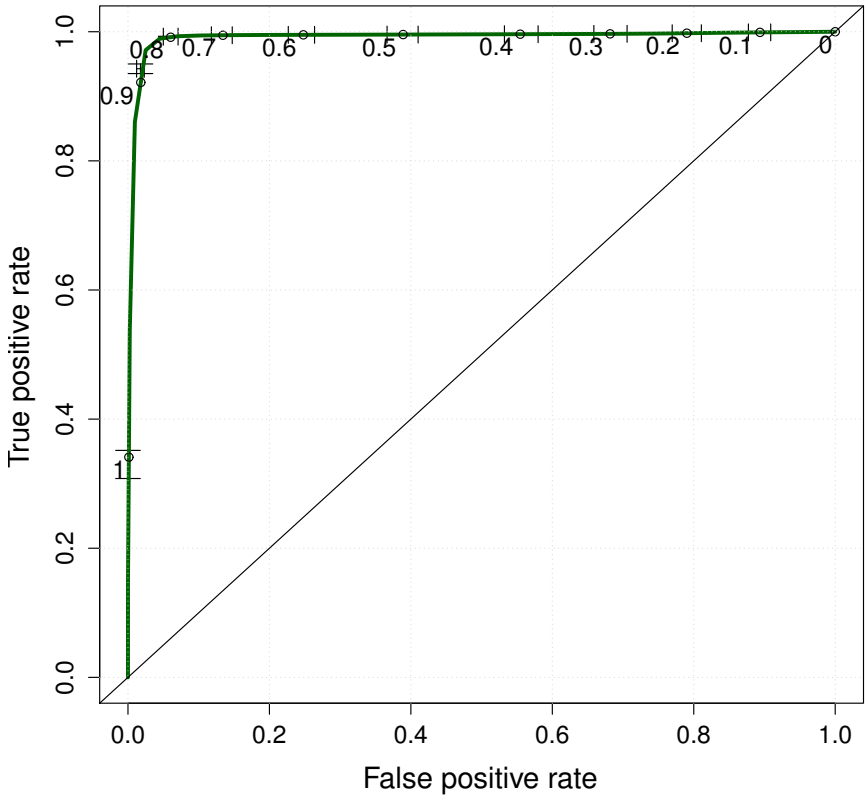

Figure 4.3: Threshold-average ROC curve of the boosted decision trees derived using each of the 989 possible data splits.

positive rate of 0.05. The average accuracy of the decision model is 98%. Though this is a very positive result, our aim is to design a classification algorithm which generalizes well by accurately predicting for *any* previously unknown case of AS originating *LVPs*. Given that for a new AS we do not have ways to learn the distribution of intended and uninteded prefixes, we choose as optimal operating point the value of the threshold where the performance of the algorithm is the highest for any possible distribution of prefixes per class. In other words, we choose the value of the threshold which gives the best performance under the worst known conditions. This point is the break even point, where the threshold value is equal to 0.6. In this operating point, the decision algorithm has an average true positive rate equal to 0.99, and an average false positive rate of 0.24. The average accuracy of the tree-based model at the break-even point is 95%.

Though we observe a slightly weaker performance than in the best operating point, we ensure that the decision algorithm at the break even point achieves optimal performance for new cases of ASes originating *LVPs*. We further refer to the boosted tree-based classification model using the 7 most-important features and operating with a decision

threshold of 0.6 as the *Winnowing Algorithm*.

4.2.6 Performance on the Hold-Out Dataset

To further asses the Winnowing Algorithm performance, we test the model on the held-out independent dataset, which has not been previously used for training or for validation. We train the prediction model on all the available ground-truth data, encompassing both validation and training datasets. We then test the boosted decision tree on the tuples in the held-out dataset. The performance of the winnowing algorithm is characterized by an average true positive rate of 0.951, with a 95% confidence interval of [0.87, 0.99] and an average false positive rate of 0.01, with 95% confidence intervals of [0, 0.02]. We further calculate the accuracy of the Winnowing Algorithm, by evaluating the overall proportion of tuples correctly identified. We obtain an average accuracy on the held-out test set equal to 97.2%.

4.3 Discussion on the Machine Learning Approach

Though the machine learning approach is gaining popularity for Internet-oriented applications, it is sometimes hard to understand the functionality of the mechanism. In this section, we provide the intuition behind the decision rules implemented in the winnowing system.

4.3.1 On the Visibility Features

One particularity of the Winnowing Algorithm is that, for classifying, it builds upon the 7 most important visibility features of the available ground-truth *LVPs*. We further observe that the set of features is consistent with the operational status of the routing system. For example, it has been long observed that accidental leaks usually generate a large number of prefixes at once. This explains why, in the context of the Winnowing Algorithm, the most important feature used to pinpoint *unintended LVPs* is the average number of *LVPs* injected by the same origin AS. Also, a high variation in the total number of *LVPs* from the same origin AS hints that the prefixes may not be expressions of stable long-lived routing expressions, but merely a side-effect of errors or faulty routing policies. Additionally, we use the visibility degree and the proportion of active monitors from the daily sample detecting *LVPs*. These features capture the prefix visibility dynamics caused by the variations in the daily set of active monitors used. For example, we have observed that majority of unintended prefixes have a stable visibility degree of about 3, which is consistent with the fact that misconfigurations usually affect the routing policies of the ASes in the direct vicinity of the origin. Furthermore, discarding the last two features, namely the *TimeActive* and the *VisibilityLabel*, can be rationalized using the ground-truth

data. For instance, as previously depicted in Figure 3.8, the lifetime of *unintended LVPs* is much longer than the lifetime of easily-noticeable anomalous events, which are quickly fixed by the origin. For this reason, the lifetime of unintended LVPs is consistent with the lifetime of intended LVPs which appear as a result of the routing policies configured. Thus, the parameter *TimeActive* does not discriminate well between the two classes of *LVPs*.

4.3.2 On the Data Structure

One of the restrictions we impose in the data structure proposed in the machine learning study design is that the *LVPs* originated by the same AS be all included in the same dataset, namely training, validation or hold-out test data. This restriction ensures that we are correctly using our winnowing mechanism to distinguish between *LVPs* from **new ASes** that might be suffering from unforeseen events. However, it is also important to accurately classify **new *LVPs*** from a network which already provided feedback used for deriving the Winnowing Algorithm. In order to verify the performance of the classification model on new *LVPs* originated by ASes used in the training phase, we perform a very simple experiment. Namely, we split the dataset independently of the origin AS. We withhold 100 random instances of each class for testing and use the rest for training. We find that the Winnowing Algorithm derived in Section 4.2.5 performs a highly accurate classification of the test samples, only misclassifying one out of the 200 test tuples. In other words, when training on *LVPs* originated from one particular origin AS, the algorithm has a fairly easy task in classifying the new LVPs originated by the same AS.

4.3.3 On the Ground-truth Lifetime

The ground-truth dataset of 20,000 *LVPs* documents a wide range of cases of previously undetected anomalous events, affecting the interdomain entities. Though the root causes of the anomalies we detect are recurring in the Internet, their appearance in the BGP Visibility Scanner might change in time. It is unclear at this point how the validity of the ground-truth dataset and, consequently, the performance of the Winnowing algorithm would be impacted by the evolution of the routing anomalies. We leave for future work the analysis on the lifetime of the ground-truth knowledge we have accumulated and the stability of the accuracy of the winnowing system in time. Additional amount of feedback from active users of the visibility scanner can advance our understanding of the evolution of LVPs in the Internet. Furthermore, a ground-truth dataset covering a longer period of time would also allow us to enhance the capabilities of the Winnowing Algorithm, since this would offer many different examples of intended and unintended LVPs while also capturing the evolution in time of the visibility parameters of routing

anomalies.

4.4 Summary

In this chapter, we continue the efforts described in the previous chapter and propose a machine learning approach to automatically distinguish between the LVPs which are the *intended* result of applying routing policies and the *unintended* LVPs which emerge in the Internet as side-effect of routing misconfiguration or complex policy interactions. Leveraging the popular method of boosted classification trees, we use a unique ground-truth dataset in order to derive a classification model for the *LVPs* detected by the BGP Visibility Scanner. After extensive testing, we conclude that the proposed system winnows unintended *LVPs* with 95% accuracy. This further proves that visibility features are generally powerful to detect anomalies which, despite their impact on the routing system, are hard to single out. In order to develop the machine learning algorithms, we use standard libraries for R, thus reinforcing the robustness and generality of the resulting system. Additionally, we make available an anonymized version of the ground-truth dataset upon request, allowing for the reproducibility of the machine learning analysis by any interested party.

Using the Winnowing Algorithm, we can further classify new *LVPs* identified by the BGP Visibility Scanner. For example, for the set of *LVPs* retrieved on the 14^{th} of July, 2013, counting exactly 84,982 *LVPs*, the boosted decision tree classifies 25,860 *LVPs* as intended ($\sim 30\%$), and the rest of 59,117 *LVPs* as unintended ($\sim 70\%$).

Chapter 5

Reachability of Limited Visibility Prefixes

Using the Winnowing Algorithm presented in the previous chapter, we are able to accurately distinguish unintended LVPs which unexpectedly emerge in the Internet because of configuration errors or bogus routing policies taking effect. This set of anomalies not only impacts the efficacy of the intended routing policies of the ASes affected. There are cases when the limited visibility prefixes may also be globally unreachable, since there might not always be a less-specific covering prefix to ensure that the destinations attached are globally reachable, even if through an ill-preferred route. In this chapter, we aim to establish if prefix visibility at the interdomain level can be further used to alert ISPs about reachability issues their prefixes might be suffering. In other words, we analyze next if the routing anomalies that render a prefixes as limited visibility can also deteriorate the reachability of the corresponding address space. To this end, we perform reachability measurements towards prefixes in all of the three above-mentioned sets of prefixes, i.e. HVP, LVP and DP, with the goal of establishing the existence of a correlation between visibility and reachability of prefixes in the Internet.

5.1 Measurement Approach

We begin our analysis by first presenting the approach we propose to determine if a prefix is reachable from a given vantage point in the Internet. The challenge for doing this with IPv6 prefixes is that it is not a simple task to find an address that is actually allocated to a host in a given prefix. Given the current stage of density of the IPv4 Internet, this should not constitute a concern when testing the reachability of the address space. However, for consistency, we further employ the same measurement approach both for IPv6 and IPv4 prefixes.

The idea we put forward is to probe the reachability of a prefix is to perform tracer-

oute towards the first available IP address within the prefix and check if the last node responding to the traceroute belongs to the AS of the target prefix or to one of its Internet providers, as observed in the BGP AS-Path. In other words, our measurement approach is as follows.

- We send a traceroute probe towards the first available address within the target prefix.
- We say that the prefix is reachable if :
  1. The traceroute probe reaches the AS of the target prefix, i.e., the last AS which appears in the BGP AS-Path attribute[1]
  2. The traceroute probe traverses the second-last[2] AS along the BGP AS-Path for the target prefix.

We consider this latter hypothesis because there may be cases where, even if the probe does reach its destination, it might happen that the AS of the source IP for the last ICMP message received is actually the transit provider of the target AS. This happens because it is a common operational practice that ASes use addresses from their providers for their transit links. As a result, the router within the destination network that issues the last message of the traceroute process will do so using an source address from its ISP's address space. We do acknowledge that this may also be due to reachability problems in the last hop, which our methodology is unable to distinguish.

5.1.1 Traceroute Selection

We establish next which of the existing traceroute approaches is the most efficient. Traceroute is one of the most widely used network measurement tools, useful both to network operators and researchers. The original traceroute tool sends UDP probes to increasing unlikely port numbers towards the target IP address. Apart from the default traceroute [78], several other traceroute approaches are available. We verify their performance by testing from a single source the status of a large set of control addresses using various probing methods, namely UDP traceroute, TCP traceroute and ICMP traceroute. The control set contains 70,000 addresses which are known to be reachable. This is made up of addresses from many sources, including DNS entries, Alexa's top sites, and several other sources. Using all the available traceroute probing approaches, we check the reachability status of these 70,000 addresses from a machine inside a major Japanese ISP's network. Our results show that the most efficient probing method is *ICMP traceroute*,

[1] Usually, in the BGP AS-Path the last hop represent the origin AS of the prefix, while the first hop represents the AS whose routing table we analyze.

[2] Following the order of the ASes in the BGP AS-Path attribute, the second-last hop (2LH) in the AS-Path corresponds to the transit provider of the origin AS.

which successfully got responses from 99% of all the 70,000 probable IP addresses. Consequently, the traceroute probing method we further employ in our study is the ***ICMP traceroute***. This results is consistent with the observations of Luckie et al. in [63].

5.1.2 Validating the Measurement Approach

We validate our approach by checking the reachability status of a set of prefixes which are a-priori known to contain at least one reachable address. Our aim is to determine if the reachability of a single IP address is representative for the most-specific covering prefix. For each of the previously mentioned reachable anchor addresses, we map the covering prefix installed in the BGP routing tables. We use public routing data information to determine the most-specific prefixes covering each of these reachable addresses. The set of prefixes determined represents address space known to contain at least one address which is successful to ICMP traceroute probing. These prefixes form the target set of prefixes which we use for validation.

We start by running traceroute from a machine within the major Japanese ISP, for which we also have the BGP routing information corresponding to the moment we run the measurements. We send ICMP traceroute probes towards the **first available IP address** within each of the prefixes determined above. According to the proposed methodology, we consider that the traceroute probe reached its destination when the traceroute probe traverses either the origin AS of the destination address, either the second-last AS (2LH) appearing in the BGP AS-Path towards the target prefix. In order to identify the 2LH towards a prefix, we analyze the AS-Path information in the BGP routing table of the AS from which we are generating the traceroute messages.

After parsing the results of our traceroute tests, we learn that the ICMP traceroute probes successfully reached more than 96% of these *a-priori* reachable prefixes. Consequently, the methodology we propose is able to identify with 96% accuracy the reachability status of an IPv6 prefixes. For the other 4% of prefixes, our methodology is unable to determine reachability. This may be due to several reasons, including ICMP filtering or routers silently discarding packets.

5.2 Reachability Measurements and Results

5.2.1 Reachability Measurements for the Different Visibility Classes

We further aim to establish the reachability for prefixes with all of the three different classes of interdomain visibility, namely DPs, non-dark LVPs and HVPs. For this analysis, we use the set of IPv4 and IPv6 LVPs derived on the 8th of August, 2013. We perform the reachability measurements first from a single source, for which we also know the state of the BGP routing at the moment of testing. From the point of view of the measurement

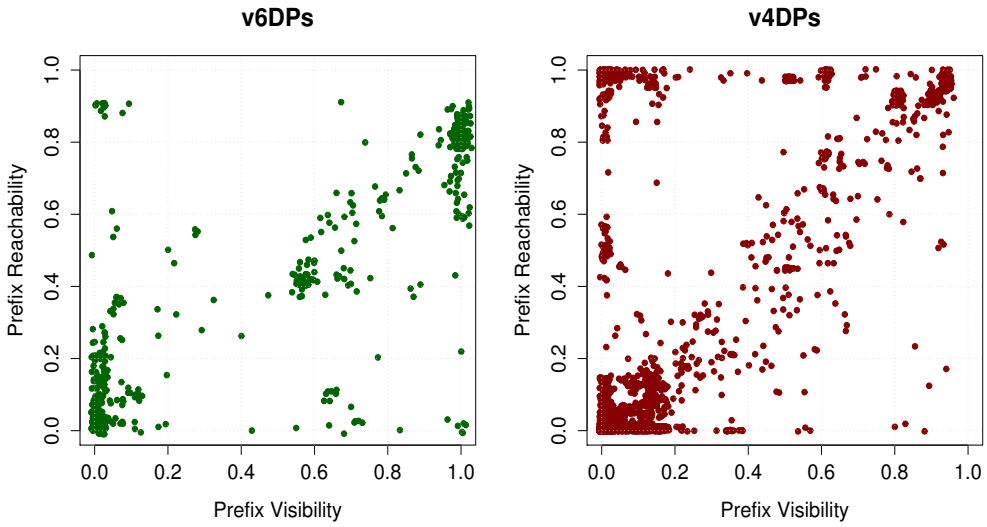

Figure 5.1: Scatterplot of reachability probability against the DP's visibility, for v6DPs and for v4DPs.

source, the High-Visibility Prefixes are the prefixes contained in its BGP routing table. There are in total 13,195 such IPv6 prefixes and 480,400 IPv4 HVPs. These prefixes may not be globally High-Visibility, since there may be other routing tables not "seeing" some of these prefixes. We label all the rest of prefixes learned from the rest of the routing tables collected from the public repositories as Limited Visibility. The latter reach a total number of 2,359 prefixes for IPv6 and 87,397 prefixes for IPv4. In order to check if any of the Limited Visibility prefixes are in fact Dark Prefixes from the point of view of the ISP, we check which LVPs have a less-specific HVP in the ISP's routing table to offer global reachability. We are thus able to single out a total number of 511 IPv6 DPs and 2917 IPv4 DPs.

In the case of the IPv6 LVPs which have a covering high-visibility IPv6 prefix (i.e., they are not dark), we observe that 94% of the prefixes are reachable from the Japanese ISP's network. For the IPv4 LVPs with covering HVP, we find that 89% were reachable from the single vantage point used. This is consistent with the precision of our tool so we cannot make any claims about reachability problems in the LVP set. We next evaluate the reachability status for the IPv6 and IPv4 DPs. We learn that for more than 95% of these IPv6 prefixes, the traceroute measurements did not reach the target. Consequently, less than 5% of the dark address space is reachable from the vantage point. This result is further consistent with the reachability measurements performed for the IPv4 dark prefixes, out of which less than 3% were reachable from the vantage point. This results shows that, contrary to the case of non-dark LVPs where there might be a less-specific covering HVP to ensure global reachability, the DPs do present serious connectivity problems, when measured from this single vantage point.

5.2.2 RIPE Atlas Measurements and Results

Previously, we have seen that, because of the lack of a covering HVP, dark prefixes exhibit serious connectivity issues, when tested from a given vantage point in the Internet. In this section, we further test is this is globally valid. In this section, we use the RIPE Atlas Platform [79] to run **larger-scale measurements for characterizing the reachability of the global dark address space**.

We zoom out from the previous localized analysis of reachability, and test the reachability of the DPs from 100 different probes active in the RIPE Atlas platform. We run the measurements both towards the globally defined set of IPv6 dark prefixes, i.e. the 473 v6DPs derived from analyzing 110 BGP routing tables, and also towards the set of IPv4 dark prefixes, i.e., 3,200 v4DPs derived from analyzing 154 global BGP routing tables. We send ICMP traceroute probes towards a target address within each of the v6 and v4 DPs. We proceed to verify the reachability results in accordance with the methodology specified in Section 5.2.1. Point (2) of the proposed methodology requires to verify if the traceroute probe traverses the provider of the origin AS for the target prefix. As opposed to the previous case where we have the BGP routing table from the AS hosting the traceroute source to analyze, we now do not have access to the BGP routing tables corresponding to the 100 Atlas probes used. In order to overcome this issue, we build a set of *probable* second-last hops which are likely to be traversed towards each of the possible destination ASes. We do so by analyzing all the available routing tables from the ASes active in RIPE RIS and/or Routeviews, and monitoring the ASes appearing as 2LHs towards every destination AS of interest, i.e., the AS of the target prefix. We then state that the target prefix is reachable if *the traceroute probe traverses **any** of the probable second-last ASes towards the origin AS of the target prefix.* In order to further understand the impact of relaxing point (2) of our methodology, we perform the following verification. We first determine the set of probable second-last hops towards every destination AS, without using the BGP routing information from the AS hosting the machine we used to run the traceroute tests in Section 5.2.1. We then verify the proportion of prefixes for which the 2LH appearing in the BGP AS-Path from the Japanese AS routing table is *not* among the set of 2LHs likely to be traversed towards the target prefix. We find that only for 0.05% of the targets, the 2LHs appearing in the BGP AS-Path of the Japanese ISP are not included in the set of probable 2LHs derived from all the available global routing tables.

After processing all the traceroute results from each of the 100 probes towards the Dark Prefixes, we conclude that the average reachability degree for a v6DP is of 46.5%, whereas for v4DPs this decreases to only 17.4%. To further understand this result, we verify how the DP reachability correlates with the visibility degree of a DP. We show in Figure 5.1 the scatterplots both for IPv6 and IPv4 DPs' reachability against their visibility within the corresponding sample of ASes analyzed. We observe that for the v6DPs, depicted in

the left-side plot, there is a stronger correlation between reachability and visibility than for the v4DPs. This happens because, for the v4DPs, we see a high number of prefixes with very limited visibility, but which are highly reachable from the sample of 100 probes chosen. We observe that in the v4 plot from Figure 5.1 there are approximatively 8% of IPv4 prefixes with visibilities smaller than 0.2 and reachability larger than 0.2. As previously noted in [80], this may be due to default routing in IPv4. In Section 3.3, we explain many of the real-life operational reasons for which this type if v4DPs emerge in the Internet. For example, we observe in the lower-left corner of the IPv4 plot in Figure 5.1 a very large number of v4DP (approximatively 72% of all the v4DPs) with a reduced visibility degree and a corresponding low reachability degree. These v4DPs may be route leaks which, as we learn from the operational use cases explained in Section 3.3, often occur in the Internet. Consequently, the lack of reachability observed for v4DPs is largely explained by the fact that these prefixes are unintended to be visible in the Internet to begin with. At the same time, even if the v6DPs do not follow the known symptoms of route leaks or anomalies previously learned from the IPv4 cases, they do struggle with important lack of reachability. This further supports the hypothesis that, while in IPv4 the DPs are in majority results of mistakes or slips in the network configuration, for IPv6 we understand this as a side-effect of the early stages of development of the network.

In an effort to establish the possible correlation between visibility of v4DPs and v6DPs, we further separate only the dark prefixes which are generated by ASes active both in v4 and v6. In other words, we focus on analyzing the reachability of DPs whose ASes originate both IPv4 and IPv6 dark address space. In total, we test the reachability of 214 dark prefixes, out of which 88 are v6DPs. We learn that, in average, for the v6DPs there is an average probability of 40% of being reachable from a vantage point in the Internet, which is consistent with the general reachability result for all the v6DPs. This probability decreases to 20% for the v4DPs, also consistent with the overall reachability result for IPv4. Consequently, there is no apparent correlation between the reachability of v4DPs and v6DPs originated from the same AS, these actually following the general reachability trends previously established.

5.3 Summary

In this chapter, we focus on characterizing the LVPs identified using the BGP Visibility Scanner. To this end, we perform an extensive analysis of how BGP route propagation affects the global reachability of the corresponding address space. We propose a methodology to measure the reachability status of the active LVPs, which represent address space that is not present in all the global routing tables of the operational networks. We find that, while the fraction of limited visibility address space is similar in the IPv4 and the IPv6 Internet (about 20% of the prefixes), the proportion of dark address space in the

IPv6 Internet is significantly larger than in the IPv4 Internet (3.75% versus 0.6%). We find an important correlation between the limited visibility of a dark IPv6 prefix and its reduced reachability. Moreover, while the IPv4 dark address space can be largely explained as route leaks or mistakes, this is not valid for the v6DPs. We believe that this is a serious problem for the IPv6 Internet, as limited reachability of a non-negligible set of prefixes undermines the global connectivity of the Internet. In future work we expect to investigate the reasons behind the large amount of dark address space in the IPv6 Internet.

Chapter 6

The Inadvertent Economic Impact of Strategic Deaggregation

We previously observed that the majority of prefixes with limited visibility identified with the BGP Visibility Scanner includes prefixes that are covered by less-specific prefixes which, in general, have high visibility at the interdomain level. Often, network operators intentionally limit the visibility of distinct fragments of their address block by selectively announcing the more-specific prefixes to different upstream providers, while still injecting the covering less-specific prefix to all the providers. This is usually driven by severe traffic engineering needs, e.g., load balancing the traffic by originating several more-specific prefixes and announcing different prefixes via different AS paths. In this chapter, we focus on the detection and analysis of *strategic deaggregation*, which constitutes of the action of splitting the owned address block to a certain variable degree and selectively injecting different more-specific prefixes to different disjoint subsets of providers.

Generally, address space fragmentation, also known as *prefix deaggregation*, offers a very high granularity for traffic manipulation. This technique allows networks to divide their assigned address blocks in different-sized sinks of traffic which can thereafter be easily maneuvered. For example, using this technique, geographically-spread networks can divert different amounts of traffic corresponding to different points of presence (PoP), thus attracting traffic into their network through the PoP closest to the final destination. By using this type of techniques, network operators intentionally limit the visibility of the more-specific prefixes at the interdomain level. Consequently, this phenomenon partially explains the large number of more-specifics identified as more-specifics with the BGP Visibility Scanner presented in Chapter 3.

In this chapter, we observe the collateral benefits which emerge as by-products from the use of deaggregation, without, however, constituting the central drive for ASes to deploy such strategies in the first place. We merely acknowledge the popularity of deaggregation in the Internet and investigate the possibility of an economic gain strictly from

the point of view of the transit traffic bill for the deaggregating party. Regardless of the main goal to be achieved through deaggregation, we do observe that, in certain conditions, the deaggregating AS can indeed enjoy a decrease of its transit traffic bill within certain conditions as a by-product of the deaggregation strategy deployed.

Though majority of Internet routes is very stable in time [81], it has been observed that for each destination there is a subset of unstable traffic sources large enough to cause important traffic shifts in the interdomain [29]. We further show in this chapter that by deaggreagating, the customer AS reduces the path diversity towards each more-specific prefix by selectively announcing it to a subset of providers. This, in turn, compels the traffic originated by any source towards a destination within a more-specific prefix to flow through the preferred set of transit providers. This happens because the *longest-prefix match rule* over-rides any other routing policy applied to the covering less-specific prefixes. This translates into a more deterministic traffic pattern, which decreases the traffic fluctuations on a transit link and, implicitly, in the monthly traffic bill.

In this chapter, we study the impact address-space fragmentation has on the transit traffic bill of the networks originating the more-specific prefixes, from a theoretical point of view. We continue our analysis throughout Chapter 7 where, through the analysis of real-world data from an operational ISP, we focus to identify and quantify the impact of actual occurrences of prefix deaggregation.

We propose an analytical model to analyze the effect of different deaggregating strategies on the traffic stability and ultimately on the transit cost for the deaggregating ASes. The model accounts for the route dynamics which are responsible for large traffic shifts in the interdomain, like previously observed in [29]. We integrate in the Internet model three important elements, i.e., the interdomain routing model, the traffic model and the cost model, whose entanglement offers the necessary underlying structure for the analysis of these Internet phenomena. This general Internet model disencumbers our analysis of the complex Internet phenomena, maintaining a continuous focus on the impact of different deaggregating strategies on the transit traffic stability and ultimately on the transit cost incurred on the customer ASes.

We find that, as a result of the unique interaction between the path dynamics in the current Internet, the asymmetrical popularity of traffic sources and the popular billing method which relies on the 95^{th} percentile of traffic [31], [30], the ASes which engineer their incoming traffic using strategic deaggregation might enjoy one collateral benefit which, to the best of our knowledge, has not been previously studied: *the inadvertent decrease of their transit traffic bill.* We stress the fact that we analyze the existence of an economic **side-effect** of prefix deaggregation. We do not perform a study of the central motivations driving operators to perform prefix deaggregation in the first place, nor do we defend or encourage the usage of deaggregation in the Internet. We merely acknowledge the popularity of this strategy in the Internet and further investigate the possibility of an

inadvertent economic gain for the deaggregating party. Regardless of the main goal to be achieved though deaggregation, we observe that, in certain conditions, the deaggregating AS can indeed enjoy a decrease of its transit traffic bill as a by-product of the deaggregation strategies deployed. We show that, in certain conditions, through deaggregation, network operators can reduce the route diversity towards each prefix announced and, consequently, also the traffic fluctuations on the corresponding transit link, thus further impacting the monthly traffic bill paid to the transit providers.

6.1 Toy Example

Analyzing the Internet ecosystem is a challenging task, since it presents with many dynamic elements acting at different timescales. In order to achieve a better understanding of the impact of prefix fragmentation on the transit bill, we unburden our analysis of the complex Internet characteristics and intuitively present the setup we aim to analyze. We introduce next a toy example to illustrate how a network changing its strategy from non-deaggregation to strategic deaggregation can benefit from a decreased transit traffic bill, and possibly impact the revenues of its providers.

In order to clarify the main phenomena we observe in this chapter, let us consider the simple case of one destination network announcing the same prefix 1.1.0.0/16 over two different transit links, like we can see in Figure 6.1.a. We reduce the number of sources of the interdomain at two, out of which one is generating $\frac{3}{4}$ of the whole traffic T consumed by the destination network, and the other one, the rest. We analyze next the traffic distribution on each of the two transit links. We consider the 95^{th} percentile pricing model, the most widely used method for charging the IP transit, in which the monthly bill is the function of the peak level of traffic, not the average usage.

We monitor the level of traffic on each link during one month. We consider that source AS 1 is sending its traffic on link l_1 for half of the period, after which, due to a routing change, it starts forwarding its traffic on link l_2. Source AS 2 suffers the opposite events, namely it sends the traffic during the first half of the month through link l_2 and for the second half of the month it switches to link l_1. The transit traffic cost is calculated using the 95^{th} percentile rule for each of the two links. As a result, because the traffic on link l_1 had a level of $\frac{3T}{4}$ for more than 5% of the billing period, the transit traffic bill for link l_1 is $c\frac{3T}{4}$. Similarly, the transit traffic bill for link l_2 is also $c\frac{3T}{4}$, as for more than 5% of the billing period the traffic level was $\frac{3T}{4}$. Therefore, the total cost paid for the consumed traffic T is $c\frac{3T}{2}$, which is with $c\frac{T}{2}$ higher than the cost cT paid based on the 95^{th} percentile rule if no routing changes would happen.

We show in this chapter that, through selective deaggregation, the destination AS can inadvertently avoid the fluctuations of traffic due to routing changes and thus also decrease its transit traffic monthly bill. Consider that the destination AS divides its

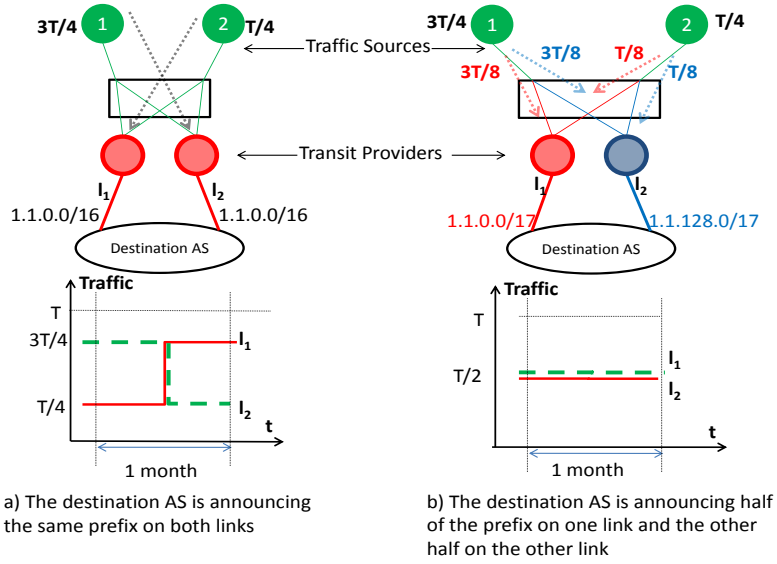

Figure 6.1: Toy example representation.

address space into two more-specific prefixes and announces each on a separate link, i.e. announces 1.1.0.0/17 through link l_1 and 1.1.128.0/17 through link l_2, like we can observe in Figure 6.1.b. If we assume uniform distribution of incoming traffic for the prefix, this means that each more-specific prefix receives half of the traffic generated by each source. In this scenario, the routing changes do not artificially increase the 95^{th} percentile and the transit traffic monthly bill for the destination AS is cT.

In the real Internet, the number of independent sources is much higher that the number assumed in this toy example. Because of this, one may think that due to the large number of sources in the interdomain, the routing changes characteristic to in the global routing system will only lead to small relative fluctuations of traffic. However, the skewness of the traffic distribution on sources has an important effect on the amount of traffic switching between transit links. In other words, if a large source of traffic becomes instable due to interdomain routing changes, then important amounts of traffic shift between different routes, thus heavily impacting the traffic distribution on the incoming links towards a destination.

In [27], the authors actually verify the amount of routing changes possibly affecting how traffic flows towards a destination. Based strictly on the information contained in public BGP routing tables, the authors go to show that those routing changes increase the transit bill of given customer with an average of 5%. Clearly, in the operational Internet some of these customer ASes are much more affected by routing changes than others. However, this average gives us a somewhat concrete idea on the manner in which the combination of routing changes, the skewed distribution of traffic on sources [32] and the popular 95% percentile billing scheme [30, 31] inflate ones transit bill.

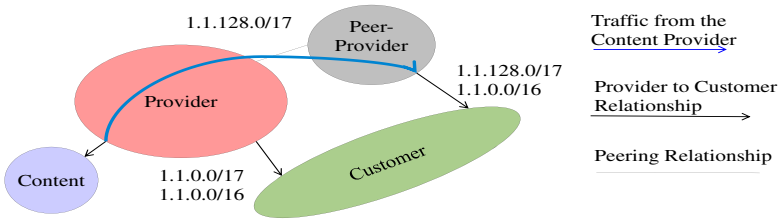

Figure 6.2: Strategic prefix deaggregation may have additional implications in terms of costs incurred by the provider.

The reasons for deploying the deaggregation strategy may include a wide variety and may or may not be related to decreasing ones transit traffic bill. For example, when analyzing the topology from Figure 6.2, one reason might be the need to avoid the capacity upgrade on the link between the Customer and the Provider, or even the need of the Customer to receive traffic for the more-specific locally from the Peer-Provider, thus avoiding hauling the traffic within his own network. In the scenario from Figure 6.2, the Provider network also supports an additional cost implied by having to haul the customer traffic through its own network towards the Peer-Provider. Studying the motivation for employing this mechanism is, however, out of the scope of our analysis. We focus instead on the economic impact of deaggregation, regardless of the main reason behind deploying this strategy in the first place. In the following section we propose a general model to analyze the savings in the monthly transit traffic bill incurred by an efficient deaggregating strategy.

6.2 Model Description

We establish, in this section, the settings of the general Internet model proposed for the study of the impact of prefix deaggregation on the transit traffic bill. By combining three important elements, i.e. the interdomain path changes, the 95^{th} percentile billing rule broadly used in today's Internet and the skewed distribution of the traffic demand on sources, the model offers the underlying structure for the analysis of the phenomena associated with the deaggregating strategy intuitively captured by the toy example. An initial version of this model was previously presented in [27].

We model the Internet at the AS level, where the networks consist of N sources and one destination AS, as we can observe in Figure 6.3. This assumption does not not impact the generality of our model as, in the current Internet, paths are calculated independently for each destination. Therefore, we focus our analysis on the case of one destination network with n transit links[1] which are accommodating the traffic demand distributed

[1]Without restricting the generality of the analysis, one might consider that one transit link corresponds to a distinct upstream transit provider for the destination AS. However, the proposed Internet model can easily be used to analyze more realistic situations where the number of links between a transit provider

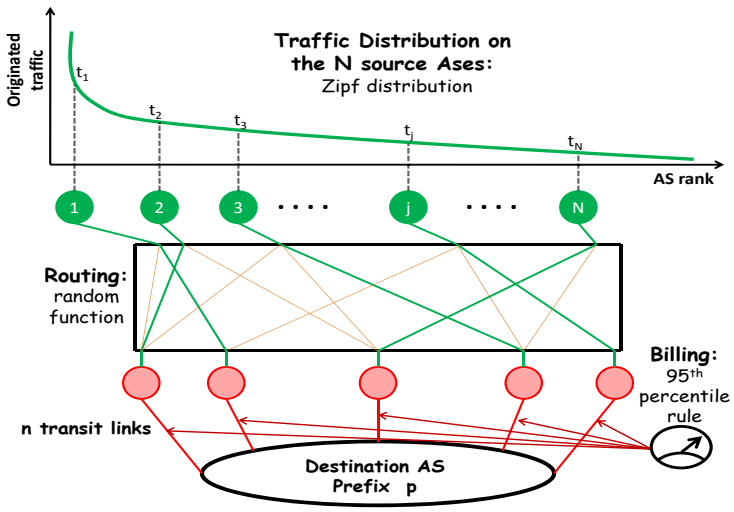

Figure 6.3: Graphical representation of the proposed Internet model.

over N sources in the interdomain. We assume a symmetric model where all the links have the same capacity and are equally likely to be a part of the path from a source to the destination in question. For the ease of the presentation, we assume an uniform distribution of incoming traffic on the destination address space. As depicted in Figure 6.3, we integrate three important elements in the model, i.e. the interdomain routing model, the traffic model and the cost model. Their entanglement offers the necessary underlying structure for studying the influence of various deaggreagation techniques on the traffic fluctuations and the interdomain traffic bill.

6.2.1 Deaggregation Strategies and the Model for Interdomain Routing Changes

We model several different behaviors with respect to the deaggregation of announced prefixes. First, the AS can decide to announce one aggregated prefix through all its transit links. Alternatively, the AS can decide to divide the n transit links into λ link sets (where $\lambda = 2, ..., n$) and announce a single more specific prefix over all the links of each link set. The result is that it announces λ more specific prefixes P_1, P_2,..., P_λ over λ disjoint links sets l_1, l_2,...,l_λ.

We also assume that the assigned address space can be divided evenly between the available number of link sets and announced by the destination AS as a single more-specific prefix separately on a different link set[2]. Moreover, we assume that the announced prefixes

and its customer is higher than 1.

[2]Due to the manner in which the prefix can be split, this is true in the case when the number of links is equal with a power of two. In the other cases, while it is not always true that the evenly divided address space can be announced only as a single aggregated prefix, we can find a particular fragmentation of the address space that would allow us to achieve the uniformity desired and announce the smallest number of more-specific prefixes in the link set.

are propagated as injected by the origin i.e. the other ASes honor the origin deaggregation, which is aligned with current operational practices [38]. Additionally, we assume that all the announced prefixes are reachable from every AS in the Internet. This means that every AS in the interdomain receives routes for the λ prefixes corresponding to the originating AS and selects one route for each prefix.

The path selection dynamics towards the destination of interest are the result of the complex interaction between the Internet topology dynamics and the policies of different ASes along the paths between the source and the destination AS. The changes in the selected paths often happen as a consequence of, for example, topological modification in the network or individual routing policies changes. Usually, the timescale characteristic for these routing changes is of a few hours or days.

In order to better understand the impact of the route dynamics on the cost for transit traffic, we analyze the path changes using a timescale relevant for the billing process, namely a month period. In particular, if we consider that the destination AS announces a given prefix P_x over all the links contained in the set of links l_x, we care about which ingress link of the ones contained in l_x is a part of the path selected by the source AS to send traffic towards the prefix P_x.

We model the BGP path dynamics towards the destination of interest as follows. We define the initial state of the interdomain routing and the transitive states of the routing process in the analyzed time period. The initial state consists of the paths used at the beginning of the analyzed time interval by each source AS to reach the prefixes announced by the destination AS. We model this initial set of routes as a random selection between the available BGP paths between each source AS towards any destination prefix. We assume that at the beginning of the interval all the available transit links have the same probability of being a part of the path selected by the a source AS. In other words, if a destination AS announces a prefix over x different transit links, then the probability that any of those links is further a part of the forwarding path from a source towards the destination at the initial state is $\frac{1}{x}$. This implies that if the sink AS with n links is announcing a single prefix through all its links(i.e. $\lambda = 1$), then any source AS will have to randomly select a single route in order to forward its traffic the entire address space of the destination network. If the sink AS is announcing λ fragments of the address space over λ different link sets, then the source AS must randomly select one path for each most-specific prefix announced (i.e. λ different paths) in order to forward its traffic towards the destination AS.

In the rest of the time interval, we analyze the routing state using a time-slotted model. We divide the month into 5 minutes slots, which is consistent with the 95^{th} percentile billing rule currently used in the Internet. We encompass the dynamics of the routing process due to topology or policy changes[3] by considering that in every time slot all the

[3]We do not consider the routing changes due to equipment failures, as these changes cannot be ac-

source ASes are independently repeating the route selection process. We consider that with a given probability p the result of the random selection process is different from the initial-state path. This implies that, for a proportion p of time, traffic may shift away from the initial-state transit link towards another of the remaining equiprobable transit links.

We further intent to quantify the cost paid by the destination AS for performing or not deaggregation in the interdomain by comparing the case in which only one prefix is announced over all links, thus allowing for many routing choices for the traffic sources, and the case in which an unique prefix is announced over one link only, thus strictly reducing the path diversity towards that particular set of addresses. Consequently, we calculate the amount of traffic on each incoming link towards the analyzed destination, while accounting for the path changes that may cause traffic to shift towards or away from the transit link.

6.2.2 Traffic model

In this section, we analyze the traffic distribution on the available incoming links, depending on the manner the destination AS injects its prefix(es) in the interdomain. We assume that the total amount of traffic T received by the destination AS from the N sources is uniformly distributed across its prefix P. This means that if T traffic is sent to P, then if we split P into two more specific prefixes P_1 and P_2, the expected amount of traffic for each of these more specific prefixes is $\frac{T}{2}$. In the case of an uneven traffic distribution, it can be easily proved that a correspondingly proportional prefix fragmentation can be found such that the amounts of traffic per more-specific prefix are comparable. According to [32], the cumulative distribution of inter-domain traffic contributed by all the origin ASes in the Internet approximates a power law distribution. We assume that each source network j included in our model generates an amount of traffic t_j towards a given destination in the interdomain, as depicted in Figure 6.3. We assume that the generated traffic t_j follows a Gaussian distribution in time characterized by the statistical mean μ_j and a variance σ_j^2, which is in line with the results from [82].

6.2.2.1 Distribution of Traffic on Sources

We assume that the traffic generated towards a given destination is distributed among the existing sources according with Zipf's law, as previously described in [83]. This assumption is consistent with the traffic measurements in [32], as the Zipf distribution is a particular case of a power law distribution. The Zipf distribution is one particular example of a power law [84]. A simple description of data following such a distribution is

counted as potential savings since any operational viable deaggregation strategy must support backup links.

the existence of a few elements that have very high values, a medium number of elements with medium values and a huge number of elements with very low values (therefore, the probability of larger values is very low and the probability of low values is very high). Given a ranking of the Internet entities, the Zipf law states that the traffic generated by a network is inversely proportional with its rank. For any destination network we assign the following amount of incoming traffic from AS with rank j:

$$t_j = \frac{\frac{1}{j^\alpha}}{\sum_{k=1}^{N} \frac{1}{k^\alpha}} T = z_j T, \tag{6.1}$$

where z_j is the j ranked element in a Zipf distribution corresponding to AS j. The Zipf distribution includes a parameter α that controls the skewness of the traffic distribution on destination networks. The total amount of transited traffic received by the destination AS can be expressed as the sum of all the traffic contributions $T = \sum_{j=1}^{N} t_j$, for all sources j in the Internet.

6.2.2.2 Distribution of Traffic on Transit Links

The total amount of traffic T consumed by a particular AS in the Internet consists of the contribution of all the sources in the interdomain. We analyze here the traffic distribution on the n ingress links of a destination AS. We capture both the case in which the destination AS deaggregates to different degrees and the case in which the AS does not fragment its address space, and we compare the results.

We begin our analysis by characterizing the distribution of traffic on the incoming links of a destination AS that announces its address space as *one single aggregated prefix*. Consequently, any of the available links towards the destination network can be a part of the traffic forwarding path. We include in Figure 6.4 an example of the traffic dynamics captured in the distribution of traffic per transit link. For a given destination AS with n links we define the subset s_i of sources which have as initial state path a route which includes link i, where $i = 1, n$. For example, in figure 6.4, subset s_1 includes all the source networks that have chosen transit link 1 in the initial phase of the model. Due to the fact that each link has the same probability of being chosen by each source for traffic forwarding in the initial state of the interdomain routing process, the expected value of the size of source sets s_i is of $\frac{N}{n}$ ASes. Consequently, when announcing the same prefix over all the links, the incoming traffic on each link in the initial state of the routing process has a statistical expected value of $\frac{T}{n}$.

When dividing the month in many equal-sized time-slots, we further consider that the route selection process happens for each source AS in every slot. Therefore, at the beginning of every time interval in the analyzed period, with a probability p the newly chosen forwarding path is different from the one used in the initial state. This would

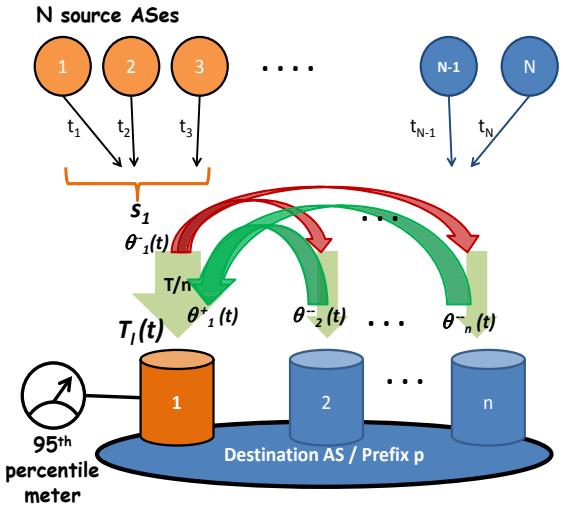

Figure 6.4: Traffic dynamics for each transit link.

trigger the shift of a certain amount of traffic from link l to the rest of the links for the destination AS and the other way around.

We denote with $\theta_l(t)$ the random variable which represents the traffic leaving at moment t from link i and dividing among the rest of the transit links, as we can observe in Figure 6.4. The unstable traffic $\theta_l(t)$ leaving link i at moment t can be further expressed as $\sum_{j \in s_l} q_j(t) t_j$, where t_j represents the amount of traffic generated by source AS j, q_j is either 1 if at moment t link i is a part of the forwarding routed used by the source AS j or 0 in the contrary case and s_l represents the set of sources with initial state path including link i. Formally,

$$\begin{aligned} &P(q_j = 1) = p; \\ &P(q_j = 0) = 1 - p. \end{aligned} \tag{6.2}$$

Consequently, the unstable traffic leaving any link l at time t follows a Binomial distribution, i.e. $\theta_l \approx Binomial(\frac{N}{n}, p)$, with $l = 1, n$. Thus, the mean and variance of the unstable traffic leaving a link, i.e. $\theta_l(t)$ with $l = 1, n$, has the following expression for any of the n links:

$$\begin{aligned} &\tilde{\mu}_l = p\frac{T}{n}; \\ &\tilde{\sigma}_l^2 = p(1-p)\sum_{j \in s_l} t_j^2. \end{aligned} \tag{6.3}$$

When analyzing the traffic on a link we also have to consider the traffic moving towards the current link i from the rest of the links $k \neq i$. This incoming traffic represents only a fraction of the entire instable traffic moving away from any link $k \neq i$. We denote with

$\theta_k(t)$ the traffic leaving any link k, where $k \neq i$. Similar to the case of link i, we can express $\theta_k(t)$ as $\sum_{j \in s_k} q_j(t) t_j, k \neq i$, where q_j is either 1 or 0 depending if at moment t link i is a part of the forwarding routed used by the source AS or not. The traffic shift probability is equal to the probability of path change p, i.e. $P(q_j = 1) = p$. The total instable traffic is represented by $\sum_{k \neq i} \theta_k(t)$. This amount evenly splits between all the $n-1$ equiprobable alternative links, including the analyzed link i. Consequently, the expected value of the incoming traffic on link i is represented by the $\frac{1}{n-1}$ part of all the total unstable traffic, i.e $\frac{1}{n-1} \sum_{k \neq i} \theta_k(t)$. This random variable also follows a Binomial distribution, as it represents the addition of $n-1$ independent Binomially distributed random variables.

We can now express the total volume of traffic on each link towards the destination, which changes at every time-slot t like showed in the following expression:

$$T_i(t) = \frac{T}{n} - \theta_l(t) + \frac{1}{n-1} \sum_{k \neq i} \theta_k(t), \tag{6.4}$$

where $\theta_l(t)$ represents the traffic leaving link i and $\frac{1}{n-1} \sum_{k \neq i} \theta_k(t)$ represents the expected value of the traffic shifting from the rest of the links to link i. This yields the following expressions for the mean and variance of the total incoming traffic on a given link l:

$$\begin{aligned} \mu_l^{(i)} &= p\frac{T}{n}; \\ \sigma_l^{(i)2} &= p(1-p) \left(\frac{1}{n-1} \right)^2 \sum_{k \neq l} \sum_{j \in s_k} t_{ij}^2. \end{aligned} \tag{6.5}$$

Therefore, the expressions for the statistical mean and variance for the total traffic on link i when a single prefix is announced over all the available links are:

$$\begin{aligned} \mu_i &= \frac{T}{n}; \\ \sigma_i^2 &= p(1-p) \left[\left(1 - \frac{1}{(n-1)^2} \right) \sum_{j \in s_i}^{|s_i| = \frac{N}{n}} t_j^2 + \frac{1}{(n-1)^2} \sum_{j=1}^{N} t_j^2 \right]. \end{aligned} \tag{6.6}$$

When each origin AS deaggregates the assigned address block into λ more-specific even prefixes, where $\lambda \leq n$, and announces them over different link sets, different parts of the address space are reachable only through a subset of the total incoming links. Consequently, the source ASes split their traffic evenly for the deaggregated prefixes and choose one path for each fraction of the address space. The size of the set of source ASes with an initial path including one of the incoming links towards a destination prefix is equal with $|s_i| = \lambda \frac{N}{n}$. These sources do not send all their traffic on the chosen link from a

given link set, but they only send the corresponding fraction of traffic for the destination prefix, i.e. $\frac{t_j}{\lambda}$.

When the destination AS deaggregates its address block in a number of prefixes smaller than the number of available transit links, we have λ sets of links including the $\frac{n}{\lambda}$ different links. Consequently, the amount of traffic on each transit link T_l at time t has the following expression:

$$T_i(t) = \frac{T/\lambda}{n/\lambda} - \theta_l(t) + \frac{1}{\frac{n}{\lambda} - 1} \sum_{j \neq i} \theta_j(t). \tag{6.7}$$

The unstable traffic from link i to the other $\frac{n}{\lambda} - 1$ links in the same set is now $\theta_l(t) = \sum_{j \in s_i}^{|s_i| = \frac{\lambda N}{n}} q_j(t) \frac{t_j}{\lambda}$, where $|s_i| = \frac{\lambda N}{n}$ represents the number of sources in the source set s_i, directly proportional with the number of announced prefixes in the interdomain. We observe that, while the mean value of the incoming traffic remains the same as in the previous case, i.e. $\mu_{l,\lambda} = p\frac{T}{n}$, the expression of the traffic variance becomes:

$$\begin{aligned} \sigma^2_{i,\lambda} =& p(1-p) \left(\frac{1}{\lambda^2} - \left(\frac{1}{n-\lambda} \right)^2 \right) \sum_{j \in s_i}^{|s_i| = \frac{\lambda N}{n}} t_j^2 + \\ &+ p(1-p) \left(\frac{1}{n-\lambda} \right)^2 \sum_{j=1}^{N} t_j^2. \end{aligned} \tag{6.8}$$

When evaluating the traffic variance with the expression in (6.8) relative to the one in (6.6) we observe that as the number of announced prefixes λ is increasing, the amount of traffic on each link becomes more stable, as its standard deviation is decreasing. This phenomenon is explained by the fact that, as the number of prefixes increases, so does the number of disjoint link sets. We have observed that in the case where only one prefix is announced over all links, there is only one link set including all the available transit links. Contrariwise, when announcing more prefixes, the number of link sets is increasing, also implying that the amount of incoming traffic and the number of different links included in each set are proportionally decreasing. This further translates into a decreasing number of routing choices for the traffic to reach the more-specific prefix announced on a specific link set.

Comparing the two expression from (6.6) and (6.8), we obtain the following ratio between the two variances:

$$\gamma = \left(\frac{n-1}{n-\lambda} \right)^2 \frac{1 + \left[\left(\frac{n-\lambda}{\lambda} \right)^2 - 1 \right] \frac{\sum_{j \in s_l}^{|s_l| = \frac{\lambda N}{n}} t_j^2}{\sum_{j=1}^{N} t_j^2}}{1 + ((n-1)^2 - 1) \frac{\sum_{j \in s_l}^{|s_l| = \frac{N}{n}} t_j^2}{\sum_{j=1}^{N} t_j^2}}. \tag{6.9}$$

For $2 \leq \lambda < n$ and approximating $\sum_{j \in s_l}^{|s_l|=\frac{\lambda N}{n}} t_j^2 \approx \lambda \sum_{j \in s_l}^{|s_l|=\frac{N}{n}} t_j^2$, we conclude that $\gamma < 1$, which indicates that when the degree of deaggregation is increasing, the variance of the traffic on a link is smaller. This results in a diminution of the traffic burstiness and even smaller fluctuations in the total amount of chargeable traffic.

In the extreme case when the AS is announcing as many more-specific prefixes as number of transit links, the size of the set of sources with a route that includes link i in the initial state is $|s_i| = N$. In other words, every source AS installs in its routing table a stable path for each transit link for the destination AS. This implies that the traffic shifting from one link to the others is zero and, similarly, the traffic incoming from the rest of the links is also null. Therefore, the variance of the traffic on each link resulting from route changes is zero $\sigma_i^2 = 0$, as the traffic forwarding paths are very stable. Consequently, the incoming traffic on each link equal with $\frac{T}{n}$ does not fluctuate during the analyzed period, as this amount of traffic is confined to the preferred incoming link.

6.2.3 The Cost Model

The 95^{th} *percentile rule* is currently the most widely-spread billing method among ISPs [30]. This method usually implies that the agreed billing period (usually a month) is sampled using a fixed-sized window, each interval yielding a value that denotes the traffic transferred during that period. The resulting intervals are sorted and the 95^{th} percentile of this distribution is used for billing [30]. Consequently, this billing method is considered as a compromise between billing a customer based on the absolute traffic usage or based on the capacity of the transit links and the peak rates.

A recent transit cost survey [85] has shown that the price per unit of transfered traffic, denoted here by c_t, decreases with the increase of the expected volume of transit traffic following a convex dependency. However, this is only true when the increase of the expected amount of traffic significant i.e. one order of magnitude. In the case where the increase of expected traffic volume is in the same size range as the initial traffic volume, the cost per traffic unit remains constant. We assume that the variations in traffic do not change the order of magnitude of the received traffic, therefore we can also assume a linear cost function for the transit traffic.

In our model we include a cost function with the following expression:

$$C = c_t * V, \tag{6.10}$$

where V is the charging traffic volume (i.e. the 95^{th} percentile of the monthly traffic) of the destination AS i and c_t is the corresponding transit traffic unit cost. We consider that the total charging traffic volume for any destination AS, represents the addition of all the chargeable traffic volumes on each incoming link, and therefore can be expressed

as

$$V = \sum_{i=1}^{n} \left(\mu_i + 1.96\sigma_i\right), \tag{6.11}$$

where n represents the number of incoming link for the destination AS, and μ_i and σ_i have the expressions from (6.6). Given the fact that the traffic on link l follows a Binomial distribution $B(N, p_i)$, we can approximate it with a *Normal (Gaussian) distribution* $N(\mu_i, \sigma_i^2)$. The expression $\mu + 1.96\sigma$ from (6.11) represents the estimation of the 95^{th} percentile of a Normal random variable $N(\mu, \sigma^2)$ representing the individual traffic volume on the incoming links.

In order to capture the full impact of deaggregation on the transit traffic bill, we focus on the amount of chargeable traffic in the two extreme cases: (i) no deaggregation: $\lambda = 1$, (ii) strategic deaggregation: $\lambda = n$. We calculate next the total amount of chargeable traffic on each link, i.e. the 95^{th} percentile of the link traffic, when no deaggregation is performed by the destination AS, i.e $v_i|_{\lambda=1}$ and when the number of prefixes announced is equal to the number of available links, i.e. $v_i|_{\lambda=n}$:

$$\begin{aligned} v_i|_{\lambda=1} &= \frac{T}{n} + 1.96\sqrt{p(1-p)}\sqrt{\sum_{j\in s_i} t_j^2 + \frac{1}{(n-1)^2}\sum_{k\neq i}\sum_{j\in s_k} t_j^2}; \\ v_l|_{\lambda=n} &= \frac{T}{n}. \end{aligned} \tag{6.12}$$

We can easily observe that the additional traffic on each link is

$$\begin{aligned} \gamma_i &= v_i|_{\lambda=1} - v_i|_{\lambda=n} = \\ &= 1.96\sqrt{p(1-p)}\sqrt{\sum_{j\in s_i} t_j^2 + \frac{1}{(n-1)^2}\sum_{k\neq i}\sum_{j\in s_k} t_j^2}. \end{aligned} \tag{6.13}$$

Furthermore, the difference in the total charging traffic volume for the analyzed destination AS with n links can be expressed as the sum of the traffic fluctuations in all the links, i.e. $V = \sum_{l=1}^{n} \left(v_i|_{\lambda=1} - v_i|_{\lambda=n}\right)$. This yields the following expression for the total volume of additional chargeable traffic:

$$\gamma = \sum_i \gamma_i. \tag{6.14}$$

The savings in transit traffic bill represent the cost c paid for the burstable, unstable traffic. Consequently, the additional cost emerging from path instability in the interdomain is

$$c = \gamma c_t. \tag{6.15}$$

Henceforth, the saved amount in the transit traffic bill represents a fraction of

$$RS = \frac{\gamma}{T + \gamma} \tag{6.16}$$

out of the actual price paid for the consumed traffic without deaggreagation. Substituting the generated traffic t_j for every source AS j with the expression in (6.1) yields that the relative transit traffic savings are a function of the number of links towards the destination AS, the instability probability p and the Zipf distribution skewness parameter α, which does not depend on T : $RS = f(n, p, \alpha)$.

6.3 Quantifying the BGP Path Dynamics

In this section, we use the previously proposed model to obtain a first approximation of the economics impact of strategic deaggregation. We then validate the model by contrasting these results derived from the model with results both from simulations and from real data-driven estimations using publicly available BGP traces.

In order to obtain numerical values for the potential savings in the transit costs resulting from strategic deaggregation, we need first to assign realistic values to three parameters, namely, N - the number of traffic sources, α - the skewness parameter and p - the instability probability. N stands for the number of ASes in the Internet, which can directly be extracted from the records maintained by the Internet Registries. At the time of the analysis, it is in the order to $36,000$. α is the skewness parameter for the Zipf distribution on the traffic source for which the current state of the art [86, 83, 14] assigns a value of 0,9.

p - on the other hand - is a parameter specific to our model and requires additional work for its estimation. As p is the probability of a change in the ingress link used by the source ASes to send traffic towards the destination, we estimate it by analyzing real BGP data. We will next present the data set used and then we estimate p.

6.3.1 Data Set

The data set used includes the BGP routing table snapshots, i.e. the instance of a routing table taken at a certain time, taken every 8 hours from a sample of of 66 different ASes present in the RIPE database [22], during 6 months, from December 2010 until May 2011. This adds up to a total of more than $35,000$ snapshots of full routing tables, containing the BGP routing information of the sources in question towards more than $300,000$ destination prefixes. Some relevant statistical properties of our data set are the distribution of sizes of the sampled ASes and the distribution in the number of providers the destination ASes have, as we describe next.

The sample of 66 analyzed source ASes includes networks with ranks that vary between

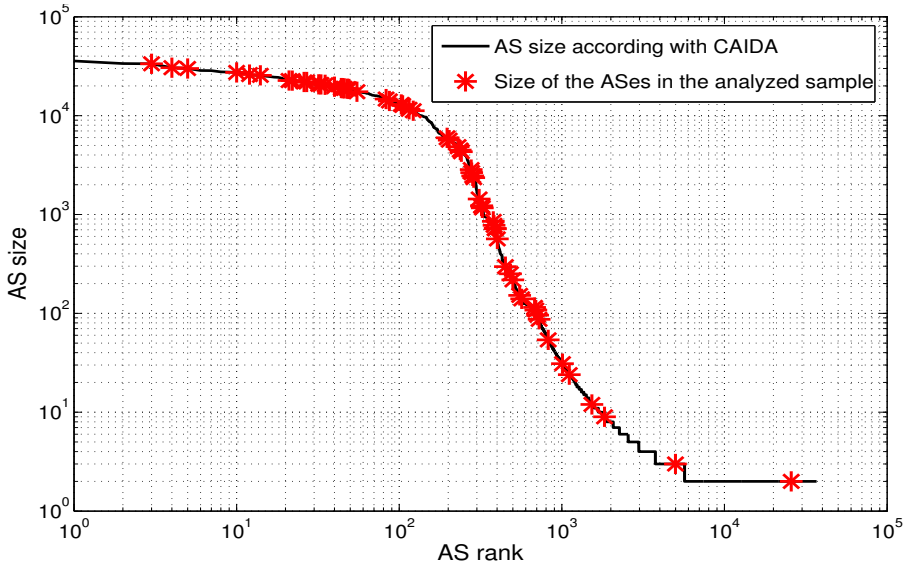

Figure 6.5: Distribution of the considered AS sample on the entire interdomain space.

3 and 25,700, according with the CAIDA ranking [87,88] of ASes. The ranks are assigned in function of the customer cone size (i.e., the proportion of IPv4 prefixes which can be reached by an AS following only transit links). We observe in figure 6.5 the distribution of the AS sample on the entire interdomain space. Even if the sample size only includes 0.2% of the whole set of ASes present in the interdomain, they are responsible for generating around 10% of the interdomain traffic. We make this approximation based on the assigned CAIDA ranks and assuming traffic proportional to the customer cone size. Consequently, the sample of analyzed source ASes is a representative sample both from the point of view of the amount of generated traffic and also the ranks distribution on the entire CAIDA rank set.

We estimate the number of different upstream providers per destination by analyzing the routing information dataset corresponding to the month of January and identifieng the *second last-hops* (2LH) in the paths installed in the routing tables. For each network receiving traffic from all the source ASes in the analyzed sample, we find the set of unique 2LHs used by the source networks to reach the prefixes announced by the destination.

In Figure 6.6 we can observe the CDF corresponding to the number of different second last hops per destination network.

For each different number of transit links per AS identified in the data set we extract from Figure 6.6 a weight which represents the number of destination networks which are reached through that number of different transit links. For example, 42% of the destination ASes are reached through two different transit links. These are the weights we are going to use in the following section to generate a weighted average of the overall savings throughout the Internet. We do so in order to be able to compare the savings

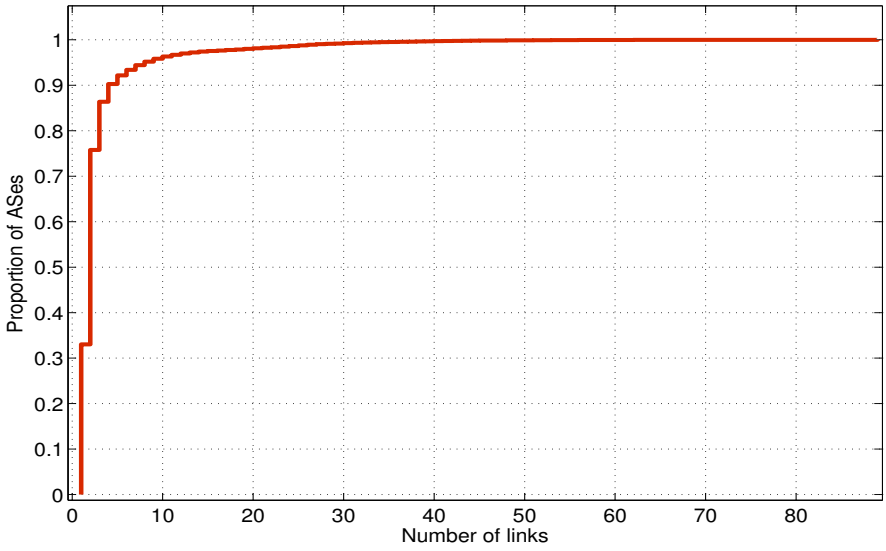

Figure 6.6: Empirical cumulative distribution function of the number pf links per destination AS.

obtained in the model and the simulations with the ones obtained from analyzing this particular data set. Additionally, we observe from Figure 6.6 that approximatively 95% of all the analyzed networks in the interdomain have at most 10 transit links. For this reason, we focus next only on studying the behaviour of the networks with at most 10 transit links. Using the assigned weights for different values of the number of links allows us to correctly quantify the mean amount of savings for a destination network in the Internet which we calculate using the weighted average.

6.3.2 Estimation of the instability probability

In order to estimate the probability of transit link instability, we further observe the changes in the 2LHs included in the AS paths towards the destination prefixes installed by the source in the analyzed BGP routing tables. For each source-destination pair of ASes, we calculate the probability that in a given interval the source AS is not using the transit link selected in the initial state of the interdomain routing towards the destination prefix. For each of the 66 sources analyzed, we evaluate the relative time the source AS is not using the path announced in the first time slot of the analyzed period towards every destination prefix in the interdomain. Next we match every destination prefix to the originating AS and average the time spent on an alternative path for all the prefixes announced by the destination AS. We thus obtain the probability that a source uses a path towards each destination AS which is different from the path used in the initial state of the routing process. We approximate the probability parameter in our model with the mean value of the transit link instability probability over all the observed sources,

yielding a value of $p = 3.5\%$.

6.3.3 Analytical Model Savings Quantification

We observe in Figure 6.7 the model estimated savings for a destination AS with a variable number of links. For an instability probability of $p = 3.5\%$ in the interdomain and a Zipf distribution of traffic with a skewness parameter of 0.9, a destination AS with $n = 1, 10$ transit links may have an additional cost incurred by the route instabilities in the interdomain that can reach up to 6.5% of the transit traffic.

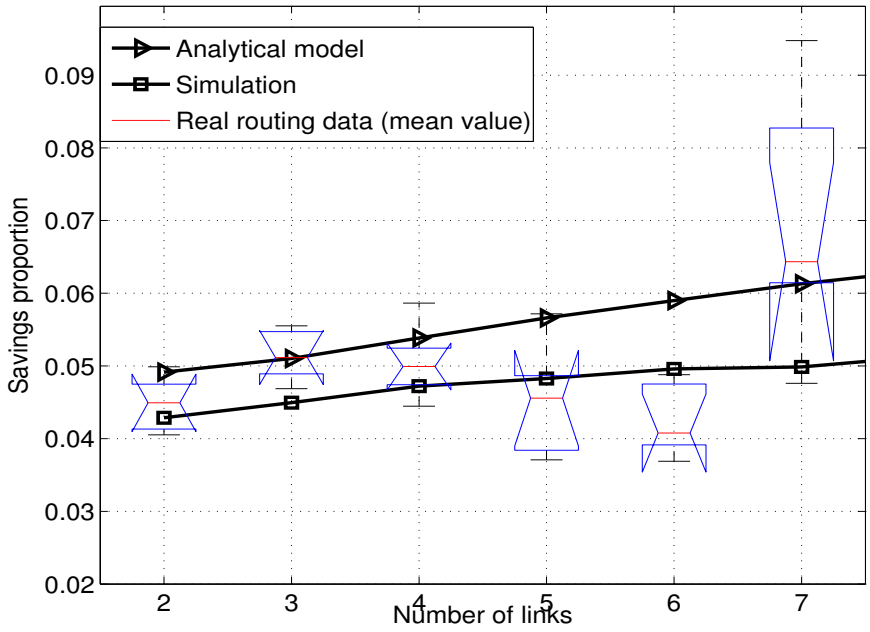

Figure 6.7: Model generated savings curve for an AS with at most 10 transit links, considering a transit link instability probability of 3.5% and a skewness parameter $\alpha = 0.9$. Comparison with simulation results and data-driven savings approximation.

We calculate next the weighted average of savings in the interdomain. We weight the approximated mean savings for a destination AS with n transit links, with the corresponding weights in Figure 6.6. We obtain an average value of the savings equal with 5.2% of the usual transit traffic bill.

6.3.4 Model Validation through Simulation

In order to check the accuracy of the results obtained by using the analytical model, we simulate the model for a destination AS with n transit links, where $n = 1, 10$. We consider that the destination ASes receive traffic from 36,000 source ASes, following a Zipf distribution with skewness parameter equal with $\alpha = 0.9$, as previously approximated

in [86].

We perform the sampling of the level of traffic on each transit link in 100 different equal-sized time-slots which cover the entire period of one month. In the first time-interval each source AS randomly chooses one of the n equiprobable providers in order to reach the destination AS. In the remaining time-slots we simulate the BGP selection algorithm as a random process in which with a probability 3.5% the source ASes use a different link from the initially chosen one. We sort in increasing order the samples for the values of the traffic and define the 95^{th} percentile of the traffic on each link. Afterwards, we use the formula in (6.16) to evaluate the proportion of savings if deaggregation would be performed in an efficient manner.

We run this simulation for 100 times with the same parameters for each different number of links $n = 1, 10$. We thus obtain the mean value for the savings percentage of each destination AS with n transit links with less than 1% margin of error at 95% confidence level. We observe in Figure 6.7 the curve of average savings for a destination with n links, where $n = 1, 10$. We next calculate the weighted average of the savings on the destination ASes, like in the previous sections. We obtain a value of 4.5% of savings on the transit traffic bill.

The difference between the model and the simulation results which can be observed from Figure 6.7 can be explained by the fact that the analytical model uses as an approximation for the 95^{th} percentile the Normal-based approximation confidence interval, i.e $\mu + 1.96\sigma$. This does not occur in the model simulation, where we have all the samples of the discrete Binomial Distribution, for which we can easily define the 95^{th} percentile of the link traffic level. We verify that this is the root of the discrepancy by comparing the by checking that the the mean and the standard deviation of the traffic on a transit link in the model and in the simulation match. For example, for a destination AS with two transit links, we have assumed in the model evaluation a value of the mean traffic of 0.5 of the entire incoming traffic T, which accurately matches the value estimated by the model simulation. The standard deviation of the traffic on a link represents 0.0130 of the whole incoming traffic in the model evaluation, and 0.0100 of the whole incoming traffic in the simulation evaluation.

6.3.5 Savings Quantification using Real Routing Data

In this section we contrast the previously estimated numerical values for the transit link instability costs with approximations performed based on actual routing information. For this purpose, we process the BGP routing data present in the RIPE database corresponding to *6 months*, from December 2010 until May 2011 generated by 66 different sources. From comparing the routing tables we can evaluate the actual amount of routing changes towards a given destination AS in the Internet. Therefore, we no longer generalize the path change dynamics for all the ASes in the sample by considering the same

value of the change probability p. Consequently, the path changes described by p are here substituted with genuine path changes inferred from comparing the routing tables over 6 months time. However, one issue with the data set is that it includes changes in routing due to failures in the ingress links. These changes cannot be accounted for potential savings since any operationally viable deaggregation strategy must support backup links. This means that this type of fluctuations in the routes also affects the case where the origin AS is injecting deaggregating prefixes. In our data analysis, we do not account for routing changes which are due to failures in the ingress links, since any operationally viable deaggregation strategy must support backup links. In order to filter out these cases, we remove from our analysis the destinations with a non-constant number of transit links present in each monthly data set. This approach is likely to remove a superset of the ingress-link failure cases, making our result to be only a *lower bound* of the potential savings.

When performing the reality based approximation of the savings, the Binomial approximated distribution of traffic is no longer needed, as we can infer the amounts of traffic on each link from evaluating the actual contribution of each source on every link. From the Zipf distribution with 36,000 elements and $\alpha = 0.9$, we extract only the 66 elements corresponding to the sample of ASes.

In Figure 6.8 we observe that the percentage of savings for the destination ASes relative to the traffic generated by the 66 sources is similar over the 6 months period we have analyzed. This suggests that the transit link dynamics in the interdomain maintain the same per-month profile during the 6-months analyzed period.

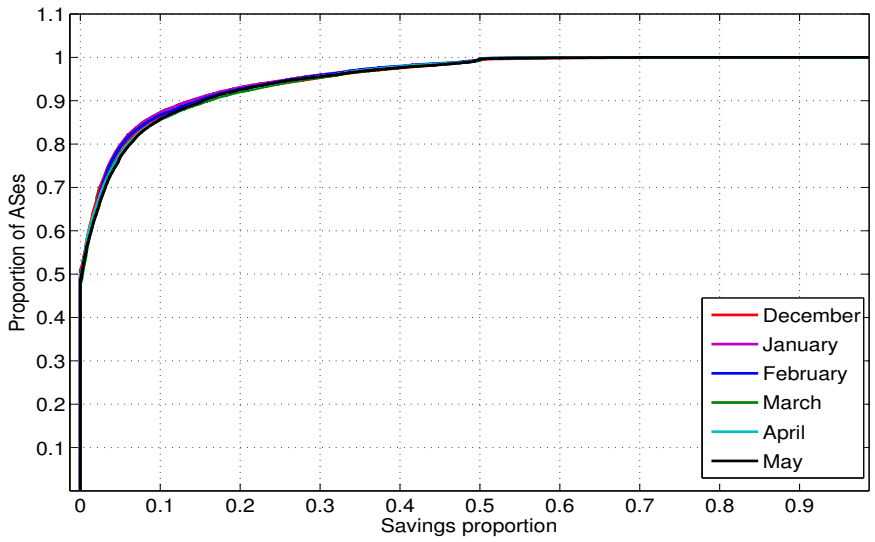

Figure 6.8: The empirical CDF of the savings proportions corresponding to the 6 months of analyzed routing data.

In Figure 6.7 we observe the savings estimated with real routing data for the destina-

tion ASes with $n \in [2, 7]$ transit links. The boxplot for each case shows the savings over the 6 months, where the central mark is the median, the edges of the box are the 25^{th} and 75^{th} percentiles, the whiskers extend to the most extreme data points not considered outliers. With 95% confidence level, the average amount of savings for an AS with 2 upstream providers lies in the confidence interval $[4.2\%, 4.77\%]$, which is consistent with the previous approximations.

6.4 Summary

In the previous chapter, we looked at the correlation between the global prefix visibility and its global reachability. We have found that the DPs are the most affected by limited reachability, especially in the case of IPv6 prefixes. However, majority of LVPs which are more-specific prefixes covered by less-specific HVPs are generally reachable, even is the traffic follows the ill-favored path corresponding to the covering prefix. In this chapter, we further direct our attention towards the impact that more-specific prefixes with intended limited visibility have on the routing system. The selective advertisement of deaggreagated prefixes, defined as strategic deaggregation, is widely used as a traffic engineering technique in the Internet. We show here that, independently of the main reason driving the adoption of this technique, strategic deaggregation presents with one particular collateral benefit for the networks deploying it, namely a reduction of their transit bill.

The proposed Internet model allows our analysis to focus on the impact of different deaggregating strategies on the transit traffic stability and ultimately on the transit cost incurred on the customer ASes. The results presented in this chapter are the direct consequence of the current operational status of the Internet, as the latter is characterized by the unique mixture between the BGP-specific routing mechanism, the billing model and the difference in amount of generated traffic between different source networks. Consequently, the relative savings approximated with the proposed model depend on three parameters, namely N (the total number of ASes in the Internet), α (the skewness parameter of the Zipf distribution of traffic on sources) and p (the path change probability). Based strictly on the information contained in public BGP routing tables, we show that strategic deaggregation may decrease the transit bill of a given customer by 5% in average. Clearly, in the operational Internet some of these ASes that were taken into account for this approximation are much more affected by routing changes than others, and thus may experience . However, this average gives us a somewhat concrete idea on the manner in which the combination of routing changes, the skewed distribution of traffic on sources (analyzed in [32]) and the popular 95% percentile billing scheme [30, 31] inflate ones transit bill.

Chapter 7

Strategic Deaggregation Detection

In this chapter, we focus on identifying real-life occurrences of the strategic deaggregation scenario observed and analyzed in the previous Chapter 6. Here, we take the point of view of a transit provider to a deaggregating customer and ask a two-staged question:

(1) How extensive is the use of prefix deaggregation among the customer networks? We further propose a methodology to identify cases of deaggregated prefixes within the customer base of an operational ISP within a certain time-window. We enable any operator with the necessary tools to detect the customers which are new deaggregators and and monitor their behavior in time.

(2) Can it be verified that deaggregation combined with selective advertisements could decrease the transit bill of some customers? Customers which exhibit this behavior may be able to game the 95^{th} percentile billing rule and possibly have a negative impact on the business of their ISPs. We propose a passive measurement approach to detect strategic deaggregation events and to asses their economic consequences.

Our approach requires obtaining and processing routing, topology, traffic and billing information and molding it in order to reach the correct level of understanding on the impact different customers might have on the business of their providers. The novelty of this methodology is the manner in which it merges different types of information characteristic to a transit provider, in order to have a complete picture on the operations of its customer networks.

Any ISP interested in detecting the occurrence of this phenomena within its customer base can construct the dataset containing all the various batches of different data and apply the proposed processing methodology. We show how any ISP can monitor the amount of deaggregation generated by its customers and quantify the impact strategic deaggregation may have on its own revenues. The central reasons for the customer network operator to deploy the deaggregation strategy in the first place may include a wide variety and may or may not be related to decreasing ones transit traffic bill. The transit provider, however, might be impacted by its customers' choices in terms of deaggrega-

tion. As previously proven in the Section 6.2, the unique interaction of three important characteristics of the current Internet, namely the interdomain path changes, the 95^{th} percentile billing rule broadly used in today's Internet and the skewed distribution of the traffic demand on sources, make if possible for networks to inadvertently decrease their transit traffic bill using strategic deaggregation. We further argue that, by simply detecting the cases of strategic deaggregation, a transit provider can identify the customers who may unknowingly impact in a negative way its revenues. Furthermore, detecting a large number of such cases in ones customer base might provide the necessary incentives for the adoption of a more suited billing model than the sub-optimal 95^{th} percentile billing rule.

We propose a passive measurement approach for the detection of strategic deaggregation events and to asses their economic consequences. The novelty of the approach is the manner in which it merges different types of information characteristic to an ISP in order to have a complete picture on the operations of its customer networks. This requires obtaining and processing routing, topology, traffic and billing information and molding it in order to reach the correct level of understanding on the economic impact different customers might have on their transit providers. Any ISP interested in detecting the occurrence of this phenomena within its customer base can employ the proposed methodology.

The methodology is structured in three parts, each conveying relevant results concerning deaggregation dynamics within the customer base of a transit provider. We summarize in Figure 7.1 the steps taken in the methodology and show which type of information is required for each part.

Step 1: *Detect more-specific prefixes.* First, we detect ASes which change their behavior and start using deaggregation within a predefined time-window. For this step we require the BGP routing information from the ISP (i.e., the global BGP routing table from the transit provider), as depicted in the first processing block in Figure 7.1. We expand on the mechanism in section 7.1.

Step 2: *Detect strategic deaggregation.* Second, we check for selective advertisements of the more-specific prefixes previously identified. As depicted in the second processing block from Figure 7.1, in this step we use all the BGP routing information from the monitors active in the the RIPE RIS and RouteViews projects.

Step 3: *Evaluate economic impact.* Third, we try to determine if performing strategic deaggregation leads indeed to economic benefits for the customer network. For the cases of strategic deaggregation, we monitor the traffic data both (i) before deaggregation, when the address block is injected as one prefix to all providers (i.e. *no deaggregation*), and (ii) after the strategic deaggregation, when the address block is fragmented into as many more-specific prefixes as the number of transit providers and each more-specific is selectively advertised to a different provider (i.e. *strategic deaggregation*). It is important

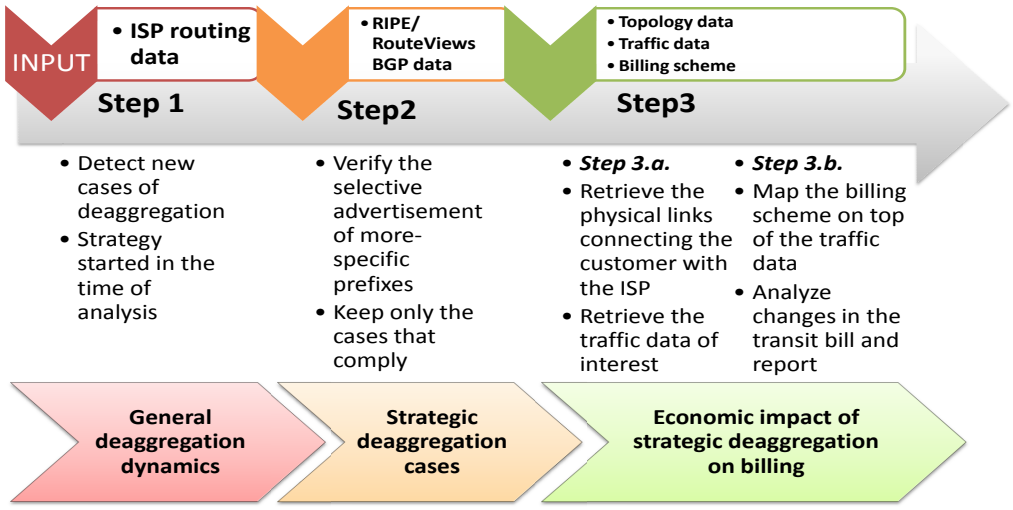

Figure 7.1: The methodology steps: at each step we require a different input dataset depicted at the top of each processing block. At the bottom of each block, we can see the results we obtain at each step.

to capture both these states, in order to be able to correctly quantify the economic impact of strategic deaggregation. We evaluate the transit bill for each case and compare. This is depicted in the third processing block from Figure 7.1.

Step 3.a: We extract the traffic data on all the links connecting the provider with the identified customer which is deploying strategic deaggregation. This requires a previous mapping between customers and transit links from the ISP. We obtain this topology data after parsing all the router configuration files provided by the ISP. This step is further depicted in the first sub-block of the third processing block in Figure 7.1.

Step 3.b: Finally, we move to *estimating the bill for the aggregated and deaggregated traffic patterns*. Thus, by applying the ISP's billing scheme to the traffic traces, we can quantify the impact of strategic deaggregation on the transit traffic bill. This step is depicted in the last processing block in Figure 7.1.

The methodology is aimed at working with a large and diverse collection of real data. Any ISP interested in detecting the occurrence of this phenomena within its customer base can build the dataset and employ the proposed methodology. Moreover, the tools we have developed are publicly available[1] for the research community.

7.1 Detection of Deaggregation Events

The detection algorithm we propose in *Step 1* for the identification of more-specific prefixes performs a comparative analysis of the BGP information obtained from the ISP.

[1]The code is available to be downloaded from `http://fourier.networks.imdea.org/people/~andra_lutu/ITC25_code/` .

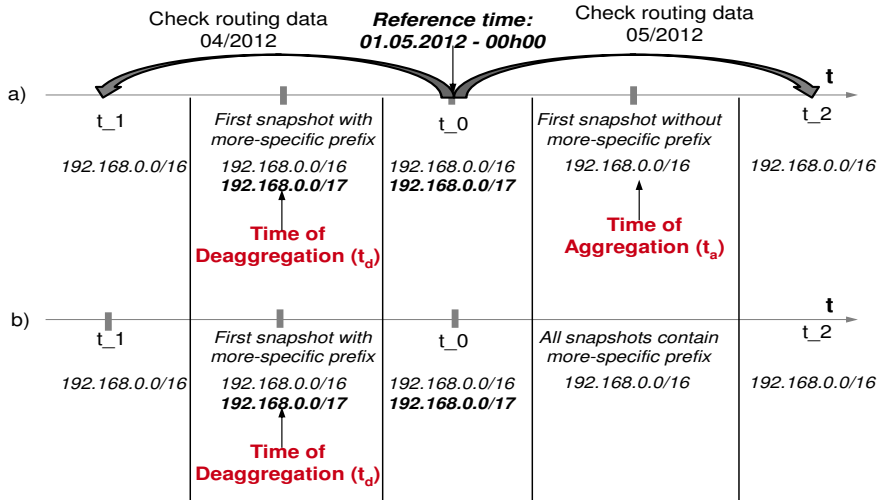

Figure 7.2: Algorithm used for detecting new customer deaggregation events. In the case where the algorithm detects that the more-specific prefix has been re-aggregated, i.e., the more-specific no longer appears in the routing table snapshot during one month at least, as depicted in case a), then the deaggregation event is discarded. We aim to detect cases of deaggregation which last at least for one billing period, i.e., where the more-specific prefix can be seen in the routing table snapshot for at least one month after the inferred time of deaggregation, as depicted in case b).

The different states of the algorithm are depicted in Figure 7.2. We begin by choosing a *reference routing table*. The time-stamp of the reference routing table represents the *reference time*. The detection algorithm identifies the customer prefixes based on the information from the provider (for example, customer routes are tagged with specific informational communities). We assume deaggregated prefixes exist at the reference time and we verify if the more-specific prefixes started to be advertised within the month prior to the chosen reference time. We progressively contrast the content of the reference routing table with each of the previous routing tables collected for a certain period before the reference time. As depicted in Figure 7.2, we verify the routing information from as much as one month before the reference time in order to capture the dynamics of prefix deaggregation in a timescale that is consistent with the billing period. The analysis of the prefixes advertised by the customer ASes during this particular time-window allows us to separate the *newly injected* more-specifics prefixes, which first started to be injected in the month prior to the reference time. It further separates this cases in more-specifics that are not active for at least one months post-deaggregation (i.e., the situation depicted in Figure 7.2.a)) and, contrariwise, more-specifics that are active for one month post-deaggregation (i.e., the situation depicted in Figure 7.2.b)). This also enables us to determine the presence of a covering prefix injected by the customer network and the approximative moment of deaggregation.

The algorithm can be run on longer timescales (e.g. two months, three months, one

year etc.), thus allowing the ISP to get a bigger picture on the deaggregation dynamics within its customer base at different timescales.

7.1.1 The Two-by-Two Routing Tables Comparison

We contrast the entries from the reference routing table with any other routing table collected in the period of analysis, to which we further refer as a *pair* routing table. We begin by first defining the set of prefixes present *only* in the reference routing table by separating the prefixes advertised only at the reference time and not present in the pair routing table, i.e

$$\Delta_i = P_{\text{ref}} - P_i \tag{7.1}$$

where P_{ref} represents the set of prefixes in the reference routing table and P_i, the set of prefixes installed in the paired routing table. For each of the prefixes in the Δ_i set defined above, we use a digital tree search [89] to identify the covering prefixes among the entries in the pair routing table. Assuming that no network is less specific than a /8, we are thus able to rapidly build the covering digital tree corresponding to each of the prefixes of interest. From each tree, we retrieve the least-specific prefix, i.e. the tree root, which we further use in the traffic data analysis. We do not examine the intermediate prefixes (shortly appearing intermediate phases in the deaggregation process), since for these there exists a more-specific prefix which can influence the manner in which traffic flows towards the destination.

By performing this comparative study using all the periodically collected routing tables from the ISP, we obtain an accurate picture of the evolution of the prefix deaggregation dynamics within the customer base of the provider. We monitor the changes of the previously defined prefix sets Δ_i during the analysis interval. The approximative time of deaggregation is, at the latest, the collection time of the first routing table snapshot which contains the candidate more-specific route known to already be installed in the reference routing table. This moment is marked in the time-line depicted in Figure 7.2 as the first moment where the more-specific prefix and the covering prefix are both present in the pair routing table.

7.2 Sifting the Results

In order to correctly identify the long-lived deaggregation events which may have an economic impact, we need to make sure that the retrieved more-specifics are not sporadic events. We discard from our analysis the cases of prefixes which, as depicted in Figure 7.2.a), get re-aggregated in their less-specific covering prefix shortly after the deaggregation was performed. What we are interested to analyze further are cases of

deaggregation which match the setting in Figure 7.2.b), where the more-specific is active for a month after the time of deaggregation.

We apply the same detection algorithm to identify potential re-aggregation cases of more-specifics into their covering prefixes which might happen in the month after the moment of deaggregation. We perform this latter step in order to assure that from the results provided by the algorithm we select only the more-specific prefixes that remain installed in the routing table for at least one month from the moment of deaggregation, and thus may impact the transit traffic bill. For avoiding cases of dynamic deaggregation-aggregation behavior, we filter out prefixes with intermittent presence in the routing tables, i.e. with a presence time lower than 5% of the billing period.

Past the reference time, the previously described two-by-two comparison algorithm actively detects cases of re-aggregated more-specific prefixes in the Δ_i set. We approximate the time of re-aggregation with the collection time of a pair routing table which contains only the covering prefix after the reference time.

7.2.1 Validation of Selective Advertisements

The selective advertisements validation process is further integrated in Step 2. We combine the internal routing view from the ISP with the external views taken from the ASes participating in the RIPE RIS and RouteViews project. In particular, we identify all the active providers used for reaching both the covering prefix and the more-specific prefix from Step 1.

We aim to check if the covering prefix is injected to all the active providers and the deaggregated prefix is selectively injected. To this end, we analyze all the routing information retrieved during the corresponding time period (one month prior to the moment of deaggregation and one month after) from all the monitors whose routing tables we were able to retrieve from the public collectors. We monitor the routing information from each external AS towards the customer prefixes. Thus, we can infer the approximative number of active transit providers for the destination prefix by identifying the list of unique *second last-hops* (2LH) in the `AS-Path` BGP attribute after removing AS-Path prepending. The 2LH is the AS which we see before the destination AS in the `AS-Path`. This represents the provider used to reach the destination from the traffic source (i.e for some of the paths, this 2LH should be the ISP providing the data for this study).

We accept a certain error in the inferred connectivity degree of each customer, since we only have partial information on the interdomain routing. Given that the number of monitors active within the RIPE RIS and RouteViews project is limited [15], we have only a partial picture of how external sources of traffic reach the interest prefixes. However, since the sample of monitors is biased towards large Tier-1 networks, we assume that this is a reasonable approximation. We discuss how it influences our results, along with other limitations of the methodology in Section 7.4.

7.3 Applying the Proposed Methodology

In this section, we show how we can apply the proposed methodology on real data obtained from an operational major Japanese ISP. The network dataset includes BGP routing tables, traffic data, topology data and the billing scheme from the ISP looking to monitor the behavior of its customers. The analysis of this data, corroborated with an external view from the monitors active within the RIPE RIS and RouteViews projects, offers the information necessary for the detection of deaggregation strategies and the analysis of their economic impact.

7.3.1 The Dataset

In order to properly apply the proposed methodology, we need access to various information from the ISP. The primary set of data we integrate in our study, the BGP routing data, is periodically collected from a monitor inside the ISP's network. Every two hours we obtain the complete routing information from the ISP. The routing snapshot (i.e. the complete BGP routing table taken at a certain moment in time) offers an accurate perspective on the dynamics of the customer prefixes which are of interest for our study. We assume that if a prefix is present in consequent snapshots it was also there between the snapshots. In addition, prefixes not present did not appear between the snapshots. The two-hours timescale offers a small enough granularity in order to capture the long-lived changes in the deaggregation strategy of the customer network. In order to correctly separate the *customer* network information from the BGP snapshots, we use the internal community tags the ISP uses for the routes received from its customers. We are thus able to identify all the customer networks with public AS numbers, which we monitor in our measurements using the proposed methodology. We target only networks with public AS numbers, since it is likely that they also have multiple providers.

The collection of transit links through which each of these customers connects to the provider is necessary when extracting the traffic data corresponding to the detected cases of strategic deaggregation. In order to extract the topology information, we parse all the configuration files from the provider's edge routers, characteristic to different vendor-specific operating systems. We thus obtain a mapping of the customer AS number and the set of physical links where the customer connects to the Japanese provider.

The traffic data is collected in NetFlow format and spans over the two months period of May - June 2012, capturing two different billing cycles. The sampling rate used for most routers is $\frac{1}{8,192}$. However, for some routers this may differ, depending on the traffic load on the router and its processing power. We analyze the traffic data that corresponds to the two different billing-compatible time intervals, i.e., one month before and another month after the deployment of strategic deaggregation. This limits us to detecting cases of customer networks deploying the deaggregation mechanism in the time-window corre-

sponding to the two months of the study. This limitation comes from the characteristics of the major ISP itself, which stores the traffic data for its customer only during the latest two months.

Finally, we add to our analysis the type of billing scheme employed by the ISP. Generally, the billing method relies on the 95^{th} percentile rule and the exact interval used for billing is the calendar month.

7.3.2 The Results

We illustrate the use of the proposed methodology using as an input the complete dataset described in the previous section. First, we perform an extended analysis of "new" deaggregation strategies initiated within a period of 6 months (i.e., from May until October). This is aimed at providing a better understanding of the dynamics concerning deaggregation within the customer base of the Japanese ISP. We thus quantify the amount of more-specifics injected by customers of the ISP within the previously-mentioned period and monitor their evolution in time. First, we iteratively select as a reference time the *last* snapshot time-stamp taken within each month, from May to October. By applying the algorithm described in Section 7.1, we are further able to identify the set of customer networks that start to deploy deaggregation within the month previous to each of the 6 reference times. For being able to asses the impact of deaggregation on the transit traffic bill, it is also important to make sure that the newly injected more-specific prefixes are active throughout a whole billing period after the moment of deaggregation. To this end, we verify the routing data provided by the operational ISP for one month after each reference time, i.e., from June until November.

We summarize the detection results in Table 7.1. For example, we note that during August there were 6 different customer ASes which started to inject 19 new more-specific prefixes to the Japanese provider. We conclude that, generally, there are few customers deaggregating. And even more, the number of more-specifics injected to the ISP for each of the months analyzed is generally low, as observed in the third column from Table 7.1. Overall, we observe 212 new more-specifics being injected throughout the 6 months analyzed.

Given that we have traffic data available only for two months, we present the analysis of the economic impact for deaggregation strategies identified in this particular period. In order to differentiate the cases of strategic deaggregation, we merge the results of the previous analysis with the external routing data from the monitors active in the RIPE RIS and RouteViews projects. We use the results corresponding to the prefixes deaggregated in May, which also persist in the routing table for the next month.

Overall, we detect *154 more-specific prefixes* injected by the customers of the Japanese ISP during the month of May. The number of more-specifics injected in May is larger that in the other months due to a heavy deaggregator, which injects 120 more-specifics

Month	No. of customer ASes	No. of more-specifics
May	7	154
June	1	3
July	2	3
August	6	19
September	5	42
October	2	12

Table 7.1: Number of deaggregating customer ASes and total advertised deaggregated prefixes per month.

out of the total identified. The prefixes are injected by 7 of the networks purchasing transit from the Japanese provider, as noted in Table 7.1. Among the 154 more-specific prefixes first injected in May, we are able to identify one case of deaggregation combined with selective advertisements, which fulfills all the requirements imposed. Our analysis shows that on the 28th of May, at around 16:00 hours, a customer prefix is deaggregated and the resulting more-specific prefix is injected to only one of the providers (i.e. the major ISP providing data). Moreover, the more-specific prefix is not re-aggregated into his covering prefix at any point during the following month of June .

For the quantification of the impact of strategic deaggregation on the transit bill, we compare the traffic pattern for the identified prefix during a month prior to the moment of deaggregation (i.e. May) with the traffic pattern for the more-specific during a month after the moment of deaggregation (i.e. June). Since the billing period used by the Japanese ISP is the exact calendar month, we compare the bill from May with the bill from June. In order to extract from the traffic collection the data that interest us, we must first identify the physical links connecting the customer network under study and the provider. By parsing all the router configuration files, we obtain the identity of all the interfaces on the routers connecting the two networks. We then evaluate the chargeable amount of traffic for each case using the 95^{th} percentile billing rule. We conclude that, even if the expected amounts of traffic for the two prefixes are comparable, the transit bill is **20**% **lower** for the customer AS after selectively injecting the deaggregated prefix, as observed in Figure 7.3.

The difference in the chargeable volume of traffic per month may be due to the surge we observe in the traffic profile depicted in Figure 7.3 during the first analyzed month. In order to check that this increase is caused by routing changes that influence the way large sources send their traffic towards the destination AS, we would need a complete view of the evolution in time of the BGP routing tables for the source networks. However, this type of information is unavailable at this point. Instead, we observe the changes in the number of active sources out of the top 20 which forward their traffic to the destination prefix via the Japanese provider, as depicted in Figure 7.3. We extract this information

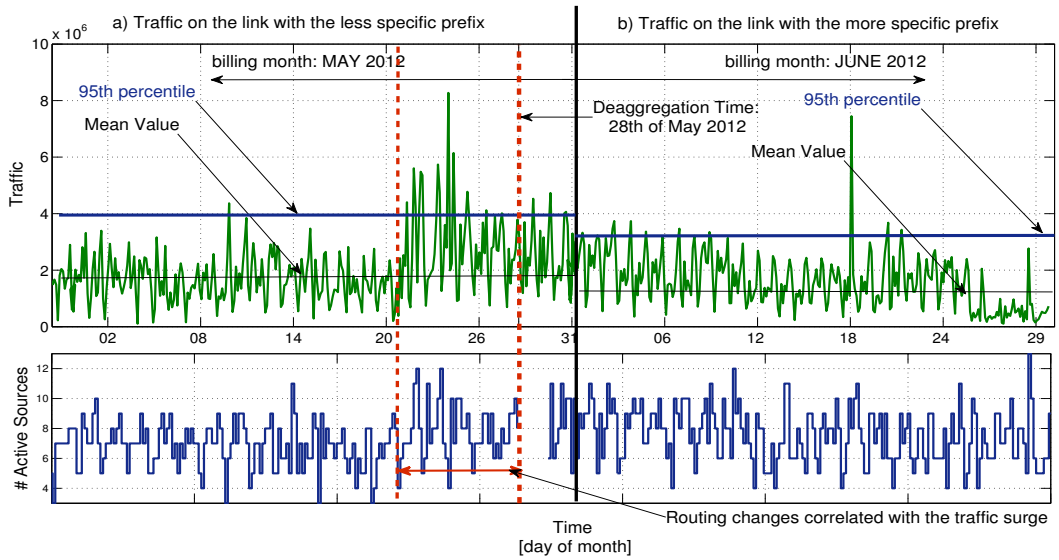

Figure 7.3: Study case: result identified using the proposed methodology.

from the NetFlow traffic data of the Japanese ISP. The analyzed sources are prefixes with length 24 and are responsible for more than 50% of the total traffic towards the destination prefix. After the injection of the more-specific prefix, the traffic has a more stable behavior than in the previous case and, also, the number of active traffic sources is more stable in time. We can also notice that there is a symmetry between the surge of traffic and an increase in the number of sources that forward their traffic through the transit link. The observed correlation between routing changes and traffic fluctuations supports the hypothesis according to which the 95^{th} percentile billing rule can be gamed by the customer networks by restricting the choices of transit links diversity towards the destination prefix. However, we cannot demonstrate the causality between the changes we observe in the traffic pattern and the deaggregation strategy being deployed because of the lack of interest cases which fulfill the model requirements.

Based on the single perfect match for the strategic deaggregation previously identified, we can only conclude that the study result supports the analytic observations for the economic impact of deaggregation.

7.4 Discussion

We proposed a novel methodology to identify all the cases of strategic deaggregation generated by the customers of any ISP and measure their economic side-effect. The methodology comes also with a number of limitations and challenges, which we hereafter explain and address.

In order to identify real occurrences of the interest scenario, we demonstrate the use

of this methodology on the real data from an operational ISP. We obtain all the necessary information from a major Japanese ISP. This includes routing data that enable us to monitor how customers advertise their address space over time and the corresponding traffic traces, router configuration information needed to identify on which links to look for the traffic data and finally, the billing scheme used. The quality of the results is conditioned by the quality of the data. Though the amount of information we handle is very large, it does not offer perfect information regarding the operations of the customers.

The Japanese ISP maintains fine-grained traffic information for its customer prefixes only for the latest two months prior to the moment of analysis. Consequently, the complete dataset from the major active ISP spans over a period of two months. Since we require the traffic traces both before and after the strategic deaggregation mechanism was deployed, this limits the traffic analysis only to cases of strategic deaggregation that have occurred as far as one month previous to the moment of analysis.

We validate that the more-specific prefixes are selectively advertised only towards the Japanese ISP using all the routing information gathered from ASes that are active in the RIPE RIS and RouteViews projects. Given that the number of monitors active within the RIPE RIS and RouteViews project which provide their full routing tables is limited to approximatively 150, we have only a partial picture of how external sources of traffic reach the prefixes identified. Consequently, a prefix may be thought to be selectively advertised when it is in fact advertised to multiple providers. In this case, though, we should not see a lower transit bill than in the aggregated case.

When analyzing the real data from the Japanese ISP, we do not observe many cases of strategic deaggregation occurring within the time-window of interest. Though we do find a number of general deaggregation cases, the corresponding prefixes look like expressions of stable strategies, justified by general operational practices. Consequently, this does not allow for an extensive evaluation of the impact of this deaggregation strategy at the economic level.

All together, in the context of the Japanese Internet community we conclude that strategic deaggregation is generally not a practice used actively within the customer base. The results of our study show that the customers of the Japanese ISP do not make an extensive use of prefix deaggregation in general, and even less in the strategic form defined in this paper. There are a number of reasons why this may be the case, including the general pressure of the community regarding the negative impact of deaggregation, the unwillingness to increase the overall complexity of the Internet or even the lack of basic necessary expertise which would allow the deployment of such strategies at the interdomain level. It is, however, important to keep in mind that these conclusions are based on data from one major ISP, and the results may be different for other entities from other geographical regions.

7.5 Summary

In this chapter, we propose a novel methodology for identifying cases of prefix deaggregation generated by the customers of any ISP within a predefined time-window. In order to identify real occurrences of the interest phenomena, we demonstrate the use of this methodology on the real traffic and routing data from a major Japanese ISP.

Overall, we do not observe much deaggregation generated from the customer networks of the ISP. And even more, despite the fact that the proposed analytical model does successfully support the observed phenomena, we do not identify many cases of strategic deaggregation performed by the ISP's customers. We do, however, distinguish and analyze a strategic deaggregation case that fulfills all the constraints imposed in the methodology. Regardless of the main goal to be achieved through deaggregation, we observe that, in certain conditions, the deaggregating AS can indeed enjoy a decrease of its transit traffic bill as a by-product of the strategic deaggregation deployed within certain conditions. In this representative case of strategic deaggregation detected within the customer base of a major Japanese ISP, the deaggregating AS enjoyed a 20% decrease on its transit bill.

Though this supports the results of the Internet model analysis, we cannot generalize these findings to all other networks performing deaggregation. In other words, one can think that the vast amount of deaggregation we generally observe at the interdomain level may not be nearly as strategic as the one captures in the model presented in this chapter. However, the results presented do point to the fact that the 95^{th} percentile billing model can potentially be, once again, gamed to the advantage of the deaggregating party. This can provide further incentives to transit providers to monitor their customers and adopt further policies to deal with their customers aggressive deaggregation strategies, which may impact their business.

Chapter 8

Conclusions and Future Work

In this thesis, we perform an extensive analysis of the intricacies emerging from the complex netting of routing policies at the interdomain level, in the context of the current operational status of the Internet. Abundant implications on the way traffic flows in the Internet arise from the convolution of routing policies at a global scale, at times resulting in ASes using suboptimal ill-favored paths or in the undetected propagation of configuration errors in routing system. In Chapter 3, we prove that monitoring *prefix visibility* at the interdomain level can be used to detect cases of faulty configurations or backfired routing policies, which disrupt the functionality of the routing system. The BGP Visibility Scanner has already demonstrated its ability to trigger valid visibility alarms and helping operators debug their routing policies. We were able to help identify more than *18,000 unintended LVPs* and assist the origin networks in identifying their causes. This is an ongoing service, since the BGP Visibility Scanner is still currently maintained. We conclude that, though legitimate routing policies of an AS can constrain the visibility of its prefixes in the Internet, the LVPs often stem from human operator errors or unpredicted interactions with the policies of other Internet players. Such prefixes can be easily missed and thought to be legitimate events at the interdomain level. They are often overlooked as valid expressions of intentional events. For example, an ISP was able to learn that 4,000 of its prefixes were leaking through some of its direct peers and were visible in the Internet since at least 6 months before the query was performed. After further investigation, the network operators identified and corrected a misconfiguration in the outbound prefix-filters, which should have otherwise discarded those prefixes. Such events may stem as a consequence of the merger between large ISPs whose configurations are consequently changing. This type of transition may affect the visibility of some prefixes, as it has been observed in the case of the Level3-Global Crossing merger [90].

In light of the observed perpetuity of such anomalous interdomain events, we tackle in Chapter 4 the acute need for a simple warning system for faulty configurations and/or problematic external routing conditions to assist operators in optimizing the performance

of their routing policies. We show that the lack of global prefix visibility can offer early warning signs for anomalous events which, despite their impact, often remain hidden from state of the art tools, e.g., [7, 8]. We thus rely on machine-learning design a *Winnowing Algorithm* able to predict with 95% accuracy if a LVP is intended or unintended. We leverage the robust machine learning concept of *boosted classification trees* [24] to train the system on 20,000 ground-truth *LVPs*, and thus enable it to learn the patterns of misconfigurations and bogus routing policies which are normally hard to detect. Furthermore, the classification model uses only visibility-related per-prefix features in order to predict the class of the *LVPs*.

The ground-truth dataset of 20,000 *LVPs* documents a wide range of cases of previously undetected anomalous events, affecting the interdomain entities. Though the root causes of the anomalies we detect are recurring in the Internet, their appearance in the BGP Visibility Scanner might change in time. We leave for future work the analysis on the lifetime of the ground-truth knowledge we have accumulated and the stability of the accuracy of the winnowing system in time.

While affecting the global visibility of prefixes, misconfigured or unintended routing policies also impact the global reachability of prefixes. In Chapter 5, we show that such unintended Internet behavior not only degrades the efficacy of the routing policies implemented by operators, causing their traffic to follow ill-favored paths, but can also point out problems in the global connectivity of prefixes. More specifically, we research how the visibility degree of a prefix impacts its global reachability. From multiple vantage points in the Internet, including 100 RIPE Atlas active probes, we test the reachability of both IPv4 and IPv6 LVPs. We find that *limited visibility does not necessarily imply limited reachability*, since there could be a less-specific *HV* covering prefix that provides reachability. However, Dark Prefixes (DPs), which by definition do not have a covering less-specific prefix to ensure global connectivity, remain highly unreachable. Moreover, while the IPv4 dark address space can be largely explained as route leaks or mistakes, this is not valid for the v6DPs. We believe that this is a serious problem for the Internet, as limited reachability of a non-negligible set of prefixes cripples the fundamental function of the Internet, i.e., ensuring global connectivity for every host attached to it.

Majority of LVPs which are more-specific prefixes covered by less-specific HVPs are generally reachable, even is the traffic follows the ill-favored path corresponding to the covering less-specific prefix with global visibility. In Chapter 6, we further direct our attention towards the impact that more-specific prefixes with intended limited visibility have on the routing system. The impact of prefix deaggregation on the routing system has long been a reason of debate in the Internet community. Though usually frowned upon, this strategy is more commonly used nowadays, especially in light of the IPv4 address space depletion. In this thesis, we show that individual networks deploying prefix deaggregation may also, given certain general Internet conditions, have an economic im-

pact on their transit providers. Using a general Internet model, we analyze how, after splitting its allocated address space, a customer AS can "control" which of the transit providers is used to reach each more-specific prefix through the use of selective advertisements. This further decreases the number of choices in terms of transit link used to reach the destinations advertised. As a side-effect, this translates into a more deterministic traffic pattern on that particular transit link, which consequently means decreased traffic fluctuations and thus a smaller monthly transit traffic bill. This comes as a result of the unique interaction between several elements that are characteristic to the current operational Internet: the path dynamics in the current Internet, the asymmetrical popularity of traffic sources and the popular billing method which relies on the 95^{th} percentile of traffic. Should any of these elements change (e.g., should the provider bill its customers using a different billing model than the 95^{th} percentile), the observed collateral benefit might no longer apply. This does not mean, however, that deaggregation in the fist place will no longer be used.

The real cost of deaggregation in the Internet is not easily quantifiable. The marginal cost of injecting one more prefix at the interdomain level does not only depend on the cost of an additional entry in an already bloated routing table. Since routers are currently capable of handling very large BGP routing tables, the real threat deaggregation poses regards the convergence time of the global routing system. Consequently, when thinking about the cost of deaggregation, we need to consider its impact on the global routing system and on the Internet community. Until now, the transit providers did not have the right incentives to refrain from advertising the deaggregated prefixes as injected by their customer [38]. Taking into consideration the monetary aftermath of strategic deaggregation analyzed in Chapter 6 this thesis, the new economic incentives might be enough to push providers to change their strategy and transfer some of the costs of prefix deaggregation back to their customers. This could imply an important shift in the prefix deaggregation strategies adopted by the ASes in the Internet, moving the set of individual deaggregation strategies closer to the social welfare, where everybody enjoys increased benefits.

Though the analytical model proves the possibility of inadvertent economic benefit for the deaggregating party, we further verify for real-life occurrences of such scenarios. To this end, in Chapter 7, we propose a novel methodology to *a-posteriori* identifying cases of prefix deaggregation generated by the customers of any ISP within a predefined time-window. We focus on identifying cases of selectively advertised deaggregated prefixes. We explain how the economic side-effect of the strategic deaggregation can be measured. In order to identify real occurrences of the interest phenomena, we demonstrate the use of this methodology on the real traffic and routing data from a major Japanese ISP. Overall, we do not observe much deaggregation generated from the customer networks of the Japanese ISP. We cannot know whether this is an artifact of the particular analyzed

network providing us all the data or it is a general feature throughout the Internet. It may happen that in the case of some other transit providers, such deaggregation scenarios are much more frequent. We distinguish and analyze a strategic deaggregation case that fulfills all the constraints imposed by the methodology. We find that through selectively injecting more-specifics, the customer AS is able to smoothen the traffic variations and save approximatively 20% on its transit bill. In the long term, this may negatively impact the business of the ISP. This result supports the hypothesis of an economic impact of strategic deaggregation, but it is not sufficient for generalization.

The work outlined in this thesis is focused on developing techniques, tools and models to assist network operators, protocol designers and researchers in understanding the manner in which BGP routing policies take effect in the Internet and which may be their possible impacts on the community. As the routing necessities of ASes grow more complicated, also the relationship between networks fall out of the two well-known generic categories, i.e., customer-to-provider and peer-to-peer. In the last years, customized policies became more often in the Internet. Additionally, the traffic engineering goals of each network operator grow far beyond the use of strategic deaggregation as a brute-force solution. Thus, as a future direction of work, we focus on characterizing the complexity of the Internet and of its routing policies by quantifying the magnitude of the Limited Visibility Prefixes set within the whole Internet. Ideally, capturing the total number of LVPs would imply setting up a BGP monitor in every operational AS in the Internet and compare the contents of the obtained routing tables. Since this is not feasible, we propose the use of a statistical model that would illustrate the information gain in terms of new LVPs learned from each routing table of every AS active in the Internet. We plan to sub-sequentially analyze the level of Internet complexity captured by the interaction of convoluted routing policies interacting to further limit the visibility of prefixes at the interdomain level.

References

[1] L. Quan, J. Heidemann, and Y. Pradkin, "Trinocular: Understanding Internet Reliability Through Adaptive Probing," in *Proceedings of the ACM SIGCOMM Conference*, Hong Kong, China, August 2013.

[2] T. Griffin and G. Huston, "BGP Wedgies," 2005, RFC 4264.

[3] "How the Internet in Australia went down under." [Online]. Available: http://www.bgpmon.net/how-the-internet-in-australia-went-down-under/

[4] "Pakistan hijacks YouTube." [Online]. Available: http://www.renesys.com/2008/02/pakistan-hijacks-youtube-1/

[5] R. Bush, "Oh K can you see," in *NANOG mailing list archives*, October 2005.

[6] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, 2002.

[7] Y.-J. Chi, R. Oliveira, and L. Zhang, "Cyclops: the AS-level connectivity observatory," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 5, 2008.

[8] "RIPE Labs." [Online]. Available: https://labs.ripe.net/

[9] M. Caesar and J. Rexford, "Bgp routing policies in isp networks," *Network, IEEE*, vol. 19, no. 6, pp. 5–11, Nov 2005.

[10] T. Bu, L. Gao, and D. Towsley, "On characterizing BGP routing table growth," *Computer Networks*, vol. 45, no. 1, pp. 45–54, 2004.

[11] L. Cittadini, W. Muahlbauer, S. Uhlig, R. Bush, P. Francois, and O. Maennel, "Evolution of Internet Address Space Deaggregation: Myths and Reality," *IEEE Journal on Selected Areas in Communications, special issue on Internet Routing Scalability*, 2010.

[12] A. Lutu, M. Bagnulo, and O. Maennel, "The BGP Visibility Scanner," in *16th IEEE Global Internet Symposium (GI 2013), collocated with INFOCOM 2013*, April 2013.

[13] "BGPMON Alert Questions." [Online]. Available: http://mailman.nanog.org/pipermail/nanog/2014-April/066171.html

[14] A. Dhamdhere and C. Dovrolis, "The internet is flat: modeling the transition from a transit hierarchy to a peering mesh," in *Proceedings of the 6th ACM CoNEXT conference*, ser. CoNEXT '10, 2010.

[15] M. Roughan, W. Willinger, O. Maennel, D. Perouli, and R. Bush, "10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems," *Selected Areas in Communications, IEEE Journal on]*, vol. 29, no. 9, 2011.

[16] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "PHAS: a prefix hijack alert system," in *Proceedings of the 15th conference on USENIX Security Symposium - Volume 15*, 2006.

[17] J. Wu, Z. M. Mao, J. Rexford, and J. Wang, "Finding a needle in a haystack: pinpointing significant BGP routing changes in an IP network," in *Symposium on Networked Systems Design & Implementation*, 2005.

[18] "BGP Routing Leak Detection SystemRouting Leak Detection System." [Online]. Available: http://puck.nether.net/bgp/leakinfo.cgi

[19] RENESYS, http://renesys.com/.

[20] "BGPmon." [Online]. Available: http://www.bgpmon.net/

[21] D. Perouli, T. Griffin, O. Maennel, S. Fahmy, C. Pelsser, A. Gurney, and I. Phillips, "Detecting Unsafe BGP Policies in a Flexible World," in *International Conference on Network Protocols (ICNP)*, 2012.

[22] "RIPE RIS Raw data." [Online]. Available: http://www.ripe.net/data-tools/stats/ris/ris-raw-data

[23] "University of Oregon Route Views Project." [Online]. Available: http://www.routeviews.org/

[24] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, ser. EuroCOLT '95, 1995.

[25] G. Huston, "Analyzing the Internet BGP Routing Table," *The Internet Protocol Journal*, vol. 4, no. 1, 2001.

[26] Z. Zhang, Y. Zhang, Y. C. Hu, and Z. M. Mao, "Practical defenses against bgp prefix hijacking," in *Proceedings of the 2007 ACM CoNEXT conference*, ser. CoNEXT '07, New York, NY, USA, 2007, pp. 3:1–3:12.

[27] A. Lutu, M. Bagnulo, and R. Stanojevic, "An Economic Side-Effect for Prefix Deaggregation," in *The 7th Workshop on the Economics of Networks, Systems and Computation (NetEcon 2012), collocated with INFOCOM 2012*, 2012.

[28] A. Lutu, M. Bagnulo, C. Pelsser, K. Cho, and R. Stajevic, "An Analysis of the Economic Impact of Strategic Deaggregation," *Computer Networks, Major Revision Submitted*, August 2014.

[29] R. Teixeira, N. Duffield, J. Rexford, and M. Roughan, "Traffic Matrix Reloaded: Impact of Routing Changes," in *PAM'05*, 2005.

[30] X. Dimitropoulos, P. Hurley, A. Kind, and M. P. Stoecklin, "On the 95-Percentile Billing Method," in *Proceedings of PAM'09*, 2009.

[31] R. Stanojevic, N. Laoutaris, and P. Rodriguez, "On economic heavy hitters: shapley value analysis of 95th-percentile pricing," in *Proceedings of IMC'10*, 2010.

[32] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet Interdomain Traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 40, August 2010.

[33] C. Labovitz, A. Ahuja, and M. Bailey, "Shining Light on Dark Address Space," Arbor Netwoks, Ann Arbor, Michigan, USA, Tech. Rep. TR-2001-01, November 2001.

[34] A. Lutu, M. Bagnulo, C. Pelsser, O. Maennel, and J. Cid-Sueiro, "The BGP Visibility Toolkit: Detecting Anomalous Internet Routing Behavior ," *IEEE/ACM Transactions on Networking, Major Revision due Nov. 2014.*

[35] A. Lutu, M. Bagnulo, J. Cid-Sueiro, and O. Maennel, "Separating wheat from chaff: Winnowing unintended prefixes using machine learning," in *Proceedings of 33rd IEEE International Conference on Computer Communications*, ser. IEEE INFOCOM 2014, April 2014.

[36] A. Lutu, M. Bagnulo, C. Pelsser, and O. Maennel, "Understanding the Reachability of IPv6 Limited Visibility Prefixes," in *Passive and Active Measurement*, ser. Lecture Notes in Computer Science, 2014, vol. 8362.

[37] A. Lutu, C. Pelsser, M. Bagnulo, and K. Cho, "The Aftermath of Prefix Deaggregation ," *In: 25th International Teletraffic Conference (ITC25)*, September 2013.

[38] C. Kalogiros, M. Bagnulo, and A. Kostopoulos, "Understanding incentives for prefix aggregation in BGP," in *Proceedings of the 2009 workshop on Re-architecting the internet*, ser. ReArch '09, 2009.

[39] H. Tangmunarunkit, R. Govindan, S. Shenker, and D. Estrin, "The impact of routing policy on internet paths," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, 2001, pp. 736–742 vol.2.

[40] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, December 2001.

[41] G. Huston, "Interconnection, Peering, and Settlements," *The Internet Protocol Journal*, 1999.

[42] A. Dhamdhere and C. Dovrolis, "The internet is flat: Modeling the transition from a transit hierarchy to a peering mesh," in *Proceedings of the 6th International COnference*, ser. CoNEXT '10. New York, NY, USA: ACM, 2010, pp. 21:1–21:12.

[43] W. Mühlbauer, S. Uhlig, B. Fu, M. Meulle, and O. Maennel, "In search for an appropriate granularity to model routing policies," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07, 2007, pp. 145–156.

[44] R. Stanojevic, I. Castro, and S. Gorinsky, "Cipt: Using tuangou to reduce ip transit costs," in *Proceedings of the Seventh COnference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '11, 2011, pp. 17:1–17:12.

[45] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proceedings of SIGCOMM '05*, 2005.

[46] K. El-Arini and K. Killourhy, "Bayesian Detection of Router Configuration Anomalies," in *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, ser. MineNet '05, 2005.

[47] K. Zhang, A. Yen, X. Zhao, D. Massey, S. F. Wu, and L. Zhang, "On Detection of Anomalous Routing Dynamics in BGP," in *NETWORKING*, 2004.

[48] J. Zhang, J. Rexford, and J. Feigenbaum, "Learning-based anomaly detection in BGP updates," in *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, ser. MineNet '05, 2005, pp. 219–220.

[49] J. Li, D. Dou, Z. Wu, S. Kim, and V. Agarwal, "An internet routing forensics framework for discovering rules of abnormal BGP events," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, October 2005.

[50] J. Han, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[51] N. Feamster, J. Jung, and H. Balakrishnan, "An empirical study of "bogon" route advertisements," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 1, pp. 63–70, Jan. 2005.

[52] V. Khare, Q. Ju, and B. Zhang, "Concurrent prefix hijacks: Occurrence and impacts," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12, 2012, pp. 29–36.

[53] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding bgp misconfiguration," in *Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '02, 2002, pp. 3–16.

[54] F. Wang, Z. M. Mao, J. Wang, L. Gao, and R. Bush, "A measurement study on the impact of routing events on end-to-end internet path performance," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '06, 2006, pp. 375–386.

[55] V. Paxson, "End-to-end routing behavior in the internet," in *Conference Proceedings on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '96, 1996, pp. 25–38.

[56] C. Labovitz, A. Ahuja, A. Bose, and F. Jahanian, "Delayed internet routing convergence," *IEEE/ACM Trans. Netw.*, vol. 9, no. 3, pp. 293–306, June 2001.

[57] A. Feldmann, O. Maennel, Z. M. Mao, A. Berger, and B. Maggs, "Locating internet routing instabilities," in *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '04. New York, NY, USA: ACM, 2004, pp. 205–218. [Online]. Available: http://doi.acm.org/10.1145/1015467.1015491

[58] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "NetDiagnoser: Troubleshooting Network Unreachabilities Using End-to-end Probes and Routing Data," in *Proceedings of the 2007 ACM CoNEXT Conference*, ser. CoNEXT '07, 2007, pp. 18:1–18:12.

[59] Z. M. Mao, J. Rexford, J. Wang, and R. H. Katz, "Towards an accurate as-level traceroute tool," in *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '03, 2003, pp. 365–378.

[60] R. Bush, O. Maennel, M. Roughan, and S. Uhlig, "Internet Optometry: Assessing the Broken Glasses in Internet Reachability," in *Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '09, 2009, pp. 242–253.

[61] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '06, 2006, pp. 153–158.

[62] C. Pelsser, L. Cittadini, S. Vissicchio, and R. Bush, "From paris to tokyo: On the suitability of ping to measure latency," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13, 2013, pp. 427–432.

[63] M. Luckie, Y. Hyun, and B. Huffaker, "Traceroute probe method and forward ip path inference," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, 2008.

[64] M. H. Gunes and K. Sarac, "Analyzing router responsiveness to active measurement probes," in *Proceedings of the 10th International Conference on Passive and Active Network Measurement*, ser. PAM '09, 2009, pp. 23–32.

[65] M. Nikkhah, R. Guérin, Y. Lee, and R. Woundy, "Assessing ipv6 through web access a measurement study and its findings," in *Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies*, ser. CoNEXT '11, 2011.

[66] A. Dhamdhere, M. Luckie, B. Huffaker, k. claffy, A. Elmokashfi, and E. Aben, "Measuring the deployment of ipv6: topology, routing and performance," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12, 2012.

[67] B. Quoitin, C. Pelsser, L. Swinnen, O. Bonaventure, and S. Uhlig, "Interdomain traffic engineering with bgp," *Communications Magazine, IEEE*, vol. 41, no. 5, pp. 122–128, May 2003.

[68] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and Principles of Internet Traffic Engineering." RFC 3272, 2002.

[69] F. Wang and L. Gao, "On inferring and characterizing internet routing policies," in *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '03, 2003, pp. 15–26.

[70] "BGP Routing Table Analysis Report." [Online]. Available: http://bgp.potaroo.net/

[71] B. Zhang, R. Liu, D. Massey, and L. Zhang, "Collecting the Internet AS-level topology," *ACM SIGCOMM CCR*, vol. 35, no. 1, 2005.

[72] "Team Cymru - The Bogon Reference." [Online]. Available: http://www.cymru.com/BGP/bogons.html

[73] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of BGP multiple origin AS (MOAS) conflicts," in *1st ACM SIGCOMM Workshop on Internet Measurement*, 2001.

[74] T. Hastie, R. Tibshirani, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction.* New York: Springer-Verlag, 2001.

[75] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees.* Monterey, CA: Wadsworth and Brooks, 1984.

[76] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2013, ISBN 3-900051-07-0. [Online]. Available: http://www.R-project.org/

[77] E. Alfaro-Cortes, M. Gamez-Martinez, and N. Garcia-Rubio, *adabag: Applies multiclass AdaBoost.M1, AdaBoost-SAMME and Bagging*, 2012.

[78] V. Jacobson, "traceroute." [Online]. Available: ftp://ftp.ee.lbl.gov/traceroute.tar.gz

[79] "Ripe Atlas." [Online]. Available: https://atlas.ripe.net/

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

[81] J. Rexford, J. Wang, Z. Xiao, and Y. Zhang, "Bgp routing stability of popular destinations," in *Proceedings of IMW'02*, 2002.

[82] R. van de Meent, M. Mandjes, and A. Pras, "Gaussian Traffic Everywhere?" in *Proceedings of the IEEE International Conference on Communications, ICC'06.* IEEE Computer Society, June 2006, pp. 573–578.

[83] A. Dhamdhere and C. Dovrolis, "An agent-based model for the evolution of the internet ecosystem," in *Proceedings of COMSNETS'09*, 2009.

[84] Q. Chen, H. Chang, R. Govindan, S. Jamin, S. Shenker, and W. Willinger, "The Origin of Power-Laws in Internet Topologies Revisited," in *INFOCOM '02*, 2002.

[85] B. Norton, *The Internet Peering Playbook : Connecting to the Core of the Internet.* Dr. Peering Press, August 2011.

[86] H. Chang, S. Jamin, Z. M. Mao, and W. Willinger, "An empirical approach to modeling inter-as traffic matrices," in *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, 2005.

[87] "Caida AS Ranking." [Online]. Available: http://as-rank.caida.org/

[88] M. Luckie, B. Huffaker, A. Dhamdhere, V. Giotsas, and k. claffy, "As relationships, customer cones, and validation," in *Proceedings of the 2013 Conference on Internet Measurement Conference*, ser. IMC '13, 2013, pp. 243–256.

[89] D. E. Knuth, *The Art of Computer Programming, Volume 3: Sorting and Searching (2nd Edition).* Addison-Wesley, 1998, vol. 3.

[90] "3356 leaking routes out 3549 lately," in *NANOG mailing list archives*, March 2014.