TU WIEN Informatics

Aggregation & Analysis of IPv6 Prefixes at Internet-Scale

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

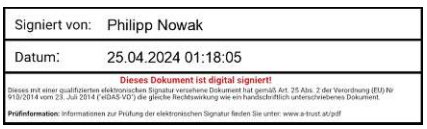
Philipp Nowak, BSc
Matrikelnummer 11776858

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Mitwirkung: Univ.Lektorin Dipl.-Ing. Dr.techn. Johanna Ullrich, BSc

Wien, 25. April 2024



Philipp Nowak

Edgar Weippl

TU WIEN Informatics

Aggregation & Analysis of IPv6 Prefixes at Internet-Scale

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Philipp Nowak, BSc
Registration Number 11776858

to the Faculty of Informatics

at the TU Wien

Advisor: Privatdoz. Mag.rer.soc.oec. Dipl.-Ing. Dr.techn. Edgar Weippl
Assistance: Univ.Lektorin Dipl.-Ing. Dr.techn. Johanna Ullrich, BSc

Vienna, April 25, 2024

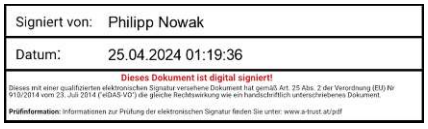
Philipp Nowak	Edgar Weippl

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.at

Erklärung zur Verfassung der Arbeit

Philipp Nowak, BSc

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.



Wien, 25. April 2024

Philipp Nowak

Acknowledgements

This piece of paper you are holding right now, or, more realistically, seeing on a digital screen, took me multiple years to produce. During this time and before it, so many great people have helped to make it possible for me to write this. Thank you to everyone for making *the* world and *my* world a better place.

I believe that a significant success factor for this project are the valuable ideas, patience, and pragmatic working mode contributed by my advisor Johanna Ullrich. Without this supportive environment, it would not have been possible for me to get anywhere near completion. Further thanks go to Markus Maier, who provided both additional input and support with the measurement nodes.

I would also like to acknowledge the scientific community in this field, where many talented individuals and teams work together to further human knowledge. Attention to detail, well-written research papers, a culture of sharing tools, and consideration for ethics do not go unnoticed.

A core aspect for success is all the indirect support I have been lucky to receive. So many individuals have contributed to the necessary environment and encouraged me to continue this 2.5-year-long journey. I deeply appreciate the support of my family, friends, and community. Countless current and past colleagues at my company have provided vital flexibility, support, and time management ideas, enabling me to complete this thesis while working.

Finally, I would like to acknowledge everybody that makes this place we call the internet more liveable. This includes so many inspiring people worldwide, both in terms of content and making it actually work on a technical level. It is important that we work together to protect this space.

Abstract

Due to its crucial importance in today's world, researchers are interested in understanding large-scale real-world usage of the internet. Concrete motivations vary, but are commonly related to security research.

The internet community is in the process of exchanging the base protocol used for internet communication, introducing IPv6. This more modern protocol significantly expands address space from 2^{32} to 2^{128} possible addresses. While there are many benefits to this modernisation, researchers have been relying on exhaustive internet measurements for decades. Existing methods cannot directly cope with the considerable expansion of search space, and the community is interested in new ideas that restore the ability to observe the entire internet. Current solutions are still characterised by trade-offs and limitations.

Structure-aware probing is an idea that addresses some of these. Instead of exhaustively and linearly inspecting every possible address, this method recursively finds interesting spaces by repeatedly splitting the measurement space, akin to binary search. A focus on high-level prefix structures allows experiments to target wide areas of the internet, while not requiring exorbitant probing rates due to re-use of previously-discovered information. The thesis proposes an algorithm that applies this method by combining a variety of existing concepts.

Evaluation against a benchmark linear measurement, ground truth data, and behavioural metrics shows that structure-aware probing is a promising idea. While there still is work to be done until this particular system can reliably scan the entire internet, results are in general promising, affirming successes from existing work with related methods.

Kurzfassung

Aufgrund der entscheidenen Bedeutung des Internets für die heutige Gesellschaft ist die Wissenschaft daran interessiert, seine praktische Verwendung im großen Rahmen zu verstehen. Konkrete Motivationen unterscheiden sich, häufig besteht aber ein Zusammenhang zu IT-Security.

Bestehende Methoden sind gefährdet, da das Internet auf das modernisierte Basisprotokoll IPv6 umgestellt wird. Dieses erweitert den Addressraum von 2^{32} auf 2^{128}. Seit Jahrzehnten etablierte Messmethoden können mit dieser beträchtlichen Erweiterung jedoch nicht umgehen, weil sie alle möglichen Adressen einzeln prüfen. Daher ist die Gemeinschaft auf der Suche nach neuen Ideen, um das IPv6-Internet vollständig messen zu können. Bestehende Lösungen sind jedoch noch durch Einschränkungen geprägt.

Die Idee des *Structure-Aware Probing* könnte einen Teil dieser Herausforderungen bewältigen. Anstatt jede mögliche Adresse zu prüfen, teilt sie den Suchraum wiederholt, ähnlich zu *Binary Search*, um interessante Regionen zu identifizieren. Durch Verschiebung des Fokus auf größere Netzwerkstrukturen und Wiederverwendung bestehender Ergebnisse können Messungen weite Teile des Internets betrachten, ohne exorbitante Bandbreiten zu benötigen. Diese Arbeit entwickelt einen Algorithmus, der diese Methode durch Kombination diverser bestehender Konzepte umsetzt.

Auswertung anhand einer linearen Kontrollmessung, einem bekannten Netzwerkplan und Verhaltensmetriken zeigt, dass *Structure-Aware Probing* eine vielversprechende Idee ist. Obwohl das konkrete System noch Verbesserungspotenzial bei der Resilizenz zeigt, sind die Ergebnisse grundsätzlich positiv und bestätigen vergleichbare Resultate verwandter Studien.

Contents

CHAPTER 1

Introduction

Due to its crucial importance in today's world, researchers are interested in understanding large-scale real-world usage of the internet. Concrete motivations vary, but are commonly related to security research. For example, seeing how often a vulnerable software version is in use can help estimate impact on productive systems beforehand and check patching behaviour afterwards. Related data might also help motivate network policies and protocol improvements. Similarly, large-scale analysis of real-world system behaviour can help uncover privacy and security issues in widely-used protocols, as demonstrated by [RBC21].

The internet is a vast space characterised by a variety of independent actors. Their behaviour is not centrally catalogued by a single entity that could be consulted for census data necessary for scientific research. Consulting every single party for a survey is infeasible, which is why scientists need to gain information differently. An alternative technique is to exhaustively probe every IP address. IP addresses are necessary to communicate on the internet, and such probing does not assume cooperation from the target.

In fact, this technique has been providing researchers with data necessary to their studies for decades. Current tools can inspect every publicly-reachable address on the internet in minutes, depending on what needs to be measured. Still, active internet research is not a solved problem. The community is in the process of adapting new version of the foundational Internet Protocol, IPv6, which still poses inherent methodological challenges. While this process has been going on since at least 1998, it has been gaining additional traction in recent years due to depletion of available addresses in the old IPv4 version. [ADSH14, Bev16, DH17, DC98]

This exhaustion of IPv4 addressing resources is a core motivation to adopt the more modern protocol. IPv6 offers relief by expanding the (theoretical) address space from 2^{32} to 2^{128} addresses. This astronomical expansion fundamentally changes addressing

philosophy, which in IPv4 often amounts to tedious management of a scarce resource. The new protocol on the other hand aims to provide an abundance of space by design. Even the smallest sub-departments could grow to the size of the entire IPv4 internet in the future without any need for renumbering. Usually, the minimum size of a single network segment in IPv6 is 2^{64} addresses. This amounts to 2^{32} *times* the entire public IPv4 internet. [DH17, DH06]

This expansion implies that address space usage will be very sparse. Even if researchers are aware of specific subnetworks that are interesting for analysis, they are highly unlikely to guess even a single active address without significant effort. The same principle also applies to active networks themselves. Due to exponential growth of effort compared to IPv4, existing methods are no longer directly applicable. Even the most efficient techniques would be unable to probe all 2^{128} addresses in any reasonable time frame. For this reason, the scientific community needs alternative techniques for internet measurement. The literature already offers some ideas, for example the collection of a public "hitlist" of active IPv6 addresses that can be probed.

Another possibility is to shift the focus from individual addresses to entire network structures. While there are still 2^{64} possible networks in theory, these are often assigned hierarchically due to the internet routing architecture, with significant portions remaining unused. An algorithm that is aware of this structure might aid in identifying interesting regions, which can then be investigated recursively to find active networks. While this idea has been explored in various contexts [RB20a, LaF15, LS16], there still does not seem to be a standard tool that applies this mechanism to IPv6. [DH06]

An interesting property of network structures is that at least larger ones form an inherent component of an organisation's network architecture, which makes them less likely to fluctuate frequently. This implies that inferences made from previous rounds of measurement can be expected to remain reasonably consistent over a longer time than individual addresses, which sometimes even rotate automatically by design.[1] Such a reliance on existing structural knowledge may allow more fine-grained probing in the long term, since higher-level structures do not need to be rediscovered with every measurement.

The method proposed in this thesis aims to combine these two ideas. A focus on high-level prefix structures allows experiments to target wide areas of the internet while not requiring exorbitant probing rates due to re-use of previously-discovered information. Due to the recursive nature of the algorithm, more granular subnetwork allocations should be discoverable than when linearly probing the space. This mechanism also reduces the number of probes sent to areas that are unlikely to reveal further topology.

[1] Small prefixes can also exhibit similar behaviour, for example end-user networks are commonly rotated by ISPs. [RBC21] Larger aggregates that are inherent to the network architecture should be more difficult to change, requiring e.g. changes to static core router IP address configuration, motivating the assumption that they do not.

Research questions that motivate the thesis' focus are laid out below. Each of the questions targets one of the two key areas of innovation.

Prefix Aggregation

1. How can prefix aggregation be performed in a meaningful yet efficient manner?
2. Is it likely that this information can be used to **enhance the scanning hit rate**?

Value Metric

3. How can **more "valuable" (i.e. higher hit probability) target addresses** be discerned based on results of previous scans?
4. How can this "value metric" be **stored and updated** over multiple scans, ideally without significantly impairing scanning or analysis rate?
5. Is it possible to **enhance the hit rate** based on this "value metric"?

Contributions made by this thesis are:

- Comprehensive Literature Research presenting the relevant state of the art.
- Proposal of algorithms and surrounding system design (Measurement Setup). Key innovations are feedback mechanism, dynamic focusing of probing into interesting subnets instead of fixed-size spaces, and confidence metric.
- Testing the architecture in multiple long-running Experiments.
- Qualitative & Quantitative Evaluation of results against a benchmark measurement, ground truth, and metrics, targeting both general concept and concrete implementation.

The remainder of this thesis is structured as follows. Chapter 2 introduces the necessary context and provides an overview of the related state of the art. Chapter 3 shows in detail the measurement setup and explains why it is set up that way. Results of the measurements are evaluated in chapter 4, and finally chapter 5 summarises results of the work and addresses the research questions.

Ethical considerations are discussed in general for this kind of research in chapter 2, and applied to the proposed setup in chapter 3.

The thesis implementation is published as open-source software at `https://github.com/literalplus/prefix-crab`.

CHAPTER 2

Background

This chapter explores the related background, laying the foundation for measurement design. This resulting design is presented in chapter 3.

Exploration starts with a general discussion of IPv6 network measurements in section 2.1. First, the necessary IPv6 networking basics are presented, with a focus on the related ICMPv6. This protocol is the primary measurement target, motivating discussion of what information can be obtained from unknown networks using this protocol, and which limiting factors exist.

With this basic understanding, learnings can be drawn from successful IPv4 measurements and early IPv6 experiments in section 2.2. As more studies focus on IPv6, simplifications turn into more nuanced trade-offs, leading to the current state of the art in section 2.3. These measurements already discover significant information at high probing rates. Focus however remains on individual addresses. This changes as the scope expands to higher-level aggregates and topology. These concepts motivate the key ideas of the proposed method, which are discussed in section 2.4.

The theoretical background is finalised by section 2.5, a discussion of general ethical implications presented by internet measurements.

2.1 Network Measurements with IPv6

IPv6 is a protocol that devices use to communicate on the global internet. Every participating device receives one or more 128-bit identifiers, called *IPv6 addresses*, that others use to indicate it as the communication target. Due to the large size, IPv6 addresses are denoted in hexadecimal, grouping two bytes with a colon, and omitting redundant zeros: `2001:db8:0:0:0:0:0:beef`. Additionally, long runs of zeroes are collapsed to `::`, resulting in `2001:db8::beef`. [DH17, KK10]

If a device wants to reach an address that it is not directly connected to, it passes the message to a special device in a process called *IPv6 Routing*. This *router* then forwards the message, with each following router again delegating to a neighbour until the target can be reached directly. The process is repeated in the opposite direction for the response. [DH17]

Core mechanisms of this communication are governed by the IPv6 standard. It defines the unit of communication as a string of binary data, called an *IPv6 packet*. Each packet follows the same format. It starts with an *IPv6 header* indicating sender and receiver along with other parameters required for delivery, similar to a physical envelope. After that, the content of the message, called *payload*, follows. [DH17]

Since not every message on the internet has the same purpose, the payload may be of various different formats itself. These are in turn governed by dedicated *upper-layer* protocols, which the sender identifies by their assigned number in the IPv6 header. Similarly, IPv6 is itself embedded in a *link-layer* protocol, which is responsible for linking directly neighbouring devices, i.e. those that don't require routing[1] to communicate. [DH17]

In addition to upper-level protocols, IPv6 packets may also embed control messages. These messages are governed by a special protocol called *ICMPv6*, which is an important part of IPv6, but represented like an upper-level protocol. Its packets do not fundamentally carry application data, but instead are used for purposes like error reporting, diagnostics, and automatic configuration. [GC06, SNNS07]

For our purposes, four of the core ICMPv6 message types are relevant: [GC06]

- Destination Unreachable signals that a router on the path was unable to continue forwarding a packet.
- Time Exceeded prevents packets from entering infinite loops by limiting the number of routers they may pass. It is sent when a packet's *Hop Limit* counter reaches zero.
- Echo Request is used for diagnostics. When such a packet reaches a device, it returns a corresponding Echo Reply to the sender. If the response arrives, the sender knows that the target is in general reachable.

While these packets are crucial to make IPv6 work as intended, attackers or researchers may use them to obtain information about a target network. One common way to do so is to deliberately cause Time Exceeded messages by sending packets with a low *Hop*

[1] This statement is also true when tunnelling over IPv6 itself. IPv6 packets cannot only be carried by "classic" link-layer protocols such as Ethernet, but (parts of) the path can also be IP links. The "outer" IP packets are then routed independently until they reach the end of the tunnel. This process is transparent to the routing of the original IPv6 packets, which only continues once they are unpacked at the tunnel exit, so the statement stands. [DH17, DC98]

Limit to begin with. Each router on the path further decrements the Hop Limit, and once it reaches zero, a Time Exceeded response reveals a router's IP address. This can be repeated, each time adjusting the initial Hop Limit, to discover all routers on the path to the destination in a process called *Tracerouting*. [Jac00]

This process can for example be triggered by Echo Request messages, but in general may use any upper-layer protocol. If a stateful protocol such as TCP is used, there is a risk of causing the target to wait for continuation of the conversation, which may be considered as more intrusive. A stateless protocol such as UDP can be an alternative, but may also cause applications to react, or a security policy to discard the traffic. For example, [LHH08, Gas17] observe significantly fewer responses to UDP probes compared to ICMP(v6). While UDP is the default in the original `traceroute` tool [Jac00], these results suggest preferring ICMPv6 Echo Requests as probe packets on the public internet.

When relying on Echo Requests to probe destinations, common results [Hol20] can be divided into three categories that each suggest different conclusions about the target:

- An **Echo Reply** indicates that this address is actively in use. [2]
- A **Destination Unreachable** error may indicate that a block of addresses is in use, but not this particular address, depending on its specific error code. It may suggest that responsive addresses can be found in the vicinity.
- Finally, the least conclusive result is **no response**, which may mean that the address block is not in use.

  A security policy that filters ICMPv6 packets on the path behaves identically. While the specification [GC06] requires that all devices implement an echo responder, it does not explicitly prohibit such filtering. In addition, this behaviour may be observed if these errors are disabled, which is explicitly permitted by the RFC, or if *ICMPv6 Rate Limiting* occurs.

Such rate limiting is explicitly mandated by the specification. This is in contrast to ICMPv4. The ICMPv6 RFC recommends a burst-tolerant algorithm to permit use cases like treaceroute, which often causes many responses from the same device in a short interval. However, a minimum allowed rate is not specified apart from an example. This means that when sending many probes to addresses on the same path, responses may be missing due to this mechanism. It is nontrivial to limit the sending rate appropriately, as the allowable value is not standardised. [GC06]

[2] This does not necessarily mean that a dedicated device is present in the usual sense. Some organisations have infrastructure that responds to whole blocks of addresses, which is called *Aliasing*. See [GSF+18] for a discussion of this phenomenon.

2.2 Learning from IPv4

In addition to the latest Internet Protocol version 6, the older IPv4 is still in heavy use today. It has been the de-facto internet protocol for a long time. As a result of this, internet measurements have historically focused on IPv4. Seriously targeting IPv6 is a development of the last few decades, motivated by slowly increasing IPv6 adoption in the real world. This section provides an overview of IPv4 measurements, explains the fundamental differences between the two protocols from a measurement standpoint, and briefly discusses early attempts to probe the IPv6 internet.

Arguably, the foundations for IPv4 topology measurements were laid around the year 2000. A key example is *Mercator* [GT00], which produced a router-level map of the internet at the time. It uses a combination of Traceroute and *Source Routing* [3]. Mercator cites a 1998 paper [PG98] as the earliest known similar project.

An important realisation from that research is that alias resolution must be applied to topology maps. This means that multiple IP addresses of the same router are normalised to reduce artefacts in the map, which might otherwise invite incorrect conclusions. The *Paris traceroute* technique [ACO+06] advances this mechanism by providing an algorithm that also removes artefacts due to load balancing. Current tools still rely on ideas produced by this research. [HRAD20, LS16]

With such modern tools, it is now possible to traceroute the entire IPv4 space in around an hour. This figure is achieved by 2016's `yarrp` [Bev16]. One of its primary innovations is the concurrency design of its tracerouting mechanism, which achieves improved performance characteristics using two key ideas. First, the necessary data for traceroute computation is attached to the measurement packets, instead of storing it in memory. This works because the original payload is quoted in Time Exceeded messages, allowing the state to be recovered exactly when and where necessary. In addition, yarrp sends probes with all Hop Limit values immediately, instead of waiting for a lower Hop Limit's response. `yarrp`'s metrics are further improved by 2020's *FlashRoute* [HRAD20].

For simple one-shot scanning without traceroute, numbers are even more promising. `zmap` [ADSH14, DWH13] already achieves measurement durations below five minutes in 2014. The difference in magnitude is explained in part by the fact that the traceroute necessary for a topology scan requires multiple packets per target address, while a one-shot scan only needs one. In this case, only one TCP probe for a single port is sent to every address on the IPv4 internet. Even with, and also because of, the statelessness achieved by yarrp, traceroute still inherently produces much more traffic.

This increase in traffic however is dwarfed by what would be required to exhaustively probe IPv6. Compared to its predecessor (2^{32} addresses), it offers a 2^{128} address space, which is 28 (base 10) *orders of magnitude* larger. The extent of this size difference is difficult

[3] Source Routing means that the sender of a packet requests a specific routing path to be taken, instead of on-path routers selecting it as usual. This was supported in early versions of the IPv6 standard, but has since been deprecated for security reasons. [NNSA07]

to visualise. For example, the smallest block of IPv6 addresses that is recommended [RHN11, IAB01, CCG+15] to be allocated comprises 2^{64} addresses. The endeavour of scanning even a single one of these networks exhaustively is equivalent to scanning 2^{32} *times* the entire IPv4 space. With the `zmap` scanning rate advertised above, such a scan would take more than 40 000 years. The same calculation can be applied to the idea of probing a single address in every such network in theoretical existence, which would take equally long. In practice, probing every *allocated* network can be feasible, because only a very small portion of the IPv6 space is currently allocated. [GC16]

Another consequence of the larger address space is that IPv6 will be populated much more sparsely than IPv4. [DH17, GC16] With an enormous 2^{64} addresses, even the largest enterprises would not be able to exhaustively fill even a single one of these networks. This means that uniformly random probing is likely not to hit many addresses assigned to actual devices. This assumption is confirmed for example in [RLB16].

On the other hand, visibility into the internal structure of networks might be expected[4] to increase. With IPv4, it is common practice to aggregate many devices to conserve address space. This is managed by a special device that acts to the outside with a single[5] public IPv4 address. Internally, it translates packets to and from the private IP addresses of devices behind it that actually communicate in a practice called *Network Address Translation* (NAT). In a properly-designed IPv6 network, such a practice should in general not be necessary because there are enough IPv6 addresses and networks to assign a public address to every device. [HS99, ZTL10, DH17]

A summary of these theoretical considerations can be found in RFC 7707 [GC16], which updates an earlier RFC from 2008 [Cho08]. These documents explain how there are circumstances that cause structure to be present inside the components of a network's addresses. Apart from technology-inherent structure, for manual assignments, humans tend to pick values that follow some schema or are easy to remember. By scanning common values first (e.g. the first address in the network), measurements can potentially hit targets earlier. This is however not useful if the goal is to determine whether the network is in use or not, because such methods may increase hit probability but cannot guarantee that other addresses are unused. For adversaries such as worms [BCK06], such tactics can still make a difference, because their intention is not to answer a research question, but to find some targets to compromise.

In addition to these search space optimisation methods, the authors also raise more creative means. While the strategies discussed so far rely on actually sending probes (Active Measurement), other procedures utilise external side-channels. Examples

[4] Independent of NAT, visibility into the internal network structure can also be reduced by employing strict security policies [ZTL10] that discard packets that would disclose internal addressing information. If the primary motivation for NAT deployment is obscurity, then it can be expected that such measures would be deployed, cancelling out the effect.

[5] Multiple public addresses managed by a single NAT device are possible. This complication is omitted here to simplify the explanation. For a detailed discussion of NAT in IPv4, refer to [HS99]. A detailed discussion of NAT in IPv6 can be found in [ZTL10].

that can reveal addresses include DNS and reverse DNS, participation in application-level protocols, routing tables, or, if some local devices are already compromised, neighbour caches and log files. Approaches that do not directly probe targets and exclusively rely on side-channels are called Passive Measurement. [GC16]

Independent of these theoretical considerations, practical scans of IPv6 networks have been published much earlier. In their 2003 paper [WCVY03], Waddington et al. demonstrate that IPv6 topology can be discovered using active probing. Instead of the public IPv6 internet known today, their target is a large IPv6 trial deployment, *6bone*. The scan relies on the now-deprecated source routing. Despite both target and method no longer being applicable, their study shows IPv6 has already been considered for measurement in early stages of its roll-out.

Since these early days, an important contributor of measurement data has been the CAIDA network research initiative. Their continuous internet probing programme, *Ark* [Cen20], begun in 2008 and continues to serve as a key foundation for a significant portion of internet research. Its basic principle relies on address block announcements from *BGP* [RHL06, MD99]. This protocol is used on the internet to negotiate routing paths and address reachability. Snapshots of its state are made available publicly by various organisations, providing insight into which IP address blocks can be reached in practice.[6] For each of these blocks, CAIDA probes a random IPv6 address in 48-hour intervals, Historical results of these measurements are available to the research community.

Early IPv6 research made possible by their infrastructure includes a 2008 study of address allocation strategies [Mal08] and a 2013 topology measurement of 49 000 routers [BBLR13].

2.3 State-of-the-art Applications to IPv6

Since the first measurements, IPv6 probing methodology has come a long way. Various methods have been proposed to produce candidate IP addresses to probe, with increasing quality. These methods, both active and passive, have been combined into bias-reduced address hit lists. This is especially beneficial because researchers can rely on established hit lists. They no longer need insider access at a network provider to acquire good-quality seed data for internet measurements. This section provides an overview of the available tools and methodology for IPv6 internet measurements, both for individual targets and for traceroutes.

As discussed, the IPv6 address space is both vast and sparse. It is crucial to find methods that guide probing towards addresses that are more likely to result in knowledge gain. Techniques of this kind can be classified into two categories. On one hand, creative use of side-channels permits retrieval of active (and thus potentially responsive) IPv6 addresses from unrelated systems, such as DNS. This is promising where available, but often

[6] As opposed to solely being allocated to an organisation, or in principle allocatable as per the IPv6 addressing standards.

requires target or network cooperation and introduces bias towards specific use cases that produce these side-channels. On the other hand, given active addresses, machine learning and statistics can be applied to predict promising other addresses. This is interesting to expand an existing dataset, but often cannot be used if no addresses are known in the first place, and might also introduce bias.

A common and obvious side-channel is DNS, and in particular reverse DNS (*rDNS*). Reverse DNS is used to map IP addresses back to DNS names. In their 2017 paper, Fiebig et al. [FBH+17] present a way to find IP addresses in the rDNS tree based on a subtle semantic difference in error messages. If the `NXDOMAIN` error is indicated, the program knows that there is nothing at that node or below. The `NOERROR` code however indicates that this node does not exist, but there are nodes below. This can be used to traverse the rDNS tree and efficiently find any addresses that have entries, which are valuable to scan. This approach yields 5.8 million addresses at the time and is deemed reliable in 2018 [FBH+18].

After publication, however, many network operators were observed deploying mitigations against this information disclosure, necessitating a more robust method. [BHFV18] This solution is proposed by Borgolte et al. [BHFV18] in 2018 and relies on DNSSEC. DNSSEC is a security mechanism added to protect DNS records against tampering. An inherent property of this protection is that there must be cryptographic assurance that there are no further records in the tree below a node. Otherwise, an attacker could omit records, and thus break integrity. The Borgolte et al. method relies on exactly this property to traverse the DNS tree. Because this method relies on an inherent property of DNSSEC, it cannot be mitigated as easily as the original side-channel. Despite that, the project only yields 2.2 million addresses, which is less than half of the rDNS approach.

In addition, it still requires a large amount of active probing. An improvement to this issue is proposed in 2023 by Rye and Levin [RL23]. Instead of actively measuring, they contribute an IPv6-enabled server to the *NTP Pool*, which is a community-supported network of time servers. From the access logs of this server, they are able to collect 7.9 *billion* IP addresses of clients and servers.

Apart from side-channels, IP addresses to scan can also be algorithmically predicted. The research community has proposed mechanisms based on various different metrics. While manual guessing of potential addressing patterns has been suggested as early as 2008 [Cho08, GC16], automating the process using machine learning and statistics has only recently gained traction.

One of this method's early adopters are Ullrich et al. [UKKW15] with their *pattern mining* approach. They use machine learning to analyse patterns in known-to-be-active addresses from the same network. Their algorithm then uses these patterns in the seed data to generates additional addresses following the same scheme. It outperforms brute-force for the same number of probes.

Structural insights remain of interest, for example for the method proposed by Foremski et al. [FPB16] in the following year, which relies on *entropy* as core metric. Their

algorithm analyses the distribution of given examples and generates further addresses from that. 40% of the generated addresses turn out to be active in their evaluation. In contrast to earlier work, this approach is also able to occasionally predict active *networks* outside the training addresses.

A third interesting metric is *density*, which Murdock et al. [MLB+17] exploit in 2017 for a third fundamental address generation approach. This means that regions with relatively more addresses are preferred for generation. During evaluation, the approach is found to be at least as accurate, and up to eight times more fruitful. A secondary result is the explicit focus on the concept of a Probe Budget. This means that researchers decide in advance the rate or number of addresses to probe. Candidates are intentionally prioritised, and more promising targets included. The density concept is later significantly improved by Song et al. [SYW+22] in 2022.

In a meta-analysis of address generation algorithms, Steger et al. [SKZ+23] find that responsiveness levels vary greatly between the methods. An important recommendation however is to, rather than trying to find a globally-ideal algorithm, consider for each scientific project what sort of addresses should be generated and which metrics to target.

Methods that identify promising addresses are now in place, but only used in isolation. It remains to combine them so that researchers can benefit from these complementary results. This step is taken in 2016 by Gasser et al. [GSGC16] with the *IPv6 Hitlist Project*[7]. They use passive measurements and publicly-available data to generate a seed set of addresses, which feeds into an active `traceroute` measurement component. The first iteration locates 150 million addresses, covering 72% of all BGP-announced prefixes for IPv6. One important by-product of this work is the IPv6 port of `zmap`, which is a crucial step for later research. Another such adaption [Kuk16] was performed independently at the same time, but did not gain similar popularity, despite the author recommending it as more flexible.

Further improvements to the hitlist are made in 2018 [GSF+18] and 2022 [ZSS+22], each time reducing bias and cleaning up measurement artefacts. These steps include normalisation of redundant clusters using a mechanism termed Aliased Prefix Detection, removal of noise produced by the Chinese Great Firewall, incorporating additional sources, and tuning the methods.

Core focus of the improvements over time is to provide a more representative hitlist. Number of addresses alone is not the sole metric for hitlist quality, because research needs to ensure that results reasonably represent the base population. This means in particular that the composition should match the general internet in usage patterns, e.g. clients, servers, routers, corporate networks, and ideally also approximately in ratio. That is, there should be no specific deployment type that is significantly over- or underrepresented in comparison to the actual general population (which is difficult to verify due to the general population being inherently unknown). The community has in the past relied on

[7] `https://ipv6hitlist.github.io` (accessed 2024-02-02)

internet-wide measurements to identify whether some condition applies to a significant portion of deployments. Examples include IPv6 adoption rate analysis [CAZ+14] and exploitability of IoT devices [SPV21]. A non-representative hitlist composition skews and potentially invalidates such results. Steger et al. [SKZ+23] evaluate this aspect and find that all considered types of deployments are represented. However, they recommend to still consider composition on a case-by-case basis. [GSF+18, ZSS+22]

Apart from composition, another concern is that the hitlist is fed by constant active measurement. While this is not generally considered harmful if care is taken, fewer probes are clearly to be preferred if all else is equal. Rye and Levin [RL23] address this aspect in their work, proposing NTP as a valuable side-channel for IPv6 address discovery. From their passive logs, they generate significantly more (however complementary) addresses than the Gasser et al. hitlist includes, without any active probing. A large potion of these hits are end-user clients, which have previously been difficult to obtain in large numbers, and still do not tend to be persistent (see also [RBC21]). Apart from the possible implications on hitlist composition, this raises ethical concerns. The authors thus release their data only on a `/48` network granularity. Due to this (rightfully!) reduced granularity, and complementary nature to the existing hitlist, it remains to be seen if and how active measurements can be significantly reduced in the process of hitlist generation.

2.4 Beyond Individual Addresses

While hitlists are an important innovation for IPv6 measurements, their focus remains on individual addresses. This is sufficient for many studies that only need a flat list of likely-still-active addresses. On some occasions, it is however interesting to gain insight into the greater structure of the internet, and focus on more constant aggregates instead of often-fluctuating single points on the map. A more structure-focused approach might also be helpful to see where active addresses are more likely to be. This may help focus scanning on fruitful regions, but also to detect and combat bias introduced by them. Knowing the governing network structure, it is possible to avoid likely-redundant probing of related addresses, reducing load and skew towards clusters. This section discusses how IPv6 addresses can be structured into networks and what the research community has done so far to uncover structures in the real world.

IPv6 addressing is in its nature hierarchical. This begins with the way that special meanings are allocated to address blocks. The IPv6 addressing architecture specification [DH06] defines a variety of such semantics based on the leftmost bits of the IPv6 address. For example, any address starting with hexadecimal `ff` is assigned the special *multicast* semantic.

Internet-wide research is most interested in addresses that have no particular meaning (Global Unicast Addresses). They refer to only one specific device, and the same one globally across the internet. Other address blocks may have local meaning only, or refer to multiple devices. Global Unicast Addresses follow the general structure shown in Figure 2.1. [DH06]

	Address Component		Width (*bit*)
subnet prefix	global routing prefix	...	n
	subnet ID	...	m
	interface ID (IID)	...	$128 - n - m$

Figure 2.1: Structure of an IPv6 Global Unicast Address [DH06]

These components' meanings are as follows. The *global routing prefix* is the global identifier assigned to the site that the address is located at. These assignments are managed by IANA, which hierarchically delegates authority to regional and local *internet registries*. These are in turn free to further subdivide their space and allocate it out customers. The general recommendation is for such hand-outs to be large enough that they can again be further subdivided into subnetworks (*subnets*), which are the smallest logical unit of aggregation. A subnet is semantically equivalent to a *link*, as used by the link-layer protocol that carries IPv6. Inside a subnet, individual devices are then identified by their *interface ID*.[8] [DH06, Int, RHN11, CCG+15]

Concrete processes that assign these IIDs vary, but most imply a fixed IID width of 64 bits. This in turn fixes the usual *subnet prefix* length at $n + m =: 64$ bits. The subnet prefix length defines the number of bits from the left that indicate the subnet, including the global routing prefix, and is denoted with a slash. For example, such a standard subnet would be called `/64`. In a usual setup, this could be part of a larger `/56` prefix for a single site, allowing for $2^{64-56} = 2^8 = 256$ subnets there. The entire organisation might have been allocated a `/52` prefix, allowing for 16 sites. [RHN11, CCG+15, DH06]

4-bit steps have the benefit that each subnet level is denoted by a single character in hexadecimal notation. For example, if the `/52` has the base address `2001:db8:beef:a::`, then a `/56` prefix adds a single character, e.g. `2001:db8:beef:ab::` . A specific prefix is denoted by appending the prefix length to the base address, for example `2001:db8:beef:abba::/64` . Addressing based on these increments is however not mandatory, and there exist valid reasons to deviate from it. [CCG+15, DH06]

"Odd" prefix lengths might be observed in the wild for example due to BGP route aggregation. This means that a router combines prefixes routed over the same path into a single rule in the *routing table*. Without this important mechanism, every router would need to keep a rule in memory for every reachable prefix, inflating the global routing table size. Preventing this growth is also an important motivation for the strictly hierarchical way that prefixes are allocated. [RHL06, Nar10]

While the fixed standard `/64` prefix size might suggest that any research into prefix

[8] As suggested by the name, the IID (and thus the IPv6 address) does not technically identify the device, but rather an *interface* of the device. For example, a router with multiple network ports would designate each of these ports as a separate interface and route packets between the attached networks. This distinction is omitted to reduce confusion. [DH17]

structure is futile, this does not need to be the case. First, this question depends on how equivalence in terms of topology is determined. It might well be that (two or more) adjacent prefixes are used in the same way, and, depending on an equivalence measure, would thus be part of a larger aggregate. Additionally, due to the vastness of not only the IID, but also the prefix space, it is very likely that many subnets remain unassigned, which could be detected and used to derive subnet allocation policies. Given an IPv6 address, it is very likely to find $(n + m) = 64$, but the particular values of n and m can still be of interest.[9]

Due to the way that prefixes are used for routing, this proposed *prefix structure* and the well-known *routing topology* are likely to be related. It may even seem that they measure the same thing, which does not need to be the case. The latter is the graph that describes how *routers* are *connected*, while the former is a set[10] of *prefixes*, *grouped* by the way in which they are used. Nevertheless, the routing topology is a proxy through which the prefix structure may be observed.

This relation to routing topology is what makes it interesting in this context to consider routing topology measurement. Important tooling for this is provided by Gaston in his 2017 Master's thesis [Gas17], where he ports `yarrp` to IPv6. The adapted version quickly becomes a standard tool, for example being used by Beverly et al. [BDPR18] to analyse routing topology.[11] That work also begins to additionally focus on prefixes instead of individual addresses, however only in the evaluation, and not to drive probing itself.

It is however neither the first nor last to pivot in this direction. Consideration of address aggregates has already been of interest in the early 2000s, at the time targeting IPv4. In [KW00], the authors use web server logs to group similarly-behaving client addresses by considering their routing topology. Another similar passive measurement [KLPS02] is already performed two years later. This analysis considers possible groupings of all IPv4 prefix sizes using statistical methods. However, both of these approaches focus on passive analysis of access logs, which differs significantly from the proposed setup for this thesis.

Active discovery of prefix structure can be interpreted as an extension of network topology measurements. This notion is introduced by Gunes and Sarac [GS07] in 2007, continuing the trend of enhancing accuracy and efficiency of topology probing at the time. Their declared main goal is to reduce redundancy in topology graphs, which could already be considered practically equivalent to our notion of prefix structure. A crucial limitation of their process is that a (potentially partial) topology graph must be collected beforehand as input.

[9] n and m are to be understood as defined above in Figure 2.1.

[10] The prefix topology can also be modelled as a tree, which we will in fact later do, but this is a simplification. Prefixes that are used in the same way do not necessarily need to be adjacent, depending on the measure of equivalence. Crucially, measurements suggest that non-adjacent prefixes are often measured as equivalent if they are unused.

[11] That paper's main author is listed as advisor on the relevant thesis.

Nevertheless, this method forms a crucial foundation for later research. From collected topology data, the authors construct candidate subnets of some size. These are then *recursively* split into all possible smaller units that still accommodate responsive addresses. Four proposed conditions are then applied to remove unlikely combinations. For example, the *accuracy* condition checks adjacency of candidate subnet members and rejects subnets where there are other hops in between members. Such topology implies routing being necessary to communicate, contradicting the notion of a shared link-layer medium and thus subnet. [GS07]

While this same general method continues to be expanded upon for IPv4 [KGO12, GD20, YYZ23], it has also been adapted for IPv6. This adaption, termed Recursive Subnet Inference (RSI) 6, is performed in LaFever's 2015 Master's thesis [LaF15]. It relies on CAIDA's Ark [Cen20] on-demand topology probing infrastructure. The principle is to start at a BGP-announced prefix and keep splitting it into smaller subnets recursively. Addresses at the quarter marks of the prefix are probed and their traceroute results compared to inform split-or-keep decisions. If they are different, the algorithm recurses to the next prefix length.

In addition to this family of methods, there is also a subtly different approach called Hobbit. This idea is introduced in 2016 by Lee and Spring [LS16], aiming to determine whether an IPv4 prefix is *homogenous*, with the ultimate goal of evaluating whether `/24` is an appropriate aggregation size for IPv4 network measurements. Their motivation is that if homogenous blocks can be identified, each of these blocks need only be probed few times to obtain all the information it has to offer. Especially for larger address blocks, significant fractions of probing budget might be wasted on redundant samples that do not lead to new information, which Hobbit aims to solve.

Their solution is different mainly because they choose to only consider the last non-destination hop on the path (Last-Hop Router, LHR). This is an interesting change due to the fact that previous methods need (at least tentative) topology information as input, which requires significantly more probes to collect. An additional benefit, and the initial motivation for this approach, is that this information is less skewed by load balancing. In particular, the Hobbit method is designed to be resistant against per-destination load balancing. Such setups divide load persistently based on the destination address, instead of making a new choice for each packet. As a consequence, even repeated traceroutes to a single address will show the same path, evading detection by earlier methods, as the authors show in an initial discovery measurement based on Paris traceroute. [LS16]

However, additional care must be taken apart from not considering the whole path. Hobbit further introduces a trick that relies on the inherent hierarchy of routing tables. This is necessary because there are cases where multiple addresses with distinct *observed* Last-Hop Routers are indeed in the same subnet. In particular, this might be caused by per-destination load balancing, which shows up in the measurement as non-contiguous sections of address space being observed to be handled by each Last-Hop Router. [LS16]

Such a result does not correspond to what will be reasonably represented in a standard

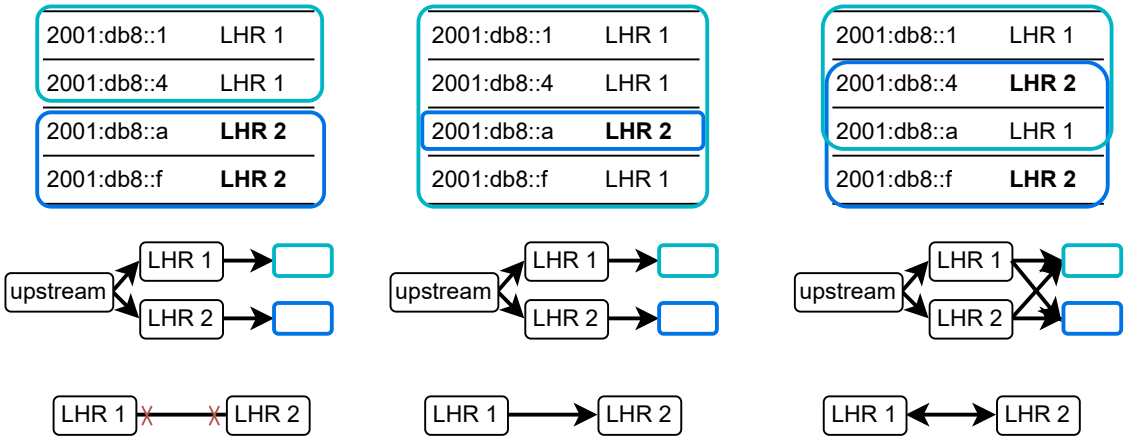


Figure 2.2: Visualisation of the Hobbit hierarchy trick. From top to bottom: Observed address ranges with associated Last-Hop Router, actual routing topology, relationship between the routers. Adapted from [LS16, Figures 1-2].

routing table. The core principle of a routing table is a that it behaves as a lookup table keyed by IP prefixes of the final destination addresses of packets. Each such prefix in the table is associated with one, or multiple in the case of load balancing, *next-hop* router(s) to forward incoming packets to. This by definition relies on and invites a hierarchical address space structure, i.e. assigning contiguous sections to each router that directly correspond to IP prefixes. If this is not done, many more rules must be installed in the table to represent the routing topology, which is undesirable both for storage size and management overhead. It is thus customary to assign IP addresses in such a way that entire prefixes are routed via the same next-hop router, making their relationship strictly hierarchical. [LS16, FLVY93, Bak95]

Assuming such a hierarchical routing architecture, address space in a prefix can only be allocated to multiple Last-Hop Routers in certain ways. In particular, the address spaces observed to be handled by two Last-Hop Routers can only be either disjoint or inclusive, i.e. one completely contains the other. In both of these cases, the prefix is *heterogenous*, meaning that it can be meaningfully subdivided, in this case into units matching the Last-Hop Router assignments. [LS16]

As a consequence, if intersecting address spaces occur in the Last-Hop Router observations, this suggests per-destination load balancing. In that case, Hobbit labels the network as homogenous. The observed address space regions, actual routing topology, and inclusion relationships between Last-Hop Routers in each of the three cases are shown in Figure 2.2. [LS16]

Inspired by Hobbit, *edgy* [RB20b, RB20a] is a study that aims to shed light on the IPv6

periphery, i.e. Last-Hop Routers. One of their motivations is that end-user devices are often not directly reachable, but their Last-Hop Router is more likely to be observable using a traceroute. Similar[12] to LaFever's method [LaF15], they split prefixes into candidate subnets and drill down into areas that seem interesting.

Their method's detailed behaviour is as follows. Preparation begins with a seed dataset, e.g. BGP prefix data or a pre-existing hitlist. Destinations in the seed are tracerouted and grouped by the observed Last-Hop Router. `/48` prefixes with non-unique Last-Hop Routerss are dropped, since they seem less likely to be subnetted, or the responses come from earlier in the topology. Remaining `/48`s are subdivided to `/56`s and random IIDs probed in each. `/48`s with many responsive sub-prefixes are promoted to the next round, while less responsive ones are discarded as unlikely to be interesting. This is repeated with sizes of `/60`, `/62`, and finally `/64`. Notably, *edgy* always probes all sub-prefixes of the whole `/48` and does not discard regions that are unresponsive, i.e. the size of the prefix-under-test is constant just like the `/24` in Hobbit. Only granularity changes. [RB20b]

An interesting extension to the *edgy* method is proposed by Rye et al. [RBC21], where the goal is to defeat address privacy introduced by periodic rotation of end-device IIDs (*IPv6 Privacy Extensions*) combined with prefix rotation performed by providers. The core idea here is that, while end-user devices often utilise randomised and transient IPv6 addresses to preserve privacy [Nar10], the Last-Hop Router in their home often does not, frequently even encoding its MAC address into the IID (Modified EUI-64 [CCG+15]) Their method targets ISP's end-host allocations efficiently by reverse-engineering addressing policies. The allocation size to the end user can be observed by probing adjacent `/64`s to a user's network and seeing if the same Last-Hop Router (= the Customer Premises Equipment at the customer's end of the line) responds. Which portion of the ISP's address space allocation is reserved for end-user prefixes is estimated by monitoring how far constant EUI-64 MAC addresses of the Last-Hop Router move across the provider's whole range.

Apart from the discovered privacy issue, a very interesting aspect to this study is that it re-uses previously-discovered information (prefix sizes, allocation pool size) to focus further probing. While some studies operate in independent rounds, others combine new results with existing knowledge. When measurements are cheap, like in IPv4 studies, there is no reason to invest additional effort and complexity into re-using past results. However, if exhaustive measurements are costly, it can prove beneficial to remember at least basic properties of the observed system. This is especially the case if a full probing round takes longer than the desired update interval, which is clearly the case for IPv6.

It thus may seem obvious to always "recycle" previous results in IPv6 probing efforts. While additional complexity added by such a system is already a deterrent, there are further inherent limiting factors to this approach. For instance, how can one determine which results to incorporate into persistent storage? How long should they be retained? Does it make sense to remember all knowledge for the same amount of time? What

[12] Note: The second author of *edgy* is credited as advisor on LaFever's thesis.

influence do inconsistencies and contradictions have on collected data's validity, and how can they be detected? The following paragraphs consider existing approaches to some of these questions in the field of network measurement.

Fluctuations in IP address behaviour are considered in a 2015 study [PB15] performed by Plonka and Berger. One of their core motivations is to determine which addresses can be observed for a long time, and which disappear quickly. This is observed primarily using requests sent by the target devices, relying on CDN logs for address behaviour analysis. A result is that single (client) addresses tend not to remain observable for long periods of time, with only 0.1% remaining stable over a year. `/64` prefixes on the other hand often stay active, even if not necessarily with the same clients (prefix rotation).

A more explicit approach to keep measurements up to date is taken by Giotsas et al. in their *Reduce, Reuse, Recycle* approach [GKF+20]. Their idea is to keep a once-measured traceroute corpus updated with topology changes, while only re-probing paths that are likely to have changed. Signals that suggest a traceroute might need updating are collected both directly through BGP changes, using the BGPstream platform, and indirectly by monitoring public traceroute results provided by RIPE Atlas.

In addition to recycling existing data for one's own research, an interesting optimisation is also to provide this information in accessible format for *other* researchers to reuse. A well-known example of this idea is the already-discussed IPv6 Hitlist [GSGC16]. Even further re-use is presented in a concept proposed by [DAM+15, CMC+16], who eliminate the need for downstream researchers to perform their own measurements entirely. These systems show concepts for shared "search engines" over topology data, where researchers can collect necessary data simply by formulating the right query over an existing database.

Finally, a recycling-adjacent notion that has received a considerable portion of the community's interest is using existing measurements to reduce redundant probes. Ever since large-scale probing has become accessible, researchers have noticed that many targets behave identically and can be aggregated to reduce the number of probes required. Effectively all studies aiming at a meaningful coverage of IPv6 must apply this principle to some extent, since even a single `/64` is impractical to exhaustively measure. This can manifest either as direct focus on redundancy, or an effort to more accurately predict where responsive addresses can be found. Many of the aforementioned studies, beside various others, cite this redundancy reduction principle as a key motivation, with some notable examples being [LS16, LaF15, BF13].

2.5 Ethical Standards for Internet Measurements

Before a measurement can be proposed and executed, it is necessary to evaluate ethical implications of internet measurements in general. With the large corpus of work in this area, spreading over decades of measurement, it might seem like the consequences would have been considered in detail. While many works, especially the more recent ones, have a dedicated section weighing their contributions against potential harm, a large portion

omits discussion. Researchers that do mention ethical considerations usually do so in a short section, but there are few papers where this is a core focus.

One reason why this might be the case is that a large portion of classical ethics research focuses on studies where humans are directly involved. In particular, this is a common theme in the medical field, where early experiments with human subjects are often considered highly problematic in today's world. The field produces the *Belmont Report* [Dep79] in 1979, which lays out standard ethical guidelines for medical research involving human subjects. It takes until 2012 for an adaption to be produced specifically for challenges of IT research involving human subjects, called the *Menlo Report* [BDKM12].[13] [PA16]

While this report already considers challenges of large-scale research and indirect stakeholders specific to the digital world, it still is not directly applicable to internet measurement research. This challenge is discussed in a 2016 ACM article [PA16]. Its main result is the suggestion to include a dedicated ethics section in every research paper utilising internet measurements. This measure should raise awareness that ethical issues must also be considered for research that does not *directly* target human subjects, and allow collection and analysis of different ethical standards in use by various researchers. Despite no clear decision framework being proposed, it is clearly visible in the literature that after the publication of this article, the dedicated ethics section has received more adoption, e.g. [SKZ+23, BAF+21, Bev16], especially when compared to the previous years when a majority of the research considered for this work fails to mention the word "ethics" at all.

An intuition for how the explicit inclusion of ethical considerations into network measurement research has developed over time can be gained in Figure 2.3. This graph indicates the ratio of research papers that do not mention ethics, point at it briefly, or include a dedicated section, by year of publication. The data is obtained by searching each entry considered in the literature research for this work, and searching for the word prefix "ethic".

It should be noted that earlier works occasionally refer to "good internet citizenship", however this term is in general a small subset of a proper ethical consideration, and thus does not in itself constitute an ethics mention or section for our purposes. An additional caveat is that there clearly are more papers, especially targeting IPv4, which are not considered in this non-exhaustive analysis. Its purpose is solely to provide a general intuition that inclusion of an ethics section has significantly increased since around 2012, and not to be an exhaustive and statistically sound examination.

Clearly, when targeting other people's infrastructure at scale, researchers have to be careful not to cause harm by accident. A number of risks is present, with most of them being well-known in the literature. The most common ones are as follows. Citations are to be understood as examples, especially for the more general risks.

[13] Both of these reports are sponsored by US government agencies. They may or may not reflect the situation in the rest of the world. Still, they provide a solid foundation for ethical analysis.

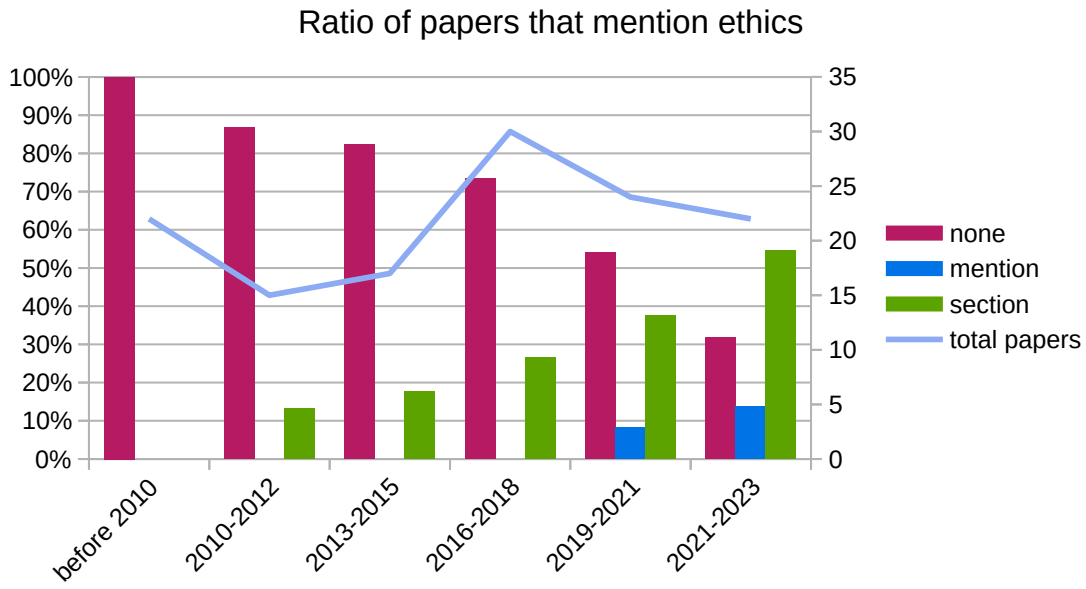


Figure 2.3: Research papers that include an ethics section, mention ethics briefly, and do not mention it, by time of publication. The overall number of papers that were considered in each timeframe is indicated by the line. Notable points in time: 2012 — publication of *Menlo Report*, 2016 — publication of ACM article.

- **Infrastructure overload.** The influx of probes targeted at a single host or router causes a service impediment. With IPv6, this can also refer to *ICMP Rate Limiting*. This limit is global, meaning that legitimate users and other research receive fewer ICMPv6 responses because the measurement is consuming the entire rate limit over a long time. Possible mitigations include spreading out load over many targets at the same time, and reducing redundant probes. Resource exhaustion has been a topic since the first internet measurements, and avoiding this is considered a core part of *good internet citizenship*. [PA16, KLWR16, LS16, BF13, DWH13]

- **Accidental breakage.** Nonstandard protocol implementations break when faced with uncommon packets. This is especially of concern for more specific application-layer measurements. For example, improper usage of stateful protocols may cause queues to overflow, e.g. when opening but never closing a large amount of TCP sessions. In some setups, these issues can also be triggered by measurements that are usually considered to be non-intrusive, such as low-volume ICMP traceroute probes bringing down a European ISP in [Mai21]. This risk can be reduced by relying on standard ICMP probes intended for diagnostics instead of application-layer protocols, and by staying as close to usual standard implementations as possible. [Mai21, DWH13]

- **Information exposure.** Analysis results, but in particular raw datasets, are used to infer private information about individuals or organisations. This especially applies to modern data mining and side-channel methods, which may expose private details based only on what is otherwise considered harmless metadata. Risks must be weighed against open sharing of data in the scientific community, which might aid future research. A mitigation can be observed at CAIDA, where a portion of datasets is only provided in full after an approval process. [Cen20, PA16, BDKM12]
- **Nefarious method usage.** Proposed methods are used for purposes that induce harm. For example, exposed vulnerabilities could be abused to compromise target machines for a botnet. This risk can be reduced by a careful ethical evaluation, but is difficult to avoid: Without a detailed method description, other researchers are unable to benefit from it. [PA16, BDKM12]
- **Investigation effort.** Suspicious probes cause unnecessary investigation efforts for network operators, in particular when they trigger automatic Intrusion Detection Systems. A mitigation is to avoid non-standard or attack-like behaviours, and to make the purpose of research as explicit as possible e.g. with reverse DNS, a website hosted at the probe address, or indications in the probe themselves. [DWH13]

A large portion of the current best practices for harm risk reduction is mentioned in the original `zmap` paper [DWH13]. The most important ones are scan rate reduction, spreading out load to different targets, clear indication of purpose, simple opt-out process, and minimal measurement duration and volume.

In general, there is no one-size-fits-all conclusion, and every probing effort must be weighed individually, as discussed in [PA16, BDKM12]. A template for questions to consider in an ethics section may be found in [PA16].

Ethical evaluation of this thesis in particular continues in the final section of chapter 3.

CHAPTER 3

Method

This chapter discusses concepts and implementation of the proposed measurement setup. Ideas are based on theoretical foundations laid out in chapter 2, while the performed experiments and their evaluation are scope of chapter 4.

The core aim of the proposed system is to discover the internet's *prefix structure*, as defined in section 2.4. The algorithm is heavily informed by a combination of previous work's methods, with the most influential inspirations being Hobbit [LS16] and edgy [RB20a]. A very similar method is proposed by LaFever's master's thesis [LaF15].

On a high level, the most important components of the process are:

- Initial Probes — Probe interesting (BGP) prefixes initially using ICMPv6 Echo Requests sent by *zmap* [DWH13].
- Follow-Ups — Refine with routing topology data from traceroutes where needed using *yarrp* [Bev16] [Gas17].
- Aggregation — Collect the results per prefix and persist in a database. Split the prefix into two if the data suggests independent subnets.
- Feedback Mechanism — Use results of previous rounds and the current state of the discovered prefix structure to drive subsequent rounds.
- Probe Budget — Be aware of ICMP rate limiting and keep a budget for how many probes can be issued per round.
- Priority Classes — Group prefixes by how likely they are to result in further knowledge and look at "more interesting" candidates first.

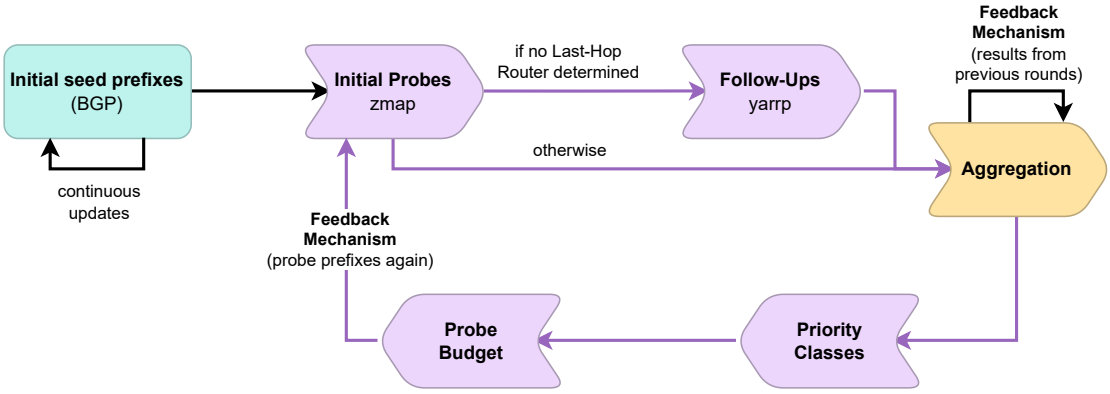


Figure 3.1: High-level overview of the interaction between method components. These making up the Feedback Mechanism's cycle are marked by purple colour.

How these components interact is outlined in Figure 3.1. Starting from the left, the journey starts with seed data. Prefixes announced using the BGP protocol are assumed to be the largest possible networks. These BGP Roots form root nodes in a forest of binary trees, and are later split into smaller units if measurement results suggest so.

A prefix then enters the measurement feedback loop (purple arrows). This begins with initial echo request probing of random addresses in the prefix. If no Last-Hop Router is evident from responses, the prefix receives follow-up traceroute probes to obtain one. Once probing is complete, the aggregation step merges new data with past results for the same prefix, and persists this information in a database.

Additionally, this freshly-merged data drives the prefix split decision. This is the second important responsibility of the aggregation component. It involves deciding whether the prefix is a homogeneous unit, and should thus be retained at that granularity, or if its two equally-sized subnets should be considered independently. The main driver here is the relation of Last-Hop Router sets observed for each candidate subnet. For example, if both subnets have the same set, they are likely equivalent, and no split is suggested. Their relation further determines the priority class assigned to the prefix. Priority classes approximate the likelihood of information gain from further probes into the same prefix. Before actually performing a split, the system computes a confidence metric to determine if sufficient evidence is available. If it is insufficient, the split is deferred until more information becomes available in a future probing round.

Probe budgeting is performed periodically, based on the priority classes of all available prefixes. This process schedules all prefix analyses of measurement round by allocating a specific portion of the budget to each priority class. This allocation is determined by the classes' relative importance, and how many prefixes are available for probing in each class. For example, fresh prefixes are preferred for scheduling, but budget is only consumed if such prefixes are actually available, leaving it to less promising classes otherwise.

This completes the feedback loop, allowing the process to repeat and continuously refine

the observed prefix structure with new data. In addition, seed prefix data can be periodically updated, and obsolete parts of the structure are taken out of service.[1]

The remainder of this chapter is structured as follows to explain how these components interact to build the measurement setup. Data sources, system architecture, and probing itself are subject of section 3.1. Exploration continues towards the prefix split mechanic, its implications, and surrounding infrastructure in section 3.2. Finally, section 3.3 applies the ethical principles from chapter 2 to the proposed setup.

3.1 Data Sources & Measurement Process

Two key aspects of every active internet measurement are how initial seed data is obtained and how probes are actually sent out. Both of them are scope of this section, starting with the former.

Due to the vastness of IPv6's address space, it is infeasible to scan all of it. However, only a small fraction is currently allocated, and even less is assigned to organisations. Which address prefixes are currently reachable on the public internet can be determined from state exports of the BGP routing protocol. Such exports are published by various institutions participating in BGP, drastically reducing the space measurements need to consider.

For public routing purposes, the internet is divided into so-called *Autonomous Systems* (AS), which are assigned globally-unique numbers (ASN). Protocols that manage inter-organisational routing (i.e. across AS borders) rely on these numbers to identify paths. ASNs are assigned by the Internet Assigned Numbers Authority (IANA), which delegates large blocks to regional organisations. IANA publishes block assignments at `https://www.iana.org/assignments/as-numbers/as-numbers.xhtml` at the time of writing (2024-03-11). Regional *internet registries* each make their own registrations available to the public, for example the North American ones can be found at `ftp://ftp.arin.net/pub/stats/arin/delegated-arin-extended-latest`.[2][Int]

The proposed system relies on a service that aggregates these publications into a structured format. It scrapes official datasets and commits an update to GitHub daily at `https://github.com/ipverse/asn-ip`. Seed data is informed by the BGP Roots that each AS announces. While the service seems to be maintained by an individual, provided data seems to be sufficiently reliable for purposes of this thesis. Manual monitoring would require significant effort. Should updates cease in the future, it is still possible to implement a custom scraper at that time.

In addition to organisation names and originated IP address ranges, IANA also collects country association for AS assignments. This means that organisations of individual

[1] This feature is disabled during measurements performed for this thesis to reduce the number of variables influencing system behaviour.

[2] A collection of these URLs for all Regional Internet Registries is located at `https://github.com/ipverse/feedback/discussions/3` (accessed 2024-03-11).

countries can be targeted. Clearly, this will neither include all devices in that country, nor reliably exclude devices from other countries. In a globally-connected world, many organisations operate internationally, and will not have separate allocations for every country they operate in. Still, such a filter can provide a baseline reduction of measurement space. The proposed implementation relies for simplicity on aggregation performed by `https://whois.ipip.net/iso/AT` instead of manual parsing of source data, similar to the AS-prefix mappings.

While this filter reduces measurement space, it does not exclude particularly sensitive usages. There is no particular reason to expect that `traceroute` and Echo Request probes would cause significant harm such usages. Nevertheless, this is a new method that has not been applied to real-world networks before, and experiments should avoid disturbing sensitive usages at all cost. Similar considerations motivate a reduction of measurement scope to AS registered in Austria. Due to the small number of remaining AS, this exclusion can be performed manually, relying on publicly available information about the organisations behind each AS. Examples of excluded usages are military, international organisations (UN), and power grid. An auxiliary result is a rough classification of the remaining AS into general industries, which is summarised in Table 3.1.

Along with the main measurements relying on this filter, additional measurement campaigns are performed for this work with varying goals. During initial development of the solution, a series of exploratory measurements allows local testing against a virtualised setup in the GNS3 network simulation software[3], denoted `E-x`. From this point, each

[3]Available at `https://www.gns3.com/` as of 2024-03-14

Classification	# AS	Size
ISP, Small or Regional	75	$\approx 1.7 \times$ `/24`
IT Services, General	64	$\approx 1.7 \times$ `/25`
Other and Unknown	31	$\approx 1.4 \times$ `/27`
Business, General	24	$\approx 1.5 \times$ `/27`
IT Services, Hosting	20	$\approx 1.5 \times$ `/26`
Excluded from Measurement	17	$\approx 1.3 \times$ `/26`
Government	15	$\approx 1.4 \times$ `/28`
Business, Manufacturing	12	$\approx 1.1 \times$ `/27`
Business, Financial	9	$\approx 1.6 \times$ `/29`
Science and Education	9	$\approx 1.1 \times$ `/29`
Business, Transportation	7	$\approx 1.3 \times$ `/29`
ISP, Large	6	$\approx 1.2 \times$ `/26`

Table 3.1: Rough classification of measured Austrian AS by fields of business. Size measures stem from addition of the number of `/64` networks contained in AS of the group. The number of AS in each class and its combined size are not always related. For example, the six large ISPs have more address space than the 24 general businesses.

#	Targets	Nature	Algorithm	Start	Duration
E-x	simulation	development	100	*various*	—
U-0	university	exploratory	101	2023-12-03	8 days
U-1	university	exploratory	102	2023-12-26	10 minutes
U-2	university	exploratory	103–111	2023-12-26	13 days
U-3	university	validation	112	2024-01-09	2 days
AT-10	Austrian AS	evaluation	112, 120	2024-01-18	1.5 months
AT-11	Austrian AS	evaluation	120	2024-03-11	10 days

Table 3.2: Overview of measurement campaigns performed for this work. U-* re-use their *measurement trees* (but not *prefix trees*; see section 3.2), while AT-* both start from scratch.

campaign gradually expands measurement scope. Algorithm fine-tuning against a real-world target happens with campaigns U-0 through U-3, targeting an Austrian university network in cooperation with insiders. The full-scope main campaigns AT-10 and AT-11 target the selection of Austrian AS outlined above. An overview of all campaigns is presented in Table 3.2.

Despite varying scopes and minor adjustments to the split algorithm, the overall setup remains the same for all measurements. Probes are sent out from a single measurement server[4] located at a well-connected datacentre in Vienna, running Ubuntu Linux with 256 GB RAM and a 10 Gbit/s uplink. All custom services are written in the Rust programming language for optimal performance and stability. `yarrp` and `zmap` run bare-metal, while all other services and infrastructure components are containerised with `podman` to simplify management.

An overview of the system's architecture is presented in Figure 3.2. The measurement is driven by four custom Rust services, which each implement one or more components of the high-level concept. Seed Guard updates the database with changes to the seed data. Aggregator periodically selects which prefixes to probe in each round. Additionally, it aggregates and stores probe results, which it further uses to feed split and confidence algorithms. Probe results are provided by Yarrp Buddy and Zmap Buddy, which drive probing using their respective lower-level tools. Finally, the client application Crab Tools provides researchers with an interactive view of the current state and results of the campaign.

[4]Refer to https://aim.sba-research.org/scanserver.html (accessed 2024-03-14) for details.

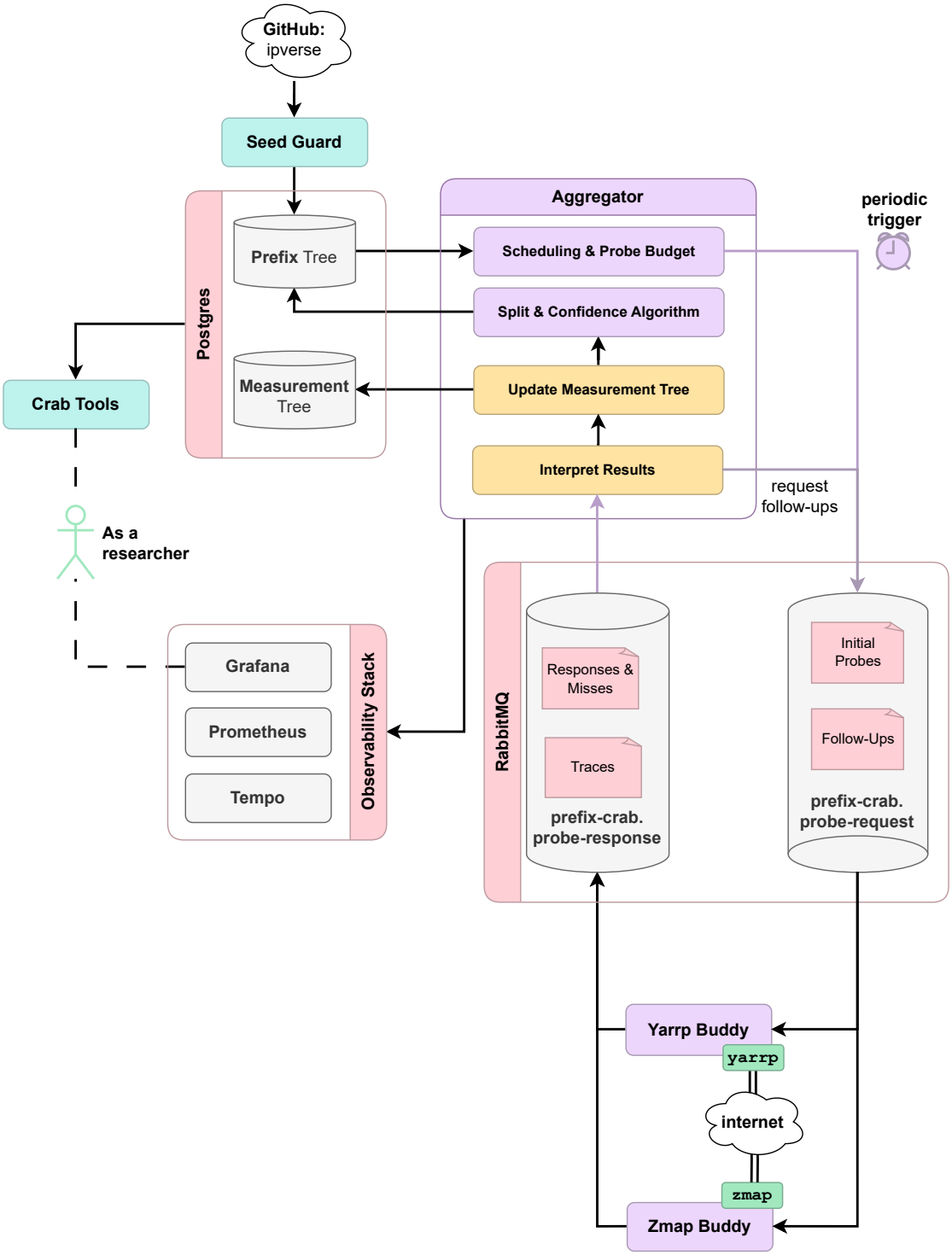


Figure 3.2: Technical overview of the measurement process. Infrastructure components are marked in pink and grey. The remaining colours correspond to Figure 3.1.

Details for each service are discussed in different parts of this thesis, along with their respective high-level concepts. The current section focuses on Seed Guard and the Buddies, while responsibilities of Aggregator are discussed in section 3.2. The remaining Crab Tools becomes relevant in chapter 4.

Apart from the business services, various off-the-shelf infrastructure components are critical to the measurement process. RabbitMQ provides message queues that services use to pass work and results. Its management interface allows researchers to observe load and trends in the system. Additional observability into business metrics for monitoring and evaluation is provided by Grafana and Prometheus, while Tempo aids performance analysis with trace[5] data. Postgres stores the state of both prefix and measurement trees, along with other data needed for operation.

In general, the system architecture is scalable primarily due to properties of the message queues. While services run on a single host for our purposes, they could theoretically be distributed to multiple machines for improved throughput. The Aggregator needs additional configuration in such a setup, because the periodic trigger should only happen once. This is achieved by pinning the scheduler to a specific instance. For the Buddies, care must be taken not to mix responses for different instances on the same link. This is relevant both between the two tools and replicas of the same tool. Dedicated source addresses for each of the tools provide separation in this thesis' measurements. Both tools natively ignore packets destined to unknown addresses. Additional network latency is induced when moving infrastructure components to different hosts, which must be weighed against throughput improvements. For this work, all components apart from the observability stack remain on the same physical host.

Scalability becomes increasingly relevant when historical measurement results accumulate during long-running campaigns, but also with higher target probing rates. When scaling measurement throughput, care must be taken to scale all key components to consistent capacities, either automatically or manually. This is necessary because, otherwise, some parts of the system will be idle and consume unnecessary resources, while others become overloaded, forming a bottleneck that threatens overall stability.

Message queues can in theory support *back-pressure*, meaning that producers automatically throttle their production rate when consumers are at capacity. RabbitMQ does implement such a mechanism to protect its own health[6], but this seems to be focused on ensuring that messages are properly stored in the queue, and not on limiting the backlog of consumers. Regardless, dynamic flow control is not possible because the proposed system relies on a fixed *prefix scheduling budget* for each measurement round, which is entirely unaware of how many messages are queued for processing. This choice is made both for simplicity and due to the fact that network throughput will always be limited to the fixed link speed of the node(s) hosting the buddies. [CC09]

[5] Tempo provides distributed system tracing, not to be confused with `traceroute` data.

[6] Refer to `https://www.rabbitmq.com/docs/flow-control` and `https://www.rabbitmq.com/blog/2020/05/04/quorum-queues-and-flow-control-the-concepts` (both accessed 2024-03-14) for more information.

This means that parameters for this *static* flow control mechanism must be tuned for all components to have compatible production and consumption rates. As mentioned, an inherent limiting factor is the network link capacity of the single measurement node. Because it seems unlikely a priori that other factors will significantly limit throughput, the method relies on exactly this metric as baseline for capacity calculation.

However, `10 Gbit/s` is not a meaningful baseline for throughput calculation. The sending rate must be further reduced to prevent ICMP Rate Limiting. If a probe's target fails to respond due to this mechanism, no information is gained and the allocated measurement budget goes to waste. In addition, the rate limit is global to all clients on the internet and "legitimate" consumers are also deprived of responses that may be necessary for their operation. It follows that the limit should be safely avoided by a reasonable safety margin.

With this information, key parameters can now be computed that determine flow rates. The first set controls ICMP Echo Requests sent by `zmap`. All parameters are relative to a single scheduling round. The following enumeration calculates the **rate limit for a single Autonomous System** (AS) in a scheduling round, based on some constants and parameter choices.

1. **Scheduling delay between rounds** $T := 120$ s. *Destination Unreachable* responses are delayed by neighbour discovery timeouts, which are occasionally high in practice. Usual[7] values seem to range from $2 - 23$ s, of which the upper bound is chosen for safety. A relatively high scheduling delay reduces the relative impact of this waiting period $T_{w,z}$, but must be balanced with reaction latency. If the delay is too high, progress on individual prefixes will be slow, limiting the speed at which structures are discovered.

2. **zmap runtime per round** $T_z = T - T_{w,z} = 97$ s.

3. **Target zmap AS data rate** $r_z := 75$ packets per second (pps). This choice is based on an assumed ICMP rate limit of 100 pps, motivated by results from [AOR17, Hol20, Mai21] and reduced by a safety margin of 25%. This is in relation to a single router. Internal routing structure of the AS is unknown before measurement, so the conservative choice is to apply this limit for the entire AS.

4. **Number of packets for a single prefix analysis** $l_z := 32$ packets/prefix. This choice is effectively arbitrary. Each candidate subnet receives 16 probes, which aims to balance reasonable coverage with low budget usage.

5. **Permissible AS prefix analysis frequency** $f_z := r_z / l_z \approx 2.34$ prefixes/s.

6. **Permissible prefixes per schedule and AS** $n_z = T_z * f_z \approx 227.34$ prefixes.

[7] Preliminary results provided by Johanna Ullrich.

#	zmap				yarrp		
	send rate		prefixes per round	per-AS prefix limit	send rate		prefixes per round[9]
	kbit/s	pps			kbit/s	pps	
U-⋆	**63**	112	320	—	27.3	38	≈ 18
AT-10a	**100**	190	541	100	288	400	≈ 184
AT-10b	**200**	360	1060	75	576	800	≈ 366
AT-11	**300**	550	1620	75	720	1000	≈ 459

Table 3.3: Global flow parameter choices for each relevant campaign. zmap kbit/s target rates (bold) are the calculation input. How remaining values are derived is discussed in enumerations starting on page 32.[10] For AT-10, the measurement rate is increased after 22 hours without issues, resulting in the a/b split.

A result of this calculation is that the per-AS rate limit is the same for every measurement campaign, as the formulae do not depend on overall send rate. This principle is not adhered for measurements U-0 through U-3, where an intuitive allocation was relied upon, based on an assumption about the network topology.

For these measurements, it seemed like the load was handled by two load-balanced routers[8], suggesting a likely total ICMP rate limit of 200 pps. Reduced by a safety margin, a send rate of 150 pps was divided between the two tools with a ratio around 3:1, zmap receiving the larger share. This ratio choice was informed by the assumption that a minority of prefixes would require follow-up probing, which the measurement results show to be incorrect. The productive AT-⋆ campaigns' allocate a larger share of the bandwidth to yarrp becaused of this observation.

Unlike AS-level rate limits, global flow parameters depend on the overall target sending rate. As confidence in the method increases, later measurements choose higher send rates. Global flow parameter choices for both tools are presented in Table 3.3. While the U-⋆ campaigns rely on the arbitrary allocation mechanism described above, AT-⋆ rates are informed by a more sophisticated calculation.

For zmap, the primary input is the target send rate in kbit/s. All other parameters are derived from this, rounding appropriately to add safety margins. These margins are especially important to give consumers the opportunity to recover from processing lag.

[8] Measurement results reveal that these routers only handle a specific portion of the network.

[9] A more meaningful metric for follow-up requests is *prefixes per second*, since these are not driven by a static scheduling budget. We use the same unit for both tools to enable comparison.

[10] A program that calculates these parameters is provided in the Crab Tools command rate-calculate.

Global parameters for `zmap` are calculated as follows.

1. **Target send rate conversion** $R_z[\text{pps}] = R_z[\text{kbit/s}]/s_z$, where the packet size $s_z := 560$ bits for `zmap`, comprising a 14-byte Ethernet header, a 40-byte IPv6 header, an eight-byte ICMPv6 header, and eight bytes of payload.

2. **Prefix frequency** $F_z[\text{prefixes/s}] := R_z[\text{pps}]/l_z$. Converted to rate of prefix analyses as intermediate step. This represents an average over time since `zmap` requests are batched in reality – results are thus sent out in bursts and not at a consistent rate.

3. **Prefixes per scheduling round** $F_z[\text{prefixes/round}] = F_z[\text{prefixes/s}] * T_z$. This value controls how many prefixes are selected in a single scheduling run overall.

Lower per-round AS-level rate limits for F_z (see Table 3.3) are applied for campaigns where multiple AS are targeted. Measurements start with a lower value than the theoretical limits calculated on page 30 for two reasons. First, `yarrp` send rate is only controlled indirectly by the `zmap` send rate, and bursts are thus plausible, especially if there is an interruption in result processing. Second, early campaigns remain conservative to avoid potential calculation or algorithm design mistakes from causing strain on individual targets, with the intention of increasing the value as confidence in the method increases. In practice, structural properties of a few AS cause them to consume maximum budget consistently over the entire measurement duration. This actually motivates a limit reduction starting with `AT-10b`.

Unlike `zmap`, `yarrp` probes are driven by follow-up requests scheduled by the results of echo response analysis. This means that an additional per-round prefix budget is not necessary. The same applies to per-AS rate limits, which are loosely enforced because the number of echo requests limits how many echo responses arrive, forming an upper bound for how many traceroutes will be requested.

Global parameters for `yarrp` are calculated as follows.

1. **Maximum packets for a single follow-up analysis** $l_y := 32$ traces/prefix $* 15$ hops. This is intended to match the `zmap` probe count. However, a traceroute needs one packet per target *Hop Limit* to find all hops on the path. A usual hop number of 15 is selected *(Hop Limit values 2-16)*. `yarrp` is additionally configured in *fill mode* (see [Bev16]), meaning that it may expand up to 32 hops if no Time Exceeded packet is encountered by then. This should be a rare case and is thus disregarded in the calculation.

2. **Expected packets per trace request** $\mu_y := p * l_y = 240$ packets/prefix. Due to the significantly higher packet count, each probe is requested only with probability $p := 0.5$.

3. **Target send rate** $R_y[\text{pps}] := F_z[\text{prefixes/s}] * \mu_y$. The theoretical `yarrp` send rate is determined by how many probes are scheduled by follow-up requests. If this is chosen too low, then follow-ups will be unable to keep up. High values must also be avoided, because, as discussed for the AS-level rate limit, echo responses are published in batches, and such bursts would reflect in probes if not controlled by the send rate. In practice, these risks motivate a significantly lower overall threshold than theoretically possible. Traceroute send rate has not proven to be a bottleneck during measurement campaigns, and it was not necessary to increase this rate. Values given in Table 3.3 already include this adjustment, R_y^*. Theoretical limits can be calculated using the provided formula.

4. **Target send rate conversion** $R_y^*[\text{kbit/s}] = R_y^*[\text{pps}] * s_y$, where the packet size $s_y := 720$ bits for `yarrp`.

5. **yarrp runtime per round** $T_y := T - T_{w,y} = 110s$. Traceroute results do not expect or rely on Destination Unreachable responses, so no NDP timeout is relevant, and a lower $T_{w,y} := 10s$ can be chosen.

6. **Expected prefixes per scheduling round** $F_y^*[\text{prefixes/round}] = R_y^*[\text{pps}]/\mu_y * T_y$. Due to the R_y^* adjustment, this value is not the same as F_z and must be calculated back from the actual target send rate. Further, it is merely an expected value due to the probabilistic sending controlled by p and not every echo response necessitating a follow-up trace.

As mentioned above, both types of probing are performed in batches. This is because the underlying tools are designed for large-scale long-running internet measurements with fixed target lists, and not for dynamic single-shot probing. While it may be possible to re-implement the tools with support for this type of workload, it remains out of scope for this work for two primary reasons. First, the implementation work required to effectively rewrite both tools is significant, and would require either porting their features to Rust, or adding two more programming languages (C and C++) to the tech stack. Second, it makes sense to rely on proven tools. Especially in the cases of `zmap` and `yarrp`, these programs have been used for a large variety of research studies for nearly a decade, so it is reasonable to assume that their implementation is robust and results can be compared to others research in the field.

Both tools are controlled by their respective Buddy applications. The buddies abstract away message queue handling and batching, which the underlying tools are not designed to handle. In addition, Zmap Buddy performs subnet splitting and random address selection. Yarrp Buddy follow-ups simply reference addresses selected in this way. Once a batch of measurement requests has built up, triggered either by probe count or a timeout, the buddy writes all target addresses to a temporary file and launches the measurement as subprocess. Results are piped via `stdout` and aggregated on the fly. At termination of the sub-process, all results are sent out to the queue for processing, aggregated to the same target-prefix-level granularity as the requests.

This simple calling paradigm mimics how these tools would be used in a classical measurement. Thus, their existing implementations can be used with the system, and behaviour will likely be similar to previous research. For `yarrp`, a minor adjustment is necessary so that the waiting time after all probes have been sent is configurable. The original distribution[11] hard-codes a value of 60 seconds, which is reduced to the parameter value $T_{w,y}$.

An alternative calling paradigm might be to keep the programs running constantly and provide inputs via `stdin`, blocking the stream until further targets are received. This could in theory be possible, since both tools are multithreaded, but is not evaluated in practice. Such a setup would require additional scheduling logic in the buddies to decide when all results have been collected for a request, complicating the implementation, and increasing potential for errors. Additionally, recovery from crashes might prove to be more difficult. Since the simple calling paradigm is sufficient for the performance requirements at hand, this alternative is not further considered. It may provide additional throughput for future research.

For `yarrp`, an obvious question is whether ICMPv6 or UDP traceroutes should be used, since the tool supports both. As discussed in chapter 2, prior work, including the original `yarrp6` research, observes significantly higher response rates for ICMPv6 probes, which motivates relying on it for the proposed measurements.

It should also be mentioned that *FlashRoute* [HRAD20] provides an extension to `yarrp` that promises more efficient probing by targeted re-introduction of state. This might have been[12] a reasonable alternative tool, but was not discovered during initial research for this thesis. While the authors name as a limitation that their state grows too fast to scan even the entire IPv4 internet, this is not necessarily a blocking concern with our short-running measurement runs that have relatively small batches.

Once probing is done, it remains to decide what the responses mean. This interpretation differs fundamentally between one-shot Echo Request probes and traceroutes. Since the system only schedules traceroutes in response to Echo Requests that do not provide the requested information, the cases that should reasonably occur with traceroutes are limited to these where a follow-up request would be issued in the first place.

For Echo Requests, chapter 2 has already established that there are three common responses. Since the primary purpose of Echo Request probing is Last-Hop Router discovery (as suggested by Hobbit [LS16]), the most favourable reply is Destination Unreachable, as its sender reveals itself as the last reachable router on the path towards the destination. What this means in detail depends on the specific subtype, as explained

[11] Constant `SHUTDOWN_WAIT` in `yarrp.h` – https://github.com/cmand/yarrp/blob/a8df9f3d76d1e495d503f128c52f791ab3ca2fd3/yarrp.h#L95 (current as of and accessed on 2024-03-16)

[12] The main reason why this was missed is that literature research focused on IPv6 methods, and it does not seem like there is a published paper for the IPv6 adaption of FlashRoute. A Rust clone of FlashRoute, also seemingly unpublished, is available at https://github.com/BugenZhao/flashroute.rs (accessed 2024-03-17).

Code	Name	Remarks
0	No Route	Subnet not in use (BGP route aggregation [Hol20])
1	Administratively Prohibited	Security policy, subnet may exist
2	Beyond Scope	Should not happen, source address is global
3	Address Unreachable	Subnet in use, address is not
4	Port Unreachable	Should not happen, ICMPv6 has no ports
5	I/E Policy Failed	Sub-type of 1, same meaning
6	Reject Route	Sub-type of 1, subnet does not exist

Table 3.4: Potential interpretations of Destination Unreachable subtypes. Compare for example [Hol20, RBC21, GC06].

in Table 3.4. An additional possible error response is Time Exceeded, which should not usually occur because it is not provoked by an artificially low hop limit. As such, these cases are treated as weirdness, for example caused by actual routing loops or similar misconfigurations. [RBC21, Hol20]

An Echo Response is not very helpful. Despite revealing a reachable address, it does not indicate the Last-Hop Router, which the system is actually looking for. Such cases require follow-up probing. Since random IIDs are chosen for each targeted network, it is unlikely that legitimate end hosts would frequently be found. This is sufficient for these purposes because we are only interested in active subnets[13], and not active hosts. However, there are some configurations where this may happen more frequently, for example *Aliasing* setups discussed in [Hol20].

Finally, it is likely [Hol20] for the most common case to be that no response is received at all. This means that a follow-up traceroute will be necessary to determine the last reachable hop. This hop may or may not be very meaningful. If a large network outright blocks all ICMP traffic, traceroutes will not provide much insight, only revealing upstream transit routers. However, this should at least permit to classify which portion of a prefix employs this policy, since the Last-Hop Router(s) will be the same. A missing response can only be determined because the buddy keeps state of which targets were requested.

For `yarrp` traceroute results, interpretation is trivial. In absence of ICMP Rate Limiting, which is assumed due to the conservative AS-level rate limits, the only necessary logic is to pick the last responsive hop in the trace that is not the target itself.

Attentive readers might wonder why it is necessary to schedule a full traceroute in the first place. On one hand, `zmap` does not currently support specifying different *Hop Limit* values for each probe, necessitating an adjustment to the implementation, which might introduce errors if not properly verified. Relying on the well-established `yarrp` is simpler and less risky. On the other hand, initial probes are sent with a usual value for the *Hop*

[13] Compare for example [RBC21].

Limit (here: 128). Which value to send in a follow-up probe is not inherently derivable from an echo response, as the received TTL only refers to the return path.

Hobbit [LS16] includes a heuristic that yields a reasonable guess at which *Hop Limit* to send. This works by assuming from the received incoming *Hop Limit* the initial *Hop Limit* value of the target host, rounding up to the next common value, e.g. 58 to 64. In addition, it assumes that forward and return paths are of the same length, which need not be true in practice. Such techniques are inherently of heuristic nature and require additional attempts if the guess is incorrect, and a verification probe even if not, significantly complicating implementation. It results that it is impractical to drive the follow-up process with `zmap` for purposes of this thesis.

3.2 Prefix Split Mechanic

Now that probes can be sent and responses interpreted, it remains to discuss how to convert this knowledge into a prefix structure. As discussed, seed data forms a forest of tree roots, with each root being a network prefix originated by an Autonomous System (AS). This section develops the method that refines these roots into a prefix tree based on measurement results. At the heart of this mechanism lies the *prefix split decision*.

Figure 3.3 presents a high-level overview of how the prefix split decision is informed. Subject of the decision is an original **prefix**, which is to be split into two **candidate subnets** if the decision is positive. Evidence for the decision is collected from the **measurement tree**, recursively looking up Last-Hop Routers (LHRs) and weirdness

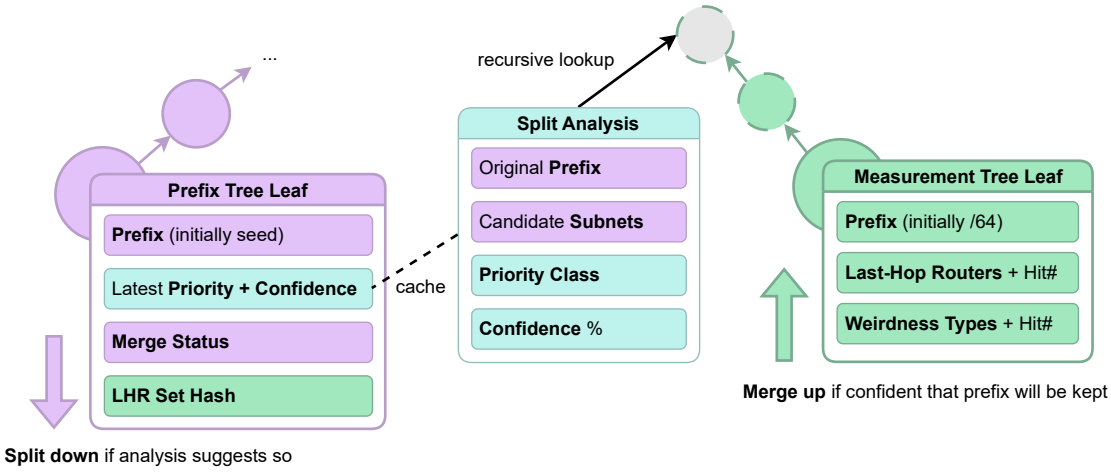


Figure 3.3: Illustration of the split decision's context. Which analyses are scheduled is driven by the prefix tree (left). It keeps a cache of the latest split analysis for prioritisation. The analyses themselves are informed by the measurement tree (right), which keeps information initially on a `/64` granularity, merging up if that level of detail is (likely) no longer needed.

relating to each of the subnets. Depending on what information is available and how it compares between the subnets, the prefix receives a **priority class**. Each class suggests either to keep or to split. Whether this suggestion is acted upon is determined by a confidence metric. Its computation depends on the priority class and how much evidence is available supporting the observation relative to the prefix's size.

Generally, the decision to consider *two* candidate subnets is arbitrary. During initial development, a value of four was also considered, but discarded because there is no obvious reason to choose that specific value. Conversely, the value two is the lowest possible value, offering maximum split resolution (but also requiring maximum node storage). Networks that have many subnets could arrive at their target split sizes faster if this value was increased. However, additional edge-case logic is needed for non-binary splits, and evaluation appears more complex due to additional cases.

There does not seem to be an a priori "correct" value. Rather, the split size is a trade-off between conflicting goals, and should be weighed based on real-world results. For example, it is not possible to decide whether a one-bit resolution is necessary without knowing which prefix boundaries are used in practice. The question of split size is also discussed and evaluated by LaFever [LaF15], who performs recursive prefix splitting similar to the proposed method. For purposes of this thesis, a binary split seems much more straightforward to implement and evaluate, while the implications of higher split sizes remain unclear upfront. This motivates to remain with binary splits for the proposed method, deferring evaluation of larger sizes to future work.

The core components of the split decision workflow are priority class assignment and confidence metric computation. Their underlying process is shown in Figure 3.4. If available, Last-Hop Router sets are compared between the two candidate subnets, falling back on observed weirdness otherwise. If no information is available at all, a `LowUnknown` rating is produced. Overall, the algorithm and confidence metric are primarily motivated by an upfront consideration of possible cases, except for the same-set heuristics, which are informed by experience from the `U-*` measurements.

The graphic is structured fundamentally in columns. Processing begins at *Last-Hop Router set difference* on the top left and continues down and right, passing various conditions. The final *Detail Condition* of each branch serves both as a final decision point and an overall description of the case. It points toward the suggested action in the *split decision* column, which is constant for each *priority class*. On the very right, the *confidence metric* is shown for each priority class. This describes inside the box how the metric of available evidence is computed, and next to it the *threshold* for the suggestion to reach 100% confidence.

This confidence metric is loosely motivated by the question "How likely is it that a single `/64` with different behaviour is missed?". A statistically proven threshold and metric selection is preferable for optimal performance. This is however nontrivial to obtain, especially without the final system being available for evaluation. Values are thus chosen arbitrarily based on a priori reasoning. More detailed analysis is deferred to future work.

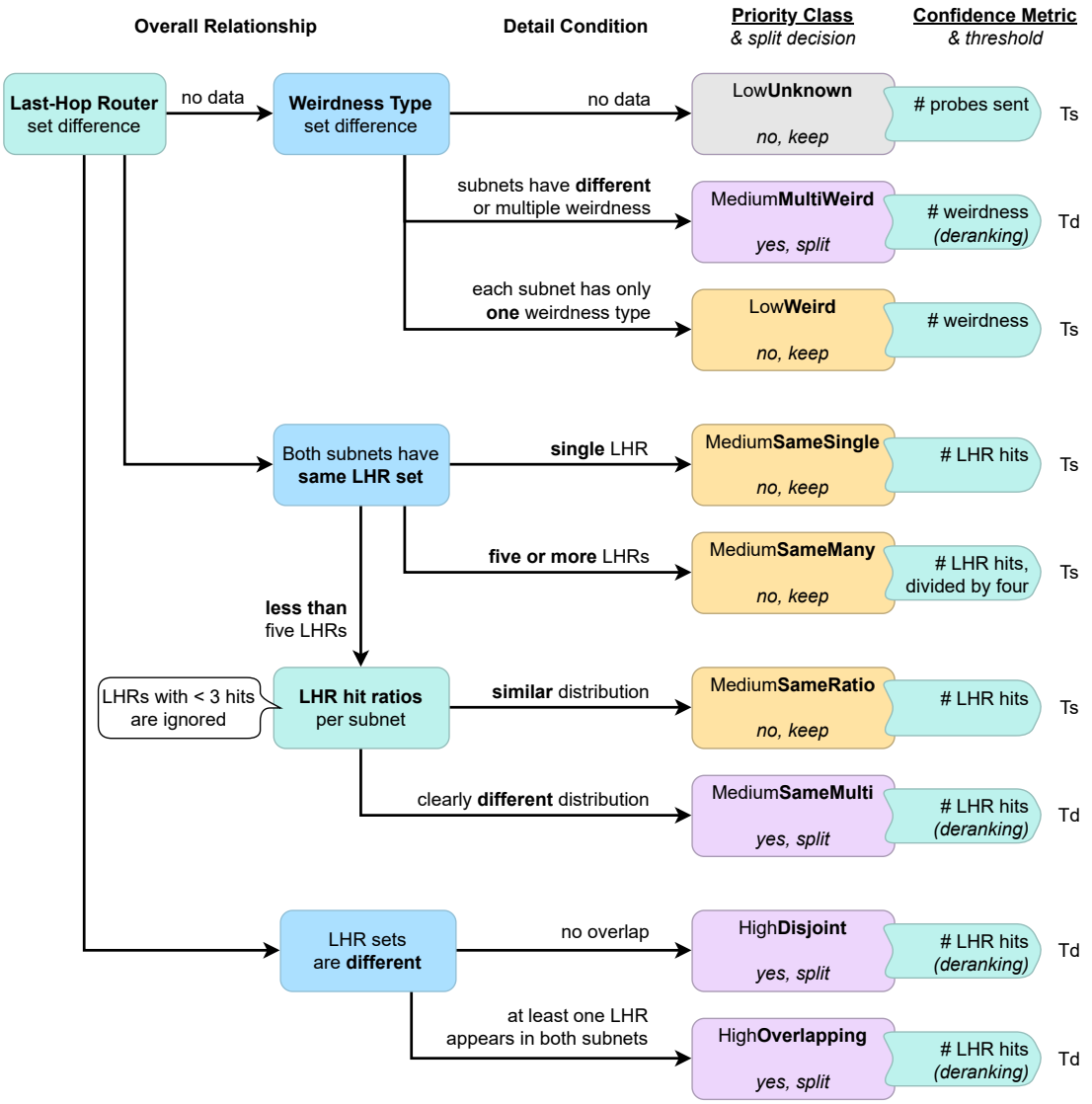


Figure 3.4: Flowchart describing how prefixes are sorted into priority classes based on information from the measurement tree. Values of *Ts* (Threshold for *same* behaviour) and *Td* (Threshold for *different* behaviour) depend on the prefix size. The *(deranking)* indicator denotes that the hit count of the most popular entity is halved during confidence calculation. This grants more influence to entities that are different from it.

With this information, it is possible to discuss and motivate the underlying process, which is described by the arrows. It starts by evaluating the *Last-Hop Router set difference*, whose possible results are explained from top to bottom. If no data is available, it remains to consider the types of recorded weirdness. If both subnets have the same single type of weirdness, the observations suggest that they are structured equally. Further investigation is unlikely to lead to new information, so the split is rejected. Otherwise, both subnets behave differently and a split is justified.

In both cases, the relevant confidence metric is how many weirdness hits have been registered. For the split case, confidence refers to the statement "There are multiple kinds of weirdness", so the most popular kind is deranked by half. Further observations of the same kind make it more likely that the prefix is mostly the same. However, with this setup, the only possible outcome is to delay a split. A future extension might choose to reject it if very few other observations appear.

Continuing down, the next case is that both subnets have the same set of Last-Hop Routers (LHRs). Intuitively, it may seem that a split can be rejected immediately in this case. However, it may be that the distribution of responses between the LHRs is different, which may hint at further subnetting. For example, some LHR A might only be responsible for a `/64` in one subnet, and an entire `/48` in the other, while another LHR B handles everything else. How often this occurs in practice is subject to evaluation. Still, a single conflicting observation may introduce another LHR, placing the prefix in a different class entirely.

In either case, it may be that additional information is hiding in subnets with equal LHR sets, so three additional heuristics are applied before deciding that a split is not necessary. If only a single LHR is observed in both halves, there is no ratio, so a split is rejected, using overall hits as a confidence metric. In addition, experience shows that if five or more LHRs are observed, the algorithm misbehaves, splitting many times, but rarely finding new information. This heuristic uses the same metric, but requiring four times as many hits for safety. What remains are cases with 2–5 LHRs, where the relative distributions can reasonably be compared. Outliers with fewer than three hits are ignored in this comparison. Such artefacts are sometimes introduced by upstream routers if ICMP Rate Limiting occurs very rarely. If the ratio is similar across the subnets, a split is rejected.

Finally, the only remaining set relation is that the LHR sets differ across subnets. More specifically, they can either be disjoint or overlap. While neither algorithm nor confidence distinguish between these two cases, they are recorded separately for evaluation purposes.

With class selection and their confidence metric computation explained, it remains to choose concrete thresholds. Between *same* and *different* observations, slightly different functions are used, both dependent on prefix length. Again, the concrete choices are mostly arbitrary. Due to exponential growth of potential subnets with decreasing prefix length, it is intuitive to choose such a growth pattern for the thresholds as well.

A few factors are interesting to inform parameter choices for this exponential function.

From real-world testing during the `U-*` campaigns, an important result is that the exponential growth must be significantly dampened to avoid waiting days for each split in the beginning when nets as large as `/31` are of interest. In addition, the implications presented by each threshold are different, *depending on the proposed decision*. If the priority class suggests a split, one single result over `100%` confidence triggers the action, so a higher standard is required. For keep decisions, evaluation is repeated until the decision is considered final at `255%`, allowing a lower value. Even though a rudimentary back-tracking (merge) mechanism is implemented for the most obvious cases, splitting is still expensive, so the higher threshold remains.

Formulae for both split thresholds are as follows.

$i :=$ prefix length	limited to be between /12 and /64
$j := 64 - i$	distance from smallest realistic network
$Ts := 64 * 2^{j/4}$	threshold for same
$Td := 256 * 1.4^{j/4}$	threshold for different

Finally, this priority information informs scheduling of prefix analyses. As discussed, a fixed overall budget of F_z[prefixes/round] is available for every scheduling round. This budget is allocated to individual priority classes using lottery scheduling, with every allocated slot being drawn independently. Every class that has leaves available receives tickets up to either their count, or its fixed limit from Table 3.5.

Once winning tickets are divided across priority classes, each slot available in the overall budget is allocated one-by-one according to the resulting probabilities. Each class then

Priority Class	Maximum tickets
High – Fresh	25
High – Overlapping	13
High – Disjoint	12
Medium – Same, single	13
Medium – Same, multiple	23
Medium – Same, ratio	10
Medium – Same, many	7
Medium – Same, multi-weird	10
Low – Weird	2
Low – Unknown	2

Table 3.5: Number of scheduling tickets assigned at most to each priority class. Classes that are expected to lead to more discovered structure receive more tickets. The classes are explained in Figure 3.4. Fresh refers to prefixes that have never been analysed, either due to splits or new seed data.

takes leaf prefixes in database order[14] up to its allocated budget. Before scheduling these prefixes, AS-level rate limits are applied. If slots are left open due to this process, the scheduler repeats it up to four times, incrementally excluding already-limited AS from the database query for each further attempt.

The budgeting process additionally ignores prefixes where confidence is already sufficient. This is trivial for splitting decisions, because the original prefix ceases to be a leaf, and is thus ignored by the scheduling process. Such decisions are executed once `100%` confidence is reached. For keeping decisions, an explicit and arbitrary limit of `255%` is introduced. Otherwise, a significant portion of the measurement budget would be invested in adding additional confidence to decisions that the algorithm is already confident in.

3.3 Ethical Considerations

With the method fully described, it is now possible to apply the general Ethical Standards for Internet Measurements from section 2.5 to the proposed measurement setup. Being an active topology measurement, the experiment does not directly involve human subjects, but rather important infrastructure that many stakeholders rely upon. As such, all feasible measures must be taken to avoid interfering with productive systems, which may cause harm. This section discusses which measures are taken to reduce the risk of this happening based on the common risks discussed in chapter 2. With this information, risks are weighed against benefits.

In general, the following measures are taken to reduce risk and potential impact of the study. These are inspired by best practices from previous work in the field, and in particular suggestions from [DWH13].

1. **Low probing rate.** Both overall and per-AS probing rates are chosen conservatively, and with appropriate safety margins. This reduces the risk of overloading infrastructure and also makes it less likely to trigger monitoring, avoiding investigation effort from network operators.

2. **Indication of purpose.** A webpage is deployed describing the nature of scans and listing source IPs. Due to infrastructure issues, reverse DNS is only available for `AT-11`, but a Google search for the measurement source address immediately points to the provided webpage[15].

3. **Non-intrusive protocol.** ICMPv6 is intended for diagnostics and it is common practice to block this traffic from outside if not desired [RLB16]. No stateful connections are opened, and no applications are targeted. The protocol is only used as designed.

[14] Proper randomisation is nontrivial and is thus deferred to future work. No evidence is available that suggests this being an issue in practice.

[15] Tested in a private browser window at the beginning of public measurements and verified multiple times afterwards.

4. **Opt-out.** An opt-out possibility is provided on the webpage, and the abuse contacts monitored. Existing opt-outs from previous measurements of the same research group are respected.

5. **Removal of sensitive organisations.** The target AS list is manually reviewed and particularly sensitive organisations are removed.

6. **Cooperation with network operators.** The network and infrastructure provider is informed about the measurements. For the `U-*` campaigns, operators of the target organisation have agreed to be scanned. This approach lowers the risk of only noticing methodical issues during public measurements, where it is much more difficult to receive feedback and react.

7. **Review with advisors.** Before measurements against productive infrastructure are started, the method is reviewed with advisors that have performed similar measurements in the past.

8. **Limited scope.** The campaigns only target organisations registered in Austria, so the scope of potential harm is limited compared to a worldwide measurement.

Personally Identifiable Information is not directly collected by the method, lowering the risk of a data breach exposing highly sensitive data. Collected data is in general publicly available, and most is also easily obtainable, e.g. AS-IP assignments. The only categories that might be of concern are reachable IP addresses and network structure.

IP addresses are retained by the system in two ways. First, all responses are stored in a special probe archive for flexibility in evaluation, revealing the IP address of each hit. This information can be removed after evaluation is complete, but is unlikely to be more sensitive than the IPv6 Hitlist which is already publicly available. Second, observed Last-Hop Router (LHR) addresses must be retained in the measurement tree as a precondition for LHR set comparison. This cannot be directly[16] avoided, but they will usually point towards network infrastructure and be discoverable by simple `traceroute` in a targeted attack. A potential exception are Customer Premises Equipment addresses, as discussed in [RBC21]. Due to the low probing rate and lack of focus on this aspect, it seems unlikely that results would contain significant volumes of such data. The retained data is not likely to be so sensitive that it must be destroyed or encrypted in relation to the relatively low risk of a data breach on the measurement infrastructure.

Discovered network structure might provide hints for targeted attacks against these AS. A sufficiently resourceful attacker can however perform the analysis themselves using the underlying tools, which are available publicly. The concepts underlying the

[16] Technically, addresses could be hashed when deploying this system to production. Hiding this information would however present a serious obstacle during evaluation of the thesis. If hashing is applied in the future, size of the input space must be considered. The IPv4 address space for example may be vulnerable to brute-force attacks or rainbow tables. Similar attacks may be possible for IPv6 by relying on public traceroute datasets for search space reduction.

proposed method are also known in isolation already, which an attacker that finds this (probably relatively obscure) thesis would probably be aware of as well. For large AS, it is additionally unlikely that our measurements result in structure observations at the detail level necessary for a targeted attack, especially since actual host addresses are not included.

Based on these considerations, it is possible to answer the questions suggested for an ethics section in [PA16]. As outlined, it is unlikely that the dataset collected by the author, or the act of collection, causes tangible harm, especially considering the measures taken to further reduce risks. The study does not rely on datasets not collected by the author. (IANA allocation data does not constitute a dataset in this sense.) The author is not aware of a generally-applicable technique that could reveal private information about individuals from the collected data. There is in theory potential that lightly confidential information about the network structure of organisations is revealed if a data breach occurs.

In conclusion, it seems like there is very low risk associated with the experiment. The benefit for society is abstract. It is unlikely that the study will directly result in useful knowledge, but learnings and tools could be the foundation for more tangible results in the future. Due to the very low risk, this abstract motivation is sufficient to justify execution of the study.

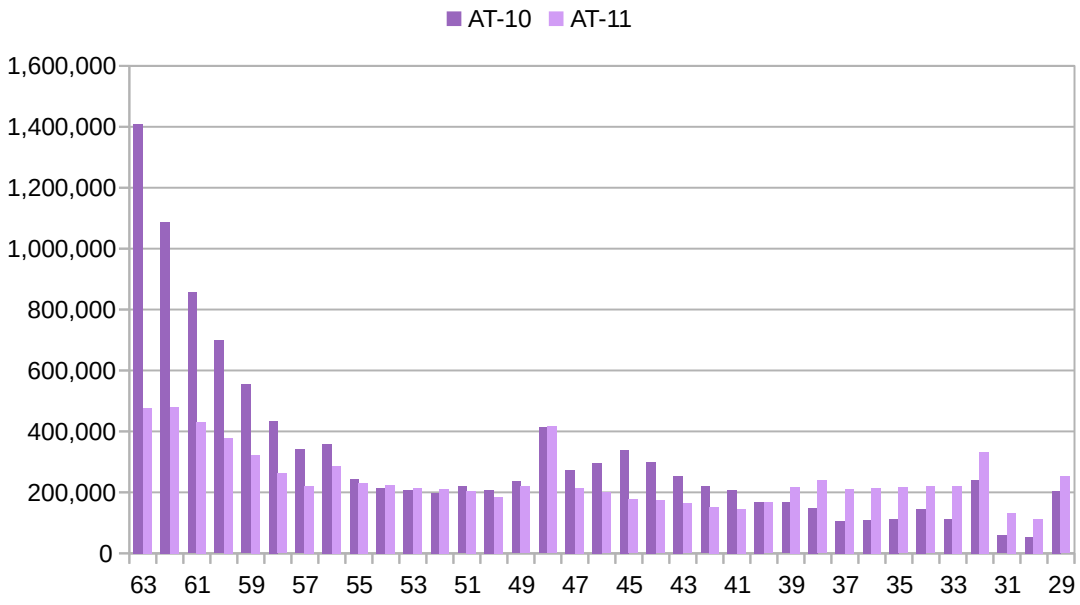
CHAPTER 4

Evaluation

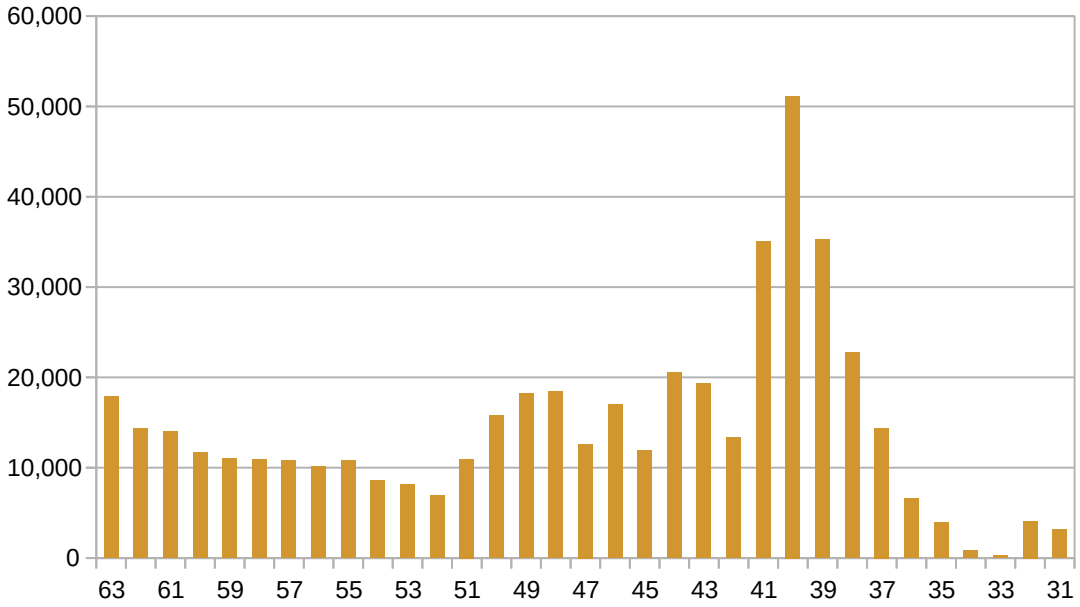
This chapter evaluates the data collected from the measurement campaigns discussed in chapter 3. Further interpretation of the results is later performed in chapter 5. Most of the data used in the current chapter stems directly from the various campaigns' Postgres databases, with aggregated results extracted primarily using plain SQL. Some of the more complex metrics require custom Rust scripts, again reading from the databases. Comparison against uninformed linear probing relies on an additional `yarrp` measurement, which is driven by a fork of Yarrp Buddy. Finally, some aspects are evaluated qualitatively by observation. An overview of evaluation strategies and underlying artefacts for this chapter is presented in Table 4.1.

Description	Strategy	Artefact
Targeted prefix lengths	qualitative	histogram
BGP Root original prefix lengths	qualitative	histogram
Response rates (overview)	quantitative	overall metrics
- distribution across BGP Roots	quantitative	histogram
Available prefixes & budget usage	qualitative	time series
Budget usage (top-10 AS)	qualitative	time series
Confidence distribution	quantitative	behaviour
Priority class changes	qualitative	behaviour
Prefix tree case studies	qualitative	similarity
Prefix tree comparison	quantitative	similarity
Linear probing measurement	quantitative	benchmark
Case study against ground truth	qualitative	network plan

Table 4.1: Overview of data sources for various aspects of the evaluation.



(a) Number of probing rounds issued per target prefix length, for each of the independent `AT-*` campaigns.



(b) Number of probing rounds issued per target prefix length, for the `U-*` campaigns overall. The response archive was not reset with the prefix tree.

Figure 4.1: Issued probe rounds per prefix length for all three evaluated measurement campaigns. Note that this does not indicate the number of individual probes issued.

Focus of the evaluation is the longest AT-10 measurement. Its sister campaign, AT-11, serves as a benchmark for stability evaluation. While AT-10 has the longest runtime, AT-11 benefits from performance optimisations based on the experiences from AT-10 and a higher probing rate. For these campaigns, results are most meaningful to discuss aggregated per BGP Root, since each of these exhibits different behaviour.

The earlier university campaigns U-⋆ are primarily of exploratory nature for algorithm development, but their results are interesting as a case study and due to available ground truth data. While the AT-⋆ campaigns start entirely from scratch, U-⋆ retain the measurement tree, which has effects on the perceived analysis behaviour that need to be considered. Cross-prefix metrics do not apply here since only a single BGP Root is targeted. For this reason, U-⋆ is excluded from a majority of discussions.

The remainder of this chapter is structured as follows. Evaluation begins with an overview of high-level observed behaviour. Algorithm performance is discussed in section 4.2. Finally, section 4.3 presents a case study on the U-⋆ campaigns' results, referring to a network plan and a linear topology measurement as ground truth artefacts.

4.1 System Behaviour & Overall Metrics

Before detailed aspects of the collected data can be considered, it is useful to gain an overview of how the system behaved during measurement. This section discusses what areas the algorithm focused on during probing and which responses it received. These aspects concern overall behaviour and do not drill down into individual organisations, which is why AT-10 is the main focus of this discussion.

While probing, all responses sent by the Buddies are archived into a dedicated table. This Response Archive provides the basis for an analysis of which prefix lengths the algorithm focused on while measuring. Histograms of this behaviour are presented in Figure 4.1.

For the original AT-10 campaign (Figure 4.1a), a clear bias towards smaller networks is visible. AT-11 shows this tendency to a lesser extent. Due to the shorter measurement time, it cannot be ruled out that the same pattern would emerge here as well. Another explanation might be the exclusion of disruptive networks whose behaviour caused the prefix tree to degrade entirely into /64 leaves. Analysis of confidence metrics may provide an intuition into which of these options applies.

Apart from this obvious artefact, peaks are visible around commonly-used aggregate prefix sizes. These clearly include /48, /32, and /29. For /56, a slight peak is visible for both measurements, separate from the tail bias. While the overall shape is very similar in most regions, differences are obvious in two places. AT-10 exhibits a bump around /45, while showing a valley around /36. The reference measurement remains mostly constant in both areas.

Comparison of these observations to the second chart in Figure 4.1b motivates the assumption that measurement behaviour is dependent on the topology that is being

probed. This is to be expected since probing focus is adjusted based on discovered structure. When many organisations are targeted, this effect may even out due to different root prefix lengths and allocation strategies, which is not the case if only a single prefix is analysed.

The specific university distribution permits some conclusions into what the algorithm decided to probe. Observed subnet lengths range begin at the root prefix length `/31` and range until the smallest analysed subnet size, `/63`. `/64` networks are of no interest for analysis: They cannot be further split, and it is thus unnecessary to probe them. The number of potential splits is inherently lower for larger subnets (right) due to the tree structure. If the algorithm suggested every decision to split, a triangle shape would be visible over the entire diagram, similar to the tail of Figure 4.1a, but more extreme. This is not the case here, however some areas do show this pattern, for example `/63–/52`.

When analysing system behaviour, a key realisation is that a single metric does not necessarily show the full picture. For example, the large peak ending at `/40` might invite the conclusion that many networks of this size are present. This is however not the case. Neither leaf nor overall node distributions show significant peaks in this range. An explanation is provided by the split analysis distribution: For `U–3`, two thirds of analyses are performed in the `/37–/40` range. This suggests that the algorithm was not quickly confident to suggest a split in this area, for example (but not necessarily) because only small subsections of the measured space exhibited different behaviour, requiring many probes before the final decision could be reached. In addition, for the university campaigns, it must be taken into account that prefix tree and split analyses were reset, but measurements and probe archive were not. Additional artefacts may result from this.

For the `AT–⋆` campaigns, an overview of what BGP Roots were targeted can be obtained by plotting their prefix lengths. Such a plot is presented in Figure 4.2. A summary of this kind has no value for the university measurement, because only a single `/31` is considered there.

Overall, the selected number of prefixes is similar over both measurements. The later campaign removed some networks, including the only `/45`. Announcements cluster significantly around common “round” lengths, notably `/48`, `/32`, and `/29`. Additionally, networks slightly larger than those sizes are common, corresponding to aggregation of two smaller networks, although the true extent of this effect is exaggerated by the logarithmic scale.

Of the BGP Roots selected for `AT–⋆`, there are four subnet relationships present in both measurements. One additional shared relationship was discovered and removed at measurement start. Four pairs are only relevant for the `AT–10` campaign, and removed for an unrelated reason for the second one. None of these pairs reside in identical AS, but all except one belong to identical organisation, most even with identical AS names. The

[1] This counts the Last-Hop-Router hits from addresses that are located in the BGP root that is being probed. Misses may for example be due to unallocated space (upstream infrastructure), security policy, or ICMP Rate Limiting.

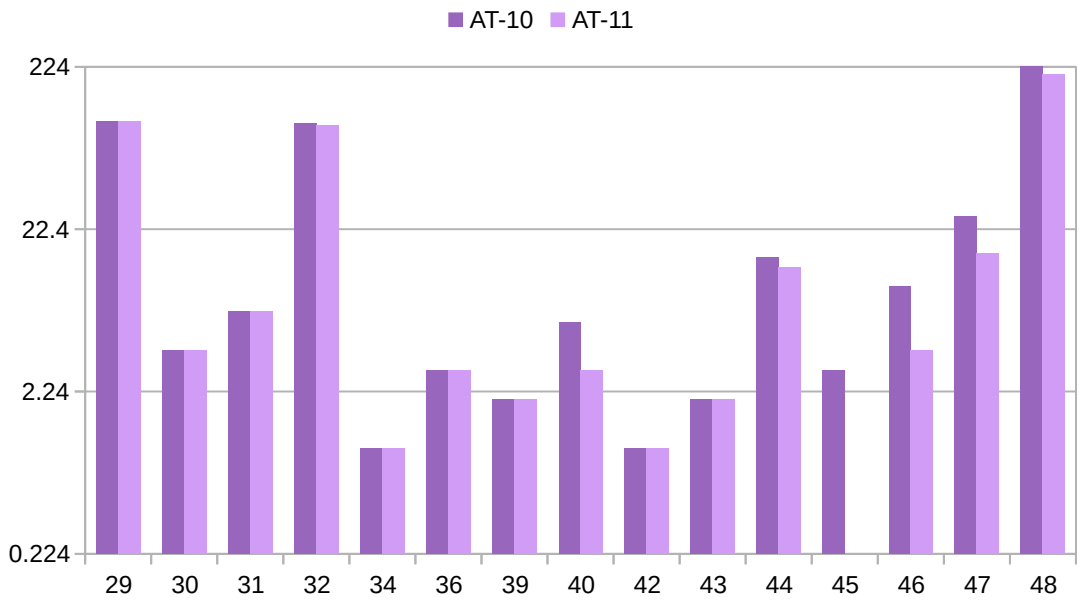


Figure 4.2: Distribution of the BGP roots' prefix lengths for each of the `AT-*` measurements. The vertical axis is scaled logarithmically because of the significant peaks.

	U-*		AT-10		AT-11	
`zmap` sent	7 543 008	(100%)	186 428 544	(100%)	140 716 280	(100%)
- received echo	0		178 478	(0.1%)	145 345	(0.1%)
- received error	6 315	(0.1%)	24 928 410	(13%)	21 094 974	(15%)
`yarrp` sent	2 095 023	(100%)	58 444 197	(100%)	41 233 802	(100%)
- no response	274	(0.01%)	1 392 691	(2.4%)	1 846 954	(4.5%)
- in-prefix LHRs[1]	364 956	(18%)	3 581 796	(6%)	5 851 489	(14%)
probes overall	9 638 031		244 872 741		181 950 081	

Table 4.2: Hit rates per campaign, attributed to tools and response types. These numbers correspond to individual probe targets (not rounds, not packets), as opposed to Figure 4.1, and are computed from the response archive. `zmap` probes send a single packet, while `yarrp` might send between 15 and 31.

four larger networks are of sizes /29 (2), /32, and /36, encompassing networks of sizes /34, /47 (2), and /48 (5). Due to the relatively feq targeted AS, it was not initially deemed necessary to filter on the BGP Roots for this condition. An example approach for how BGP Root data cleaning could have been performed can be found in [Mai21].

Probed prefixes did not always respond to probes. In fact, it is much more common for networks to ignore requests, as can be seen in Table 4.2. Overall, behaviour is similar in related measurements, but different across types. For the university, it is clear that the infrastructure is configured either not to send errors, or to filter them at the network border. On the other hand, in-prefix Last-Hop Routers are more visible in comparison to the AT-⋆ campaigns.

Despite only running for 22% of the time, AT-11 reaches over 70% of the probe count for both tools. This is higher than expected when only correcting for duration and the 50% increase in probing budget (Table 3.3), which suggests an expectation of 33%. It is likely that performance improvements implemented as a result of observability metrics introduced during AT-10 led to this difference.

Despite this difference in probe count, the two sister measurements show similar ratios for zmap metrics overall. Clear differences to the university campaign suggest that this behaviour might be different depending on what network is measured. Whether this is true can be evaluated by looking at the distribution of metric values across individual BGP Roots. For the error rate, this distribution is shown in Figure 4.3. There is a clear split between networks that send errors in most cases and networks that rarely send them, with a non-negligible portion distributed in between. More than half of the probed BGP Roots do not usually respond with errors, with 53% not sending a single one, similar to the behaviour seen in the university campaign. The distributions differ in details, but overall have a similar shape.

The echo response rates show a different picture. For the university campaign, not a

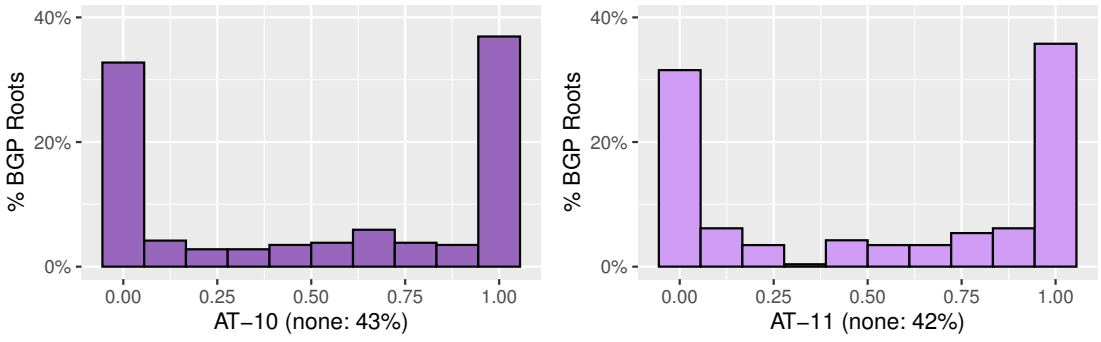
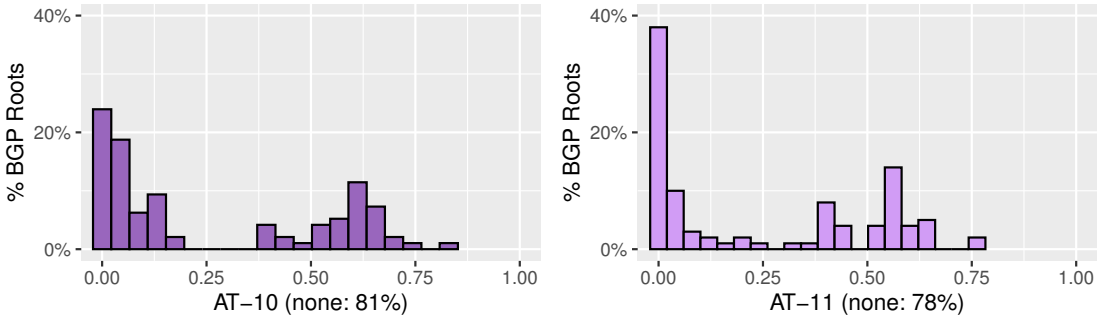
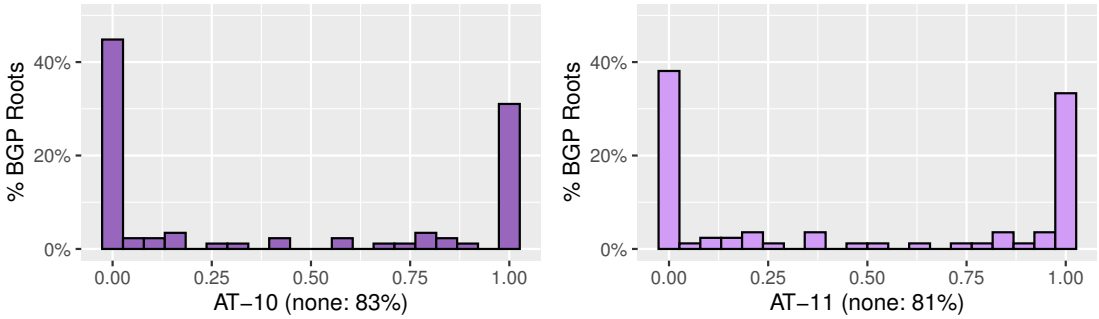


Figure 4.3: Distribution of error reception ratios (compared to zmap probes sent) over all BGP Roots measured during both campaigns. The frequency is given as percentage to correct for a slightly different number of BGP Roots. Entries with no observations are excluded to aid legibility.



(a) Histogram of: % `yarrp` probes that resulted in no response at all



(b) Histogram of: % `yarrp` probes that resulted in an in-prefix Last-Hop Router

Figure 4.4: Partial distribution of the `yarrp` metrics across all BGP Roots, for both campaigns. The majority of BGP Roots does not exhibit either behaviour. These are excluded for legibility.

single echo response was received by `zmap`, and 98% of BGP Roots in `AT-*` follow suit with rates less than 0.2%. This is to be expected, since it is extremely unlikely to guess an allocated IPv6 address even in a single `/64` due to the sheer number of possibilities. The remaining 2% echo-respond to more than 99.9% of all queries. All measured BGP Roots fall distinctly into one of these two clusters, with no in-between.

On `yarrp` side, distributions also differ between BGP Roots. The vast majority of prefixes yields responses for every traceroute, with around 20% behaving otherwise in each campaign, as shown in Figure 4.4a. For `AT-10`, two clusters are visible, each comprising around half of the remaining roots. While one cluster is close to the expected behaviour of yielding a response for every trace, the other is centred at around 60% miss rate. `AT-11` shows a similar pattern, with the cluster around zero more accentuated and the higher cluster split in the middle. Out of the 96 (resp. 100 for `AT-11`) depicted prefixes, around 75% exhibit non-responsive yarrp probes in both campaigns.

These overall response count statistics invite computation of an overall hit rate. Such a combined rate is however not meaningful due to the nature of traceroutes. These send

repeated probes with different *Hop Limits*, which will receive responses from transit prefixes as the limit decreases. Even if the target prefix never responds, a last hop might result from such a transit prefix, making a straightforward binary classification of "hit" mostly meaningless. For purposes of research question 5 (hit rate improvements), a more interesting metric is the ratio of last hops with addresses inside the BGP Root.

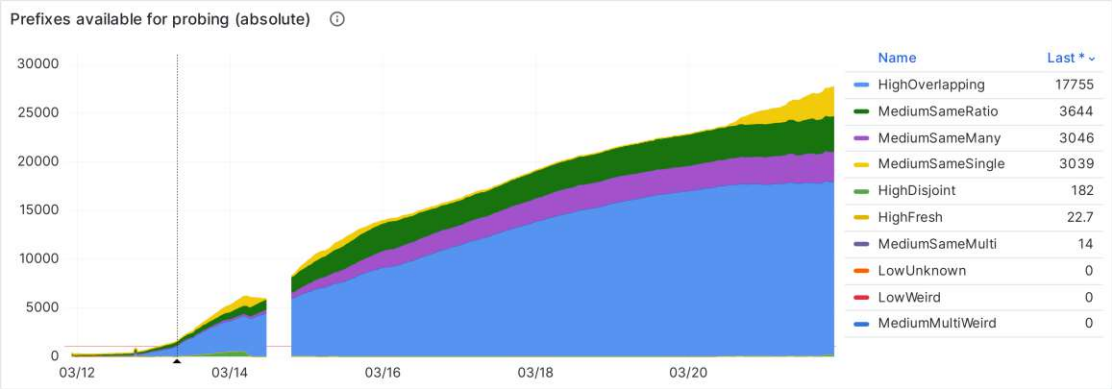
The distribution of this metric value across BGP Roots is shown in Figure 4.4b. In both campaigns, more than 80% of BGP Roots never yield a single in-prefix Last-Hop Router. While this ratio is not better than for the missed responses metric discussed above, it is still likely more meaningful. There may be different reasons why prefixes never include an in-prefix router, and a useful structure might still be discernible. For example, an operator might number border routers from the upstream provider's address space, hiding further internal structure using security policies. In this case, at least site-level topology might be discernible if each site has a dedicated uplink. Another possibility is that historically separate AS share core routing infrastructure. AS names in the seed data suggest that this is commonly the case after mergers of ISP companies. Finally, it may be that a large portion of the space is unallocated, and the algorithm has not (yet) found the interesting areas. Further conclusions might be obtainable for these assumptions by grouping related BGP Roots and explicitly labeling transit prefixes in future work.

Similar to the remaining metrics, in-prefix ratio distributions are similarly-shaped for both measurements. Peaks are around very low values, supporting the mostly-unallocated hypothesis, and very high values, where nearly every trace reaches an in-prefix router. This pattern suggests that the ratio of in-prefix Last-Hop Routers might be network-inherent property, and thus should not be considered in isolation. That is, a low in-prefix rate does not necessarily mean that no meaningful topology was discovered.

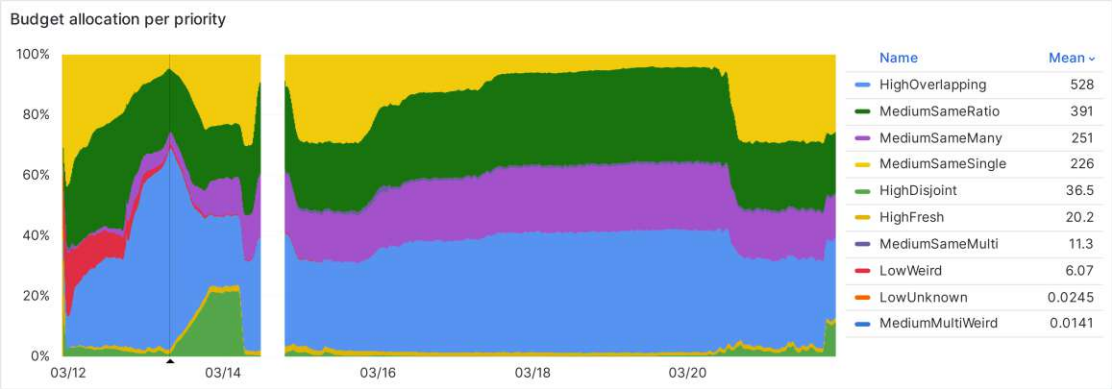
System behaviour is characterised not only be distributions of result data, but also by changes over time. Such temporal data unveils local at-the-time views of the algorithm, which may help explain certain behaviour. The proposed setup achieves observability of this kind using custom metrics, which are periodically sent to *Prometheus* and visualised in *Grafana* (see Figure 3.2). Collection of these metrics only started while `AT-10` was already running, as there was no immediate need before. The concrete motivation for increased observability was that, after around three weeks of runtime, queues started to build up backlogs with no discernible explanation. Additional business metrics helped resolve underlying performance issues.

Due to the aforementioned incomplete data for `AT-10`, the main focus of this analysis is `AT-11`, which has metrics available from the start.[2] Overall focus of the algorithm is driven by how many and which prefix tree leaves are available to analyse. How this focus developed during the campaign can be observed in Figure 4.5. A key reference value necessary to understand these graphs is the prefix budget per round. This threshold

[2]There is a gap in the data, caused by an eight-hour outage of the cluster hosting the observability infrastructure. Only metrics collection was affected, the measurement continued without interruption.



(a) Number of prefixes available for probing, attributed to priority classes. The horizontal line indicates the per-round prefix budget of `1620` for reference.



(b) Relative allocation of probing budget to priority classes.

Figure 4.5: Graphs describing available and allocated prefixes for the probe budget, over the entire runtime of `AT-11`. The black vertical line indicates when available prefixes first exceed the per-round budget.

is indicated in Figure 4.5a as a horizontal line near the bottom. An intersecting black vertical line marks when the system first reached its target probing rate, 33 hours after start. Following this ramp-up stage, search space increases quickly. After some more days, the increase of potential prefixes slows down, making the overall curve shape similar to logistic growth.

While this observation is true about the overall growth, individual priority classes behave differently. Curve shape is mainly driven by the High – Overlapping class, which constitutes the majority of probing potential. Medium – Same, ratio and Medium – Same, many behave similarly, although at lower absolute values, and with additional deviating artefacts. Medium – Same, single shows some velocity initially, but starts growing rapidly as the experiment concludes. Remaining classes are not clearly discernible in Figure 4.5a due to low absolute values. High – Disjoint has a large peak around March 14$^{\text{th}}$. The small yellow area above it corresponds to the mostly constant contribution of High –

Fresh, which shows that fresh splits are made at a regular rate.

Due to the per-round prefix budget, impact of high-contribution classes on actual allocation is limited. Figure 4.5b provides more visibility to developments in other classes. A notable break is visible once probing potential exceeds the prefix budget, indicated again by a vertical black line. Apart from that, most visible patterns are a consequence of the class-level behaviour discussed above. The large red section in the beginning is due to the relative impact of few Low – Weird prefixes, which diminishes as more other candidates become available. Starting around March 16th, Medium – Same, single prefixes are less available, shifting the distribution. Relative availability returns to normal one day before termination with a large increase also visible in the absolute chart.

Apart from priority classes, an important insight is how budget is allocated to individual Autonomous Systems (AS) in the measurement. Figure 4.6 shows this aspect for both `AT-*` measurements, so that behaviour can be compared. One experience from `AT-10` is that a single AS constantly consumes the entirety of its probing budget. An obvious resulting action is to exclude this constant consumer from measurement, focusing probing on the remaining AS. This measure is taken for `AT-11`, and the result is discernible in its chart, Figure 4.6a.

In general, measurement budget seems to be divided more evenly across the top-10 AS in `AT-11`. Some still hit their budget and do not scale down before termination, but others do seem to be satisfied after a while. Notably, AS197634 (orange) initially consumes a large portion of the budget, but ceases to do so close to termination. This suggests that there are cases where the algorithm is at some point content with the discovered topology and schedules fewer prefixes. Another factor that might have caused this improvement in diversity may be the 50% increase in probing rate, providing more space for diverse AS.

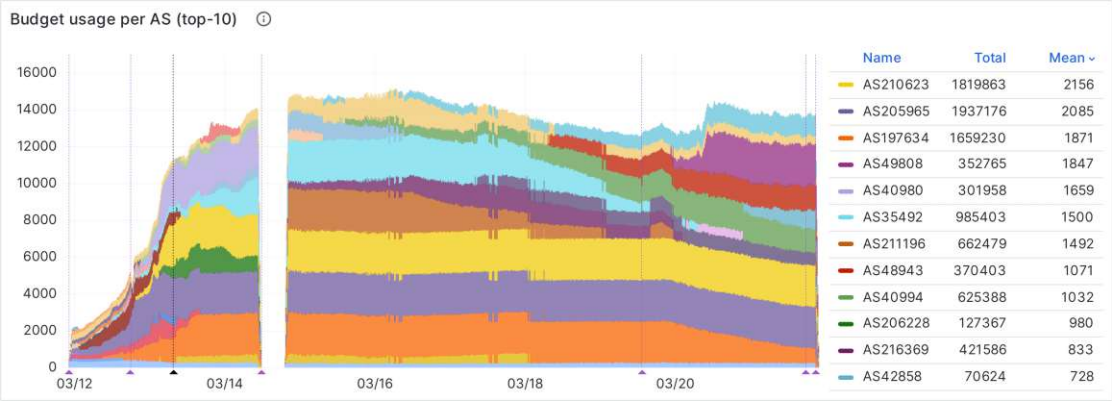
Discussion of split behaviour is omitted due to space constraints. Available metrics for this concern are shown in Appendix C. Split Behaviour Graphs for reference.

4.2 Algorithm Performance

One key interest of this research is the performance of the proposed algorithm and system. This aspect is evaluated in the section that follows. Focus lies first on confidence metrics. The question here is whether the mostly-arbitrary thresholds are appropriate, and if different values might improve performance. The second focus is a stability comparison, which is the core motivation for launching the `AT-11` campaign. This aims to answer whether the system arrives at the same conclusions for repeated measurements of the same space.

Confidence metric value distribution across different prefix lengths is a performance aspect of the metric. That is, what ratings do nodes usually receive, and what influence do the cut-off values have? The relevant thresholds for this analysis are 100, the minimum confidence needed to execute a split suggestion, and 255, the maximum confidence, after which keep suggestions will not be further verified.

(a) AT-11



(b) AT-10

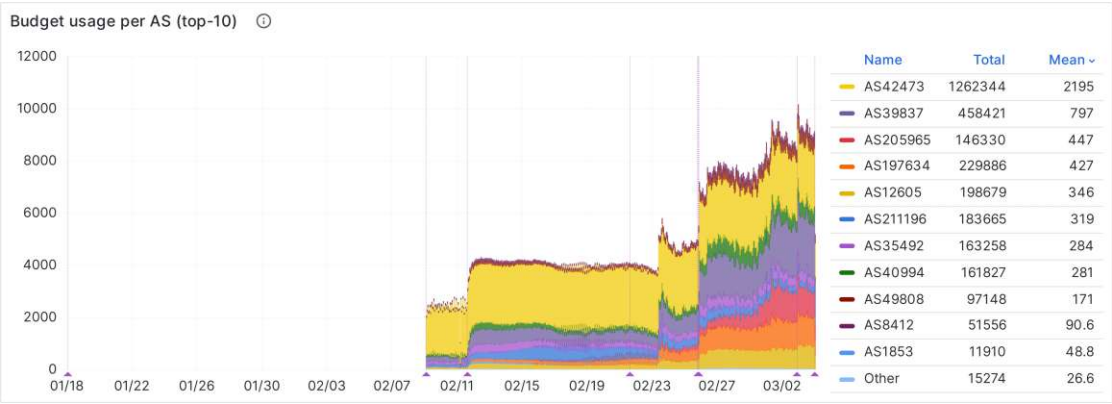


Figure 4.6: Allocation of actual probing budget to ASes, top 10 targets per round – both campaigns for comparison. Mean counts refer not to individual rounds, where the limit is 75 per AS, but to one-hour smoothing intervals used for analysis. The highest possible allocation is 2250 rounds per hour. Less-popular AS are omitted to improve legibility.

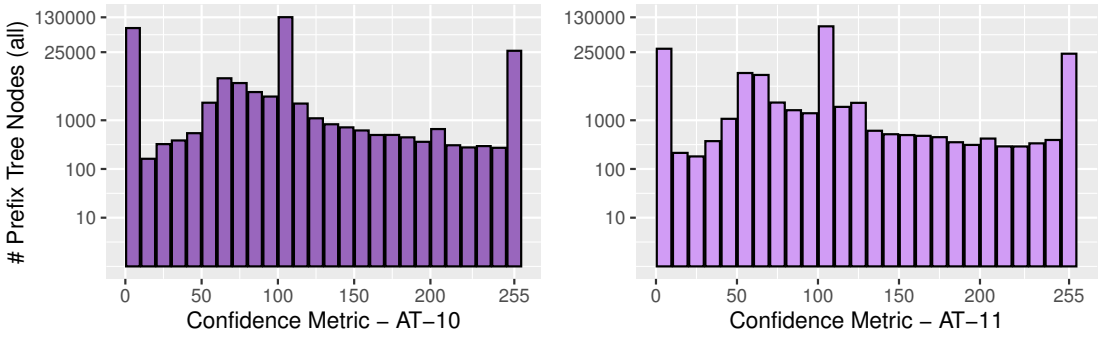


Figure 4.7: Histogram of confidence metric values across the prefix tree, for both `AT-*` measurements. Values are combined into bins of width 10, and the node count is plotted on a logarithmic scale due to differences in magnitude.

The confidence distribution across prefix tree *nodes* is shown in Figure 4.7. This includes both leaves and internal nodes that have already been split. In a converged measurement, no nodes below 100 would be expected. Ideally, there would be one cluster close to 100, indicating an optimal split threshold with no late discoveries, and another peak exactly at 255, indicating confident keeps. In practice, some activity between 100 and 255 will be difficult to avoid in larger networks, because we are unlikely to guess the location of small sections with different behaviour, e.g. if only a single `/56` of a `/31` is currently in use.

Most of the prefix tree at campaign termination revolves around the significant values 0, 100, and 255, in both cases. The cluster around zero includes prefixes that have not been analysed yet, are in a low priority class, have reached the maximum `/64` granularity, or are very large and thus need many probes to progress. The nature of the cluster around 60 is not entirely obvious, but it may indicate "the average node" moving towards the split threshold, as the system gains confidence in the discovered structure on average. This might also explain why this cluster is skewed slightly to the left in `AT-11`.

The 100 cluster comprises nodes that have been split immediately, while values above indicate that information motivating a split arrived above the confidence threshold. Diminishing returns of validating keep decisions are suggested in `AT-11`, where node count declines visibly. `AT-10` does not show this transition as unambiguously, with the above-100 distribution looking more constant, and even increasing slightly before the keep threshold. Whether either of these claims is significant requires detailed statistical analysis which is out of scope here. Care must be taken because the logarithmic scaling of the node count obscures the true extent of differences.

Statistics targeting only leaves that can be further split are included in Appendix D. Confidence Distribution of Leaves due to space constraints.

Another interesting aspect of the confidence metric is how often the verdict changes before analysis concludes. If this number is low, and changes happen early in the prefix's lifespan, a lower threshold is possible. On the other hand, if changes frequently occur

	U-*		AT-10		AT-11	
prefix tree nodes*	567	(100%)	189 559	(100%)	142 122	(100%)
- never changed	548	(96.6%)	173 157	(91.3%)	135 889	(95.6%)
- changed once	18	(3.2%)	12 167	(6.4%)	5 260	(3.7%)
- changed more often	1	(0.2%)	4 235	(2.2%)	973	(0.7%)
maximum changes	2		32		8	

Table 4.3: Number of changes to priority class assignments per prefix.
* Excluding `/64` nodes, where a further split is not allowed, and no analyses are scheduled.

above the confidence threshold, it may be wise to increase it. As indicated in chapter 3, the threshold depends both on prefix length and the recommended split decision (keep or split). These parameters thus suggest themselves as groups for analysis.

Evaluation of split decision changes is not easily possible directly in the database. The following discussion relies on a Rust script in the Crab Tools analysis toolkit to collect this data. Results are indicated in Table 4.3, which, for each measurement, groups the number of nodes depending on how often their verdict changed. An interesting result is that, for every measurement, a large majority of nodes never change their verdict, and most of the remaining nodes only change it once.

Metrics are slightly worse, in that changes happen more frequently, for `AT-10`, which suggests that changes could be more frequent with longer run times. Other factors such as algorithm improvements or higher probing rate might also influence these metrics, with the actual cause not being obvious. It may be that priority classes behave differently, motivating a further grouping based on the last priority class that the prefix received. This grouping is presented in Table 4.4.

The concrete implication of "last" in this context differs based on split result and confidence. If the metric is below the threshold (100% for split and 255% for keep), then the last analysis is the latest one performed before termination, indicating that insufficient probes were made to confidently decide on this prefix. For confident keep decisions, it is the last analysis that confirmed the split at 255%. For confident split decisions, it is the first over-100 analysis that suggested the split.

Overall, differences between the two campaigns are minor. An auxiliary result is that the distribution of nodes across priority classes is subtly different. `AT-11` has fewer overlapping nodes, but more disjoint, same-single, and same-ratio results. Both campaigns assign High – Overlapping to a significant majority of nodes, skewing overall distribution.

Priority classes in general exhibit different change count distributions. For example, both High classes skew heavily towards no changes. This may be an example of survivorship bias, since repeated measurements of these classes may lead to a transition to Medium – Same, where changes are more common. A more involved analysis might consider the paths that individual prefixes take across priority classes. This seems feasible in general

Last Priority Class	AT-10				AT-11			
	Nodes	0	1	+	Nodes	0	1	+
High – Overlapping	77%	95	3	1	70%	97	1	0
High – Disjoint	2%	99	0	0	5%	99	0	0
Medium – Same, single	4%	99	1	0	8%	99	0	0
Medium – Same, multiple	0%	19	29	51	0%	56	42	2
Medium – Same, ratio	12%	75	19	6	14%	89	9	2
Medium – Same, many	5%	59	34	7	3%	56	42	2
Medium – Same, multi-weird	0%	0	100	0	0%	0	100	0
Low – Weird	0%	89	8	3	0%	88	12	0
Low – Unknown	0%	100	0	0	–			

Table 4.4: Number of changes to priority class assignments per prefix, grouped by the last resulting priority class. The first columns indicate how many nodes are members of the priority class at termination. The following three columns show the number of changes, which are also given in percent, but without the sign to aid legibility. Highlighted rows indicate classes that suggest a split.

due to the low average number of changes: It would only need to consider a one change to cover most of the space. Such an analysis is however deferred to future work.

Change count distribution does not seem to depend exclusively on whether the class suggests split or keep. Classes that recommend a split are highlighted in the table and do not show similar distribution, except among the High classes. Other than that, Medium – Same, single behaves similarly to High. 99% of nodes that terminated on this class received the same rating initially. This result does not necessarily justify a threshold reduction. For such a decision, the initially-assigned priority class needs to be considered.

What remains for this section is to consider stability of the prefix tree across `AT-*` measurements. This can be evaluated by comparing the resulting trees. In general, trees can be compared using edit distance metrics [YKT05]. These evaluate how many changes need to be made to mutate one tree into the other. Such metrics alone are likely not meaningful for purposes of stability analysis, since nodes higher up in the tree represent much larger networks, and as a result, differences there have significantly more impact. An erroneous split on `/63` level has practically no consequence, while failing to split a `/33` might obscure entire departments of a large organisation.

Lack of a single metric means that multiple aspects need to be considered. The following analysis first considers 1:1 node comparison, and then pivots to limited-scope qualitative analysis of small subtrees. A first look at the data relies on the former strategy, targeting the entire tree. In particular, the evaluation compares prefix tree nodes one-by-one across both `AT-*` campaigns. Based on split verdict and presence, each node pair is sorted into a presence relation class. Nodes that are present on both sides receive Same or Different

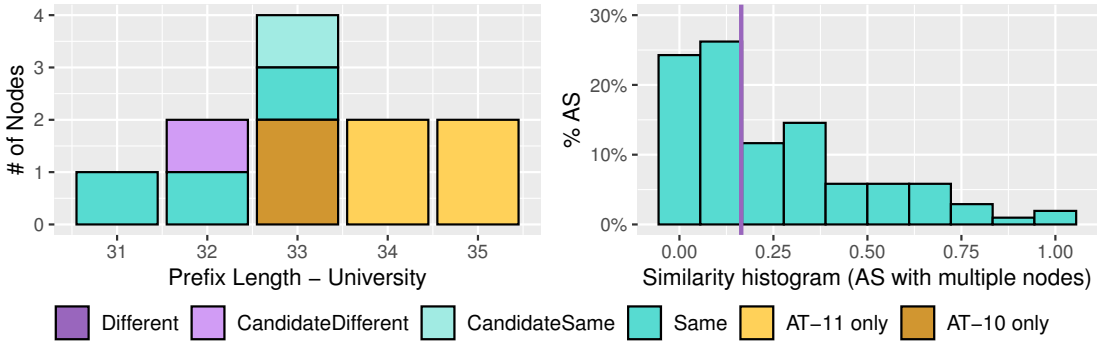


Figure 4.8: Left side: Prefix tree comparison across the evaluated AT−10 measurement and the reference campaign AT−11. The chart shows how many nodes of each length receive which comparison result, targeting the university from U−⋆. Right side: Distribution of similarity ratio over all AS that have multiple nodes. The ratio compares nodes with Same rating against the total node count in that AS.

by comparison of their split verdict across both campaigns. If the decision is below the confidence threshold on either side, the relation is downgraded to CandidateSame or CandidateDifferent. Finally, nodes that are present only in one of the trees are marked as such. Subnets that are not present in either tree are ignored.

An obvious choice for a target network is the university prefix analysed in the U−⋆ campaigns. Knowing how it behaved in the AT−⋆ measurements may be a benchmark for discussion in Case Study: University (section 4.3). The comparison result is shown in Figure 4.8, left. This chart reveals that, unfortunately, the large-scale campaigns were not able to gain sufficient confidence to pivot significantly into the network structure. For the longer AT−10, analysis reaches the /33 level, while AT−11 gets as far as /35. The parts of the structure that it discovered match the ground truth network plan.

A majority of the difference visible in the chart stems from AT−10 erroneously splitting an unallocated /32 network with High – Overlapping priority at 105% confidence. AT−11 keeps it correctly at Medium – Same, ratio, observing only two Last-Hop Routers instead of three, however only reaching 89% confidence at termination. With a longer duration, it is possible that AT−11 would have made the same error. In reality, only a single Last-Hop Router is responsible for that prefix, and the other two observed routers are further up in the trace. They are likely only observed due to ICMP Rate Limiting.

For reference, the right part of Figure 4.8 shows the overall similarity distribution across all measured AS. It excludes AS that only contain a single node in both measurements.[3] Overall, the similarity in terms of overall node count is low for most networks, with the median (vertical line) being around 1/5 nodes being rated the same across both

[3]The correct granularity for this analysis is the BGP Root, as it begins the prefix tree. However, this data is not readily available for this analysis. AS with 1.0 similarity due to multiple BGP Roots are manually removed.

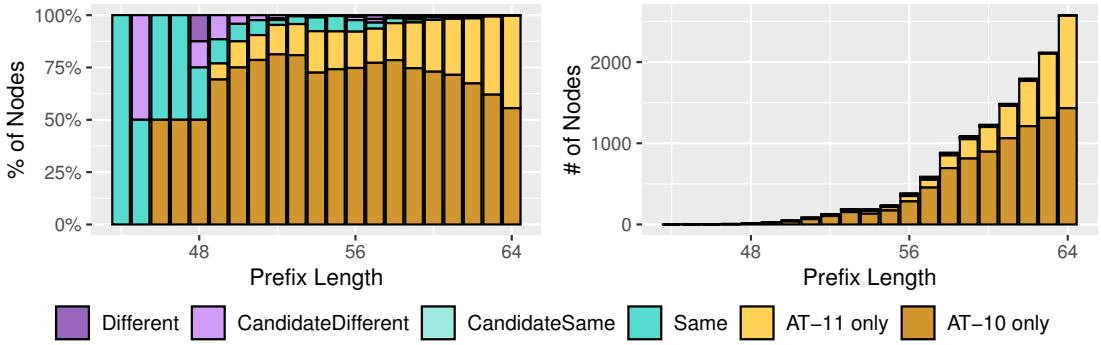


Figure 4.9: Prefix tree comparison for a highly-split AS. Both charts shows how many nodes of each length receive which comparison result. The left one shows relative distribution for each prefix length, while absolute counts are displayed on the right.

campaigns. A factor for this result might be that the algorithm is not very resilient towards Last-Hop observations higher up in the trace, for example caused by ICMP Rate Limiting or service disruptions.

Since this case only exhibits a very shallow tree, the analysis is not very insightful, motivating a second case study. As discussed on page 54, there are some networks that initially consumed their entire probe budget in `AT-11` and later stopped. This behaviour suggests that the algorithm is satisfied with the obtained confidence and thus slows down discovery. Evaluation results for the aforementioned network are presented in Figure 4.9.

Overall, there seems to be little agreement between the trees on this network, as indicated by the Same ratings disappearing as prefix length increases. Inspection with the `prefix-inspect` command of Crab Tools reveals that both studies encounter a large diversity of around 40 Last-Hop Routers, with most only receiving a single-digit number of hits. Further research reveals that many of the discovered routers are owned by transit providers, suggesting that they are located further up in the trace. If transit routes are consistent, as with the university case, the algorithm should be able to reduce this noise using the ratio heuristics (Figure 3.4). In this case, it seems that the nature of fluctuations is not handled well by the heuristics.

For larger networks with high confidence thresholds, effects are cancelled out once the Medium – Same, many heuristic is triggered. If additional Last-Hop Routers only appear slowly, as seems to be the case here, splits quickly accumulate. An investigation into the split behaviour of this prefix shows that once splits are made, the algorithm usually returns to non-split heuristics on the leaves. If too many splits were made already, this normalisation only occurs when subnets are already very small, increasing the impact of measurement artefacts.

A recovery mechanism exists to merge back adjacent nodes with the same priority class and identical Last-Hop Router set, but it does not trigger in this network for two reasons. First, occurrence of diverse transit routers prevents adjacent leaves from arriving at the

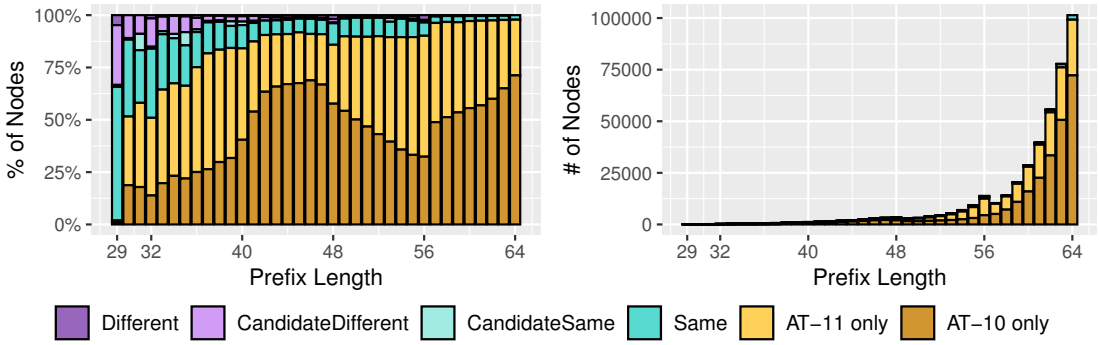


Figure 4.10: Prefix tree comparison across the evaluated `AT-10` measurement and the reference campaign `AT-11`. The left chart shows the ratio for each prefix length, while absolute node counts are presented on the right for context.

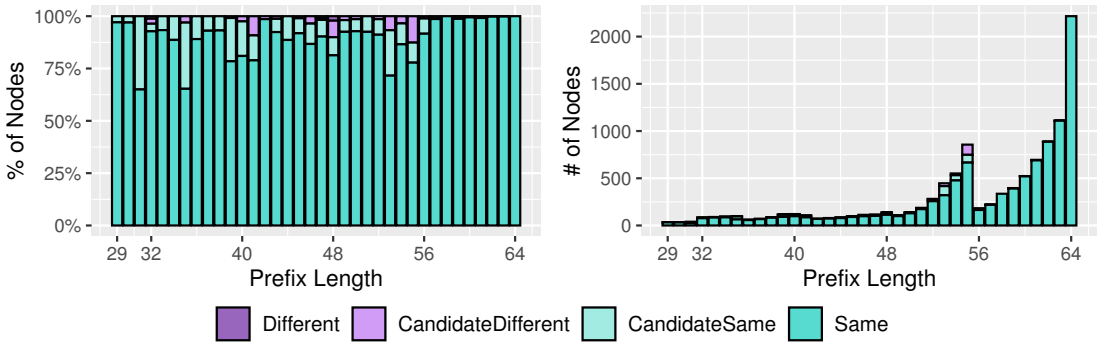


Figure 4.11: The evaluation from Figure 4.10, but excluding all nodes that have a Medium – Same class on either side. Note that this graph only shows 10 000 nodes in total, a tiny fraction of the overall prefix tree.

same Last-Hop Router set if the path has changed already. Similarly, adjacent leaves are sometimes spread between Medium – Same, ratio and Medium – Same, many. The merge algorithm is not aware that these are closely related. Second, in this case, splitting quickly reached the `/64` level. Prefixes split this far are never analysed again because no further split is possible. This means that the merge algorithm is not triggered for these subnets, and additionally would also be missing the necessary evidence. Development of a more resilient recovery algorithm for such cases might be a useful improvement in future work.

Both subtree analyses unveil special cases without providing a clear picture of overall stability. This suggests invocation of the alternative analysis method, coarse-grained overall statistics. While detailed conclusions are unlikely to result from this, it can provide a general overview of stability behaviour. The result of comparing all nodes across both campaigns one-by-one is shown in Figure 4.10. Overall, it is clearly visible that stable structures are popular only for very large networks. As subnets get smaller, splits tend to

disagree, with a majority being disjoint across measurements. These disagreements may again be caused by measurement artefacts that the sameness heuristics are not resilient against.

As a result of this suspicion, stability of the algorithm in cases other than Medium – Same becomes interesting. An adjusted analysis, excluding nodes that receive such a rating in either tree, is shown in Figure 4.11. Stability is significantly better in these cases, but this is only an insubstantial fraction of the tree's nodes, comprising a mere 10 880, which is dwarfed by the total of 423 629. Such a low ratio is surprising, considering that a large portion of nodes in each tree is rated High – Overlapping (Table 4.4). A major factor for this number is that most nodes occur only in one of the trees. This may suggest that the algorithm is not resilient against false splits, resulting in an explosion of erroneously-created nodes.

Collected data suggests that the algorithm produces stable results for the few cases where only a single Last-Hop Router is observed, and unstable results otherwise. With this information, it is not easily possible to discern which splits were made erroneously in general. Results of the case studies above suggest that a significant portion may be caused by measurement artefacts, suggesting that the algorithm itself may be viable if resilience against these factors is improved. The next section analyses algorithm behaviour in a limited case study where ground truth is available and this aspect can be evaluated.

4.3 Case Study: University

While metrics and black-box analysis of network behaviour can provide an intuition towards the performance of the proposed method, they are not a robust benchmark, as the correct structure remains unknown. This section relies on ground truth data related to a single organisation (the university scanned in the `U-*` measurements) to gain qualitative insights into algorithm performance in practice.

Ideally, this evaluation would be performed against a detailed topology actually discovered by the productive `AT-*` measurements, showing that real organisations can be sufficiently analysed in such a study. Unfortunately, the prefix tree found in these campaigns is very shallow, and analysis must revert to the original exploratory `U-3` measurement. This does not include all fine-tuned heuristics, but is sufficient to discuss the general concept.

The first part of this section evaluates against ground truth obtained by linear uniform probing of all `/48` subnets of the target prefix. Its main focus is to decide if the algorithm missed substantial parts of the prefix structure, compared to the results obtainable by similar means, but without applying structure-aware prioritisation. The section's remainder compares the obtained prefix tree to the available knowledge about address allocation policies in the network.

Data collection for linear uniform probing builds upon `yarrp`, similar to the campaign itself. This has the effect that any issues inherent to this tool, or traceroutes in general, will also occur in the benchmark data, enabling a fair comparison. A probing rate of 75

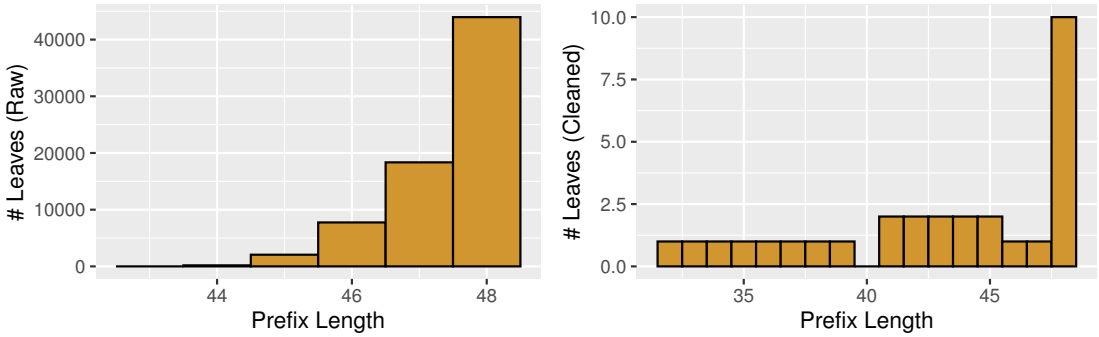


Figure 4.12: Prefix Length distribution for leaf nodes aggregated from benchmark measurement, raw (left) and cleaned of ICMP Rate Limiting artefacts (right).

packets per seconds is chosen to mimic the main campaign.[4] Again borrowing from the main experiment, a Rust program drives the measurement, however this time writing the result directly to a CSV file for later analysis. `/48` is the fixed target granularity for uniform probing, which strikes a balance between measurement time and detail. The measurement for this evaluation was started on March 21st, 2023, and lasted around 4.5 days. A benchmark on `/52` granularity would already take more than two months, infeasible for purposes for this evaluation.

131 072 `/48` prefix entries result from this measurement. Each has received 16 probes, of which nearly all were successful. In addition, the observed Last-Hop Routers from these probes are recorded. This information enables aggregation of the flat structure into a tree representing subnet structure of the organisation, combining neighbouring prefixes that share the same Last-Hop Router set.

An initial aggregation of the collected data suggests that the benchmark measurement is also affected by ICMP Rate Limiting artefacts. This can be assumed because raw results do not produce meaningful aggregates, which is known to be false due to the ground truth network plan. Figure 4.12, left shows an overview of the observed aggregation behaviour. Inspection of some prefixes that were expected to be merged reveals that 99.9% of `/48`s are served by some combination of two Last-Hop Routers, `B:1101:1007::1`[5] and `L:c1c:804a::2`. Notably, the latter never appears alone, suggesting it to be noise. Traceroute data (Figure 4.13) shows that it is in fact an upstream router, which is likely only visible if responses from the actual last hop are dropped due to ICMP Rate Limiting.

With this knowledge, these measurement artefacts can be removed by ignoring this

[4]The original idea here is that `AT-*` would find substantial structure, and evaluation could be performed primarily against these measurements. By evaluating against `U-3`, the benchmark receives a slight disadvantage with regard to ICMP Rate Limiting, as its probing rate is higher than the `yarrp` throughput of `U-3` (Table 3.3).

[5]Addresses in the following are stripped of their publicly-routable prefixes, with only local parts shown. This prevents information presented here from trivially revealing network structure. Compare section 3.3.

upstream router. The resulting network structure is shown in Figure 4.12, right. Much fewer leaves are discovered, and the observed topology is consistent with the network plan, albeit coarse-grained due to the `/48` aggregation. This technique cannot be 1:1 applied during measurement because the data needed for this qualitative analysis is not available to the application. It might however be possible to perform deduction based on a corpus of full traceroutes to the network, e.g. detecting that a specific router is upstream based on some metric. How this would work in practice remains unclear, and this question is thus deferred to future work. In addition, this information would need to be regularly invalidated. Despite running on the same host only two months apart, the concrete upstream router observed during rate limiting is different between `U-3` and the benchmark measurement.

The prefix structure resulting after this data cleaning is shown in Figure 4.14. The parts that it discovers are correct, but a substantial number of `/48`s are not identified. In particular, there should be four additional networks below `U::/40` and five in `U::100::/40`. Apart from this, nine networks are active below `U:1000::/40`, none of which are identified. A testing network at `U:4049::/48` is also missed, however this may be very difficult to discover due to the usage pattern. Overall, the (cleaned) benchmark measurement with 16 probes per `/48` is able to discover 20% of active `/48` networks, and does not falsely suggest any. Its confusion matrix is shown in Table 4.5.

This data suggests that Last-Hop-Router set measurements can in this case discover some prefix structure. It is possible that increasing the probe count might increase the true positive rate. Linear probing alone does not however suggest where such additional probes should be focused, making further discovery difficult because the entire prefix must be scanned. The tree generated from measurement data can serve as an indication where interesting networks may be located, similar to the primary algorithm. In this case, the jump from `/31` to `/48` is significant, and confidence in unused areas might be improved by introducing further increments, as suggested by [LaF15, RB20a].

Having obtained an overview of the benchmark algorithm's performance, it is now time to compare the prefix tree suggested by `U-3` to benchmark and ground truth. Overall, the algorithm discovers 31 Last-Hop Routers, whose different behaviour is summarised in Table 4.6. While the majority of observed *routers* is in-network, i.e. likely genuine, most *observations* are attributed to out-of-prefix routers in the `L'`[6] and `B` networks. The algorithm is able to uncover substantially more routers than the benchmark, which already suggests that more focused probing is beneficial. A difference in probe count does not explain this. Both `U-3` and the benchmark measurement by chance probed a very similar number of targets (2 095 023 vs. 2 097 152).[7]

[6] During the `U-*` measurements, the routing path shown in Figure 4.13 had two other prefixes in the `L` position, owned by the same upstream provider. These are referred to as `L'`.

[7] Half of these probes go towards the upper half of the measured `/31`, which seems to be entirely unused, putting the benchmark at a disadvantage in reality. Awareness of this fact however already is a result of the informed probing strategy. True linear probing would only know this after measurement has finished.

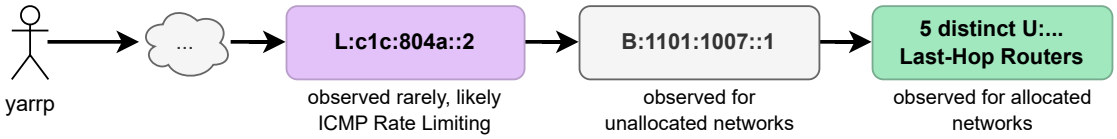


Figure 4.13: Observed routing path to the university network, informed by benchmark results and traceroutes. Publicly-routable prefixes are redacted by consistent letters. This is not a complete picture due to limitations of the benchmark (see Table 4.6).

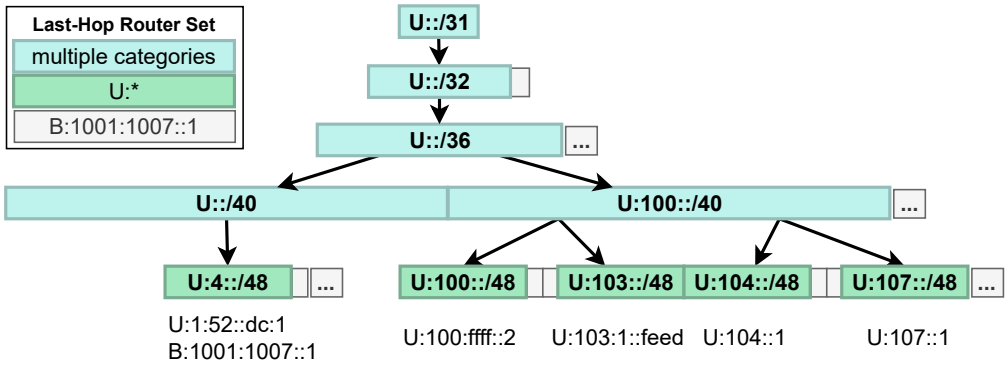


Figure 4.14: Prefix structure of the university network, as discovered by the bechmark measurement, on `/48` granularity, corrected for ICMP Rate Limiting artefacts. Seemingly unused networks (`B:1001:1007::1`) are only hinted at to improve legibility.

		Prediction (Benchmark)	
		In use	Not in use
Ground Truth	In use	5	19
	Not in use	0	131 048

Table 4.5: Confusion matrix for the uniform benchmark measurement. A network is considered "in use" by the benchmark if the observed cleaned Last-Hop Router set is different from `{ B:1001:1007::1 }`. The ground truth considers it in use if it is contained in the addressing plan.

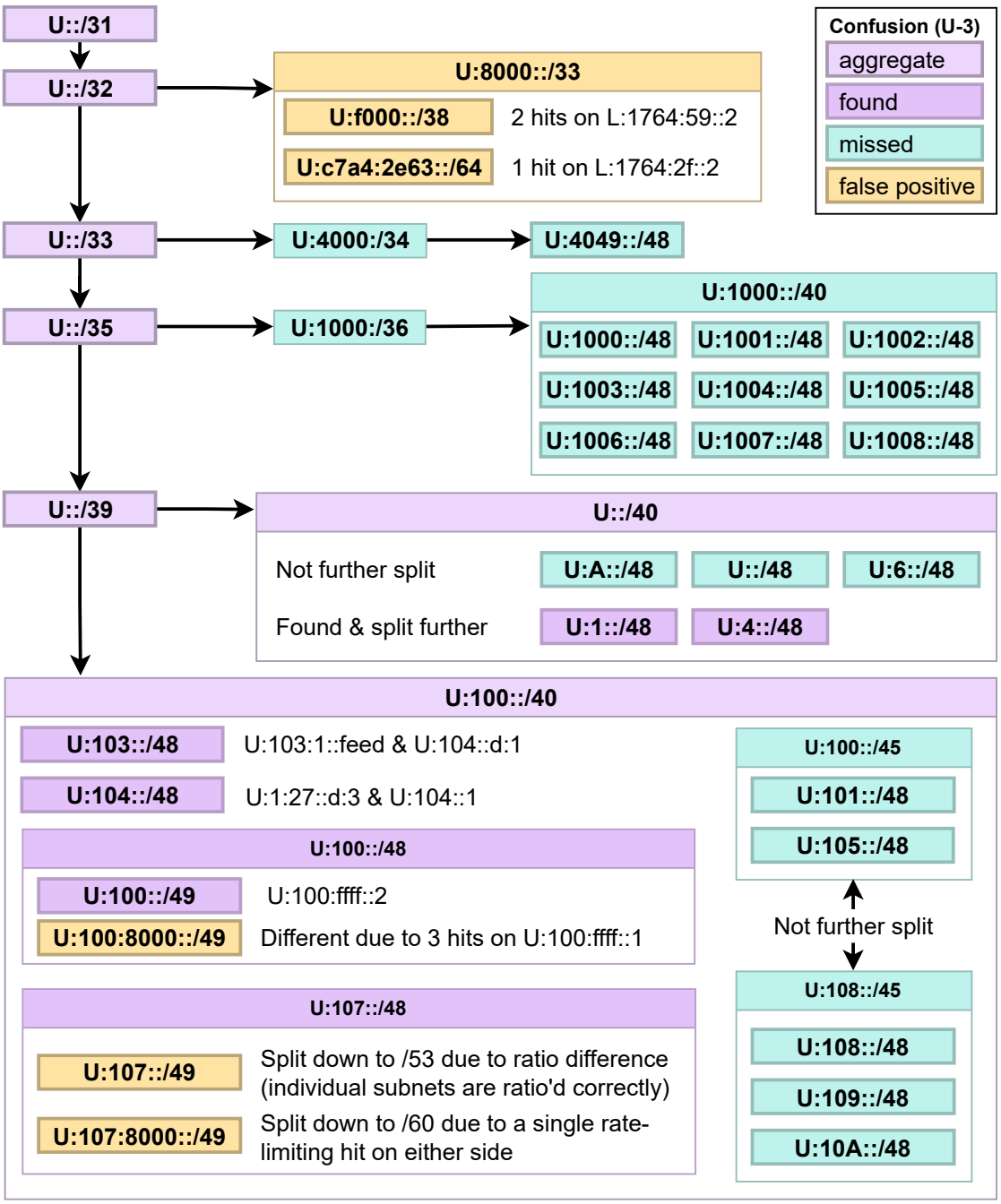


Figure 4.15: Prefix structure of the university network, as discovered by the `U-3` campaign, on `/48` granularity. Seemingly unused networks (`B:1001:1007::1` + 5% `L'` routers) are omitted to improve legibility. Confusion is indicated by colours, as explained in the legend on the top right.

Last-Hop Router(s)	#	#Hits	Remark
B:1001:1007::1	1	1 143 097	not allocated / border
B:1001:1007::2	1	7 818	next hop for U::/39
L′	3	64 422	upstream / rate limit
U:1:⋆	14	57 637	individual networks
U:4:⋆	6	1 057	individual networks
U:10x:⋆	4	21 580	individual networks
U:103:1::feed	1	286 217	U:103::/48

Table 4.6: Individual Last-Hop Routers observed in the U−⋆ measurement. The first section is out-of-prefix, while the remainder of Last-Hop Router *addresses* are located inside the university network. It is not clear why the last entry received such a substantial amount of traffic from the U−3 prefix tree. It is possible that that /48 was wrongly split in U−0 or U−2.

A comparison of the prefix tree generated by U−3 to the ground truth data is presented in Figure 4.15. False positives are an interesting aspect here, as in most of the shown cases, they are ”one-hit wonders“: caused by a single rate-limited probe. This behaviour has since been resolved by introducing an absolute minimum for ratio consideration in the Medium − Same heuristics. This fact motivates to disregard the false positives introduced in the subnets of U:8000::/33, as they would not occur with the current algorithm version. The artefacts inside the two /48 networks are not relevant for an analysis on /48 granularity, and their inclusion is for illustrative purposes only. Both containers are treated as found.

Overall, the proposed algorithm finds the same /40 aggregates as the benchmark measurement. On /48 granularity, it identifies all subnets found in the benchmark, suggesting that informed probing is not worse than uniform on this granularity. In addition, it finds significant structure inside U:1::/48, which the benchmark fails to identify. The confusion matrix is not repeated due to its high similarity to the benchmark (Table 4.5). The slightly increased true positive rate is 25%. This is a representation only of results at /48 granularity and does not take into account further discoveries the algorithm made at higher detail levels.

From the collected data, it is not possible to unambiguously identify why 18 false negatives remain. Manual inspection of these cases shows that no conflicting Last-Hop Router probe was received for any of them. In two cases, the erroneous *keep* decision occurred far up in the tree, which may suggest that the confidence threshold should be increased for large prefixes. However, with only such a small portion of the space being allocated, it remains plausible that the particular probes sent failed to guess the correct area, i.e. the algorithm was unlucky. A possible mitigation for this issue might be to randomly direct probes at large networks that are already at 255% keep confidence. This may also help with the remaining missed prefixes, which are mostly rejected at /48 granularity.

In these cases, a general increase of the keep threshold could also lead to improvements.

Overall, the benchmark shows that the algorithm performs on par with uniform probing at `/48` granularity, while additionally providing deeper structural insights. This is consistent with existing results in the literature [LaF15, RB20a] and suggests that structurally-informed probing is a viable technique, also if expanded to shift focus only to interesting areas[8].

Detailed discussion of the discovered tree is omitted due to space constraints. An overview of two `/40` networks is shown in Appendix B. Granular Analysis of `U-3` Splits.

[8]This is an important distinction. [LaF15, RB20a] only change measurement granularity but not the overall region that is targeted.

CHAPTER 5

Conclusion

Structure-aware probing is a promising idea that can help focus internet probes on more interesting areas of the search space. This work demonstrates that long-running measurements that feed back already-obtained knowledge into a prioritisation process can be successful. While there is still work to be done until this particular system can reliably scan the entire internet, results are in general promising, affirming successes from existing work with similar methods.

This chapter returns to the research questions postulated in chapter 1 and proposes answers based on the results obtained in chapter 4. Aspects that remain out of scope and might be interesting for future work are discussed in section 5.2. In particular, this includes adjusting the system to be more resilient against measurement artefacts, and scaling it to probe the entire IPv6 internet in practice.

5.1 Interpretation

Prefix Aggregation

1. *How can prefix aggregation be performed in a meaningful yet efficient manner?*

   The proposed method organises prefixes announced via BGP in a forest of *prefix trees*. Each leaf of the tree is evaluated for splitting using continuous measurement, building a reusable knowledge base of probe results. A split is made once a confidence metric indicates that the evidence is sufficient to suggest that the two direct subnets behave differently. This method relies on `zmap` [DWH13] and `yarrp` [Bev16] for measurement, and takes inspiration from previous structural probing efforts such as [LaF15, RB20b, LS16]. Technical throughput limitations and measurement artefacts still impede applicability, but the method is already shown to be at least as effective as linear uniform probing, while providing more meaningful insights with the same amount of probes.

2. *Is it likely that this information can be used to* ***enhance the scanning hit rate****?*

   The term "hit rate" is not directly meaningful to structural data. Comparison against linear probing shows that the proposed method can aid in unveiling more detailed structures in the search space than otherwise possible. There is a clear trade-off between breadth and depth of scanning. Reliable structural information, if obtainable, can be used to focus a limited probing budget towards regions with more structural features (depth) while still considering large search spaces overall (breadth).

Value Metric

3. *How can* ***more "valuable" (i.e. higher hit probability) target addresses*** *be discerned based on results of previous scans?*

   The proposed algorithm relies primarily on perceived structural properties to focus probing toward more interesting areas. This does not directly rate addresses. Results of previous scans can be used to build a *Last-Hop Router set* for each subnet of a prefix of interest. Comparison of these two sets can provide an intuition whether the subnets are structurally different. The proposed method introduces an algorithm that relies on probe results from both subnets to sort prefixes into *priority classes* that approximate how likely it is that additional information can be obtained by inspecting the subnets more closely.

4. *How can this "value metric" be* ***stored and updated*** *over multiple scans, ideally without significantly impairing scanning or analysis rate?*

   It is not feasible to dynamically recalculate an absolute priority metric for each candidate prefix in every round. Expected value of measuring a prefix again is approximated by the assignment of a priority class. Prefixes are not further ranked inside a class, but rather chosen arbitrarily for simplicity. Lottery ticket scheduling assigns the available probing budget to candidates of each class in proportion to the relative potential of the class.

5. *Is it possible to* ***enhance the hit rate*** *based on this "value metric"?*

   Results from a case study show that the informed probing performed by the proposed method is able to unveil additional prefix structure when compared to linear uniform probing. However, the true positive rate remains at a low 25%, with improvements not being substantial on a `/48` network granularity. Measurement artefacts such as ICMP Rate Limiting remain a challenge. Heuristics are introduced to limit the impact, but their effectiveness remains unknown. Confidence metrics used to reject regions as uninteresting require additional fine-tuning. In a real-world measurement, the algorithm is easily misled by unstable paths or similar artefacts. It may focus disproportionate measurement effort towards networks with fluctuating or diverse Last-Hop Routers.

Overall, the proposed method seems promising in general, but still tends to be confused by real-world behaviour. Existing heuristics do not seem to fully solve sensitivity to complex last-hop behaviour, suggesting that a conceptual expansion of the algorithm is necessary. This may for example involve collection of full traceroutes to detect ICMP Rate Limiting artefacts.

5.2 Future Work

Issues left open for future work can be broadly classified into two categories. On one hand, there are potential expansions of the general concept which may be interesting to investigate. On the other hand, shortcomings exist with the concrete implementation that are either parameter choices or of technical nature. In particular, while in theory suitable for internet-wide measurements, database performance is already an issue with the larger campaigns performed for this work.

For the general concept, a concrete issue observed during evaluation is resilience towards measurement artefacts. Despite parameter choices that should avoid ICMP Rate Limiting, artefacts of it are observed in the data. These disruptive factors likely cannot be avoided with slow probing rates, since limit budget consumed by others (measurements and actual users) is out of our control. It follows that the algorithm must be more resilient against it. While ignoring Last-Hop Routers with very few hits is a start, observations show that artefacts still occur. For example, the university campaign reports a consistent 5% upstream router leakage ratio across unallocated networks.

These leakage phenomena, which are the primary implication of ICMP Rate Limiting for this method, cannot be directly detected without additional information. As such, a promising expansion of the method may be to store not only the Last-Hop Router, but also upstream routers. This data could then potentially be used to decide whether observed last hops are from higher up in the trace. While this idea seems interesting, and is already possible since `yarrp` collects full traces as a by-product, concrete implementation will likely be more involved. The concept is primarily informed by behaviour of the university measurement and there may be cases where upstream routers are legitimately observed.

A related area of potential is general handling of cases where multiple last hops are observed in at least one subnet. While algorithm behaviour is clearly motivated in the remaining cases, this case may be caused by various different factors without an obvious correct choice. Evaluation shows that this is by far the most common case, so substantial improvements can be expected by a proper conceptual solution to this issue. Especially cases with a substantial number of distinct Last-Hop Routers, such as the highly-split AS from Figure 4.6 significantly confuse the current algorithm, often causing splits down to `/64`, and blocking substantial probing budget in the long term.

An interesting partial aspect, related to the full-trace suggestion, is whether the qualitative reasoning performed in the university case study could somehow be automated. If the algorithm were aware of the reverse-engineered semantics assigned to some Last-Hop

Router combinations, e.g. hash `e7c5` meaning unassigned, it may be able to better focus probing.

Grouping networks by fingerprints (= hash of Last-Hop Router set) may be a step in that direction. This could be combined with a modification of the binary split paradigm, as suggested by [LaF15, RB20a]. In practice, a larger number of subnets, for example 16, could be considered at once. When sufficient evidence is available, they are grouped by fingerprint and split accordingly. Evaluation suggests that many intermediate nodes are created by the binary splitting paradigm, which could be avoided if prefix granularity was more dynamic. An initial motivation for the binary split is that the decision only has two possible outcomes, which might aid in evaluation (but did not).

Another conceptual issue is that prefixes which have obtained maximum confidence are never re-probed. A motivation for this decision is that this simplifies scheduling, and higher confidence bounds may have a similar effect. Evaluation results suggest that this is likely detrimental to long-term quality of the observed structure. Allocation of a fixed budget to re-evaluation of maximum-confidence prefixes, ideally prioritising larger aggregates, might be a step in the right direction.

This consideration can be further expanded to raise the question of changes in the underlying topology, which are not accounted for in the current mechanism. A solution for this should consider a sustainable method for error correction. The algorithm is currently able to back-track when adjacent prefixes have the same fingerprint and priority class, but this mechanism is very limited, for example not being robust against repeated split-merge cycles. Such a solution needs to ensure that conflicting splits and merges do not occur in parallel on the same node when replicated.

In addition to these conceptual challenges, multiple known issues exist with the concrete implementation of the system. Tracing data collected by the observability stack suggests that circumstances exist where aggregation is a bottleneck, in particular identifying database query time as a culprit[1]. The naive solution deployed during measurement is to scale Aggregator with additional replicas. Five instances were needed at the end of `AT-11` to cope with aggregation requests, which suggests that this implementation will not reliably scale to full-internet measurements without performance improvements. Ideas include proper partitioning of workloads to enhance parallelism, introduction of a cache, or changes to the database schema.

Further improvements might be accomplished by questioning the usage patterns of `yarrp` and `zmap`. For example, the dedicated university measurement only received a negligible amount of responses to `zmap` probes. A mechanism could be introduced that decides for each BGP Root whether `zmap` probes usually receive responses, skipping directly to `yarrp` probing if not.

[1] Prefix aggregation was observed to occasionally block for ≈ 800 milliseconds on individual database queries, despite NVMe SSDs being in use. Experiences during evaluation show that an 8 GB memory limit for Postgres was accidentally left in place, which may be a factor in this behaviour. Increasing to 96 GB resulted in substantial performance gains with evaluation workloads.

Regarding `yarrp` in particular, the current setup produces complete traceroutes as a by-product. Improvements may be possible if the scope is reduced to explicitly target the Last-Hop Router, for example utilising the mechanism proposed in DoubleTree [DRFC05]. Additionally, the FlashRoute [HRAD20] expansion to `yarrp` might improve performance. It already relies on this idea.

Finally, prefix structure could be primed by importing additional existing measurement results. In particular, a CAIDA Ark [Cen20] importer comes to mind, which may help kickstart probing with historical data. Such an integration needs to take care that old data is properly invalidated or ignored.

List of Figures

List of Tables

Bibliography

[ACO+06] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding Traceroute Anomalies with Paris Traceroute. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 153–158, New York, NY, USA, 2006. Association for Computing Machinery. event-place: Rio de Janeriro, Brazil. doi:10.1145/1177080.1177100.

[ADSH14] David Adrian, Zakir Durumeric, Gulshan Singh, and J. Alex Halderman. Zippier ZMap: Internet-Wide Scanning at 10 Gbps. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*, San Diego, CA, August 2014. USENIX Association. URL: https://www.usenix.org/conference/woot14/workshop-program/presentation/adrian.

[AOR17] Pablo Alvarez, Florin Oprea, and John Rula. Rate-limiting of IPv6 traceroutes is widespread: measurements and mitigations. *Proc. IETF*, 99, 2017. URL: https://www.ietf.org/proceedings/99/slides/slides-99-maprg-rate-limiting-of-ipv6-traceroutes-is-widespread-measurements-and-mitigations-01.pdf.

[BAF+21] Kevin Bock, Abdulrahman Alaraj, Yair Fax, Kyle Hurley, Eric Wustrow, and Dave Levin. Weaponizing Middleboxes for TCP Reflected Amplification. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3345–3361. USENIX Association, August 2021. URL: https://www.usenix.org/conference/usenixsecurity21/presentation/bock.

[Bak95] Fred Baker. Requirements for IP Version 4 Routers, June 1995. Issue: 1812 Num Pages: 175 Series: Request for Comments Published: RFC 1812. URL: https://www.rfc-editor.org/info/rfc1812, doi:10.17487/RFC1812.

[BBLR13] R Beverly, W Brinkmeyer, M Luckie, and J Rohrer. IPv6 Alias Resolution via Induced Fragmentation. In *Passive and Active Network Measurement Workshop (PAM)*, pages 158–167, March 2013. doi:10.1007/978-3-642-36516-4_16.

[BCK06] Steven M. Bellovin, Bill Cheswick, and Angelos D. Keromytis. Worm Propagation Strategies in an IPv6 Internet. *login Usenix Mag.*, 31, 2006. URL: https://www.cs.columbia.edu/~smb/papers/v6worms.pdf.

[BDKM12] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. The Menlo Report. *IEEE Security & Privacy*, 10:71–75, March 2012. URL: https://www.dhs.gov/sites/default/files/publications/CSD-MenloPrinciplesCORE-20120803_1.pdf, doi:10.1109/MSP.2012.52.

[BDPR18] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P. Rohrer. In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery. In *Proceedings of the Internet Measurement Conference 2018*, IMC '18, pages 308–321, New York, NY, USA, 2018. Association for Computing Machinery. event-place: Boston, MA, USA. doi:10.1145/3278532.3278559.

[Bev16] Robert Beverly. Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pages 413–420, New York, NY, USA, 2016. Association for Computing Machinery. event-place: Santa Monica, California, USA. doi:10.1145/2987443.2987479.

[BF13] Thomas Bourgeau and Timur Friedman. Efficient IP-Level Network Topology Capture. In Matthew Roughan and Rocky Chang, editors, *Passive and Active Measurement*, pages 11–20, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. URL: https://hal.inria.fr/file/index/docid/827174/filename/PAM2013-NTC.pdf.

[BHFV18] Kevin Borgolte, Shuang Hao, Tobias Fiebig, and Giovanni Vigna. Enumerating Active IPv6 Hosts for Large-Scale Security Scans via DNSSEC-Signed Reverse Zones. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 770–784, May 2018. ISSN: 2375-1207. doi:10.1109/SP.2018.00027.

[CAZ+14] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. Measuring IPv6 Adoption. In *Proceedings of the 2014 ACM Conference on SIGCOMM*, SIGCOMM '14, pages 87–98, New York, NY, USA, 2014. Association for Computing Machinery. event-place: Chicago, Illinois, USA. doi:10.1145/2619239.2626295.

[CC09] Rebecca L. Collins and Luca P. Carloni. Flexible filters: load balancing through backpressure for stream programs. In *Proceedings of the Seventh ACM International Conference on Embedded Software*, EMSOFT '09, pages 205–214, New York, NY, USA, 2009. Association for Computing Machinery. event-place: Grenoble, France. doi:10.1145/1629335.1629363.

[CCG+15] Brian E. Carpenter, Tim Chown, Fernando Gont, Sheng Jiang, Alexandre Petrescu, and Andrew Yourtchenko. Analysis of the 64-bit Boundary in IPv6 Addressing, January 2015. Issue: 7421 Num Pages: 24 Series: Request for Comments Published: RFC 7421. URL: https://www.rfc-editor.org/info/rfc7421, doi:10.17487/RFC7421.

[Cen20] Center for Applied Internet Data Analysis. Archipelago (Ark) Measurement Infrastructure, May 2020. URL: https://www.caida.org/projects/ark/.

[Cho08] Tim Chown. IPv6 Implications for Network Scanning, March 2008. Issue: 5157 Num Pages: 13 Series: Request for Comments Published: RFC 5157. URL: https://www.rfc-editor.org/info/rfc5157, doi:10.17487/RFC5157.

[CMC+16] Italo Cunha, Pietro Marchetta, Matt Calder, Yi-Ching Chiu, Bruno V. A. Machado, Antonio Pescapè, Vasileios Giotsas, Harsha V. Madhyastha, and Ethan Katz-Bassett. Sibyl: A Practical Internet Route Oracle. pages pp. 325–344, Santa Clara, CA, March 2016. USENIX Association. URL: https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-cunha.pdf.

[DAM+15] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, CCS '15, pages 542–553, New York, NY, USA, 2015. Association for Computing Machinery. event-place: Denver, Colorado, USA. doi:10.1145/2810103.2813703.

[DC98] Dr Steve E. Deering and Alex Conta. Generic Packet Tunneling in IPv6 Specification, December 1998. Issue: 2473 Num Pages: 36 Series: Request for Comments Published: RFC 2473. URL: https://www.rfc-editor.org/info/rfc2473, doi:10.17487/RFC2473.

[Dep79] Department of Health, Education, and Welfare. Belmont Report, 1979. URL: https://www.hhs.gov/ohrp/sites/default/files/the-belmont-report-508c_FINAL.pdf.

[DH06] Dr Steve E. Deering and Bob Hinden. IP Version 6 Addressing Architecture, February 2006. Issue: 4291 Num Pages: 25 Series: Request for Comments Published: RFC 4291. URL: https://www.rfc-editor.org/info/rfc4291, doi:10.17487/RFC4291.

[DH17] Dr Steve E. Deering and Bob Hinden. Internet Protocol, Version 6 (IPv6) Specification, July 2017. Issue: 8200 Num Pages: 42 Series: Request for Comments Published: RFC 8200. URL: https://www.rfc-editor.org/info/rfc8200, doi:10.17487/RFC8200.

[DRFC05] Benoit Donnet, Philippe Raoult, Timur Friedman, and Mark Crovella. Efficient Algorithms for Large-Scale Topology Discovery. In *SIGMETRICS 2005 - 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 327–338, Banff, Alberta, Canada, June 2005. ACM. URL: https://hal.archives-ouvertes.fr/hal-01491665, doi:10.1145/1064212.1064256.

[DWH13] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *22nd USENIX Security Symposium (USENIX Security 13)*, pages 605–620, Washington, D.C., August 2013. USENIX Association. URL: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric.

[FBH+17] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. Something from Nothing (There): Collecting Global IPv6 Datasets from DNS. pages 30–43, February 2017. doi:10.1007/978-3-319-54328-4_3.

[FBH+18] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, Giovanni Vigna, and Anja Feldmann. In rDNS We Trust: Revisiting a Common Data-Source's Reliability. In *Proceedings of the 19th Passive and Active Measurement Conference*, PAM, March 2018. URL: https://escholarship.org/uc/item/6822t9g9.

[FLVY93] Vince Fuller, Tony Li, Kannan Varadhan, and Jessica Yu. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy, September 1993. Issue: 1519 Num Pages: 24 Series: Request for Comments Published: RFC 1519. URL: https://www.rfc-editor.org/info/rfc1519, doi:10.17487/RFC1519.

[FPB16] Pawel Foremski, David Plonka, and Arthur Berger. Entropy/IP: Uncovering Structure in IPv6 Addresses. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pages 167–181, New York, NY, USA, 2016. ACM. event-place: Santa Monica, California, USA. URL: http://doi.acm.org/10.1145/2987443.2987445, doi:10.1145/2987443.2987445.

[Gas17] Eric W. Gaston. High-frequency mapping of the IPv6 Internet using Yarrp. Master's thesis, Naval Postgraduate School, Monterey, California, US, March 2017. URL: https://hdl.handle.net/10945/52982.

[GC06] Mukesh Gupta and Alex Conta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, March 2006. Issue: 4443 Num Pages: 24 Series: Request for Comments Published: RFC 4443. URL: https://www.rfc-editor.org/info/rfc4443, doi:10.17487/RFC4443.

[GC16] Fernando Gont and Tim Chown. Network Reconnaissance in IPv6 Networks, March 2016. Issue: 7707 Num Pages: 38 Series: Request for Comments Published: RFC 7707. URL: https://www.rfc-editor.org/info/rfc7707, doi:10.17487/RFC7707.

[GD20] Jean-François Grailet and Benoit Donnet. Virtual Insanity: Linear Subnet Discovery. *IEEE Transactions on Network and Service Management*, 17(2):1268–1281, 2020. doi:10.1109/TNSM.2020.2976859.

[GKF+20] Vasileios Giotsas, Thomas Koch, Elverton Fazzion, Ítalo Cunha, Matt Calder, Harsha V. Madhyastha, and Ethan Katz-Bassett. Reduce, Reuse, Recycle: Repurposing Existing Measurements to Identify Stale Traceroutes. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, pages 247–265, New York, NY, USA, 2020. Association for Computing Machinery. event-place: Virtual Event, USA. doi:10.1145/3419394.3423654.

[GS07] Mehmet H. Gunes and Kamil Sarac. Inferring subnets in router-level topology collection studies. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, IMC '07, pages 203–208, New York, NY, USA, 2007. Association for Computing Machinery. event-place: San Diego, California, USA. doi:10.1145/1298306.1298334.

[GSF+18] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. 2018. Publisher: arXiv. URL: https://arxiv.org/abs/1806.01633, doi:10.48550/ARXIV.1806.01633.

[GSGC16] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. *Proc. of 8th Int. Workshop on Traffic Monitoring and Analysis*, July 2016. URL: https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/gasser2016ipv6hitlist.pdf.

[GT00] R. Govindan and H. Tangmunarunkit. Heuristics for Internet map discovery. In *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, volume 3, pages 1371–1380 vol.3, March 2000. ISSN: 0743-166X. doi:10.1109/INFCOM.2000.832534.

[Hol20] Florian Holzbauer. IPv6-Reconnaissance. Master's thesis, FH St. Pölten, St. Pölten, 2020. URL: https://phaidra.fhstp.ac.at/o:4272.

[HRAD20] Yuchen Huang, Michael Rabinovich, and Rami Al-Dalky. FlashRoute: Efficient Traceroute on a Massive Scale. In *Proceedings of the ACM Internet Measurement Conference*, IMC '20, pages 443–455, New York, NY, USA,

2020. Association for Computing Machinery. event-place: Virtual Event, USA. doi:10.1145/3419394.3423619.

[HS99] Matt Holdrege and Pyda Srisuresh. IP Network Address Translator (NAT) Terminology and Considerations, August 1999. Issue: 2663 Num Pages: 30 Series: Request for Comments Published: RFC 2663. URL: https://www.rfc-editor.org/info/rfc2663, doi:10.17487/RFC2663.

[IAB01] IAB. IAB/IESG Recommendations on IPv6 Address Allocations to Sites, September 2001. Issue: 3177 Num Pages: 10 Series: Request for Comments Published: RFC 3177. URL: https://www.rfc-editor.org/info/rfc3177, doi:10.17487/RFC3177.

[Int] Internet Assigned Numbers Authority. Number Resources. Accessed: 2024-03-12. URL: https://www.iana.org/numbers.

[Jac00] Van Jacobson. traceroute, September 2000. Version 1.4. Originally released on 1988-12-20. Accessed: 2024-01-29. URL: ftp://ftp.ee.lbl.gov/traceroute.tar.gz.

[KGO12] Hakan Kardes, Mehmet Gunes, and Talha Oz. Cheleby: A subnet-level internet topology mapping system. In *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*, pages 1–10, 2012. doi:10.1109/COMSNETS.2012.6151326.

[KK10] Seiichi Kawamura and Masanobu Kawashima. A Recommendation for IPv6 Address Text Representation, August 2010. Issue: 5952 Num Pages: 14 Series: Request for Comments Published: RFC 5952. URL: https://www.rfc-editor.org/info/rfc5952, doi:10.17487/RFC5952.

[KLPS02] Eddie Kohler, Jinyang Li, Vern Paxson, and Scott Shenker. Observed Structure of Addresses in IP Traffic. pages 253–266. ACM Press, 2002. URL: http://conferences.sigcomm.org/imc/2002/imw2002-papers/189.pdf.

[KLWR16] Johannes Klick, Stephan Lau, Matthias Wählisch, and Volker Roth. Towards Better Internet Citizenship: Reducing the Footprint of Internet-Wide Scans by Topology Aware Prefix Selection. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pages 421–427, New York, NY, USA, 2016. Association for Computing Machinery. event-place: Santa Monica, California, USA. doi:10.1145/2987443.2987457.

[Kuk16] Christoph Kukovic. *IPv6 High Performance Scanning*. Diploma Thesis, TU Wien, Wien, 2016. URL: https://doi.org/10.34726/hss.2016.35201.

[KW00] Balachander Krishnamurthy and Jia Wang. On Network-Aware Clustering of Web Clients. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '00, pages 97–110, New York, NY, USA, 2000. Association for Computing Machinery. event-place: Stockholm, Sweden. doi:10.1145/347059.347412.

[LaF15] Blake W. LaFever. Methods for intelligent mapping of the IPv6 address space, March 2015. URL: https://hdl.handle.net/10945/48133.

[LHH08] Matthew Luckie, Young Hyun, and Bradley Huffaker. Traceroute probe method and forward IP path inference. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, IMC '08, pages 311–324, New York, NY, USA, 2008. Association for Computing Machinery. event-place: Vouliagmeni, Greece. doi:10.1145/1452520.1452557.

[LS16] Youndo Lee and Neil Spring. Identifying and Aggregating Homogeneous IPv4 /24 Blocks with Hobbit. In *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pages 151–165, New York, NY, USA, 2016. Association for Computing Machinery. event-place: Santa Monica, California, USA. doi:10.1145/2987443.2987448.

[Mai21] Markus Maier. *Investigating Router Misconfigurations on the IPv6 Internet*. Diplomarbeit, TU Wien, Wien, October 2021. URL: https://doi.org/10.34726/hss.2021.77442.

[Mal08] David Malone. Observations of IPv6 Addresses. pages 21–30, April 2008. doi:10.1007/978-3-540-79232-1_3.

[MD99] Pedro R. Marques and Francis Dupont. Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing, March 1999. Issue: 2545 Num Pages: 5 Series: Request for Comments Published: RFC 2545. URL: https://www.rfc-editor.org/info/rfc2545, doi:10.17487/RFC2545.

[MLB+17] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. Target Generation for Internet-Wide IPv6 Scanning. In *Proceedings of the 2017 Internet Measurement Conference*, IMC '17, pages 242–253, New York, NY, USA, 2017. Association for Computing Machinery. event-place: London, United Kingdom. doi:10.1145/3131365.3131405.

[Nar10] Dr Thomas Narten. On the Scalability of Internet Routing. Internet-Draft draft-narten-radir-problem-statement-05, Internet Engineering Task Force, February 2010. Backup Publisher: Internet Engineering Task Force Num Pages: 25. URL: https://datatracker.ietf.org/doc/draft-narten-radir-problem-statement/05/.

[NNSA07] George Neville-Neil, Pekka Savola, and Joe Abley. Deprecation of Type 0 Routing Headers in IPv6, December 2007. Issue: 5095 Num Pages: 7 Series: Request for Comments Published: RFC 5095. URL: https://www.rfc-editor.org/info/rfc5095, doi:10.17487/RFC5095.

[PA16] Craig Partridge and Mark Allman. Ethical considerations in network measurement papers. *Commun. ACM*, 59(10):58–64, September 2016. Place: New York, NY, USA Publisher: Association for Computing Machinery. doi:10.1145/2896816.

[PB15] David Plonka and Arthur Berger. Temporal and Spatial Classification of Active IPv6 Addresses. In *Proceedings of the 2015 Internet Measurement Conference*, IMC '15, pages 509–522, New York, NY, USA, 2015. Association for Computing Machinery. event-place: Tokyo, Japan. doi:10.1145/2815675.2815678.

[PG98] Jean-Jacques Pansiot and Dominique Grad. On Routes and Multicast Trees in the Internet. *SIGCOMM Comput. Commun. Rev.*, 28(1):41–50, January 1998. Place: New York, NY, USA Publisher: Association for Computing Machinery. doi:10.1145/280549.280555.

[RB20a] Erik C. Rye and Robert Beverly. Discovering the IPv6 Network Periphery. In Anna Sperotto, Alberto Dainotti, and Burkhard Stiller, editors, *Passive and Active Measurement*, pages 3–18, Cham, 2020. Springer International Publishing. (shortened 16-page conference paper). URL: https://doi.org/10.1007/978-3-030-44081-7_1.

[RB20b] Erik C. Rye and Robert Beverly. Discovering the IPv6 Network Periphery, 2020. (full 19-page version). URL: https://arxiv.org/abs/2001.08684, doi:10.48550/ARXIV.2001.08684.

[RBC21] Erik Rye, Robert Beverly, and K C Claffy. Follow the Scent: Defeating IPv6 Prefix Rotation Privacy. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, pages 739–752, New York, NY, USA, 2021. Association for Computing Machinery. event-place: Virtual Event. doi:10.1145/3487552.3487829.

[RHL06] Yakov Rekhter, Susan Hares, and Tony Li. A Border Gateway Protocol 4 (BGP-4), January 2006. Issue: 4271 Num Pages: 104 Series: Request for Comments Published: RFC 4271. URL: https://www.rfc-editor.org/info/rfc4271, doi:10.17487/RFC4271.

[RHN11] Rosalea Roberts, Geoff Huston, and Dr Thomas Narten. IPv6 Address Assignment to End Sites, March 2011. Issue: 6177 Num Pages: 9 Series: Request for Comments Published: RFC 6177. URL: https://www.rfc-editor.org/info/rfc6177, doi:10.17487/RFC6177.

[RL23] Erik Rye and Dave Levin. IPv6 Hitlists at Scale: Be Careful What You Wish For. In *Proceedings of the ACM SIGCOMM 2023 Conference*, ACM SIGCOMM '23, pages 904–916, New York, NY, USA, 2023. Association for Computing Machinery. event-place: New York, NY, USA. doi:10.1145/3603269.3604829.

[RLB16] Justin P. Rohrer, Blake LaFever, and Robert Beverly. Empirical Study of Router IPv6 Interface Address Distributions. *IEEE Internet Computing*, 20(4):36–45, July 2016. doi:10.1109/MIC.2016.52.

[SKZ+23] Lion Steger, Liming Kuang, Johannes Zirngibl, Georg Carle, and Oliver Gasser. Target Acquired? Evaluating Target Generation Algorithms for IPv6. In *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA)*, June 2023. Place: Naples, Italy. URL: http://www.net.in.tum.de/fileadmin/bibtex/publications/papers/steger2023targetacquired.pdf.

[SNNS07] William A. Simpson, Dr Thomas Narten, Erik Nordmark, and Hesham Soliman. Neighbor Discovery for IP version 6 (IPv6), September 2007. Issue: 4861 Num Pages: 97 Series: Request for Comments Published: RFC 4861. URL: https://www.rfc-editor.org/info/rfc4861, doi:10.17487/RFC4861.

[SPV21] Shreyas Srinivasa, Jens Myrup Pedersen, and Emmanouil Vasilomanolakis. Open for Hire: Attack Trends and Misconfiguration Pitfalls of IoT Devices. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, pages 195–215, New York, NY, USA, 2021. Association for Computing Machinery. event-place: Virtual Event. doi:10.1145/3487552.3487833.

[SYW+22] Guanglei Song, Jiahai Yang, Zhiliang Wang, Lin He, Jinlei Lin, Long Pan, Chenxin Duan, and Xiaowen Quan. DET: Enabling Efficient Probing of IPv6 Active Addresses. *IEEE/ACM Transactions on Networking*, 30(4):1629–1643, August 2022. doi:10.1109/TNET.2022.3145040.

[UKKW15] Johanna Ullrich, Peter Kieseberg, Katharina Krombholz, and Edgar Weippl. On Reconnaissance with IPv6: A Pattern-Based Scanning Approach. In *2015 10th International Conference on Availability, Reliability and Security*, pages 186–192, August 2015. doi:10.1109/ARES.2015.48.

[WCVY03] Dan Waddington, Fangzhe Chang, Ramesh Viswanathan, and Bin Yao. Topology discovery for public IPv6 networks. *Computer Communication Review*, 33:59–68, July 2003. doi:10.1145/956993.957001.

[YKT05] Rui Yang, Panos Kalnis, and Anthony K. H. Tung. Similarity evaluation on tree-structured data. In *Proceedings of the 2005 ACM SIGMOD International*

Conference on Management of Data, SIGMOD '05, pages 754–765, New York, NY, USA, 2005. Association for Computing Machinery. event-place: Baltimore, Maryland. URL: https://www.comp.nus.edu.sg/~atung/publication/treematch.pdf, doi:10.1145/1066157.1066243.

[YYZ23] Wei Yao, Hai Yao, and Jing-Jing Zhao. Discovering and Mapping Subnet Level Topology. *Journal of Internet Technology*, 24:291–303, March 2023. URL: https://jit.ndhu.edu.tw/article/download/2869/2894, doi:10.53106/160792642023032402008.

[ZSS+22] Johannes Zirngibl, Lion Steger, Patrick Sattler, Oliver Gasser, and Georg Carle. Rusty Clusters? Dusting an IPv6 Research Foundation. page 15, Nice, France, 2022. ACM. doi:10.1145/3517745.3561440.

[ZTL10] Lixia Zhang, Dave Thaler, and Gregory M. Lebovitz. IAB Thoughts on IPv6 Network Address Translation, July 2010. Issue: 5902 Num Pages: 15 Series: Request for Comments Published: RFC 5902. URL: https://www.rfc-editor.org/info/rfc5902, doi:10.17487/RFC5902.

Appendices

Appendix A. Reproduction Instructions

This appendix describes how the data analysed in chapter 4 was obtained, in the hopes that it may be useful for reproduction. Any file paths refer to the `evaluation` directory in the `prefix-crab` monorepo that contains the thesis' implementation.

Configurations used to run the measurements themselves are version-controlled in the same `git` repository. Deployment is documented in the `deploy` directory. The observability stack relies on an external *Kubernetes* cluster, where the *Helm* charts located in `deploy/observability-stack` are installed.

1. **Figure 4.1** is generated from `sql/pfxc-eval-a.sql`, *A1 Result*. The chart is generated by LibreOffice in a straightforward way.

2. **Figure 4.2** is based upon `sql/pfxc-eval-c.sql`, *C1 final*. The roots are grouped by prefix length using a Pivot Table in LibreOffice, producing the chart.

3. **Table 4.2** is a programmatic evaluation generated by Crab Tools like `crab-tools hit-count OUTFILE.csv`.

4. **Information about subnet relationships between BGP Roots on page 48** stems from `sql/pfxc-eval-a.sql`, *Subnet Relationship Detection*.

5. **Figure 4.3 and Figure 4.4** are the PDF files generated by `rstats/hitcounts.R` (in order 1-3-2), operating on the output of item 3.

6. **Figure 4.5 and Figure 4.6** are generated by *Grafana v10.3* and the *Grafana Image Renderer* companion service. Each image was produced by using the panel context menu (top right) and `> Share > Link > Direct link rendered image`. The query parameters of the resulting URL were adapted to include `&scale=8&width=1000&height=350&theme=light`. An export of the original dashboard is located at `grafana/prefix-crab.dashboard.json`.

7. **The suggestion that Figure 4.5a mimics logistic growth** is obtained by manually overlaying the exported graph with a logistic growth curve in *GeoGebra* with $f(x) := \frac{L}{1+e^{-(k(x-x_0))}}$ where $L := 30, k := 0.07, x_0 := 40$. The lower two corners of the images are placed in the coordinate system at the origin and $(120, 0)$.

8. **Figure 4.7** follows from `rstats/b2_confidence.R`, `PerNodeCount.pdf`.

9. **Table 4.3 and Table 4.4** build upon another programmatic evaluation generated by Crab Tools as `crab-tools edge-analyse ::/0 OUTFILE.csv`. This raw data is further processed using `rstats/e_flappy_analyses.R`. Both tables summarise the content of the generated text files.

10. **Figure 4.8, Figure 4.9, Figure 4.10, and Figure 4.11** again use Crab Tools via `crab-tools tree-compare ::/0 OUTFILE.csv`. The university data is collected separately by limiting to its prefix, while the remainder is derived from the full-tree data. Analysis and processing for this aspect is implemented in `rstats/f_ftree_compare.R`. Manual AS filtering for the similarity chart in Figure 4.8 checks whether all nodes for an AS are of type `unsplit_root`, or merged back to such.

11. **The uniform measurement relied upon as ground truth for Case Study: University** builds upon the purpose-built Yarrp Evaluator modules via `yarrp-evaluator --subnet-size=48 U::/31 /scans/2024_03_prefix-crab-evaluation/G-university-48s/2024-03-21_university48s.csv`.

12. **Aggregation of this data into a minimal tree (Figure 4.14)** further processes this data with `crab-tools uniform-merge 2024-03-21_university48s.csv uni_merged_tree.csv`.

13. **The statement that `U-3` and the benchmark measurement probed a similar number of targets on page 64** calculates the benchmark target count as $2^{48-31} * 15$, and the `U-3` target count is taken from `crab-tools hit-count`. Its accounting is based on the probe archive, so it does take into account data left over from `U-0` and `U-2`. `U-3` additionally issued 7.5 million `zmap` probes, but only a few thousand of these received a response (errors only), so they are unlikely to have contributed in any meaningful way.

14. **Network structure comparisons (Figure 4.15, Figure I, Figure II)** rely on manual inspection of the observed tree with `crab-tools prefix-inspect U::/31`. This data is compared against the network plan of allocated networks, which is available for different regions at varying granularity, as indicated in the figures. Table 4.6 is extracted directly from Crab Tools and Table 4.5 stems from the visualisations.

15. **Statistical analysis of both raw and aggregated benchmark results** is implemented in `rstats/g_university_check.R`.

Appendix B. Granular Analysis of `U-3` Splits

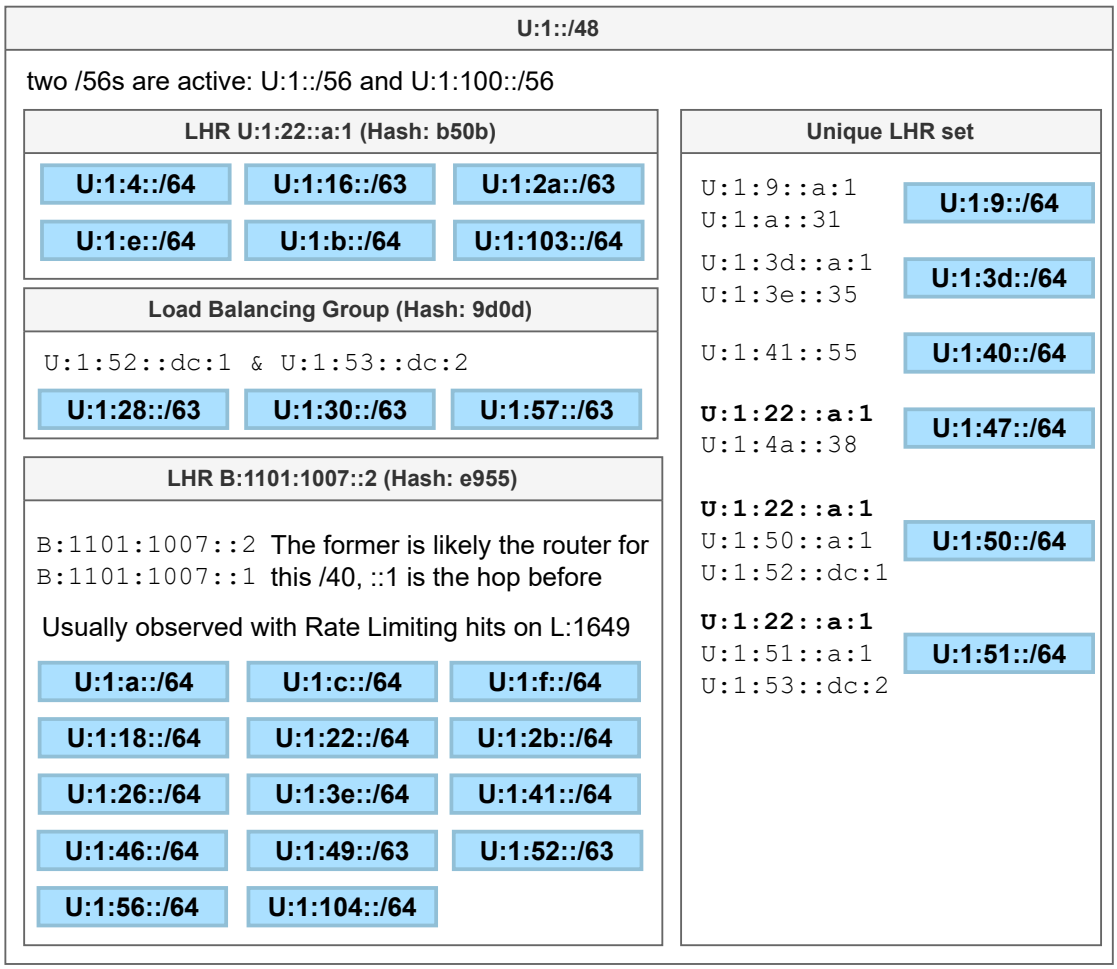


Figure I: Prefix structure of `U:1::/48`, as measured by the algorithm in `U-3`. No granular ground truth is available for this portion of the network.

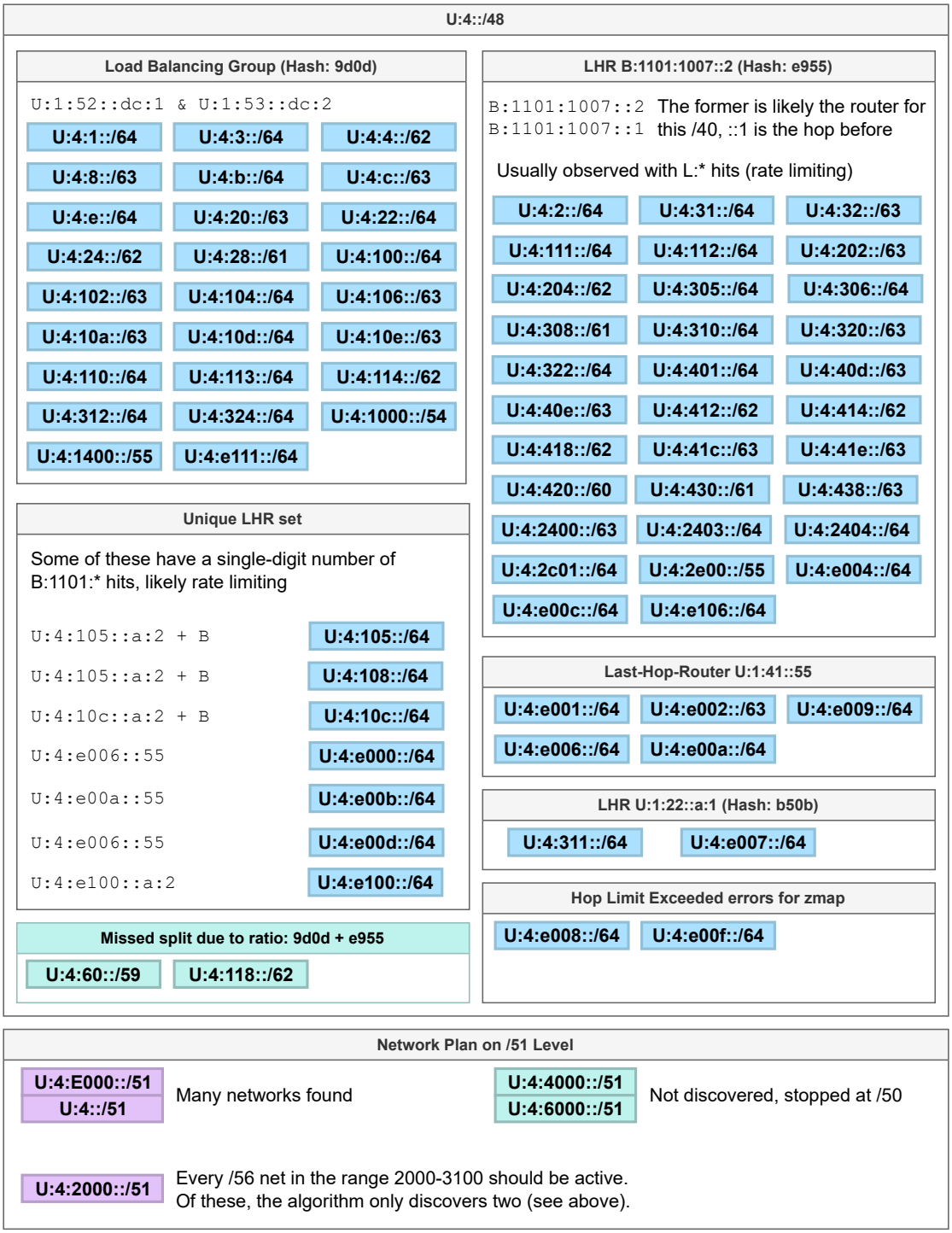


Figure II: Prefix structure of `U:4::/48`, as measured by the algorithm in `U-3`. No `/64`-level ground truth is available for this portion of the network. The lower box is based on information at `/51` granularity, and `/56` for a single network.

Appendix C. Split Behaviour Graphs

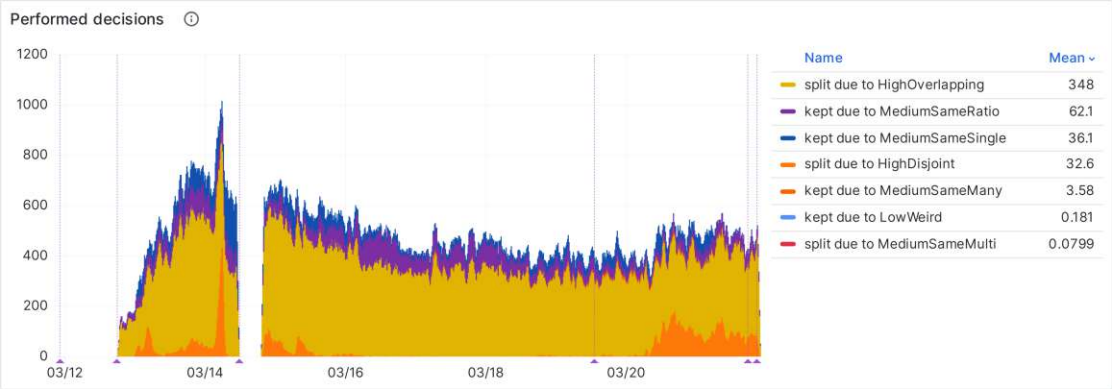


Figure III: Number of decisions actually executed per hour during the `AT-11` measurement. Keep decisions count as executed once they reach the 255% threshold. Gaps are due to delayed start of data collection and an outage of the observability system.

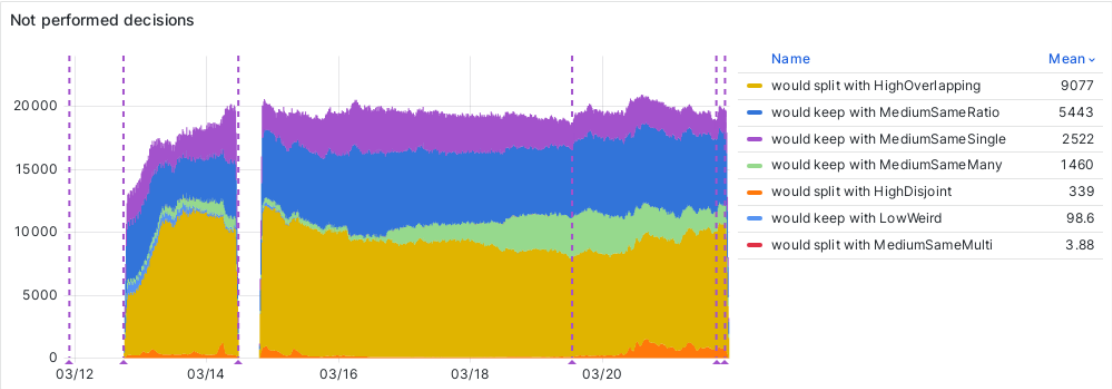


Figure IV: Number of analyses that led to decision that was not executed, per hour, during the `AT-11` measurement. Execution is deferred if confidence is below the relevant threshold. Gaps are due to delayed start of data collection and an outage of the observability system.

Appendix D. Confidence Distribution of Leaves

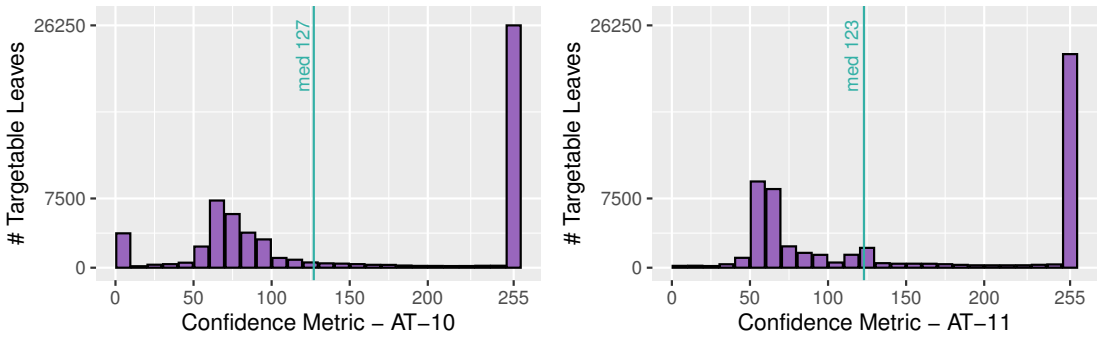


Figure V: Histogram of leaf counts per confidence value. This excludes `/64` leaves, which cannot be further targeted for splits since they are already at maximum granularity.