ISP Probing Reduction with Anaximander

Emeline Marechal[1][0000−0002−9554−8655], Pascal Mérindol[2][0000−0003−2750−376X], and Benoit Donnet[1][0000−0002−0651−3398]

[1] Université de Liège, Montefiore Institute, Belgium
{emeline.marechal, benoit.donnet}@uliege.be
[2] Université de Strasbourg, iCube, France
merindol@unistra.fr

Abstract. Since the early 2000's, Internet topology discovery has been an active research topic, providing data for various studies such as Internet modeling, network management, or to assist and support network protocol design. Within this research area, ISP mapping at the router level has attracted little interest despite its utility to perform intra-domain routing evaluation. Since Rocketfuel (and, to a smaller extent, `mrinfo`), no new tool or method has emerged for systematically mapping intra-domain topologies.
In this paper, we introduce Anaximander, a new efficient approach for probing and discovering a targeted ISP in particular. Considering a given set of vantage points, we implement and combine several predictive strategies to mitigate the number of probes to be sent without sacrificing the ISP coverage. To assess the ability of our method to efficiently retrieve an ISP map, we rely on a large dataset of ISPs having distinct nature and demonstrate how Anaximander can be tuned with a simple parameter to control the trade-off between coverage and probing budget.

Keywords: Anaximander· `traceroute`· ISP mapping · Internet Topology Discovery · probing reduction

1 Introduction

For the last 20 years, Internet topology discovery has attracted a lot of attention from the research community [9, 17]. Those researches have focused on efficient data collection (e.g., Doubletree [10]), on alias resolution [21], on ISP mapping (e.g., Rocketfuel [40] and `mrinfo` [31]), or on Internet modeling [35].

Despite being man-made, much of the Internet is hidden and unknown, for the reason that it is a large and complex system that does not consist in a single authoritative entity. Rather, it is made up of more than 72,000 independent Autonomous Systems (ASes) , each having its own commercial practices, physical infrastructure, and logical design (in particular its routing and Traffic Engineering – TE – strategies). More precisely, to deploy a specific routing strategy (from best-effort traffic to more complex strategies, such as fast re-routing), ISPs generally assign an IGP weight to each link and then elaborate more or

less complex strategies to control packet forwarding according to a given set of network metrics, related constraints, and technology [36].

Consequently, blindly sampling (a subset of) the Internet is not enough to reveal and discriminate such specific topological and routing patterns, or generic ones if they are any. Instead, in order to conduct relevant TE and IGP performance evaluations [13] and showcase the performance of a given routing proposal (with simulations or analytical models), it is more suitable to rely on distinct ISP maps offering various realistic situations, rather than using an arbitrary chunk of the Internet.

To answer this requirement and offer a framework for reproducible realistic experimentation, one needs to collect intra-domain networks of distinct natures (e.g., Tier-1, Transit, and Stub ASes of diverse sizes). We thus argue it is essential to develop modern, accurate, and advanced topology discovery tools able to skillfully capture the reality of the Internet, in particular considering its atomistic technical nature. The goal is therefore to search for efficient probing designs able to reveal the specifics of any given intra-domain router level map.

While Rocketfuel [40] topologies have been the de facto dataset in use for nearly two decades, we argue that both the resulting topologies and the underlying probing methods are now outdated. Indeed, the Internet structure and practices have evolved over time, and new refined measurements tools have become available as well [2, 25]. In this paper, we pursue the same objective as Rocketfuel formerly, i.e., to map specific ISPs at the router level. More specifically, we tackle the following challenge: *Can we infer ISP router level maps with a reduced probing budget without hampering the resulting topological coverage?* Designing efficient probing campaigns is indeed essential to speed up the measurement period and so mitigate forwarding anomalies (e.g., routing changes [48, 50]), and the effects of adaptative filtering (e.g., rate limiters [37, 16]). Otherwise, the data collected may not be consistent or suffer from poor coverage.

We first point out Rocketfuel limits for capturing nowadays Internet maps but also revisit its successful components when their efficiency is still valuable. Then, we develop our own strategies and evaluate them relying on a large and recent dataset to conduct realistic simulations and support our assumptions. More precisely, we replay measurements offline using different approaches than the initial brute-force approach in order to understand what are the corresponding gains and losses. The reduction strategies we evaluate are diverse and span from the design of the initial hitlist and its scheduling to specific reduction techniques looking at prefix de-aggregation and neighbor marginal utility. To answer our research question, this paper introduces Anaximander, our new probing method able to recover the same ISP maps as obtained with a brute force approach, but with a network-friendly and efficient probing methodology. For a given set of vantage points, Anaximander is able to adapt itself to the characteristics of the ISP being mapped. It constructs and manages a target list in order to efficiently cover most of the visible part of the targeted ISP topology. Additionally, our tool offers the opportunity to easily explore the trade-off between AS coverage and probing budget, with a single parameter.

The remainder of the paper is organized as follows: Sec. 2 positions this paper with respect to the state of the art; Sec. 3 describes how we collected and processed the data used throughout the paper; Sec. 4 discusses nowadays Rocketfuel limits; Sec. 5 introduces ANAXIMANDER, our new tool for efficiently probing ISPs; finally, Sec. 7 concludes this paper by summarizing its main achievements.

2 Related Work

Most active probing tools based on `traceroute` embed some heuristics to limit the probing overhead. Such heuristics generally rely on caching previously seen IP addresses to avoid redundancy. For example, for a given prefix P, *Mercator* [14] identifies the furthest router R that was already in the map at the time the probe completed. Then, each subsequent probes to P can start at the TLL of R: if the reply comes from R, Mercator continues to probe the path, otherwise it backtracks and restarts probing with a regular TTL of one. *Atlas* [47] probes IPv6 networks using source routed IPv6 `traceroute`. For each trace, Atlas caches the hop distance to the via-router, i.e., the intermediate router used for source routing. If the same via-router is used in a subsequent trace, then the cache distance provides the initial hop distance and alleviates the need to re-probe from the vantage point to that via-router. *Scriptroute* [41] avoids retracing paths by embedding a list of previously observed IP addresses. A given vantage point stops probing when it reaches an IP address belonging to the list. More generally, Doubletree [10] keeps track of the tree-like structure of routes, either emanating from a single source towards multiple destinations or converging from multiple sources towards a single destination, to avoid probing duplication. Rocketfuel [40], probably the most well-known intra-domain topology discovery tool, relies on two reduction techniques, namely ingress and egress reductions, to reduce its probing budget. While the first considers that probes to a given destination should converge if they enter the targeted ISP at the same ingress node, the second advocates that traces from the same ingress to any prefix beyond the same egress should traverse the same path. Generally speaking, Rocketfuel relies on BGP to guide the probing and builds a router-level map of the targeted domain using Ally. However, it has been shown that Rocketfuel tends to overestimate the path diversity of the targeted domain [42]. Sec. 4 will investigate more deeply the (other) limits of Rocketfuel. Finally, it is worth noticing heuristics have also been proposed to increase the number of nodes discovered, e.g., POPsicle [11].

With respect to ISP mapping, few other tools than `traceroute` and Rocketfuel exist. There is notably [51], that focuses on the exploration of an essential component of an ISP's infrastructure: its regional access network. By combining several Internet cartography methods (such as public WiFi hotspots and public transit of mobile devices), they are able to get some insight on this specific ISP portion, although it is often remarkably opaque and difficult to measure. There is also `mrinfo` [20], that relies on the Internet Group Management Protocol (IGMP) to enable native router level query. The IGMP reply consists in a

list of local multicast interfaces and their link with adjacent interfaces. Recursively querying adjacent interfaces can thus lead to the collection of connected topological information [31]. Pansiot et al. [34] have also provided algorithms for efficiently delimiting AS boundaries to extract ISP maps from such data. Merlin [26] extends `mrinfo` by increasing its efficiency but also mixes IGMP probing with ICMP probing (Paris Traceroute [2] and Ally [40] are used to overcome `mrinfo` limitations). However, IGMP queries are now deprecated and operators filter them at their borders [27], making those techniques unusable. The Internet topology zoo [23] and similar projects like SNDlib[3] expose real intra-domain maps manually collected from operators providing their own network maps. However, although useful for TE related reproducible experiments, such datasets are often insufficient as they do not expose large and up to date maps of the Internet. Many are outdated and correspond to small, sometimes partial, IP networks not revealing all relevant information (e.g., IGP weights, node positions, or propagation delays). Eventually, Sybil [6] is a system that can serve a rich set of queries about Internet routes, including what routes go through an ISP of interest. However, in the background, Sybil needs to continuously run measurements in order to maintain its knowledge of routing. This paradigm, which requires a database of (relatively fresh) previously-issued `traceroutes`, is a great departure from the one-shot campaign that can be run more quickly and easily.

Our goal is to provide a light probing framework enabling the deployment of repeated probing campaigns enriched with all available information brought by forwarding traces.

3 Dataset

The `traceroute` data used throughout this paper has been collected by CAIDA with `TNT` [44, 25]. `TNT` is a Paris-traceroute [2] extension that is able to reveal the content of MPLS tunnels hidden to `traceroute` exploration [46].

`TNT` has been deployed on the Archipelageo infrastructure [5] between April 17$^{\text{th}}$ and 23$^{\text{rd}}$, 2021 over 14 vantage points (VPs), scattered all around the world: Europe (6), North America (1), South America (3), Asia (2), and Australia (2). The overall set of destinations, over 11 million IP addresses, is spread over the 14 vantage points to speed up the probing process.

A total of 936,944 distinct unique IP addresses (excluding `traceroute` targets) have been collected, without counting non-publicly routable addresses, which have been excluded from our dataset. As we are interested in mapping ISPs[4] (as opposed to the whole Internet), we applied `bdrmapIT` [29], a tool for annotating routers and IP addresses with AS ownership. The objective here is to delimit as accurately as possible the ASes maps from the rest of the Internet.

`bdrmapIT`'s inferences are more meaningful when the tool is provided with information about routers, and not only IP addresses found in the traceroutes.

[3] `http://sndlib.zib.de`

[4] In the remainder of this paper, "ISP" and "AS" are used interchangeably.

AS		Topology			Directed Prefixes	
ASN	Type	Links	Interfaces	Routers	Dependent	Raw number
3491		4,399	601	107	0.49 %	832,968
6830		6,215	2,985	40	0.22 %	832,808
6762		5,338	530	95	0.69 %	831,530
174		23,115	4,931	861	1.83 %	830,610
3257		8,913	1,477	310	0.83 %	829,468
1299	Tier1	11,999	1,064	204	0.86 %	829,309
6453		5,207	831	156	1.70 %	829,002
286		1,46	56	11	0.01 %	828,926
6461		5,944	1,122	209	0.56 %	771,902
12956		8,650	981	91	0.74 %	752,128
11537		631	124	33	0.00 %	21,823
6939		11,345	743	132	1.84 %	850,999
50673		306	52	5	0.08 %	845,664
4637	Transit	1,374	402	68	0.18 %	814,703
1273		2,800	760	93	0.69 %	474,677
7922		33,473	23,356	1,054	0.77 %	163,000
2856		6,959	2,381	37	12.46 %	4,423
8764		293	93	7	22.93 %	3,149
9198		881	353	44	85.13 %	1,748
5400		2,000	395	39	16.46 %	1,628
13789	Stub	235	60	7	65.26 %	685
5432		394	172	4	80.25 %	157
1241		472	224	16	91.03 %	145
2611		176	124	5	77.97 %	59
224		506	417	8	91.30 %	23

Table 1. Various statistics on ASes of interest. Within each type category, ASes are ordered by the number of directed prefixes found in the RIBs (Routing Information Bases), which is a coarse indicator of the AS's importance in the Internet. More precisely, a `traceroute` towards a directed prefix is expected to transit through the AS of interest.

Therefore, we used MIDAR [22], a tool based on similarities in the IP-ID field, to perform alias resolution (i.e., the process of identifying IP addresses that belong to the same router [21], leading so to a router-level topology) on our set of addresses. We ran MIDAR between April 29$^{\text{th}}$, 2021 and May 1$^{\text{st}}$, 2021. Out of the 900k addresses discovered by `TNT`, MIDAR found 45,977 routers involving 147,633 addresses.

Additionally, we used APPLE [28] (between May 10$^{\text{th}}$, 2021 and May 12$^{\text{th}}$, 2021), a technique for resolving router IP aliases that complements existing techniques, such as MIDAR. We deployed APPLE on EdgeNet [39], a Kubernetes cluster dedicated to network and distributed systems research, and were able

to find 26,729 routers involving 87,532 addresses. In combination with MIDAR, we were thus able to further refine our alias resolution with a total of 57,355 routers involving 192,320 addresses, which represents an increase in coverage of 25% compared to the initial results with MIDAR.

Besides, we used BGPStream [33], an open-source software framework to easily acquire live and historical BGP data. The tool provides access to BGP views from all around the world, coming from the RouteViews [38] and RIPE RIS [43] projects. We collected 44 BGP tables (from as many collectors) in the middle of the `TNT` campaign, on April 20^{th}, 2021.

Table 1 provides global statistics about the sample of 25 ASes selected for this study and discussed in the paper. We chose ASes with varying sizes and roles in the Internet (11 Tiers 1, 5 Transits, and 9 Stubs) in order to be as representative as possible.

4 Rocketfuel Limits

The main contribution of Rocketfuel was to propose and deploy the pioneer measurement techniques to infer ISP router-level maps. The second challenge was to use as few measurements as possible to speed up the campaign, not only because of the limited capacity of legacy forwarding devices, but also because ISPs (continue to) filter probes using default rate limiters (e.g., only small bursts of ICMP replies are allowed) for both performance and security reasons. Additionally, ISPs are continuously subject to routing changes [30]. In this regard, their approach was to first exploit available routing information to select traces likely to transit the ISP of interest, and second, to apply reduction techniques based on IP routing properties to eliminate traces likely to follow redundant paths in the ISP.

At the time, Rocketfuel was admittedly the best attempt at mapping an ISP, even though it already suffered from some limits [42]. These issues had to do with the inference of numerous false links, due, on the one hand, to the naive use of the basic `traceroute` implementation (later replaced by Paris `traceroute` [2]), and on the second hand, to using alias resolution techniques that are now obsolete.

Nowadays, Rocketfuel suffers from additional problems due to the massive growth of the Internet during the last 20 years, along with all the changes that came with it. More and more edge networks joined the Internet, the interconnections between core networks became denser (flattening the Internet), and networks operational practices (such as the usage of TE, multi-homing, or provider independent addresses [1]) have significantly evolved and rely now on new technologies (e.g., MPLS or Segment Routing). Following this shift in paradigm, while the core principles guiding Rocketfuel's probing remains valuable conceptually, the set of tools and strategies applied on top of them became outdated.

In the next sections, we will review how Rocketfuel's reduction techniques (namely, Egress Reduction – Sec. 4.1 –, Next-hop AS Reduction – Sec. 4.2 –, and Ingress Reduction – Sec. 4.3) are no more suited for today's Internet.

4.1 Egress Reduction

Without going into all the details, Rocketfuel's initial pool of targets is built from BGP tables (i.e., Routing Information Bases – RIBs), that allow one to select measurements expected to transit the ISP of interest. They call this technique *directed probing*, and the number of *directed prefixes* for each AS can be found in Table 1. Egress Reduction advocates that traces from the same ingress to any prefix beyond the same egress should traverse the same path. Such traces are thus redundant, and only one needs to be collected. But in order to find said egresses, Rocketfuel must conduct a pre-probing phase to discover the ISP's egress routers common to several prefixes.

This pre-probing stage is only launched on a subset of their initial pool of targets, that they call the *dependent prefixes*. Dependent prefixes are prefixes originated by the ISP of interest or one of its singly-home customers. Therefore, by definition, all `traceroutes` to these prefixes (from anywhere in the network) should transit the ISP. This allows them to launch their pre-probing phase from a single monitor, with the guarantee that the probes will indeed go through the ISP of interest.

Even though Egress Reduction is admittedly sound in principle, being able to apply it and actually find the egresses shared by several prefixes may prove to be too costly for a marginal reduction not balancing the effort. To demonstrate this, we build the initial target pool from the RIBs (following Rocketfuel's approach) and compute the corresponding portion of dependent prefixes. The raw number of targets in the pool, as well as the corresponding percentage of dependent prefixes can be found in Table 1, in the "Raw number" and "Dependent" columns respectively.

We can observe that the percentage of dependent prefixes greatly varies from one AS to the other. For large Tiers 1, less than 1% of the targets could *potentially* be reduced with Egress Reduction. This is not surprising given that Tiers 1 are involved in almost all Internet traffic (in particular with thousands of customers) and have numerous peering relationships. Additionally, the practice of multi-homing has become more and more prominent, which further explains this very small portion of dependent prefixes. For smaller Transit ASes, the potential reduction can sometimes be slightly better, but is not a panacea either. And for Stubs (or near-Stubs), the potential reduction is indeed greater (from 10% to 90%). However, given the already small number of targets in the pool, there is no use in trying to reduce them further at the cost of additional probes during the pre-probing phase.

All in all, Egress Reduction is not actually helpful. On the one hand, the reduction potential it shows for small ASes does not actually lead to a great saving in terms of absolute number of probes. And on the other hand, the gain for large ASes looks negligible (less than 1%) with respect to the already large number of probes required, as well as for the pre-probing stage.

4.2 Next-hop AS Reduction

The principle behind Next-hop AS Reduction is that the path through an ISP usually depends only on the next-hop AS, not on the specific destination prefix. According to this idea, only one trace from ingress router to next-hop AS is likely to be valuable, which means that all prefixes sharing the same next-hop AS could be reduced to a single probe.

To determine the veracity of this assumption, we evaluated whether we often see multiple different egresses for a given next-hop AS and for a given ingress. We performed the evaluation on our 25 ASes (see Table 1) containing altogether 90,009 (ingress, next-hop AS) pairs, and found that for 30% of the cases, an ingress does see more than one egress when crossing over the same AS. Note that the initial evaluation made by Rocketfuel was conducted on a smaller dataset, i.e., only one AS and 2,500 (ingress, next-hop AS) pairs, and found that the early-exit assumption was violated in only 7% of the cases [40].

To go further, we also simulated the Next-hop AS Reduction on our dataset and found that, in the worst case, this reduction can lead up to a decrease in the discovery of 32% for links, 18% for IP addresses, and 29% for routers. These results confirm that Next-hop AS Reduction now leads to too much false negatives. This can be explained by an increase in peering relationships between ASes [7], notably with IXPs, leading therefore to a flatter but more diverse Internet [8, 49].

4.3 Ingress Reduction

The idea of Ingress Reduction is that routes taken through a network are usually destination-specific. As such, when `traceroutes` from different VPs to the same destination enter the ISP at the same ingress, the path through the ISP is likely to be the same. Therefore, only one `traceroute` from one of the VP would be required.

The first consideration to have is that purely destination-oriented routing is not necessarily the only default rule in use when it comes to forwarding packets, given the rise of TE in the last few years [45]. TE strategies take into account other parameters (such as the type of traffic, its origin, or its flow-id with multipath routing[5]) to optimize the network traffic delivery performance.

Secondly, Ingress Reduction was applicable for Rocketfuel because of its probing design, that initially assigns the complete target list to each VP (meaning that all targets are probed from each VP if no reduction applies). In our case however, our reference dataset subdivides the target list among VPs in order to speed up the probing process (by design, the same destination is not probed from two VPs or more), as it has little impact on the overall quality of discovered data [15].

[5] In theory, the transit traffic flows can be load balanced among multiple (possibly inter-domains) routes according to congestion control algorithms.

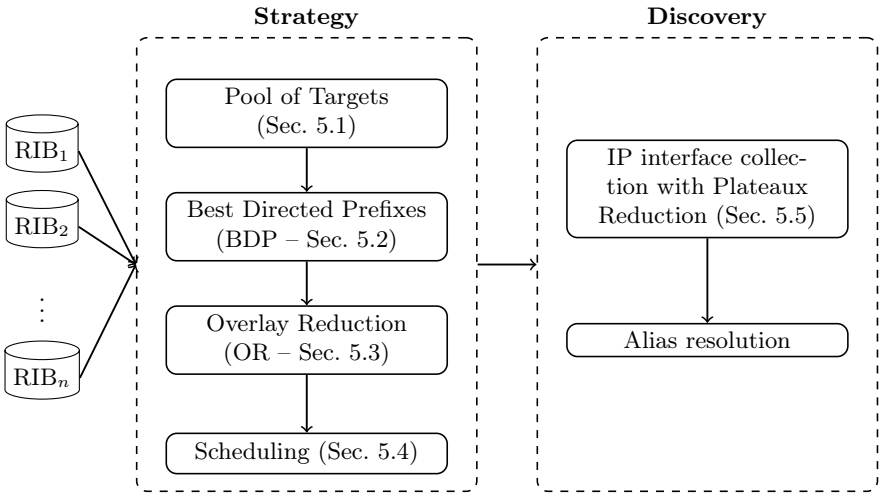


Fig. 1. ANAXIMANDER two steps process.

For those two reasons, we will not consider Ingress Reduction for our study, or any other optimization that could be done on VP placement or targets specific balancing among VPs.

5 Anaximander

This section introduces ANAXIMANDER[6], our new ISP mapping framework. In a nutshell, given a set of vantage points and a targeted ISP $\mathcal{X}$, ANAXIMANDER aims at discovering the most complete map of $\mathcal{X}$ using the minimum amount of probes to enable a quick and efficient measurement campaign.

Fig. 1 illustrates the overall behavior of ANAXIMANDER. As Rocketfuel, ANAXIMANDER starts by collecting RIBs. Afterwards comes the *Strategy* phase, which is run completely off-line and has no probing cost. In this step, ANAXIMANDER relies on three core principles: (*i*) finding an initial pool of targets expected to transit the ISP of interest (Sec. 5.1); (*ii*) applying pruning techniques to this initial pool to reduce the number of probes before probing (Sec. 5.2 and 5.3); (*iii*) sorting and scheduling the targets in preparation for the discovery phase (Sec. 5.4). Those three steps are run in sequence (the output of one is used as input for the subsequent). At the end, the Strategy phase produces an ordered list of targets for probing the ISP of interest.

Secondly comes the *Discovery* phase (Sec. 5.5), in which ANAXIMANDER will send probes based on the target list, taking advantage of the scheduling of the targets to speed up the discovery progression and possibly stop the probing in some portions when the discovery becomes marginal in said portions. Once

[6] Anaximander (610 — 546 BC) is known to be the first to have published a map of the world. See `https://en.wikipedia.org/wiki/Anaximander`

IP interfaces have been collected, ANAXIMANDER can run alias resolution for generating a router-level map of the ISP, using for instance the combination of MIDAR [22] and APPLE [28].

5.1 Initial Pool of Targets

Rocketfuel's Initial Pool of Targets We start our investigation with the same initial pool of targets as Rocketfuel, but without applying any of their reduction techniques, as we have shown they do not offer a good trade-off between coverage and efficiency for the current shape of the Internet.

The core principle for selecting targets likely to transit the ISP of interest (i.e., directed probing) is to take advantage of the routing information contained in BGP routing tables. A BGP entry is composed of a prefix (of any length) associated to several attributes, in particular the `AS_PATH` attribute. If the AS of interest appears in the `AS_PATH` attribute, sending a probe to this prefix is likely to traverse the ISP (in particular if there exists a vantage point co-located to one of the BGP collectors). The number of targets in the initial pool for each AS can be found in Table 1, in the "Raw number" column.

/24 Internal Prefixes Anticipating on the results presented in Sec. 6.2, we actually need to expand Rocketfuel's initial pool with additional targets in order to complete our exploration, given the rather low coverage resulting from it (especially for Stubs and small Transit).

A natural lead to discover most of a given intra-domain AS map is to simply add the AS's internal prefixes to the pool of targets. This time however, we consider a finer granularity and divide the raw prefixes into /24 prefixes (e.g., with prefix 109.75.120.0/22, we split it into four /24 prefixes within the range 109.75.120.0/24-109.75.123.0/24). Basically, our initial pool of target is thus composed, on the one hand, of the AS's internal prefixes (broken down into /24 prefixes); and on the other hand, of the raw[7] directed prefixes found in the RIBs.

Limitations The public BGP information we rely on is already known to be incomplete [32]. As a result, some valid targets may be skipped (i.e., *false negatives*) because of this limitation of the data, although they would have traversed the ISP.

Another important principle in BGP is that there is no single authoritative view of the Internet's inter-domain routing table – all views are in fact relative to the perspective of each BGP speaker [19]. Obviously, the ideal scenario would be to have a VP co-located to each BGP collector, in order to get the exact BGP view from the VP. But since this option is not conveniently available at large scale, we rather combine together multiple RIBs and use this merging as an approximation.

[7] Understand: not broken down into /24 prefixes.

The result of this merging can entail *false positives*, i.e., `traceroutes` that do not traverse the ISP and waste the probing budget. Indeed, the BGP collector that provided the target can be located in a very distant part of the network from the VP that will actually launch the `traceroute`. As their network views potentially do not match (for first AS hops in particular), the probe may not traverse the ISP – even though it would have, had it been launched from the BGP collector instead. These false positives sacrifice the probing budget but not the accuracy, and can be reduced later thanks to our reduction techniques (see Sec. 5.2, 5.3, and 5.4).

5.2 Best Directed Prefixes

This section introduces our first reduction technique, which is based on a simple observation of the workings of BGP routing tables. The BGP information we have access to from the RouteViews [38] and RIPE RIS [43] projects comes in the form of *routing* tables (RIBs), and not *forwarding* tables (Forwarding Information Bases – FIBs). In normal BGP operation, BGP routers typically receive multiple paths to the same prefix. All local routing information learned by a BGP speaker is maintained in the RIB. As such, a prefix can appear multiple times (and with different `AS_PATH`) in the RIB if it has been advertised by multiple BGP neighbors (see Table 2 for an example of this).

For each prefix in the RIB, the route that will actually be used to forward packets and installed into the FIB is determined by the BGP route selection process. BGP has multiple criteria for selecting the *best* route among a set of routes towards a prefix. The first selection criteria is based on local policies defined by network operators, which reflect selfish objectives. Second usually comes the shortest `AS_PATH` criteria, a globally safe criteria, which will select the route with the shortest `AS_PATH`, in order not to burden the network uselessly. If necessary, other more or less arbitrary rules are applied until a tie break is reached. Therefore, when looking at a given RIB to build our initial pool of targets, we are wasting probes on prefixes that *could* be reached through the AS of interest, but that never will, as the route inserted into the FIB can be one that potentially does not go through the AS of interest.

This situation presents the opportunity to perform a first reduction on the initial pool of targets, by building refined FIBs from the collected RIBs. Having no access to the operators' local policy, we approximate it with the *no-valley and prefer customer routing policy* [12], which is a current practice in today's Internet. This policy specifies to prefer a route through a customer AS, over a route through a peer AS, over a route through a provider AS, for economical reasons. In case the routing policy cannot be applied, or if we need a tie-break between two RIB entries, we use the second criteria and select prefixes only if the AS of interest is present in the *shortest* `AS_PATH`. More precisely, we apply this process individually for each prefix in each RIB, before merging the results together. For example, based on the use-case presented in Table 2 (and considering that the AS of interest is the AS3356), we would not select the prefix 72.249.184.0/21 for

#	Prefix	AS_PATH	BGP heuristic
1	72.249.184.0/21	9050 6762 3223 8262 36024	
2	72.249.184.0/21	1230 3223 8262 36024	X
3	72.249.184.0/21	39737 3223 *3356* 36024	

Table 2. Routing Table – Example of multiple paths towards the same prefix. AS 3223 is a pivot AS where traffic can either go towards AS 8262 or AS 3356. If AS 8262 is a customer of AS 3223, and if AS 3356 is a peer of AS 3223 (for example), entry n°3 will be discarded, according to our BGP decision heuristic. Next, entry n°1 will be discarded in profit of the entry with the shortest `AS_PATH`, i.e., entry n°2. AS 3356 being Anaximander's target, this prefix will not be selected for Anaximander's target list.

Anaximander's target list, as it is not present in the prefered path. We call this strategy *Best Directed Prefix* (BDP).

5.3 Overlay Reduction

In this section, we present Anaximander's second reduction technique, relying on a more in-depth analysis of the routing tables.

Forwarding in the Internet is usually done on a longest prefix match basis.[8] As such, a router will always prefer to forward a packet towards the most specific entry to its intended destination. For instance, a router may contain entries for prefix 10.0.0.0/8 as well as a more specific prefix 10.0.5.0/24. Packets towards 10.0.5.12 (or any other address drawn from the more specific prefix) will always be forwarded towards 10.0.5.0/24.

It is thus possible, through BGP, to announce *more specific* prefixes. Given that not all more specific advertisements serve the same purpose, Huston has proposed a classification of these more specific prefixes into three categories, based on the relationship between the more specific and its immediately enclosing aggregate advertisement [18].

The first category is that of the *Hole Punching* more specifics. These more specifics are used traditionally to advertise reachability information, in the case where a block of the aggregate prefix has been attributed to a customer AS. In the routing table, this corresponds to the case where the origin AS of the more specific route is different from the origin AS of the covering aggregate.

The second category covers *Traffic Engineering* use cases. Network operators take advantage of the longest prefix match rule to control the route choices made by other BGP speakers to direct traffic on more specific constrained paths (e.g., towards links with greater capacity, lower latency, or lower cost). In the routing table, this corresponds to the case where the origin AS of the more specific route and its covering aggregate are the same, but where the AS paths differ.

Finally, the third category is called the *Overlays*. In this category, the more specific and its aggregate share the exact same `AS_PATH` (see Table 3 for an

[8] If we do not take into account MPLS forwarding [45], for instance.

Prefix	AS_PATH
1.0.4.0/22	4608 4826 *3356* 56203
1.0.4.0/24	4608 4826 *3356* 56203
1.0.5.0/24	4608 4826 *3356* 56203
1.0.6.0/24	4608 4826 *3356* 56203
1.0.7.0/24	4608 4826 *3356* 56203

Table 3. Routing Table – Example of Overlay category with more specific prefixes. AS 3356 is ANAXIMANDER's target.

example of this). These more specific advertisements actually serve no purpose at all, as the handling of packets in the aggregate or in the more specific will be the same.[9] For this reason, the Overlays category is of particular interest for ANAXIMANDER: because there is no variation in the path towards the ISP of interest, it naturally allows us to reduce the number of probes by selecting a single prefix within a group of overlays.

We thus apply *Overlay Reduction* (OR) to BDP obtained earlier (see Sec. 5.2). To do so, we first compute the overlays groups for all the RIBs we have and combine them together to get the most complete view of the Internet. After this, we cycle through the targets in the pool and randomly select only a single prefix per group of overlays and per VP[10]. For example, based on the use-case presented in Table 3 (and considering that the AS of interest is the AS3356), we would only select one of the prefixes present in the table – prefix 1.0.6.0/24 for instance.

5.4 Targets Scheduling

After having exploited all that we can regarding routing information (with BDP and OR), we now start investigating scheduling of our targets, instead of randomly launching probes. The purpose here is twofold: the first objective is to start by probing the targets that will lead to the greatest discovery, and to put the less successful targets (i.e., those that discover only a few elements, or no elements at all) at the end of the queue. This can be useful in the context of a low probing budget, where it is necessary to stop probing as soon as possible. The second more general objective is to find an ordering or a grouping of the probes that exhibits some patterns to be exploited in order to reduce the probing budget when some explored portions becomes marginal in term of discovery.

To reach these goals, we organize the target list into four main groups: (*i*) first the targets belonging to the /24 internal prefixes of the AS, (*ii*) those

[9] It is believed network operators do this as a messy attempt to mitigate, to some extent, the risks of a more specific routing attack [18].

[10] More precisely, the OR can only be applied on a per-VP basis. Indeed, let us imagine we have two overlays. If those two overlays are taken by two different VPs, we are susceptible to find different addresses and links because of the entry point that will be different for the two VPs.

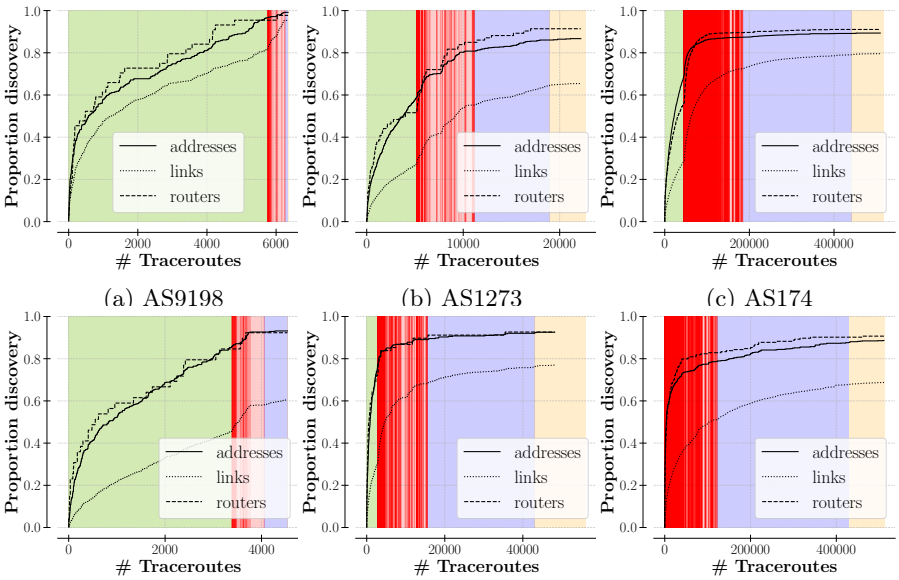


Fig. 2. ANAXIMANDER simulation with targets scheduling. (*i*) /24 internal prefixes in green, (*ii*) direct neighbors in red, (*iii*) one-hop neighbors in violet, and (*iv*) others in yellow. Separation between the ASes is shown with vertical red lines in the direct neighbors group. In the one-hop neighbors and others groups, probes are also grouped by AS, but the separation is not shown for readability purpose.

belonging to the direct neighbors, (*iii*) those belonging to the one-hop neighbors, and finally, (*iv*) the targets belonging to other ASes. Within each of the main groups, probes are gathered together by AS, with no particular order between the probes of an AS. The direct neighbors group is further sub-divided into three categories: the AS's customers, the peers, and the providers, probed in this particular order.[11] Finally, ASes within a group or category are ordered by increasing size of their customer cone, as defined by CAIDA [24, 3]. We will review in the next sections the benefits and reasons for this scheduling.

Results are depicted in Fig. 2, with the X-axis representing the number of `traceroutes` that were launched, and the Y-axis depicting the corresponding levels of discovery during the time progression. Each of the four main groups is represented by a color: green for the internal prefixes, red for the direct neighbors, violet for the one-hop neighbors, and yellow for the others. In the direct neighbors section, the separation between the ASes is shown with vertical red lines. In the violet and yellow group, probes are also grouped by AS, although the separation is not shown with vertical lines for readability purpose. A router is considered

[11] To be able to establish such a classification of ASes, we use the CAIDA AS Relationships Dataset [4].

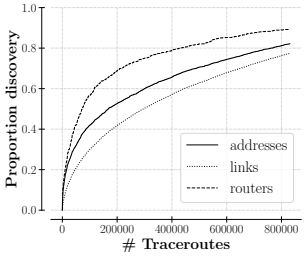


Fig. 3. Simulation with no scheduling of the probes (AS174).

as discovered if we managed to discover at least two of its addresses.[12] Due to space constraints, we present the detailed results only for six ASes (2 Stubs, 2 Transits, and 2 Tier 1) among the 25 we studied, as they are representative of the typical behaviors for their type.

This representation eases the visualization of the probing evolution: one can analyze which group of prefixes is the most important for the discovery of an AS map as `traceroutes` are launched. In the next sections, we will review the contributions of each group successively. For the interested reader, the contributions of each group individually from each others are given in Appendix B.

/24 Internal Prefixes (green group) Targets in this group are launched in no particular order. As can be seen in Fig. 2a (AS9198) and Fig. 2d (AS5400), this group represents almost all discovery for Stub ASes. This is expected given the leaf nature of Stubs, which is not to provide transit and carry traffic for others. As such, only probes directed towards the internal prefixes will hit the AS of interest. For larger ASes, internal prefixes also play a major role in the discovery (especially for addresses), with values ranging from 35% (Fig. 2f –AS1299) to 80% (Fig. 2e – AS4637). The effect for links follows the same lead, although it is less impressive. It is not surprising either that internal prefixes lead to high discovery levels for large ASes. Indeed, probes launched directly into the core of the AS naturally discover a lot of internal elements. However, probing only the internal prefixes is not sufficient in this case to discover the complete AS map.

We also notice that starting to probe with the /24 internal prefixes is beneficial, as it allows us to shift the discovery curve to the left, meaning that most of the discovery of an AS happens at the beginning of the probing campaign. To convince ourselves of this effect, let us have a look at Fig. 3, which presents a probing campaign with no scheduling for AS174. With this campaign, we can observe that the discovery curve is, in fact, already shifted to the left, with a steep increase in discovery at the beginning, followed by a slower rise. This has to do with the properties of `traceroute` exploration, which is naturally very redundant [10]. Indeed, the first `traceroutes` that are launched arrive in unchartered territory, meaning that all addresses and links discovered are new to the observer. The discovery rate at that time is thus very high. However, as time

[12] Two addresses are enough to perform alias resolution with MIDAR and APPLE.

passes, new elements are discovered for sure, but the probes nevertheless go through the same routers again and again. At this time, only a few elements per `traceroute` are thus valuable, instead of the whole `traceroute`, as previously.

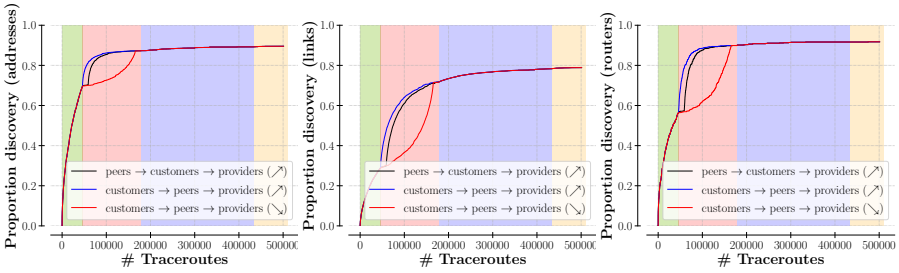
Even though it is naturally shaped to the left, we managed to increase the trend by starting to probe the /24 prefixes. Indeed, we see that for AS174 for example (Fig. 2c), we have already managed to discover 70% of the addresses after having spent only 10% of the probing budget. Compared to the campaign with no scheduling (Fig. 3), we had only discovered 40% of the addresses with 10% of the probing budget. Starting to probe with the /24 prefixes is thus an obvious first step in the right direction to get the most discovery the soonest.

Direct Neighbors (red group) The first thing we can notice is that the direct neighbors also represent a substantial part of the AS discovery (for larger ASes), with values ranging from 15% to 40% for addresses, from 25% to 40% for links, and from 10% to 35% for routers (see Fig. 2c, 2e, 2b, and 2f).

With the direct neighbors, we work on both our objectives, i.e., to group targets in some way as to find a discovery pattern that could be further exploited for reduction, and to shift the discovery curve to the left.

We meet our first objective thanks to the grouping of probes by AS. At first, when Anaximander starts to probe the prefixes of a neighbor, new IP interfaces and links are discovered. After some time though, all routes carrying traffic to the neighbor have been discovered, and it becomes useless to continue probing the neighbor's address space, because paths taken will be redundant with paths already explored. This is reflected in the simulation plots (see Fig. 2e and Fig. 2b for example) by the apparition of *plateaux* in the discovery curve. Indeed, we see that the curve presents small bursts of discovery followed by flat sections, which correspond each time with the beginning of a new neighbor probing. These flat sections correspond to `traceroutes` that were launched but that did not yield any discovery, and present thus the opportunity to be pruned from the list of targets. Some neighbors also do not present any discovery at all. The pattern is exactly the same for other ASes, but is less visible due to the scale of the plots, and the successive reductions applied to the pool of targets, which have already pruned a large number of useless `traceroutes`. In the next section (Sec. 5.5), we will see how this pattern can be exploited for probing reduction.

We meet our second objective by grouping the neighbor ASes into three categories: first the AS's customers, then the peers, then the providers, probed in this particular order. ASes within each category are ordered by increasing size of their customer cone [3, 24]. An AS's customer cone is defined as the ASes and IPv4 prefixes that can be reached through this AS by following only customer links. In other words, an AS's customer cone contains its direct customers, plus its customers' customers, and so on. The size of an AS's customer cone actually reflects the size or the influence of an AS in the routing system. After testing several combinations for the categories and ordering in both the increasing and the decreasing customer cone size, we found the optimal scheduling is indeed the one presented above. The various attempted scheduling for the direct neighbors



(a) Addresses (b) Links (c) Routers

Fig. 4. Alternative scheduling for direct neighbors (AS 174). ↗ stands for ASes ordered by increasing order of their customer cone size, while ↘ stands for the decreasing order. (*i*) /24 internal prefixes in green, (*ii*) direct neighbors in red, (*iii*) one-hop neighbors in violet, and (*iv*) others in yellow.

are presented in Fig. 4 for the particular case of AS 174 (results are similar for all other ASes, but are not presented due to space constraints).

The first thing we notice is that it is more advantageous to start probing the customers rather than the peers (note that the the position of the providers does not have much of an impact and has been left at the end of the scheduling). We explain this phenomenon with the *no-valley and prefer customer* routing policy [12], which is a current practice in today's Internet. BGP routing decisions are mostly based on business relationships and guidelines between ASes. For economical reasons, peer ASes should exchange traffic only between each other and each other's customers, as this traffic generates money for them (either the cost is null or they are paid by their customers). However, an AS should avoid forwarding traffic coming from a peer to a provider (creating so a "valley"), as it can only generate costs for the AS (no gain). For this reason, `traceroute` exploration tends to discover customer-provider links more easily than peer-to-peer links, which are subject to more constraints for the traffic they are allowed to carry. As such, launching `traceroutes` towards a peer of the AS of interest will most likely follow a route without passing through said AS of interest, because the `AS_PATH` containing the peering link is also likely to be longer (compared to a direct customer-to-provider one, if any). Links between the AS of interest and its peer will thus be harder to spot, explaining the lower discovery it brings during ANAXIMANDER's probing campaign (as can be seen in Fig. 4).

Furthermore, the increasing customer cone size order presents the advantageous burst we are looking for, followed by a decrease in the discovery rate). On the contrary, we see the decreasing order yielded the opposite trend of a slow increase followed by a speed up in the discovery. This phenomenon is due to the `traceroute` exploration process of the neighbors. More precisely, when ANAXIMANDER starts to probe the prefixes of a neighbor, new IP interfaces and links are discovered, but the discovery rate ultimately decreases as all routes carrying traffic to the neighbor have been discovered. Therefore, it is beneficial to start probing the small ASes in the AS of interest's cone (i.e., ASes with a low AS

rank) because their address space is smaller; it will thus be explored faster, and the next neighbor (with its associated discovery burst) will be tackled sooner.

One-Hop Neighbors and Others (violet and yellow group) Following the direct neighbors come the one-hop neighbors (violet group) and other ASes even further away (yellow group). Within each group, ASes are also ordered by increasing order of their customer cone [3, 24]. Separating the two groups has no effect whatsoever on the efficiency of the probing, but we present it this way to realize what is the contribution of each group to the AS discovery. The global contribution of next-hop neighbors and other ASes is much lower than the two previous groups, with values ranging from a few percents to a small 5%. Although not very visible in the plots, plateaux are also present in those two groups.

Alternative Scheduling The existence of plateaux may suggest that our scheduling is suboptimal. We have tried several alternative probing scheduling to understand to which extent we can improve the current scheduling. However, our results showed that none of those alternatives have positive effects on the results (see Appendix A). As it happens, grouping together redundant probes to create those plateaux will prove to be a useful characteristic in order to take decisions on marginal benefit while probing. We thus choose to keep the current scheduling, and to work on exploiting this pattern by reducing as much as possible the plateaux, in which no new discovery is made (see Sec. 5.5).

5.5 Discovery Phase with Plateau Reduction

The Strategy phase ends with an ordered list of target, for each ISP of interest. This list, obtained without any probing effort, serves as input for Anaximander second phase, the *Discovery step*.

In the Discovery step, Anaximander may rely on any probing mechanism. In our implementation, we use `TNT` [25], so that we can also reveal the MPLS usage [46] in the targeted AS. Then, from `TNT` IP level data, we run alias resolution (such as Midar [22] and Apple [28]) for providing a router level map.

For its probing, Anaximander will take advantage of the plateaux that appeared in the discovery curve due to the grouping of the probes by AS (see Sec. 5.4). When we reach a plateau during the probing of a particular AS, Anaximander will simply jump to the next AS and continue probing, as the AS marginal utility has become null. Of course, during the actual probing, it is impossible to know whether we are truly encountering the final plateau indicating the end of the discovery, or if additional elements will still be discovered afterwards. On the one hand, if we jump too soon to the next AS, we risk to skip some `traceroutes` that would have discovered new elements. But on the other hand, if we jump too late, we waste the probing budget for nothing.

We therefore introduce a simple *threshold parameter*, τ, that allows Anaximander to control the trade-off between maximizing the discovery and reducing the probing budget. The threshold parameter τ belongs to the interval $[0, 1]$ and represents the length of the plateau (expressed as the proportion of the probed

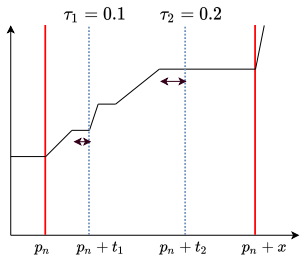


Fig. 5. Plateau Reduction (PR): Effect of parameter τ on probing.

AS address space) after which we jump to the next AS. In other terms, the lower the parameter, the soonest we jump to the next AS. On one hand, when $\tau = 1$, it means ANAXIMANDER does not take into account the plateau and continues probing even if nothing new is discovered for a group (e.g., prefixes belonging to a neighbor AS). On the other hand, when $\tau = 0$, ANAXIMANDER stops probing the plateau as soon as a single probe is useless. We call the effects of τ on probing the *Plateau Reduction* (PR).

An example of the effect of τ is given in Fig. 5. This figure presents a portion of a (fictitious and simplified) discovery curve during a probing campaign. The two vertical red lines delimit the current AS being probed (for example, a direct neighbor of the AS of interest, let us call it AS $\mathcal{N}$). The first probe belonging to AS $\mathcal{N}$ is p_n and the last probe is $p_n + x - 1$. The address space of AS $\mathcal{N}$ thus contains x probes. If we select $\tau_1 = 0.1$ as the threshold parameter, it means we will stop probing AS $\mathcal{N}$ after having encountered a plateau whose length is greater than 10% of the AS's address space. In this case, we thus stop at the first vertical blue line, i.e., at probe $p_n + t_1$. However, if we select $\tau_2 = 0.2$, we stop probing after having encountered a plateau whose length is greater than 20% of the AS's address space. This corresponds to the second vertical blue line, at probe $p_n + t_2$. In this scenario, we see that a threshold value of $\tau_2 = 0.2$ is more appropriate because it allows us to discover all there is to discover, and to prune the remaining of the plateau, thus reducing the probing budget. On the other hand, if we select $\tau_1 = 0.1$, we will jump too soon to the next AS and possibly lose some information.

6 Evaluation

6.1 Methodology

To assess ANAXIMANDER efficiency, we simulate it on the `TNT` dataset (see Sec. 3). More precisely, we replay measurements offline in order to understand what are the respective gains and losses of our probing reduction techniques. Our comparison is thus relative and we consider the brute-force approach (probing of the entire Internet at a /24 granularity) as a baseline offering an upper bound on the probing coverage one cannot outperform by construction. While this baseline provides the coverage upper-bound, it is not able to cover the whole topology (for

example, backup links are not visible if no failure occurs during the campaign): our goal is to offer the same coverage but with a reduced probing budget.

In practice, we do not assume that the `TNT` dataset provides a complete picture of the router-level topology of a given ISP. As a matter of fact, different VPs placements can lead to discovering different portions of the AS. There could even be some unlucky sets of VPs that provide very poor visibility for a specific AS of interest (for example, if all VPs have a certain Tier 1 as their primary provider while the goal of the study is to map another Tier 1).

However, such considerations are outside the scope of this work. It is not our goal here to study VPs placement strategies. Rather, we designed Anaximander to remain a flexible tool that can be launched on any set of VPs, and that will yield the best possible results given that set. This appears to us as the most sensible approach for designing a probing tool, given the difficulty of obtaining VPs to launch a campaign (and the even greater difficulty of placing VPs in strategic locations that would suit the specific purpose of said campaign).

In short, if the initial set of VPs provides very poor visibility into the AS of interest, the resulting maps will obviously not offer high quality absolute coverage. But this is independent of the probing strategies employed (and of Anaximander, *de facto*), and the maps would not have been any more complete with a brute force approach (or with any other probing strategy). This evaluation argues that Anaximander is able to recover (almost) the same ISP maps as obtained with a brute force approach, but with a much more efficient probing methodology.

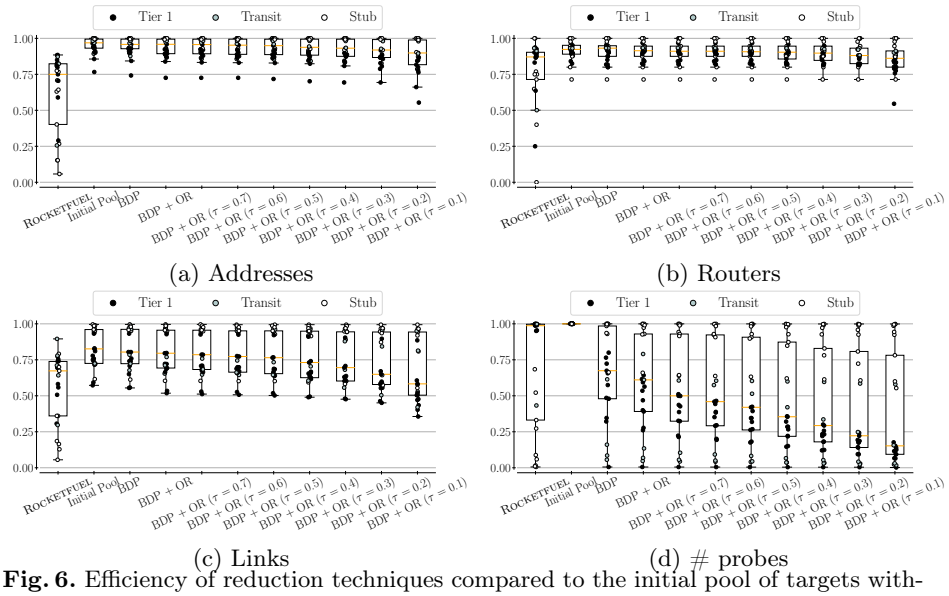
For our evaluations to be relevant nonetheless, we selected 25 ASes (see Table 1) for which we attained good coverage in terms of IP interfaces, links, and routers, given the set of VPs of the `TNT` dataset. We chose ASes with varying sizes and roles in the Internet (11 Tiers 1, 5 Transits, and 9 Stubs) in order to be the most representative. We evaluate our strategies based on two metrics: the percentage of discovery (i.e., completeness) compared to the complete AS map in the `TNT` dataset[13]; and the number of `traceroutes` sent (i.e., probing reduction) compared to the initial pool where no reduction is applied.

6.2 Results

We present in Fig. 6 the simulation results for all ASes and all reduction strategies applied successively (first BDP, then OR, then PR, as shown in Fig. 1). A comparsion with Rocketfuel is also available. Fig. 6a, 6b, and 6c present the final ASes coverage (Y-axis) for addresses, routers, and links, respectively. A router is considered as discovered if we managed to discover at least two of its addresses.[14] As for Fig 6d, it displays the corresponding reduction on the number of probes sent, relative to initial pool where no reduction is applied (see "Initial Pool" label on the X-axis).

[13] As a reminder, AS maps have been isolated with the tool `bdrmapIT`.

[14] As two addresses are necessary and sufficient to perform alias resolution with Midar and Apple.



(a) Addresses (b) Routers

(c) Links (d) # probes

Fig. 6. Efficiency of reduction techniques compared to the initial pool of targets without any reduction, and compared to Rocketfuel. "BDP" stands for "Best Directed Prefixes" (Sec. 5.2) and "OR" for "Overlay Reduction" (Sec. 5.3). The various percentages correspond to the threshold τ for PR (Sec. 5.5).

Results are presented in the form of box plots in order to study the global distribution of ASes' coverage with each strategy. Additionally, each AS is represented by a colored dot (black for Tiers1, blue for Transits, and white for Stubs) to visualize the difference in behaviors depending on the AS type.

Comparison with Rocketfuel As a reminder, Rocketfuel's initial pool of targets is composed of the directed prefixes found in the RIBs, while ANAXIMANDER's initial pool is composed, on the one hand, of the AS's internal prefixes (broken down into /24 prefixes); and on the other hand, of the raw directed prefixes. The first observation we can make for Rocketfuel in Fig. 6a, 6b, and 6c is that the final discovery levels can vary quite dramatically depending on the AS type, with values ranging from a few percents to a perfect 100%.

For Stub ASes, the global trend is to be situated in the lower part of the box plots. We see the final coverage can very often be quite low, with values that can drop around 5% for both addresses and routers, and even to 0% for links. This is actually not surprising given the edge position of the AS in the global network. Such ASes generally appear only at the end of the `AS_PATH` attribute associated to a unique prefix (internal to the AS) resulting so in very few probes launched towards the AS of interest. For large Tiers 1 however, the final coverage can reach much higher values on average, that is 80% for addresses, 70% for links, and around 87% for routers. For Transit ASes, we observe intermediate and

diverse discovery levels, with behaviors similar to Stub ASes for some but that can also span the ones of a Tier 1, depending on the size and role of the Transit AS in the global interconnection.

From these results, we can clearly see that Rocketfuel's initial pool provides quite unreliable coverage and is lacking a lot of targets in order to reconstruct the complete ISP topology. This justifies the need to expand Rocketfuel's initial pool with the /24 internal prefixes in order to complete our exploration. The final levels of discovery for Anaximander's initial pool (enhanced with /24 internal prefixes) can be found under the label "Initial Pool" on the X-axis. For addresses, routers, and links, we see the positive impact on ASes coverage brought by the addition of the /24 internal prefixes. Indeed, the box plots are much denser and higher than in the case of Rocketfuel, meaning that results are consistently better. More precisely, Stubs now almost always present a perfect coverage for addresses, routers, and links. For Tiers 1, the increase in coverage is less impressive, although still present, with a smaller 10% absolute gain. Transit ASes present once again an intermediate behavior between Stubs and Tiers 1.

These results are also coherent with those presented in Sec. 5.4, where we showed that internal prefixes represent a great part of the AS discovery (or even all discovery for Stubs). Adding the /24 internal prefixes naturally brought the box plots up for all types of ASes.

Best Directed Prefixes (BDP) Reduction Looking at Fig. 6a, 6b, and 6c for addresses, link, and routers; we notice that the BDP Reduction had little to zero impact on AS coverage. Indeed, the three box plots have been slightly elongated downwards, but this is almost imperceptible (especially for routers that are quite resistant to any reduction strategies).

If we now take a look at the reduction of probes allowed by BDP Reduction (Fig. 6d), we see it already presents a great potential in reduction depending on the type of the AS. The first result is that, for Stub ASes, there is practically no difference between the initial pool and the BDP Reduction, both in terms of probe reduction and discovery, meaning that the reduction was ineffective. Indeed, we can see across the four figures that all Stub ASes globally remain at their position in the box plots. In fact, this is not surprising given that BDP Reduction is applied to the *directed prefixes* in the pool and that their initial pool is composed mostly of the AS's internals prefixes (see Sec. 5.4 for a visualization of this). For this reason, and anticipating on the next sections, none of the reduction techniques of Anaximander will be effective for Stub ASes. Given the already low number of probes in the initial pool for Stubs, we do not consider it an issue and focus our efforts on reducing the probing budget for larger (transit) ASes.

On the other hand, for Transit and Tier 1 ASes, the effect of the reduction appears clearly with a substantial decrease of 30% on average in the probing budget, without any loss in coverage. Moreover, for some Transits and Tiers 1, BDP Reduction managed to decrease the probing budget by impressive values of up to 90%. The diversity of BGP paths does seem to introduce significant

redundancy, and targeting shortest sub-paths looks to be a good option across multiple vantage points.

Overlay Reduction (OR) As already explained for the BDP Reduction, the effects of the strategy are invisible for Stub networks (we can see all Stub ASes remain in their position in all of the box plots).

However, if we take a look at the impact of OR on Tier 1 and Transit networks, we observe, as expected, a probing reduction (more than 10%) without any significant loss in topology discovery (see Fig. 6a, 6b, and 6c).

Plateau Reduction (PR) Results for PR are presented for different threshold values (τ varying from 0.7 down to 0.1 by step of 0.1). Once again, and not surprisingly, the effects of the strategy are invisible for Stub ASes. For larger ASes however, the impact is much more significant. PR allows for an important reduction of the number of probes with more or less decrease in coverage, depending on the AS, the type of element (IP interface, router, or link), and the threshold value.

Let us look in more details at the threshold impact on the levels of discovery. For addresses and routers (Fig. 6a, 6b), we observe virtually no reduction in AS coverage up until a threshold value of $\tau = 0.4$, where the levels of discovery start to very slowly decrease. The effect of PR is a bit more detrimental for links though, where we see a slightly more important decrease in coverage compared to addresses and routers. Regarding the reduction on the number of probes (Fig 6d), we discern a very clear and steady reduction with each passing value of τ. For $\tau = 0.4$, we managed to reduce the probing budget by an average of 65% with no reduction whatsoever on coverage.

With these results, we can clearly observe the effect of the different values of parameter τ on PR. Smaller values are able to greatly reduce the probing budget, but naturally come at the cost of a decrease in AS coverage. It is up to the users to select a particular threshold value τ that best suits their needs and contraints. Intermediate threshold values might be a better option than the quite radical $\tau = 0.1$ value, as they present a reduction on the probing budget more than acceptable (between 50% and 65%) while maintaining high topology discovery levels. They can also pick a very conservative value ($\tau = 0.7$, for example) that does not decrease the topology discovery at all while still reducing the probing budget of another 10-12%.

6.3 Global Comment

Globally, the general shapes of the box plots are very promising. For addresses and routers in particular, the ASes' coverage distribution remains very high and dense across the various reduction strategies. For links, the coverage distribution remains also rather constant, although it was more spread to start with. And while the coverage remains high, the probing budget is greatly and steadily reduced with each reduction strategy.

All in all, the various reduction strategies we designed are quite effective in reducing the probing budget while still maintaining high coverage. They are also able to adapt themselves to the type of AS being mapped and thus provide flexible but consistent and reliable results across all types of AS.

7 Conclusion

Internet is a complex system made of numerous independent entities called ASes. To understand its structure and characteristics, many attempts have been proposed, developed, and deployed according to the scale and the purpose of the study. In this work, we are interested in the extraction of specific AS router level maps, with a reduced probing budget, and without hampering the resulting topological coverage. Given the difficulty of directing `traceroutes` towards an AS of interest, as well as the fact that many traces lead to redundant paths, the problem is challenging and predictions are not obvious.

To achieve this goal, we proposed Anaximander, a new efficient approach able to recover the same ISP maps as obtained with a brute force approach, but with a network-friendly and efficient probing methodology. For a given ISP and a given set of vantage points, Anaximander will design the best list of targets before actively probing the ISP. In addition, our tool also comes with a simple parameter to control the trade-off between maximizing the ISP coverage and reducing the probing budget. Overall, Anaximander is a generic tool that can adapt to the nature of the AS being mapped (e.g., Tier-1, Transit, or Stubs) thanks to its self-adaptative probing strategies and scheduling. The probing reduction we manage to achieve with our pruning techniques is significant and comes with almost no losses in term of coverage, whatever the kind of AS.

Software Artifacts

The `TNT` data we used throughout the paper is freely available on Caida website: `https://www.caida.org/data/active/ipv4_tnt_dataset.xml`. Reproducibility is therefore quite easy.

Our Anaximander implementation is tunable with a single parameter (the threshold τ for PR), making it easy to use and flexible. The simulator is developed in Go, and is available at

`https://github.com/Emeline-1/anaximander_simulator`

References

1. Abley, J., Lindqvist, K., Davies, E., Black, B., Gill, V.: IPv4 multihoming practices and limitations. RFC 4116, Internet Engineering Task Force (July 2005)
2. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding traceroute anomalies with paris traceroute. In: Proc. ACM Internet Measurement Conference (IMC) (October 2006)

3. CAIDA: AS rank v2.1. https://api.asrank.caida.org/v2/docs (March 2020)
4. CAIDA: The caida as relationships dataset. https://www.caida.org/catalog/datasets/as-relationships/ (April 2021)
5. claffy, k., Hyun, Y., Keys, K., Fomenkov, M., Krioukov, D.: Internet mapping: from art to science. In: Proc. IEEE Cybersecurity Application and Technologies Conference for Homeland Security (CATCH) (March 2009)
6. Cunha, I., Marchetta, P., Calder, M., Chiu, Y., Machado, B., Pescapè, A., Giotsas, V., Madhyastha, H., Katz-Bassett, E.: Sibyl: A practical Internet route oracle. In: Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI) (March 2016)
7. Dhamdhere, A., Cherukuru, H., Dovrolis, C., claffy, k.: Measuring the evolution of Internet peering agreements. In: Proc. IFIP Networking (May 2012)
8. Dhamdhere, A., Dovrolis, C.: The Internet is flat: Modeling the transition from a transit hierarchy to a peering mesh. In: Proc. ACM CoNEXT (November 2010)
9. Donnet, B., Friedman, T.: Internet topology discovery: a survey. IEEE Communications Surveys & Tutorials **9**(4), 2–15 (December 2007)
10. Donnet, B., Raoult, P., Friedman, T., Crovella, M.: Efficient algorithms for large-scale topology discovery. In: Proc. ACM SIGMETRICS (June 2005)
11. Durairajan, R., Sommers, J., Barford, P.: Layer-1 informed Internet topology measurement. In: Proc. ACM Internet Measurement Conference (IMC) (November 2014)
12. Gao, L.: On inferring autonomous system relationships in the Internet. IEEE/ACM Transactions on Networking (ToN) **9**(6), 733–745 (2001)
13. Gay, S., Schaus, P., Vissichio, S.: REPETITA: Repeatable experiments for performance evaluation of traffic-engineering algorithms. cs.NI 1710.08665, arXiv (October 2017)
14. Govindan, R., Tangmunarunkit, H.: Heuristics for Internet map discovery. In: Proc. IEEE INFOCOM (Mar 2000)
15. Grailet, J.F., Donnet, B.: Virtual insanity: Linear subnet discovery. IEEE Transactions on Network And Service Management (TNSM) **17**(2), 1268–1281 (June 2020)
16. Guo, H., Heidemann, J.: Detecting ICMP rate limiting in the Internet. In: Proc. Passive and Active Measurement Conference (PAM) (2018 March)
17. Haddadi, H., Rio, M., Iannacone, G., Moore, A.W.: Network topologies: Inference, modeling, and generation. IEEE Communications Surveys & Tutorials **10**(2), 48 – 69 (August 2008)
18. Huston, G.: BGP more specifics: Routing vandalism or useful? https://blog.apnic.net/2017/06/26/bgp-specifics-routing-vandalism-useful/ (June 2017), last Access: May 17th, 2021
19. Huston, G.: BGP in 2020 – the BGP table (January 2021), see https://blog.apnic.net/2021/01/05/bgp-in-2020-the-bgp-table/
20. Jacobson, V.: mrinfo (1995), see http://cvsweb.netbsd.org/bsdweb.cgi/src/usr.sbin/mrinfo/?only_with_tag=MAIN
21. Keys, K.: Internet-scale IP alias resolution techniques. ACM SIGCOMM Computer Communication Review **40**(1), 50–55 (January 2010)
22. Keys, K., Hyun, Y., Luckie, M., claffy, k.: Internet-scale IPv4 alias resolution with MIDAR. IEEE/ACM Transactions on Networking **21**(2), 383–399 (April 2013)
23. Knight, S., Hung, X.N., Falkner, N., Bowden, R., Roughan, M.: The Internet topology zoo. IEEE Journal on Selected Areas in Communications **29**(9), 1765–1775 (October 2011)

24. Luckie, M., Huffaker, B., claffy, k., Dhamdhere, A., Giotsas, V.: AS relationships, customer cones, and validation. In: Proc. ACM Internet Measurement Conference (IMC) (October 2013)
25. Luttringer, J.R., Vanaubel, Y., Mérindol, P., Pansiot, J.J., Donnet, B.: Let there be light: Revealing hidden MPLS tunnels with TNT. IEEE Transactions on Network and Service Management (TNSM) **17**(2), 1239–1253 (June 2020)
26. Marchetta, P., Mérindol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Topology discovery at the router level: a new hybrid tool targeting ISP networks. IEEE Journal on Selected Areas in Communication, Special Issue on Measurement of Internet Topologies **29**(6), 1776–1787 (October 2011)
27. Marchetta, P., Mérindol, P., Donnet, B., Pescapé, A., Pansiot, J.J.: Quantifying and mitigating IGMP filtering in topology discovery. In: Proc. IEEE Global Communications Conference (GLOBECOM) (December 2012)
28. Marder, A.: APPLE: Alias pruning by path length estimation. In: Proc. Passive and Active Measurement Conference (PAM) (March 2020)
29. Marder, A., Luckie, M., Dhamdhere, A., Huffaker, B., Smith, J., claffy, k.: Pushing the boundaries with bdrmapIT: Mapping router ownership at internet scale. In: Proc. ACM Internet Measurement Conference (IMC) (November 2018)
30. Mérindol, P., David, P., Pansiot, J.J., Clad, F., Vissicchio, S.: A fine-grained multi-source measurement platform correlatng routing transitions with packet losses. Computer Communication (COMCOM) **129**, 166–183 (September 2018)
31. Mérindol, P., Van den Schriek, V., Donnet, B., Bonaventure, O., Pansiot, J.J.: Quantifying ASes multiconnectivity using multicast information. In: Proc. ACM Internet Measurement Conference (IMC) (November 2009)
32. Oliveira, R., Pei, D., Willinger, W., Zhang, B., Zhang, L.: The (in)completeness of the observed Internet AS-level structure. IEEE/ACM Transactions on Networking (ToN) **18**(1), 109–122 (February 2010)
33. Orsini, C., King, A., Giordano, D., Giotsas, V., Dainotti, A.: BGPStream: A software framework for live and historical BGP data analysis. In: Proc. ACM Internet Measurement Conference (IMC) (November 2016)
34. Pansiot, J.J., Mérindol, P., Donnet, B., Bonaventure, O.: Extracting intra-domain topology from mrinfo probing. In: Proc. Passive and Active Measurement Conference (PAM) (April 2010)
35. Pastor-Satorras, R., Vespignani, A.: Evolution and Structure of the Internet: A Statistical Physics Approach. Cambridge University Press (2004)
36. Psenak, P., Hegde, S., Filsfils, C., Gulko, A.: ISIS segment routing flexible algorithm. Internet Draft (Work in Progress) draft-hegdeppsenak-isis-sr-flex-algo-02, Internet Engineering Task Force (February 2018)
37. Ravaioli, R., Urvoy-Keller, G., Barakat, C.: Characterizing ICMP rate limitation on routers. In: Proc. IEEE International Conference on Communications (ICC) (June 2015)
38. RIPE: Ripe ris, routing information service, see `https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris`
39. Senel, B.C., Mouchet, M., Cappos, J., Fourmaux, O., Friedman, T., McGeer, R.: EdgeNet: A multi-tenant and multi-provider edge cloud. In: Proc. International Workshop on Edge Systems, Analytics and Networking (April 2021)
40. Spring, N., Mahajan, R., Wetherall, D.: Measuring ISP topologies with Rocketfuel. In: Proc. ACM SIGCOMM (August 2002)
41. Spring, N., Wetherall, D., Anderson, T.: Scriptroute: A public Internet measurement facility. In: Proc. USENIX Symposium on Internet Technologies and Systems (USITS) (March 2002)

42. Teixeira, R., Marzullo, K., Savage, S., Voelker, G.: In search of path diversity in ISP networks. In: Proc. ACM SIGCOMM Internet Measurement Conference (IMC) (Oct 2003)
43. University of Oregon: Route views, University of Oregon Route Views project, see `http://www.routeviews.org/routeviews/`
44. Vanaubel, Y., Luttringer, J.R., Mérindol, P., Pansiot, J.J., Donnet, B.: TNT, watch me explode: A light in the dark for revealing MPLS tunnels. In: Proc. IFIP Network Traffic Measurement and Analysis Conference (TMA) (June 2019)
45. Vanaubel, Y., Mérindol, P., Pansiot, J.J., Donnet, B.: MPLS under the microscope: Revealing actual transit path diversity. In: Proc. ACM Internet Measurement Conference (IMC) (October 2015)
46. Vanaubel, Y., Mérindol, P., Pansiot, J.J., Donnet, B.: Through the wormhole: Tracking invisible MPLS tunnels. In: Proc. ACM Internet Measurement Conference (IMC) (November 2017)
47. Waddington, D.G., Chang, F., Viswanathan, R., Yao, B.: Topology discovery for public IPv6 networks. ACM SIGCOMM Computer Communication Review **33**(3), 59–68 (Jul 2003)
48. Wang, F., Mao, Z.M., Wang, J., Gao, L., Bush, R.: A measurement study on the impact of routing events on end-to-end Internet path performance. In: Proc. ACM SIGCOMM (August 2006)
49. Wang, Y., Zhang, K.: Quantifying the flattening of Internet topology. In: Proc. International Conference on Future Internet Technologies (June 2016)
50. Zhang, Y., Mao, Z.M., Wang, J.: A framework for measuring and predicting the impact of routing changes. In: Proc. IEEE INFOCOM (May 2007)
51. Zhang, Z., Marder, A., Mok, R., Huffaker, B., Luckie, M., claffy, k., Schulman, A.: Inferring regional access network topologies: Methods and applications. In: Proc. ACM Internet Measurement Conference (IMC) (November 2011)

A Alternative Schedulings

As explained in Sec. 5.4, our grouping of targets by AS has made plateaux appear during the probing, which can be exploited to prune useless probes from the list of targets in real time. However, even if we do manage to reduce the plateaux as much as possible while still maintaining a high coverage, a certain portion of the plateau (shorter or longer depending on the τ parameter) will still be explored. Indeed, Anaximander is no oracle and cannot know in advance if the final plateau has been reached or if there will still be some discovery afterwards. Therefore, it needs to explore the plateau before deciding whether it is safe to skip this portion and jump to the next AS or not. In short, plateaux are reduced, certainly, but some probes are still wasted on a regular basis in the middle of the probing.

An alternative scheduling would be to launch the exploration of ASes in parallel, rather than explore them sequentially. The Plateau Reduction would still be applied on a per-AS basis, allowing to prune useless probes as usual. At first glance, this new scheduling could have the effect of shifting all bursts of discovery sooner into the probing campaign, and to relegate all remaining plateaux towards the end. We have explored two parallel scheduling.

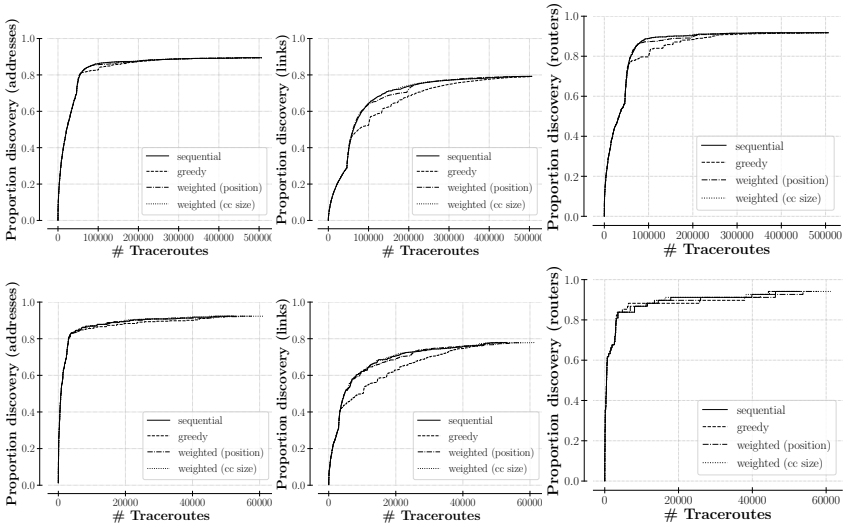
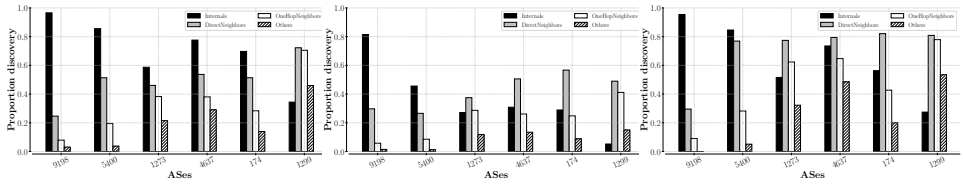


Fig. 7. Comparison of the discovery curves between the different scheduling strategies (AS174 on top, AS4637 bottom). The left column present the results on addresses, the middle column on links, and the left column on routers.

The first one, called *greedy scheduling*, will halt its probing of an AS as soon as it encounters a useless probe, and get back to it at a later time in the probing campaign. This could have, on paper, a beneficial effect on the discovery curve, by shifting it to the left and limiting the useless probes early on the campaign. Let us be careful not to confuse this concept with Plateaux Reduction. In this greedy scheme, Anaximander does not stop exploring an AS entirely at the first useless probe. It will just get back to it later, to try to relegate a potential plateau towards the end of the probing campaign. The remaining length of the plateaux are still explored at the end of the discovery curve (and they will still be reduced by Plateau Reduction).

The second one, called *weighted scheduling*, attributes a weight to each AS (based on various criteria), and explores this AS's address space in successive batches according to that weight. More precisely, based on the weight of an AS, Anaximander will explore only a given portion of that AS's address space before jumping to the next one. Once Anaximander has browsed over all ASes in such a way, it will get back to the first AS in the list and resume its probing with yet another batch. The probing of an AS will thus continue in successive batches, up until all of the AS has been probed, or until the probing is stopped because of a plateau. For our experiments, we have tested two different weighting functions, one based on the AS's relative position in the list of ASes, and the other based on its customer cone size [24]. Each function has been tested with varying parameters and results are presented for the parameters that yielded the best results.



(a) Addresses (b) Links (c) Routers

Fig. 8. Individual contribution of each of the four main groups (internal prefixes, direct neighbors, one-hop neighbors, and others).

We present our results (Fig. 7) for two ASes of Interest (one Transit and one Tier 1). Results are not presented for Stub ASes, whose pool of targets is mainly composed of /24 internal prefixes, as the scheduling of probes remains the same in this particular group. Results for the other ASes in our sample of 25 ASes lead to the same conclusions but were not presented due to space constraints.

Across all ASes and all types of elements (addresses, links, or routers), the results are unequivocal: ANAXIMANDER's sequential scheduling always outperforms (or is at least equivalent to) the other scheduling strategies. Looking in more details, we see that the greedy scheduling consistently performs worse than the other three strategies and can thus be discarded. On the other hand, with carefully crafted weight functions, we were able to get as close as possible to the same performance as ANAXIMANDER's current scheduling. More precisely, the weight function based on the customer cone size generally performs better than the one on the position of the AS. This is not surprising, as the customer cone size retains more information than simply the ordering of the AS in the global AS list.

B Individual Group Contribution

Fig. 8a, 8b, and 8c show the contribution of each group of prefixes (internals, direct neighbors, one-hop neighbors, and others) for addresses, links, and routers respectively, so that the reader can really appreciate what is the cost/-coverage ratio for each group. The same general trend remains: for any type of AS, internal prefixes are the most likely to enable a good ISP coverage for addresses (this group can even be seen as almost sufficient for stub ASes). For Tier-1 however, probing neighbor ASes becomes necessary to complete the exploration. This is particularly true for links and routers, that really benefit from probing the neighbors, and where internal prefixes are insufficient to cover the whole topology.