TU WIEN Informatics

Investigating Router Misconfigurations on the IPv6 Internet

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Markus Maier, BSc.
Matrikelnummer 01426091

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.-Prof. Dipl.-Ing. Mag. Dr. techn. Edgar Weippl
Mitwirkung: Dipl.-Ing. Dr.techn. Johanna Ullrich

Wien, 14. Oktober 2021

Markus Maier	Edgar Weippl

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

TU WIEN Informatics

Investigating Router Misconfigurations on the IPv6 Internet

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Markus Maier, BSc.
Registration Number 01426091

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.-Prof. Dipl.-Ing. Mag. Dr. techn. Edgar Weippl
Assistance: Dipl.-Ing. Dr.techn. Johanna Ullrich

Vienna, 14th October, 2021

Markus Maier	Edgar Weippl

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

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Erklärung zur Verfassung der Arbeit

Markus Maier, 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, 14. Oktober 2021

Markus Maier

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Danksagung

Ich möchte mich herzlich bei Johanna Ullrich für die Idee, die kontinuierliche Mitwirkung bei der Entwicklung sowie jeglicher Unterstützung während der Arbeit bedanken. Ebenfalls herzlich bedanken möchte ich mich bei meinem Betreuer Edgar Weippl.

Ebenfalls möchte ich mich bei SBA Research, für die Chance und Unterstützung an diesem Thema während meiner Anstellung forschen zu dürfen, bedanken.

Danke auch an meine Freunde und Arbeitskollegen die mich während der Arbeit begleitet haben. Alle Tipps und Tricks, Aufmunterungen und Ablenkungen von wichtigen Dingen sind sehr gut angekommen und werden mir immer in Erinnerung bleiben!

Ein großes Danke geht an meine Familie, die bei jedem Schritt während des gesamten Studiums und der Arbeit hinter mir gestanden sind. Danke für ihre Zeit, Geduld und Kraft die mich durch die letzten Jahre gebracht haben.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Kurzfassung

Das mittlerweile ausgereifte Protokoll IPv6, sowie dessen Vorgänger IPv4, bieten einer immer größer werdenen Anzahl an Geräten eine stabile Netzwerk Netzwerkverbindung. Unabhängig von dem Protokoll sind Stabilität und eine hohe Verfügbarkeit aller Netze von entscheidender Bedeutung und stellen Internetanbieter vor eine tägliche Herausforderung. Obwohl sich das Internet ständig weiterentwickelt, wurde ein Aspekt, der sich auf die Stabilität auswirken kann, im letzten Jahrzehnt jedoch übersehen. Bei im Vorfeld durchgeführten IPv6-Messungen wurde eine hohe Anzahl an ICMP Hop Limit Exceeded-Fehlern festgestellt. Sie sind ein starker Hinweis auf Routing Loops, bei denen einzelne Pakete immer und immer wieder im Kreis gesendet werden. Die letzte dedizierte Messung für persistente Routing Loops im Internet geht in das Jahr 2007 zurück.

In dieser Arbeit werfen wir einen Blick auf öffentlich verfügbare Messungen und führen unsere eigenen Messungen durch. Wir beschränken uns dabei nicht nur auf IPv6, sondern beziehen auch IPv4 mit ein, um beide Protokolle miteinander, sowie mit der Messung aus 2007 vergleichen zu können. Da wir Daten über die Internettopologie benötigen, können wir nicht auf übliche Scans zurückgreifen, sondern müssen alle Ziele tracerouten. Die Persistenz von Routing Loops ist von entscheidender Bedeutung, daher können wir uns auch nicht auf eine einzige Messung verlassen, sondern müssen die gefundenen Routing Loops durch wiederholtes Messen bestätigen.

Sowohl im IPv4- als auch im IPv6-Internet fanden wir deutliche Hinweise auf persistente Routing Loops. Nach der Durchführung mehrerer Messungen und des Scannens von 4 Milliarden Zielen in IPv6 fanden wir 21.227 persistente Loops in IPv6 und 25.687 persistente Loops in IPv4. Dadurch sind 31,2% des beobachteten IPv6-Internets und 9,6% des beobachteten IPv4-Internets für Störungen durch Angriffe im Zusammenhang mit Routing Loops offen.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Abstract

IPv6 is the not-so-new kid on the block, providing connectivity to an ever-increasing number of hosts next to its predecessor, IPv4. Regardless of the protocol, stability and uptime are key amongst all networks, challenging network operators on a daily basis. Even though the internet is an ever-evolving entity, an aspect that has the potential to impact stability has been overlooked in the last decade, namely routing loops. In preliminary IPv6 measurements, a large amount of ICMP Hop Limit Exceeded messages – strongly indicating routing loops – have been observed. However, the last dedicated measurement for persistent routing loops in the internet control plane dates back to 2007 and obviously lacks IPv6.

In this thesis, we take a look at publicly available measurements and plan and conduct our own full-scale measurement to investigate this anomaly. We do not restrict ourselves to IPv6 but also include IPv4 in order to compare both protocols and, further, compare the results with the last dedicated measurement from 2007. However, we cannot rely on traditional scans using only a single probe per target. As data about the topology behind the paths is required, traceroutes are performed. As the persistence of routing loops is key, we cannot rely on a single measurement either but have to confirm the found routing loops by repeatedly retracing the found paths.

We found strong evidence for persistent routing loops in both the IPv4 and IPv6 internet. After conducting several measurements and tracing to 4 billion targets in IPv6 alone, we found 21.227 persistent loops in IPv6 and 25.687 persistent loops in IPv4. These loops open up 31,2% of the traced IPv6 and 9,6% of the traced IPv4 internet for connectivity disruptions due to routing loop related attacks.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Contents

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 1

Introduction

IPv6, the successor of IPv4, increases the number of available addresses by large margins in comparison to the previous generation, paving the way for the increasing demand for IP-based devices. As registrars are finally running out of IPv4 addresses, the internet is already, though at a slow pace, migrating to IPv6, which reached technical maturity a few years ago and already carries a non-negligible portion of global internet traffic [1]. IPv6 solves the issue of availability by raising the number of addresses from 2^{32} to 2^{128} while providing additional features added within the last few years.

The increased address space allows each device to carry a public IP address, removing the need for technologies like Network Address Translation (NAT). If a native IPv6 connection is not available, transition mechanisms exist to bridge the gap and provide a preliminary connection for test and prepararing native IPv6 deployments. Many operators in the field are still not as experienced with IPv6 as they are with IPv4. Therefore mistakes in deployment happen, resulting in possible degradation of quality. Either way, we all are expecting a stable internet connectivity regardless of the protocol used to make our daily life easier and more connected.

In this thesis, we explore a way to impact the stability of both the IPv4 and IPv6 internet. Previous measurements of the IPv6 control plane showed irregularities regarding routing and reachability on operators' sides, involving multiple ISPs and other service providers. During exploration of these measurements and existing public datasets, we encountered a multitude of *Time Exceeded - Hop Limit Exceeded in Transit* messages originating from various operators around the globe. Holzbauer et al.[2] researched the ICMPv6 answer behavior and found a large number of *Hop Limit Exceeded* errors from a limited number of hosts. These *Hop Limit Exceeded* messages are a strong indication for routing loops, multiple routers engaged in a circular route, in which packets loop until the hop limit is exceeded.

To fully understand the impact of these routing loops we execute our own large-scale measurements for IPv6 and later on also for IPv4. In comparison to previous measurements, our approach differs in execution. Regular ping scans do not provide sufficient data for this analysis. The path a packet is traversing to the scanned destination has to be captured for the loop evaluation. Thus, for every target in both our IPv4 and IPv6 target lists, one traceroute was done over 25 days of scanning.

A subset of unreachable targets behind loops was selected and rescanned multiple times to verify all loops for persistence. Such persistent loops can be abused to overload routers involved in the loop and disturb connectivity for systems served by these routers. We found routing issues in both the IPv4 and IPv6 internet which could degrade traffic and connections for millions of users on the associated networks and providers as direct results of misconfiguration.

The remainder of this thesis is structured as follows:

Chapter 2 provides the background of used protocols and their characteristics. The previous work for both the existing scan tools and implemented features to scan efficiently are summarized. Further, the previous work on routing loops and their appearance in the wild is presented.

Then, chapter 3 describes the methodology on how the scan was planned and the results were extracted. Details about the implementation were added to provide further information about used processes.

The final chapter 4 discusses the found data and compares the results between IPv4 and IPv6 . Further, a comparison with the last dedicated measurement from 2007 is provided. The chapter also contains outlooks for future work.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

CHAPTER 2

Background

The following section will introduce the basics of networking and its protocols. We will discuss some aspects in detail if it is necessary to understand parts of our process and analysis.

2.1 Internet Protocol

Internet Protocol, or IP in short, is a protocol family designed to deliver packets from one host to another. On the OSI model, IP sits on layer three, the Network Layer, above the Data Link Layer with protocols like Ethernet. As it sits below the Transport Layer and thus below TCP and UDP, it does not care about ports. IP tries to deliver data to a given destination address. Packets comprise the header section containing metadata about the data transfer and the payload section with data or further encapsulated protocols.

2.1.1 IPv4 vs IPv6

Both IPv4 and IPv6 serve a similar purpose but differ in implementation. The most obvious change is the increased address length from 32 bit to 128 bit. Table 2.1 shows an example address for both protocols. This drastically improves the address space from 2^{32} to 2^{128} but also relies more on name translation as addresses can be harder to remember. To improve readability in IPv6, filling zeros can be omitted and replaced by two colons, also depicted in Table 2.1. Moreover, the header of the packets changed. Only the most necessary metadata resides directly in the header. Additional headers can be specified to include more detailed metadata or features like fragmentation and mobility support.

IPv6 reduces the total number of fields in the header, but due to the longer addresses, the header doubles in size. Both protocols contain a mandatory source and destination address and the packets lifetime. This lifetime restricts how long a packet is allowed to traverse through the routing plane as the *Time To Live* in IPv4 or the *Hop Limit* in

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Protocol	Long Form	Short Form
IPv4	192.000.002.139	192.0.2.139
IPv6	2001:0db8:0000:0000:0000:0000:0000:9890	2001:db8::9890

Table 2.1: Example IP addresses for both protocols in long and short form

IPv6. We will use both of these field names interchangeable, independent of the protocol, as they serve the same purpose. The host sending a packet can set this counter to any value between 0 and 255. Each router handling the packet will decrease this counter by one. If the counter reaches zero, the router should drop the packet and return an appropriate ICMP Error message, namely *Hop Limit Exceeded*, to the source.

Other fields such as the IP ID, encapsulated protocol and a dedicated flag field exist for various reasons, including fragmentation, but are also used for traffic shaping. Many service providers use load balancing to increase throughput, which can work on a per-packet base, but also more advanced ways exist. Instead of using various fields for this purpose, IPv6 introduced the dedicated flow label to allow easier flow control.

The notation of prefixes is used to describe address ranges and segmentation of address pools. A prefix contains a fixed part for the subnet and the prefix length. This prefix length is written as a <subnet>/X, whereas X stands for the number of bits fixed for the subnet, so for IPv4 between 0 and 32, for IPv6 between 0 and 128. For example, the prefix *172.16.0.0/24* describes the address range *172.16.0.0 - 172.16.0.255* as the first 24 bits are fixed. Same goes for IPv6, e.g. the prefix *2001:db8::/64* describes the range from *2001:db8:0:0::1* to *2001:db8:0:0:ffff:ffff:ffff:ffff*. This prefix notation will find much usage in this work.

2.1.2 ICMP / ICMPv6

ICMP and the IPv6 version ICMPv6 are important protocols to relay errors encountered during packet transfer or for connectivity diagnostics. It is not sent via TCP or UDP, and both versions are a dedicated protocol above their related IP version.

Errors are grouped in categories by their error type, while the additional error code gives a more detailed explanation. Some error types do not have error codes and are fixed on code 0. The most important errors are present in both ICMP and ICMPv6, e.g. *Destination unreachable* with multiple error codes or *Hop Limit Exceeded*. IPv6 further increases the dependency on ICMPv6 by including essential features like neighbour discovery, which replaces the dedicated *Address Resolution Protocol (ARP)* from IPv4.

Table 2.2 lists the error codes and types of IPv4 and IPv6 we use in this work.

To *"limit the bandwidth and forwarding costs incurred by originating ICMPv6 error messages"* [3][2.4f], rate-limiting has been implemented for ICMPv6. The RFC[3] in question recommends rate-limiting to be implemented in the form of a token-bucket system as it allows for a burst of error messages to be sent. The token-bucket system

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Error	ICMP		ICMPv6	
	Type	Code	Type	Code
Echo Request	8	0	128	0
Echo Reply	0	0	129	0
No Route	3	0	1	0
Administratively Prohibited	3	9,10	1	1
Adress Unreachable	3	1	1	3
Port Unreachable	3	3	1	4
Time Exceeded	11	0	3	0

Table 2.2: Table of used ICMP / ICMPv6 Error codes

Vendor	Bucket Size	Refill Rate	Packets per sec
Juniper [4]	5s (5000 token)	1000 / 1000ms	1000
Palo Alto [5]	100 token	100 / 1000ms	100
Cisco [6]	10 token	10 / 100ms	100
HPE [7]	10 token	10 / 100ms	100

Table 2.3: Default Token Settings for various Vendors

comprises two major components, a bucket that can hold up to N tokens and a refill rate that fills the bucket with X tokens per timeframe. Vendors can allow their customers to change these variables. In Table 2.3, we documented the default token refill rate and bucket size for selected vendors. To not trigger any rate-limiting, we must respect the lowest common packets per second value, which is 100 packets per second across multiple vendors default settings.

2.2 Routing Technologies

To build a small scale network with one router is easy enough, but the router needs to know where to send the packets to reach other networks or the internet. This is handled by routes the router needs to know, either by knowing a specific prefix or as a default route if the prefix is not directly known. The easiest way to configure a route is using static routes, a given IP address of another router that handles a configured prefix or the default route. For smaller and static installations, this can be viable. However, static routes are not feasible for more complex networks and the internet, as routes can frequently change and need to be pushed manually to all involved routers. Dynamic routing protocols are the easiest solution for this by offering flexibility and automatic route propagation without manual configuration. Cisco has an excellent overview of dynamic routing for the standard routing protocols within and across routing domains [8, 9].

These dynamic routing protocols are split into two groups, *Interior Gateway protocols (IGP)* within a single routing domain or *Exterior gateway protocols (EGP)* to connect

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

multiple routing domains. There are many different solutions to choose for IGP. They are categorized in how they store the routing table on each router.

Distance Vector Routing Protocols: Routing protocols in this category keep a record for each prefix with a gateway hop and how many router hops the destination prefix is away. The router chooses the lowest-cost path based on the distance and other metrics such as network latency and link bandwidth. At no point in time does any router have a complete topology of the network. It only knows the distance and the next hop for each prefix. *Routing Information Protocol (RIP)* and *Enhanced Interior Gateway Routing Protocol (EIGRP)* work on the distance vector principle, to name a few examples.

Link-state routing protocols: This category keeps a complete map of the topology in memory and calculates the paths to the destination prefix. Again other metrics like network latency and link bandwidth play a role in selecting a path. *Intermediate System to Intermediate System (IS-IS)* and *Open Shortest Path First (OSPF)* are examples of Link-state routing protocols.

For EGP, there is only one protocol. Historically, *Exterior Gateway Protocol (EGP)* (not to be confused with the entire category) was used in the early days of the internet and was superseded by BGP in 1994. BGP has since been used without alternatives and has seen many upgrades and extensions both for better protocol support and increased security.

An organization with control over serveral routers and interested in route exchange are called *Autonomous System (AS)*. Each AS receives a dedicated number called the *Autonomous System Number (ASN)*. They can announce allocated network prefixes and peer with other AS via this dedicated number. Route information exchange happens between two AS and need to be configured manually for each peer. The peering routers must be directly connected as BGP announcements are sent with a TTL of one. Routers between peers communicate via the *external BGP (eBGP)* protocol to exchange route information. If one AS owns multiple edge routers to communicate with, they need to synchronize their routing paths using *internal BGP (iBGP)*.

For each prefix a router receives, it needs to store the path to the target prefix. Instead of storing a path of IP addresses, each router stores a chain of ASN to reach the destination. The next hop is already known via the configured peering partner. BGP also features a simple loop prevention feature. If an ASN appears multiple times in the ASN path, the given route is discarded.

2.3 Migration to IPv6 & Transition Mechanisms

Internet Service Providers (ISP) want and have to migrate from IPv4 to IPv6 at some point in the future. There are a lot of possible solutions and transition mechanisms available to add some grace period as the switch is not as simple as turning off IPv4 and turning on IPv6.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Dual Stack: The easiest way to grant connectivity to both protocols is to use Dual Stack. The most desirable path is deploying IPv6 alongside the existing IPv4 infrastructure and giving their customers both an IPv6 prefix and a public IPv4 address. Dual Stack is the prefered option as all customers receive unrestricted access to both IPv4 and IPv6.

Dual Stack Lite: To preserve IPv4 addresses, ISPs decided to implement Dual Stack Lite, which limits the access on IPv4 on some parts. Let's take Figure 2.1 as an example. Like standard Dual Stack, all customers C_1 to C_n receive an IPv6 prefix and an IPv4 address. On the IPv4 network, however, the customer receives an internal IPv4 address denoted by the dedicated prefix *100.64.0.0/10* for Carrier Grade NAT. Multiple end users share the single public IPv4 address of the ISP Edge Router E managing the NAT for outgoing IPv4 connections. Thus no customer can accept new incoming, not already established connections on IPv4 as they cannot configure port forwarding on the edge router E.

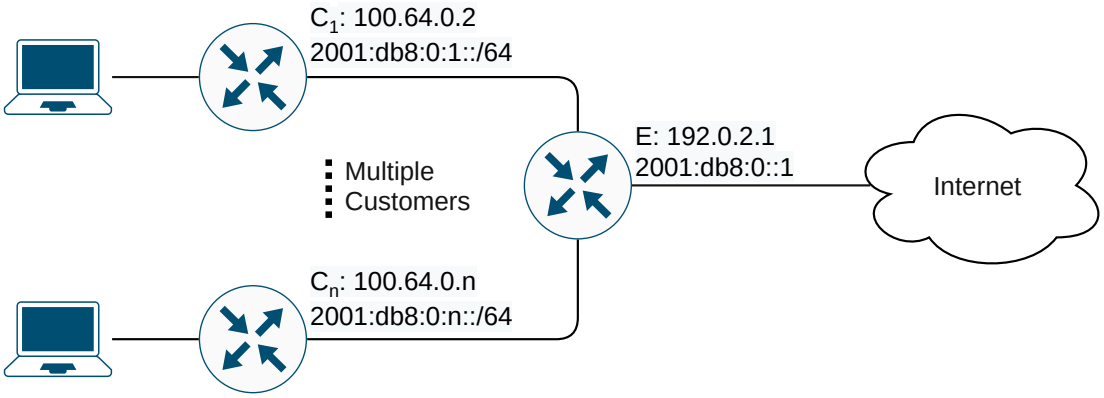

Figure 2.1: Example Network for usage of dual stack lite

Indirect Transition: There are other ways to migrate to IPv6. That include techniques like abusing NAT for cross-protocol talk and various methods of tunneling.

With NAT64, the ISP deploys an IPv6 only network, including a special DNS and NAT server to translate IPv4 only requests. If a DNS request results in an IPv4 address, the DNS64 server further translates it into an IPv6 address that contains the IPv4 within the NAT64 prefix 64:ff9b::/96. As the host sends packets to this special address, the ISPs dual-stack NAT64 server catches the IPv6 connection and starts the actual IPv4 connection to the server, returning any answers via the active IPv6 connection back to the source host.

Also possible is the usage of tunneling techniques. Again, multiple protocols allow IPv6 connectivity over an existing IPv4 infrastructure or the other way around. These protocols like 6to4, 6in4, 4in6, 6rd and many more allow tunneling one protocol over the other between two Dual Stack nodes. An example case would be a company network with IPv4 only internet access but internal Dual Stack deployment. The edge router of the corporate network can be used to connect to an external server to tunnel IPv6

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

network traffic and establish IPv6 connectivity this way. Other protocols, e.g. Teredo, allow single hosts to directly create a tunnel to an external tunneling provider to receive a dedicated, public IPv6 address. This makes it possible to receive IPv6 traffic even though the host has no dedicated IPv4.

2.4 State of deployment

The available IPv4 space is running thin. This is no new news and counter processes have been in effect since 2019. [10] RIPE NCC has had a waiting list in place since November 2019 to redistribute returned prefixes, one /24 at a time. [11] AMPRnet sold /10 for 17$ per address in mid-2019 (https://www.ampr.org/amprnet/) IPv4 address has established as a market with prefixes selling between 25$ and 40$ per address in early 2021. [12, 13, 14]

IPv6 deployment is still improving. Around 35% of all connections arriving at Google are via native IPv6, as seen in Figure 2.2. The deployment stagnated a bit and the growth switched from exponentially to linear increase. Transit and content providers provide a strong deployment at the core. However, edge providers like ISPs and enterprises are lacking due to a lack of incentives and working alternative strategies like NAT in IPv4 [15].

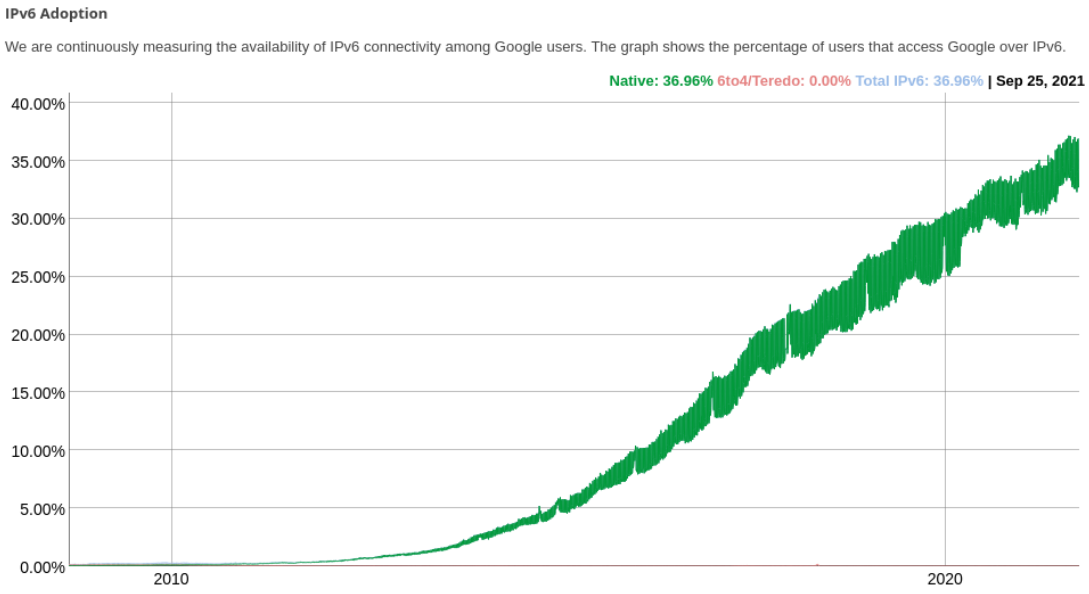

Figure 2.2: Google IPv6 Statistics - Visted 2021-09-25, Image by Google[1]

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

2.5 Definitions

In this section, we present a list of definitions used in this work. Most of them are consistent with the proposed definition by Xia et al.[16].

Traceroute: A method to get the IP address of each hop on the route to a target IP address. This is usually done by setting the Hop Limit to a low value and increasing it for each hop on the route. As each packet expires due to different Hop Limits on different hops, we can gather the IP addresses of these hops on the path to the destination. Pitfalls are load balancers, routers not answering to specification, not answering at all, or filtering packets due to rate limiting. Various techniques exist to improve the results from tracerouting (e.g. Paris Traceroute, setting fixed values for certain fields within an IP packet).

Loop: A series of routers that try to send a packet to its destination but fail to manage so, as their configured routes result in a circular path. Let's take figure 2.3 as an example and assume Router *A* and Router *B* have a route configured to 2001:db8:1000::/48 via Router *C*. Router *A* receives a packet and forwards it towards 2001:db8:1000::/48 via Router *B*. Router *B* again forwards it to Router *C*, as it has the knowledge about the given prefix via this specific Router. Router *C*, on the other hand, actually only is configured to handle the prefix 2001:db8:1000:1000::/64 within the given /48. Thus without any knowledge about the rest of the prefix, it is using its default route to further forward the packet to Router *D*. Router *D* also has no knowledge about the prefix, further sending it down its default route, reaching Router *B* again. At this point, we have reached an already known router and completed the loop. The packet trying to reach a host within the example prefix will be forwarded until the hop limit reaches zero and is discarded by a router. The router will then send an *Hop Limit Exceeded* ICMPv6 error message with the original packet encapsulated inside of it.

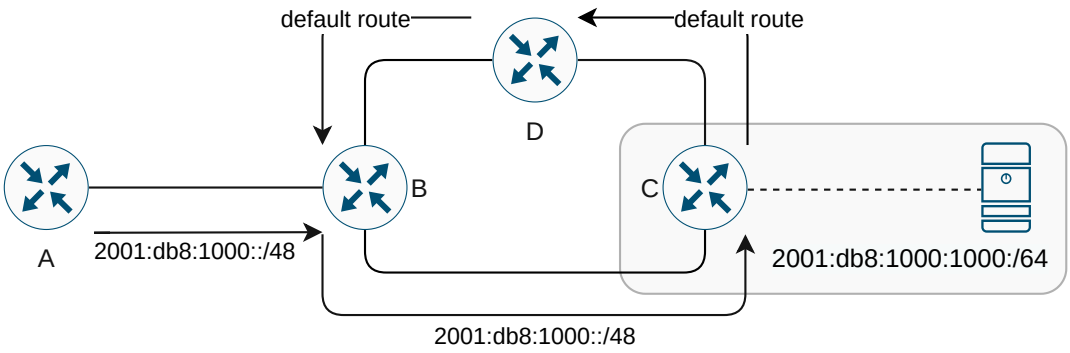

Figure 2.3: Example of a routing loop including three routers to one destination.

Transient loops: This kind of loop occurs because of automatic changes of the forwarding plane through dynamic routing protocols such as BGP, OSPF, etc. These errors usually fix themselves within a short time after the correct routes propagate between all routers

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

within the given structure. Most loops fix themselves within seconds. However, according to Teixeira et al.[17] about 35% persist up to 60 seconds.

Persistent loops: In contrast to transient loops, persistent loops are loops persisting over a long time, not being fixed by route propagations. They usually originate from manual route (mis-)configurations. E.g. a router should handle a /48 but only handles a portion of it. Its default route points back out the same interface, thus resulting in a loop. Such loops can easily be avoided by adding a pull-up route containing the entire prefix the router should handle. This pull-up route tells the router it is not supposed to forward any packets destined to this prefix but rather handle them by emitting an *Destination Unreachable* error and dropping the packet.

X-Hop Loop: A loop of a given length X, counting each participating router forwarding a packet in a circular fashion. Three routers forwarding a packet the same path over and over again is further described as a 3-Hop Loop, and so on.

Full loop: A loop of which all routers resulting in the loop are known. There are no unknown hops in between.

Candidate Prefix & Shadowed Prefix: These prefixes describe prefixes with a public route. Therefore they should be reachable by an internet-connected host. However, a loop on the path to the prefix is preventing packets to actually reach it, thus making it a candidate for a shadowed prefix. See figure 2.4 as an example. Both router A and B know about routes to the prefixes 2001:db8:1000::/48 and 2001:db8:2000::/48 via Router C. The prefix 2001:db8:2000::/48, however, is either not handled at all or not correctly by Router C, thus resulting in Router C forwarding its packets down to Router B again. This results in a loop with a loop length of two between Router B and C, making the prefix 2001:db8:2000::/48 a candidate prefix. If a Candidate Prefix and the loop shadowing it persists over multiple measurements, we upgrade the prefix to a Shadowed Prefix.

Dark Prefix: A shadowed prefix that can be exploited to attack imperiled prefixes. This means the path to this prefix contains a loop, the routers within this loop can be overloaded to attack any other forwarding path.

Imperiled Prefix: Prefixes described as imperiled are public routable prefixes and reachable by an internet-connected host. On the path to the prefix is no loop, but the route includes a router involved in a loop to a shadowed prefix, thus making it a victim if the loop is targeted and overloaded.

Back to Figure 2.4, we assume the same configuration as previously with the shadowed prefix. The correctly configured prefix 2001:db8:1000::/48 is reachable via Router A, B and C. However, due to the misconfigured route to 2001:db8:2000::/48, the route to our reachable prefix can be attacked by abusing the loop between Router B and C. As each packet loops around those two routers until the hop limit reaches zero, the impact of each packet is amplified, resulting in a lower packet count needed to overload the connection.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

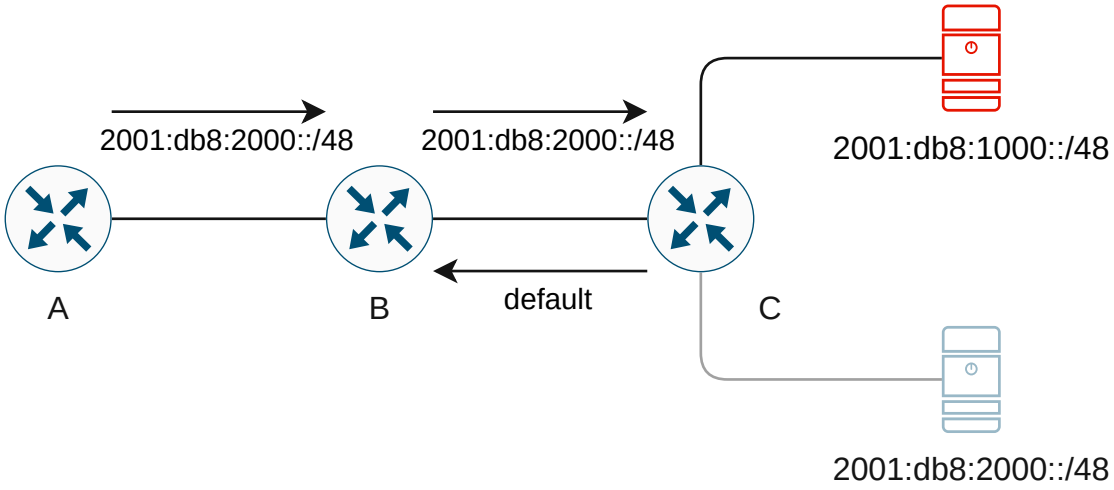

Figure 2.4: Example of one shadowed destination impacting an imperiled prefix.

Spammer: For any given route, one hop on this route is sending multiple packets back as a response to one single probe. We encountered some spammers who increased their amount of packets in response with every packet looping further on, massively amplifying the sent packages.

Load Balancer: ISPs and Service Providers use load balancers for various reasons. They can be used for Quality of Service, traffic shaping, or simply to distribute a load. This should not be an issue due to how tracerouting is implemented in yarrp. See section 2.6 for details. Even if our trace packets are load-balanced if the target is reachable or other errors than *Hop Limit Exceeded* arise, the impact is neglectable. If, however, the traceroute ends in an *Hop Limit Exceeded* error and the load balancer is involved in the path, this could impact us in two possible ways.

Load balancers with equal path length on all balancing paths still impact our unique loop detection, but the detection will report the correct loop length. However, the worst case would be an architecture like we can see in Figure 2.5. Two paths with unequal path length exist, one path over the routers $B - C1 - C2 - E$ and the other over $B - D - E$. In this case, loop lengths can report as three cases:

- All packets take the path via $C1 - C2$
- All packets take the path via D
- All packets get balanced and take the following trace as an example: $A - B - C1 - E - E - F$

Case three is due to the load balancer mixing the packets through different paths. A traceroute packet with TTL=3 enters the load balancer and ends on $C1$. The next packet with TTL=4 again enters, might take the path over D, exists the load balancer and ends on router E. Packet with TTL=5 might take the upper path via $C1 - C2$, exit the load

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

balancer and again end on router E, thus resulting in a trace that looks like it has a loop on its path. Any packet with a TTL higher than five will end on router F, finishing the trace. If, however, the trace does not end with F and, for example, enter a loop on a later occasion, it still might consider the load balancer anomaly as the smallest loop found.

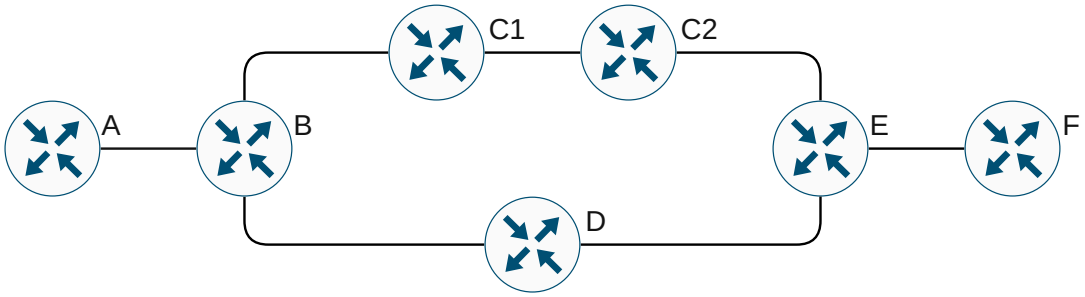

Figure 2.5: Load Balancer with unequal path lengths.

2.6 Scanning and Discovery

Donnet et al.[18] proposes *DoubleTree* as a method to scan more efficiently by abusing the tree-like structure of routes. By starting to scan near a midpoint of a route, the number of probes can be reduced dramatically as routes to these given midpoints need to be scanned only once. If finding a midpoint already scanned before, we can match the route to it without sending any additional probes.

Augustin et al.[19, 20, 21] looked at anomalies appearing during the usage of the traditional traceroute technique. Anomalies were grouped under different categories, all results of load balancing the traceroute packets, thus resulting in odd measurements. Multiple fields of TCP, UDP, ICMP and IP headers were identified to be used in load balancing. Moreover, one can use these fields to their advantage to enumerate all load balancing paths or fix the traceroute to a single path. The resulting technique and tool were dubbed *Paris Traceroute*.

scamper: Lucky et al.[22] developed scamper as a tool to efficiently and quickly trace large portions of the internet. It has a stateful design, each packet waits for a response before it is written to disk. Further, the results are gathered in a structured binary file. All traceroutes are already sorted by hop and can be analyzed efficiently. CAIDAs Ark platform uses scamper to perform their traceroutes. The analyzed CAIDA topology scan in section 3.2 is traced by scamper. The tool implements both *DoubleTree* as well as *Paris Traceroute*.

yarrp: Beverly et al.[23] also developed yarrp to traceroute a large amount of IP addresses. It also implements *DoubleTree* and *Paris Traceroute*. Further, it uses a stateless design to allow certain performance-increasing features. yarrp randomly permutates per default the input IP addresses with the given TTL values. Therefore, no trace is scanned sequentially.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

With the stateless design, no packet is waiting for any responses to arrive. This all minimizes the potential impact on routers on the path and improves results as the load from the traces is distributed over the entire target space.

ZMap: This tool allows to probe a large amount of target IP addresses while allowing very high scan rates[24]. It also works stateless, all probe packets are sent out and a global listener records all incoming packets. ZMap does not trace a path to the target. It only sends one probe to each target and records the incoming response. We are using a modified version to be able to scan IPv6 [25]. Further, we modified ZMap to extract the original destination IP address as well as timings and TTL values from the embedded packet.

2.7 Routing Loops

There has not been much attention directed towards routing loops. The last dedicated analysis goes back to 2007 by Xia et al., otherwise routing loops have been observed in a small number of papers during other measurements.

Xia et al. [16] did a detailed analysis of persistent forwarding loops and flooding attacks abusing them on the IPv4 internet in 2007. They found at least 35 million hosts shadowed by routing loops and at least 11 million hosts vulnerable to exploitable routing loops. Further, they investigated possible root causes for these loops. Missing pull-up routes seem to be the root for a large portion of loops. They performed one single large-scale scan with multiple smaller persistence checking measurements afterward to check for persistence. Moreover, they performed additional measurements from multiple vantage points to check for persistence from other origins. This work is a large influence on our methodical approach and we designed many measurements to be comparable to their data from 2007.

Rüth et al. [26] collected one week of ICMP responses to ongoing IPv4 ZMap scans to study the control plane by looking at the otherwise ignored responses. They found many misconfigured routers, sending wrong or outdated ICMP responses through the internet. Moreover, they found a large portion of the internet unreachable due to various reasons, including a large amount of *Hop Limit Exceeded* error messages. They specifically looked into these *Hop Limit Exceeded* error messages and found a large number of routing loops involving a large number of different AS. Further, they checked the persistence of the found loops. Most of them stayed and rendered over 400k /24 subnets unreachable.

Bock et al. [27] looked into traffic amplification attacks using middleboxes on the IPv4 internet. During their work, they found 53k IP addresses containing a routing loop on its path that could be abused as traffic amplifiers. Further, they identified that in 62% of all /24 prefixes containing a routing loop on its path, this one loop was the only loop found within the prefix. All other addresses responded as expected.

Nakibly et al. [28, 29] analyzed multiple automatic transition mechanisms (Teredo, ISATAP, 6to4) in regards to routing abuse. They found multiple ways to bounce packets

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

between relay servers and participating clients, effectively creating routing loops between them that could be used to interrupt service. One particular case created a self-replicating loop that would loop packets indefinitely as each packet entering the relay server resulted in a new packet, which again entered the server.

Teixera et al.[17] looked into the route changes due to an IGP (in this case OSPF) event and how they translate into BGP path changes. They analyzed the behavior of the interplay between these two routing protocols in the AT&T network and found that BGP updates can lag 60 seconds or more behind the resulting OSFP updates. Such long transition times can result in performance degradation due to a sudden increase of traffic on different routers or even routing loops due to different reaction times on route changes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 3

Scanning the Internet

In this chapter, we will talk about the various data sources we discovered, scans we executed and analysis we have done along the way.

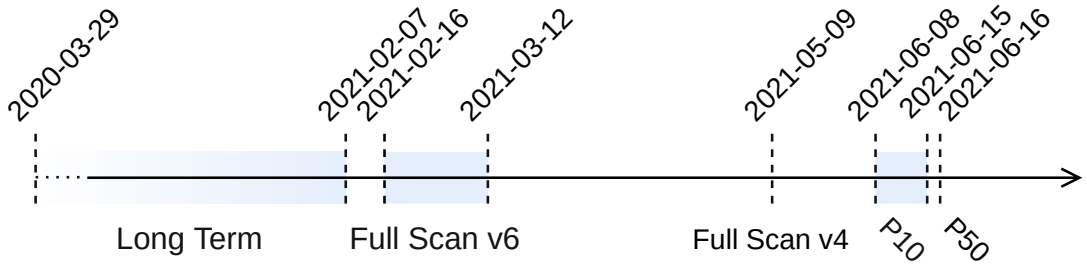

Figure 3.1: Timeline of various Scans during the Development

From previous measurements and tests, we created a small dataset of about 75000 targets that consistently returned an ICMPv6 *Hop Limit Exceeded*error. These targets were scanned twice a week from two vantage points continuously for about ten months.

At the same time, we explored other data sources, including the public datasets of CAIDA. The topology dataset spiked our interest. CAIDA sends two traceroutes to each announced prefix every 24 hours from all available ARK measurement nodes.

However, these two datasets were not satisfying due to various reasons. Thus we planned to do our own large-scale scan to cover a significant portion of the routed IPv6 net. This scan started on 2021-02-16 and lasted until 2021-03-12, recording traces for around 4.5 billion targets. Further, we later decided to replicate Xia et al.'s findings from 2007 for IPv4 and started an appropriate IPv4 Scan on 2021-05-09.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The persistence scans for both protocols started on 2021-06-08 and ended with the ZMap Persistence50 scan on 2021-06-16. Figure 3.1 shows all scans on a timeline, visualizing the duration each scan took and how much time was between each scan.

The scans were not optimally spaced. Much time went into the development of the used toolkit before continuing with the scans. We did not exactly know what data was necessary at what point in time. Each data fragment feeds back into another scan to aggregate more data and generate new target lists. Moreover, the persistence scans were further delayed as we had to port our scripts from Python to Rust, resulting in much time spent implementing analysis processes. witching language late into the process was necessary as we struggled with Python to analyze our data. We reduced the time spent on analyzing the full scan from multiple days to a few hours on Rust.

3.1 Datasets from Previous Measurements

During previous large-scale IPv6 measurements, we noticed a large amount of ICMPv6 Error message of Type 3 Code 0, also known as *Hop Limit Exceeded*, on various target prefixes. This error code is a strong indicator for a routing loop as this error only gets sent out if the hop limit of the packet reaches zero. We extracted all targets responding with such an error and used the resulting list for the first tests and measurements.

Following these tests was a long-term measurement of all 78525 targets from the original dataset. The scans lasted from 2020-03-29 till 2021-02-07 and were done twice a week, every Sunday and Wednesday. We traced the targets by using yarrp from hop 3 to 64 with a scan rate of 1.000 packets per second. Two different vantage points conducted the scans. The first vantage point, *VP1*, is our dedicated scan server located in Vienna. *VP2* is the second vantage point and is a virtual private server located in Paris. The Results of the entire scan were not as useful as hoped, mostly because the target selection is not explainable and non-reproducible. The data still gives a good overview of the longevity of the encountered loops.

Taking a closer look at the total number of unique loops and their lifetime paints an interesting image. The progression can be seen in Figure 3.2. The number of unique loops decreases consistently over ten months but stays roughly in the same ballpark around 16000 unique loops found. This is similar at both vantage points, even though *VP1* sees a slight bump in unique loops found in September 2020.

The main takeaway of this chart, however, is the persistence of loops between scans. When comparing the set of loops of each scan with the set of the first scan done in March 2020, the number of persistent loops decreases statically. However, if comparing the set of loops of each scan_n with scan_{n-1}, the number of loops within this overlap stays consistent around 13000 unique loops. Based on the consistency, it appears that about 3000 loops fluctuate between each scan. This is the difference between the total unique loops and the overlap between two adjacent scans.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

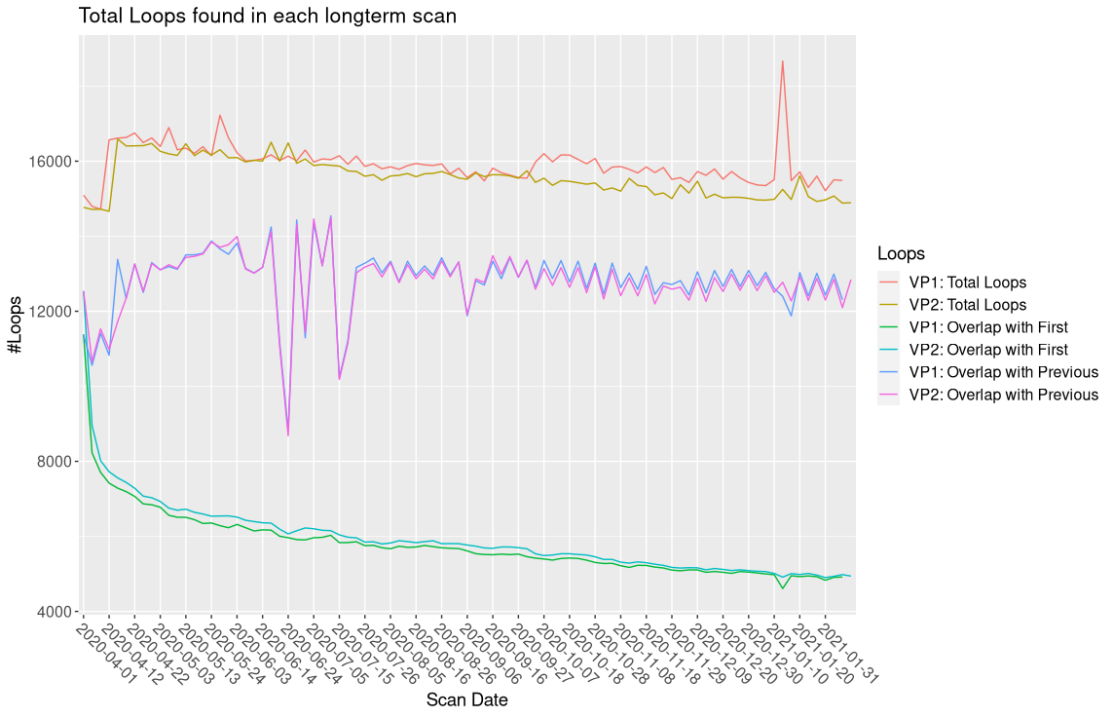

Figure 3.2: Loop IDs and Vverlap between Scans

Adding ASN information to pall targets responding with *Hop Limit Exceeded* shows the downside of this dataset in Figure 3.3. The target list was not filtered at the beginning. Therefore we introduced a heavy bias towards AS3303 (Swisscom). Thus, this analysis might not represent the internet control plane as a whole. AS3303 has by far the most targets within the dataset. 14.510 targets out of 78.525 in total originate from this AS. Around 8000, so about half of these targets resulted in a unique loop, as can be seen in Figure 3.3. The number of loops seems stable, and there is no noticeable drop in unique loops between the first and the last scan for each vantage point.

3.2 A Look at CAIDA Topology Traceroutes

With their topology dataset[30], CAIDA traces two IP addresses in each announced prefix once every 24 hours, the first address of the prefix and one random address. All scans are done with scamper[22] from all available nodes of the CAIDA Ark and publicly available in .warts files.

We first thought this dataset would be a valuable source for this analysis, but we quickly realized it was not fine-grained enough. Due to how the scans are laid out, there is considerable variance between all prefix sizes. Thus the data is only helpful for a quick overview in our case. Further, the probes for the Caida ARK are connected by people around the world, crowdsourcing the data. Therefore, the uptime is not guaranteed, but

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

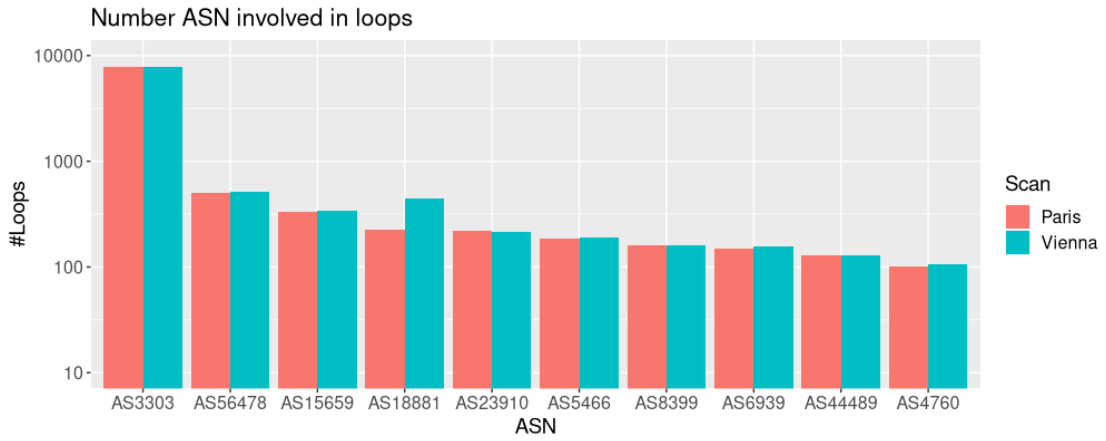

Figure 3.3: Loop IDs and Overlap between Scans

they achieve many vantage points in different parts of the world and different AS, which would otherwise be a tremendous amount of work.

Figure 3.4 shows the uptime for three selected nodes that had the most data available. The selected nodes were *ams-nl* in Amsterdam, *hkg-cn* in Hong Kong and *san-us* in San Diego. We downloaded the data from three consecutive days each month for the entire available dataset. This was done because sometimes there was no data for a single day, thus resulting in even more holes. Further, the uptime chart shows the number of received probes per node, starting slow at the beginning and gaining traction around 2017.

The data was available from as early as December 2008, with the scans still running today from 44 nodes for IPv6 probing at the time of writing. As shown in Figure 3.4, the three nodes still have some downtime with missing data, but overall the available data is usable for our overview.

The data still presents us with valuable information and strengthens the motivation to continue this work. Figure 3.5 displays the percentage of received responses of sent probes that return with *Hop Limit Exceeded*, indicating a routing loop. The graph suggests that at any given time, the number of probes being answered by an *Hop Limit Exceeded* error is around 10% of the total amount of probes sent. Also, it seems to be in line across all selected nodes around three different continents.

3.3 Full Scan

Based on the paper from Xia et al., we constructed our own large-scale measurement by scanning both the IPv4 and IPv6 internet. While it was very doable to scan the entirety of the IPv4 space, this was not feasible for IPv6. Therefore we had to find ways to reduce the given address space to a manageable size. For both protocols, we started our selection

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

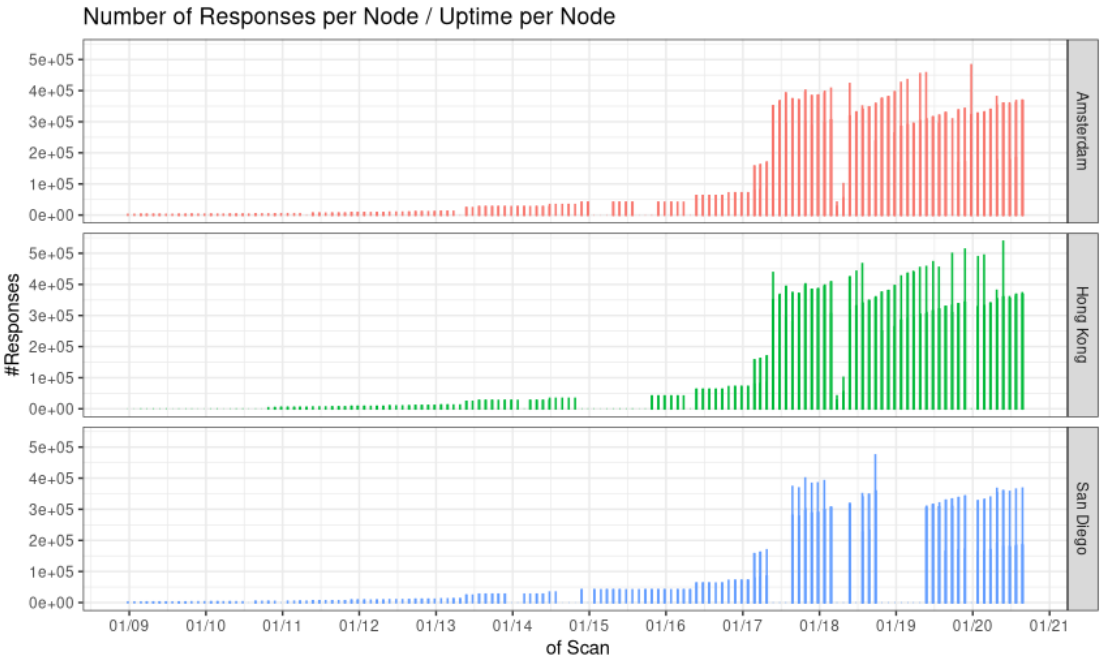

Figure 3.4: Uptime of the three selected CAIDA Topology Nodes

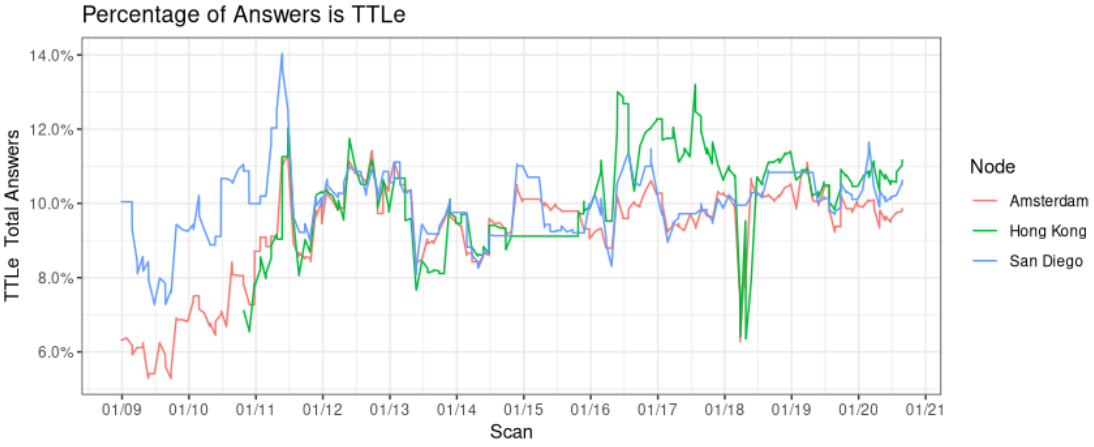

Figure 3.5: Percentage of Probes resulting in a *Hop Limit Exceeded* Error

at publicly available BGP routing data. CAIDA provides enhanced RouteViews data as a daily download containing a routable prefix, its length and the owning AS for both IPv6[31] and IPv4[32].

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.3.1 Targetprefix Size

First, we need to specify how broad we want to conduct our scan. A big piece of this puzzle is the prefix size of each subnet we want to probe. Most ISPs and other service providers filter the incoming BGP prefixes larger than /24 on IPv4 and /48 on IPv6. There are multiple recommendations[33, 34] of this configuration agreement, yet no hard limit is set or enforced by any organization or vendor.

These prefix edges can further be observed in the visibility of prefix announcements. RIPE Labs has a detailed article about announcements and their visibility within the internet [35] for both IPv6 and IPv4. Figure 3.6 shows the total announced prefixes for IPv6 and Figure 3.7 does the same for IPv4. These numbers are from 2019 and show the total number of announced prefixes. The colors represent the visibility of the prefix to each peer. Most of the prefixes up to /48 on IPv6 and /24 on IPv4 are visible to 99% of the peers. Beyond that point, announced prefixes drop in number. Further, the visibility of these prefixes is below 40$ of their peers.

The normalized data as seen in Figure 3.8 for IPv6 and 3.9 for IPv4 further confirm this cliff after /48 and /24. Thus we settled on one trace per /24 in IPv4 and one trace per /48 in IPv6.

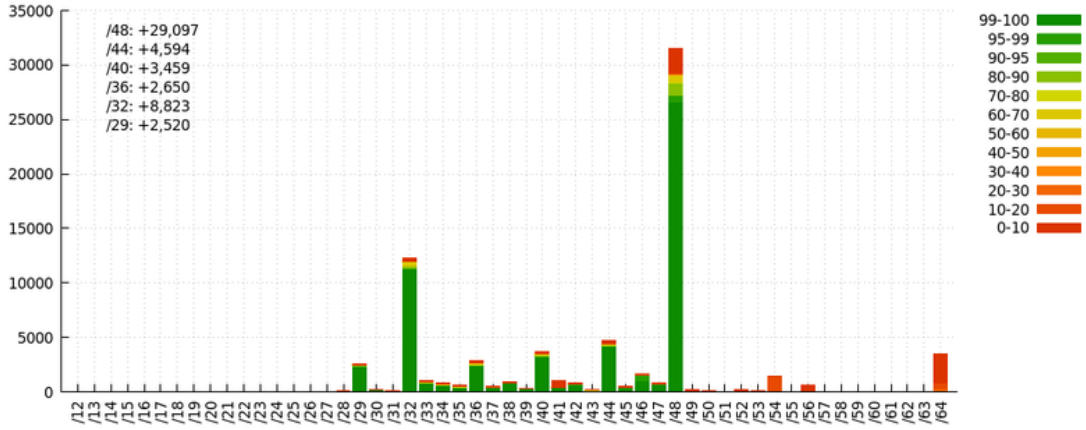

Figure 3.6: Total announced Prefixes in IPv6; Image by RIPE NCC [35]

3.3.2 Target Selection & CAIDA routeviews

The RouteViews project[36] got multiple collectors deployed in various AS to collect BGP announcements. CAIDA uses this collection and provides daily snapshots of announced prefixes containing the autonomous system announcing it. These snapshots are provided for both IPv4 and IPv6. We based our further scans and analysis on these snapshots, such as creating target lists for the traceroute scans or for further analysis later down the line. The data is structured as a TSV file containing a given prefix, its prefix length

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

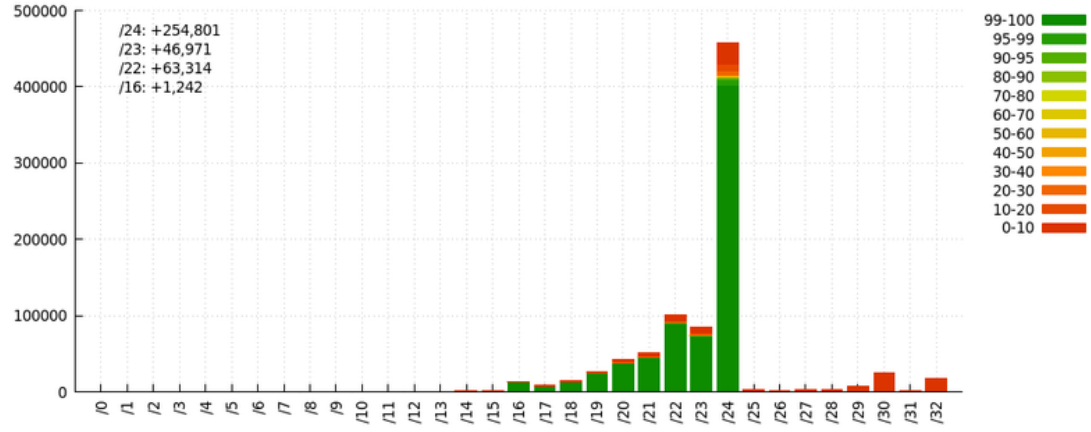

Figure 3.7: Total announced Prefixes in IPv4; Image by RIPE NCC [35]

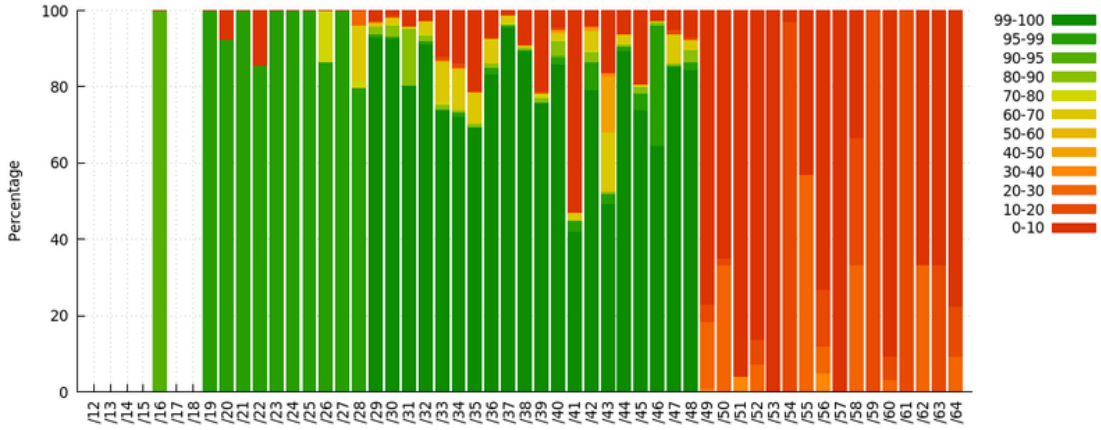

Figure 3.8: Visibility of announced Prefixes in IPv6; Image by RIPE NCC [35]

Prefix	Prefix Length	ASN
2001::	32	6939_1101
2001:4:112::	48	112
2001:200::	32	2500

Table 3.1: First three Lines for the CAIDA prfx2as BGP Announcements with AS Attribution

and the announcing AS. Table 3.1 shows the first three lines of such a given prefix to ASN file from February 19th, 2021.

We do not need the complete BGP output for our target generation but only want the most extensive, least specific prefixes. Taking all least specific prefixes works well in IPv4, as we can scan the given space in a reasonable time. However, for IPv6, we need to

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

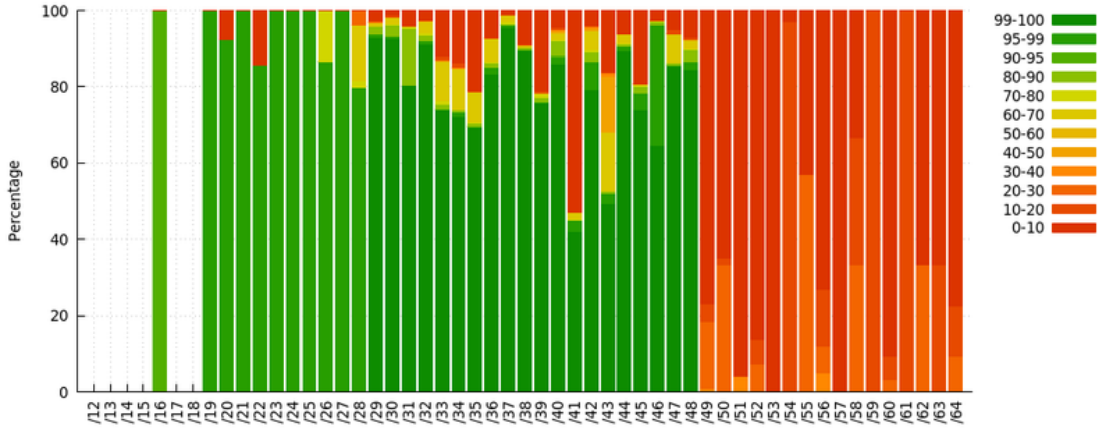

Figure 3.9: Visibility of announced Prefixes in IPv4; Image by RIPE NCC [35]

	Target Prefix	Scanrate	Duration (days)
IPv6 (no filtering)	10.319.033.933	50.000	52.55
IPv6 (no 6to4)	6.024.066.637	50.000	30.68
IPv6 (final selection)	4.463.363.543	50.000	23.42
IPv4	11.851.437	5.000	0.60

Table 3.2: Number of IP Addresses to Probe plus estimated worst case time

preselect and filter the BGP entries. Table 3.2 shows the high number of addresses and time used while scanning. The table already contains the filtered target numbers as well as the used scan rate. How we derived the scan rate will be discussed in section 3.3.3.

Using these numbers, we can calculate the worst-case duration a scan will take. We still have to consider that instead of doing a probe scan, as in we send one packet to each target destination, we need to conduct a full trace to each target. Each full trace increases the number of probes sent considerably. In the worst case, we send the maximum number of probes, which is 22 packets per target.

IPv6 Pre-selection

As previously stated, for IPv6, we settled on a granularity of /48 for our traceroute scan. We traceroute one random IP address in each /48 prefix within all announced prefixes according to the RouteViews file. Still, most of the targeted /48 prefixes reside in the biggest, least specific 32 prefixes. Therefore we need to adjust slightly and adapt our selection of the prefixes.

A flowchart in Figure 3.10 gives a visual representation of the entire process to follow along. We filtered the RouteViews file for the least specific announcements. If a prefix is of length /24 or more specific and has no supernet, we take it unconditionally. If there is a supernet, we ignore the prefix, and if the prefix is less specific than a /24, we manually

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

take a look at it. For those big prefixes, we again decided to split them up into two groups. If there are any subnets, they are taken instead, the supernet is discarded from the list. However, there are still 13 nets without any announced smaller subnets left.

The biggest prefix on this list is the /16 prefix for the transition mechanism 6to4. As traffic originating from this prefix is declining year after year, we decided on skipping it, shaving off 4.294.967.296 target destinations or about 22 days of scan time. For the last twelve prefixes without any subnets, we decided to split them into portions of /24 and scan the first IP (::1) and one random IP of each /32 slice of the bigger /24 subnet with ZMap. If there is any reaction to any of the 512 probes other than *Destination Unreachable No Route*, we add the subnet to the target list. Further, we select at least 4 /24 prefixes. If no or not enough /24 prefixes have been added so far, we complete this list by selecting further /24 from the beginning of the original prefix.

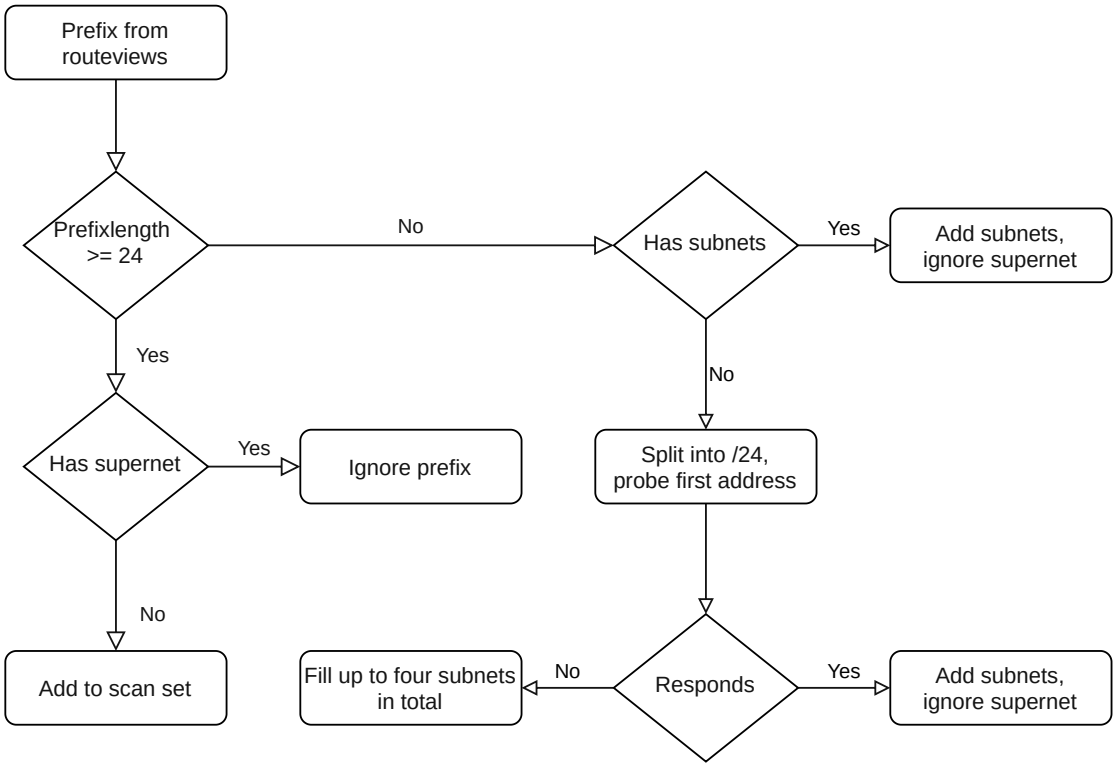

Figure 3.10: Flowchart for the RouteViews BGP Announcement Selection Process

Without filtering, there are 40 announced prefixes with sizes between /19 and /24, or if split into one common prefix length, there are 164 /24. Of these 160 prefixes, only 79 were selected for scanning by using the detailed process as depicted in Figure 3.10 and the text above. This removed 85 /24 prefixes from scanning, cutting 1.426.063.360 destinations from the target list, not counting the 6to4 prefix. In total, the scan target list resulted in 51641 subnets from /24 to /48, totaling 4.598.003.277 target destinations to scan.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

3.3.3 Scanrate

The biggest limiting factor for selecting the scan rate is the ICMPv6 rate-limiting with the implementation of the token-bucket system as described in 2.1.2. This global limit per router determines how many ICMPv6 Error Messages can be sent out and replenish over a given timeframe up to a maximum number of tokens within the bucket. Recalling the values from different vendors in table 2.3, the most restrictive value is still about 100 ICMPv6 error messages per second. We constructed a formula to determine a scan rate with a given safety margin to not disturb regular traffic activity. To begin, we need to know how hard a given router serving a given prefix is being hit as follows in 3.1.

$$\text{router packets} = 2^{(t-p)} \cdot (r+1) \tag{3.1}$$

t is the target prefix length, p is the routed prefix length and r is the number of randomized IPs. 3.1 calculates the number of targets a router with a configured prefix of length p is handling will receive if we probe several r random IPs in each subnet of length t. Further, we introduce a safety margin like following

$$\text{packets incl safety} = \text{router packets} \cdot \frac{1}{s} \tag{3.2}$$

$$\text{bucket time} = \frac{\text{packets incl safety}}{b} \tag{3.3}$$

to ensure we do not disturb packets from regular traffic. In 3.2, we introduce s as the percentage of packets our probes are received as. We selected 1\$ and 0.1% for our values of safety margins. With 3.3, we take the token-bucket rate-limiting implementation as b into consideration, which gives us a time span in which all packets are processed. For the final number of packets per second, we need to introduce the number of target prefixes of size t as n and the maximum number of hops we will trace as h. By dividing the maximum number of possibly sent probes by the given bucket time, we receive the maximum packets per second as follows in 3.4

$$\text{max pps} = \frac{n \cdot (r+1) \cdot h}{\text{bucket time}} \tag{3.4}$$

$$\text{max pps} = \frac{n \cdot h \cdot s \cdot b}{2^{(t-p)}} \tag{3.5}$$

We can further simplify the equation by eliminating the number of IP addresses per target prefix $(r+1)$ from both the fraction and the bucket time to end on equation 3.5. This simplification eliminates the number of targeted IPs per prefix.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

The resulted numbers for the maximum packets per second are plotted in Figure 3.11 for announced prefixes from /16 to /40. We set n to 4.598.003.277 as this is our number of target destinations and we set the number of traced hops h to 22. Both safety margins with 1% and 0.1% are plotted.

We decided on a scanrate of 50.000 packets per second for IPv6. In the worst case, we are reaching our safety of 0.1% packets of the enitre traffic for a router holding an entire /30 or the safety of 1% for a router handling a /27 prefix. This brings the scan duration at worst to about 24 days.

For IPv4, it did not matter as much as for IPv6 as there is no token-bucket system or similar system in place. We selected 5.000 packets per second for IPv4, 10% of the scan rate of IPv6, as it would result in the same time to complete one target file for both IPv6 and IPv4.

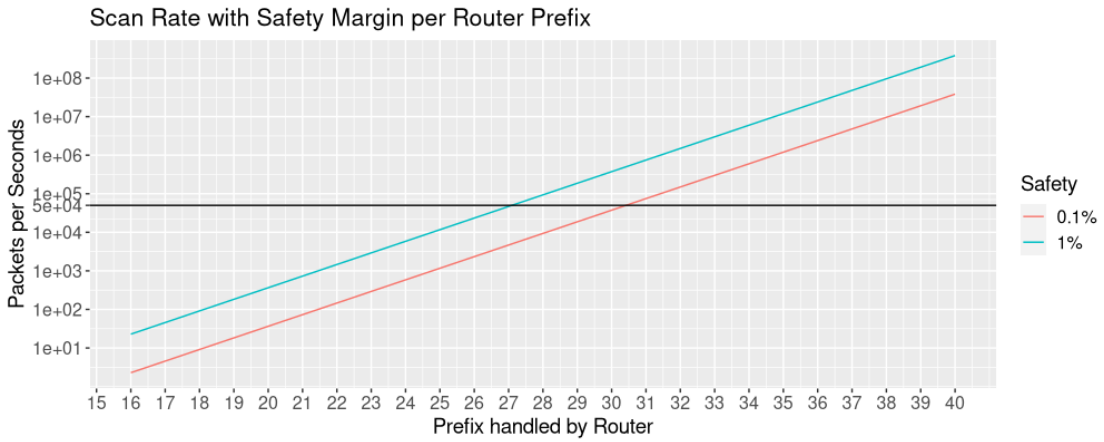

Figure 3.11: Scan Rate Safety Margins with selected Rate at 50.000

3.3.4 Target creation and load distribution

After the target selection from the BGP prefixes, we wanted to ensure we were not generating too much load onto one particular router. Thus, we decided to split the scan into multiple files. Further, this makes it possible to redo parts of the scan if one should fail. All less specific prefixes were pre-split into an intermediate prefix size of /36 and were written to disk in a round-robin fashion to different files. This resulted in about 920 files containing all target prefixes of size /36 or more specific, up to the final size of /48. That way, all less specific prefixes were stretched over the 920 files. More specific prefixes are small enough to not cause any issues during the duration of one partial scan.

These prefix lists can be used to create randomized IP Addresses for each prefix. Further, we do not need to care about the resulting address list order, as yarrp randomizes the packet sending per address and TTL. Each prefix file contained about 1.251 target prefixes

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

ranging from /36 to /48, resulting in about 4.851.482 target addresses per file. At worst, each file should take about 40 minutes to scan.

3.4 Full Scan Analysis

From this first project, we can extract a lot of first features, most of which will be used in a later stage to represent a baseline. As yarrp does traceroute to all given target addresses, we get a list of responses with the original target destination, sent and received Hop Limit.

All the lines need to be read and aggregated into usable routes, which makes analysis difficult. Keeping consistent and correct values about the found routes and unique lists of all routers as well as loops was tricky in regards to memory constraints. The analysis process had to be split into multiple runs as the dataset simply did not fit into memory. The split datasets were merged afterward into one single dataset. Yarrp produced around 3TB of probe responses from this first IPv6 scan for us to analyze.

Each scan is organized as a project, containing various files storing data about a scan. A project can be in different stages of analysis. The further we are with scanning and aggregating data, the more information we can store. Most information is only available after multiple persistence scans. For example, we can only talk about imperiled prefixes after there have been verifying persistence scans after a given time has passed.

3.4.1 Feature extraction

There is a multitude of features to extract from these output files. Most features are available after creating a full route to a destination. To be able to compare our results with Xia et al. fully, we followed their analysis process closely. However, we still modified analysis steps in the process to improve the created datasets and gain as much insight as possible. An overview of the extracted basic features can be seen in Table 3.3.

After all routes have been collected, we can start our loop evaluation. Each route is inspected hop by hop to check for a possible loop in the path to the destination address alongside other features. If we encounter a hop a second time and we do not encounter the destination address in any further packet, there is a high chance for a routing loop on this path.

If we, however, encounter a hop a second time but reach the targeted destination, we have encountered a load balancer with unequal path lengths, as shown in example 2.5 and we mark it accordingly. Further, for each hop, we check how many answers there have been. If there has been any hop with more than one answer, we mark the route to contain a spammer for further investigation.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

	Feature	Description
Route	Includes Spammer	If at least one hop answered more than once
	Has Load Balancer	If the route appears to be on path of a load balancer with unequal path distances.
	Has Loop on Route	If a forwarding loop is on path to a destination.
	Is Shadowed	If the route is shadowed by a loop, therefore not reachable.
	Is Imperiled	If a route is imperiled by loop, still reachable but can be attacked.
Loop	Full Loop	If Loop is a full loop, not missing a hop.
	Loop Members or Loop Lengths	Number of routers within this loop.
	Member of ASN	List of ASN the routers within the loop are part of.
	Shadowing Destinations	The destinations the loop is shadowing.
	Imperiled Destinations	The destinations the loops is imperiling.
Routers	Member of Loops	A list of loops the router is part of.
	Shadowing Destinations	The destinations the router is shadowing.
	Imperiled Destinations	The destinations the router is imperiling.
	Member of ASN	A list of ASN the router is part of.

Table 3.3: Available basic Features to extract per Route

Unique Loop Hash

For each loop encountered, we can create a hash unique for each sorted combination of routers. We call this feature the loop identifier or loop ID in short. This makes it possible to distinguish router combinations and create aggregations based on unique routing loops.

Each router within a loop is sorted ascending by its IP address. After that, for each IP address, all octets within the address are taken and used to calculate the MD5 hash for comparison. As we only deal with full loops, that is, only loops to which we know all hops, we do not have to worry about missing IP addresses between hops. Listing 3.1 shows different loop hashes with all containing routers. Even though they only differ in one octet, we get vastly different hashes.

```
Loop ID: 345925a5e063902488c6d1dd679d31af:
    > 2001:db8::2 - 2001:db8::1:99
Loop ID: ed3016fe470d0cef7588f1d8eef4c7c8:
    > 2001:db8::3 - 2001:db8::1:99
```

Listing 3.1: Example of Unique Loop Identifier hashing

There also exist some aggregated features over a single scan. A dictionary containing the mapping of a loop ID to its routerzz members, as well as a reverse mapping, are

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

maintained and stored for further analysis. For each router, we keep a list of shadowed prefixes, the prefixes we could not reach as the path to it contained a forwarding loop.

ASN attribution

Given that the target creation started from a BGP announcements list containing all ASN numbers, we can attribute all routers and loops to these ASNs. From these prefix-to-ASN announcements, we can build a tree structure starting from the bigger, broader prefixes containing all smaller, more specific ones.

This still isn't enough to speed up the ASN attribution, as there are a lot of BGP announcements that aren't a child prefix of any other prefix. The root node, therefore, contains a big number of first branches, slowing down any ASN lookup as we need to find the fitting branch for an address.

We thus implemented a prefilter that extracts the first octets for IPv4 and the first two octets for IPv6 and sorts the given prefixes. Essentially, a dedicated tree is created for each unique starting octet of a network. For example, for the address 172.12.0.2, we select "172" as the prefilter. For an IPv6 address like 2001:db8::5, the first number "2001" would be our selected prefilter. This makes building the tree, as well as looking up billions of addresses, quick enough to be integrated into our analysis process.

3.5 Persistence Scans

At this point, we have extracted a lot of data from one point in time, creating a snapshot of the current situation of routing loops for both IPv4 and IPv6. To create concrete evidence of persisting routing loops within this data, we need to verify all findings and check if the loops and router are persistent. Thus, we need to extract these data points and periodically trace the destinations for these given data points.

3.5.1 Persistence 1 - 10

Xia et al. originally traced all found shadowed destinations again, every twelve hours for five consecutive days. This is feasible for IPv4, however again, due to the bigger address space of IPv6, this would be too time-consuming. Thus, we had to reduce the number of targets to trace multiple times a day.

For each loop found within the full scan, we took at most five shadowed destinations to trace to. Further, we try to select targets from various different prefixes. Our selection preferred targets in /32 prefixes not already on our list. This resulted in 70.343 instead of 476.866 targets for IPv4 and 351.416 instead of 124.676.281 targets in IPv6. Just like Xia et al., we traced these targets twice a day for five consecutive days, resulting in ten Persistence scans. Further, these persistence scans were done from two additional vantage points, one in Virginia, USA and one in Sydney, AUS. The exact same target list was used to check if the loops are consistent over different continents.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

For each scan, we did the exact same loop analysis as we did for the full scan, thus resulting in a multitude of projects with loops and their router members. Over the course of ten persistence scans as well as the full scan, we could now check for overlap in loops and routers, resulting in a list of persistent elements.

3.5.2 Fluctuations in Loop Persistence

The overlap of looping routers in both IPv4 and IPv6 was as hoped. Almost all of the found routers in each persistent scan were already present in the full scan. This overlap can be seen in Figure 3.12. The category *Total Routers* is the total number of looping routers, the category *New Routers* shows routers previously not seen in the full scan and the category *In Full Scan* shows routers that we have seen previously in the full scan. Further, the total number of routers across the ten persistence scans was stable.

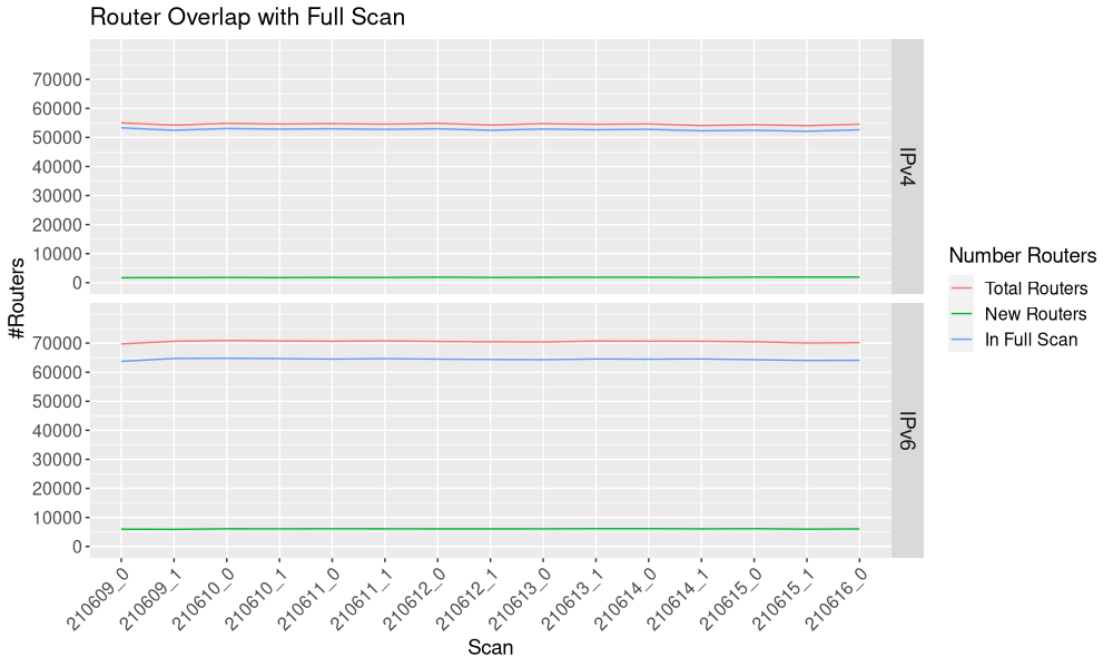

Figure 3.12: Overlap of Routers between each Persistence Scan and the Full Scan

This continues in IPv4 with the persistent loops. Again most of the found loops were already known from the full scan. For IPv6, the data shifts drastically. More than half of the loops were not previously found in the full scan, resulting in a low number of persistent loops. This is depicted in Figure 3.13. Again the category *Total Loops* is the number of unique loops found, the category *New Loops* are loops not seen in the full scan and the category *In Full Scan* are loops we already encountered in the full scan.

This shift in persistent loops does not make a lot of sense given the number of persistent routers of IPv6. We decided to look into this problem, suspecting a number of routers with slightly different IPv6 addresses than previously found, resulting in new loop hashes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

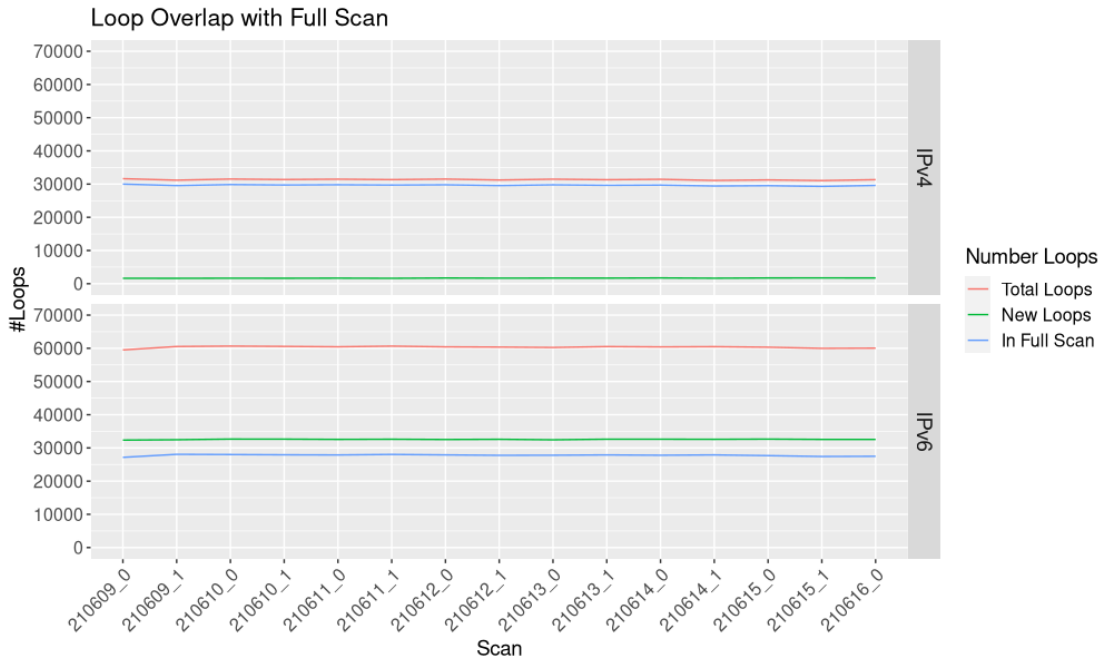

Figure 3.13: Overlap of Loops between each Persistence Scan and the Full Scan

First, we build a set containing all Loops with their member routers that are in all ten persistence scans. Each loop within this set is compared against every loop found within the full scan of the same length.

We compare all members of both loops and *Loop Length - 1* members must match for further analysis. Further, as we have the set of shadowed destinations for both the persistence scans and the full scan, we matched if the shadowed nets found in all ten persistence scans are also shadowed by the found candidate loop in the full scan.

The last two members that are not equal are then compared on octet level and we record the octet difference between both router IPs. We found 5.449 loops that matched our process, but in the end, we decided not to add them to our persistence set as we were not entirely sure about the accuracy of this process. This process still posts an interesting possibility of increasing the number of loops in the persistence set, especially with ISPs continuing the use of dynamic IP allocation instead of static addresses.

In line with Xia et al., we constructed the list of persistent loops and routers. For all eleven scans, so the full scan and all ten persistence scans, we select all loops and routers that are in all of these eleven scans. These two lists of persistent items can now be used to redo the entire analysis of the full scan to determine the actual shadowed as well as the imperiled nets.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Reference	Address	Overlap with A
A	2001:0db8:1000::1	-
B	2001:0db8:1000::100:1	48
C	2001:0db8:9000::1	32

Table 3.4: Bit Overlap for Density Calculation up to Prefix Edge of /48

3.5.3 Density

Another aspect of the shadowed and imperiled nets is how localized they are, as in, are the nets spread thin over a large number of prefixes or is there a common prefix they mostly reside in. We wanted some kind of metric to see if and how strongly localized the found nets are. With the size of the data, it is currently impossible to check if, for example, all nets shadowed by one single loop are within a common prefix. In the end, we came up with a density function as follows in 3.6,

$$\text{density} = \frac{n}{2^{p-b}} \tag{3.6}$$

where n is the number of relevant prefixes, p is the prefix length of these prefixes and b is the number of bits overlapping of the prefixes used for calculation. This calculates the percentage of the occupied number of addresses out of the maximum number of sub prefixes possible within a prefix of the size of the bit overlap.

All prefixes within this relevant set are compared to each other in regards to similar bits from the beginning of the address up to the first changing bit. This first changing bit is the b for the prefix set for the equation. An example of this can be seen in Table 3.4. Addresses A and B only differ in the 13th octet, so the difference in bits is up to the prefix edge, which in our case is /48 for IPv6. Address C looks rather similar to addresses A and B. However, the difference of one bit in the fifth octet marks the end of the combined bit overlap, which is 32 bits in this case.

3.5.4 Persistence 50

For this last persistence check, we conducted multiple ZMap scans to 50 random addresses per shadowed prefix from the full scan. This is done to get a better understanding of the sizes of shadowed prefixes, as well as to check if the entire prefix is shadowed or if there is actually a more localized problem. It further gives us more additional data points to verify if a shadowed prefix is actually shadowed or not.

IPv4 prefixes are probably smaller than /24, there are a lot of smaller prefixes in the wild, but they most likely will not be announced or the BGP announcements wont be propagated to most peers. We don't really know what to expect from IPv6, as there are a lot of different strategies for ISPs to choose from. *RIPE NCC* recommends either a one-fits-all approach by assigning every customer a /48 prefix, regardless of business or

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

residential customer. An alternative recommendation is to distribute /48 prefixes for business customers and reducing the prefix size to /56 for residential customers if the ISP so desires. Everything more specific than /56 is highly discouraged, but *RIPE NCC* further references a guideline posted by *The Broadband Forum*, which suggests at least a /60 prefix but recommends a /56 prefix for residential customers[37, 38].

First of all, we selected all candidate shadowed prefixes from the full scan. We did not filter if they have already been confirmed as persistent by the Persistence 10 scans. This resulted in all 124.676.281 prefixes selected for IPv6 and all 476.866 Prefixes for IPv4. We made sure to distribute the scan over the given prefixes as ZMap only does probe scans without automatically permutating the input space. Therefore, we implemented a scatter mode to parse the list of shadowed prefixes, trimming them to our given target prefix size and sort them into buckets of prefixes with length 36 for IPv6 and 16 for IPv4. As these buckets have a different number of target destinations in them, we merge multiple buckets together to try and reach an equal number of targets in all of them. To create the target list, we select the now merged buckets in a round-robin fashion and add an IP address from each of them to the target list. This way, we spread all target destinations from the buckets over the entire scan duration, minimizing the impact for routers that serve one particular prefix.

To now reach the 50 random IP addresses, we can use the file containing the target prefixes from the previous step and create different target IP addresses for each prefix. Instead of creating one big target file, we conduct 50 individual ZMap scans, deducting the target file from the list of target prefixes. The target file creation itself is repeatable. All target creation is done via a seeded pseudo-random number generator. The resulting CSV files feed back into our analysis process, where we look at the number and type of each response.

3.5.5 Amplification Factor

The amplification factor is a metric to determine how strong the impact of a single packet in a given loop is. It is rather simple to compute by using the following equation

$$\text{amplification} = \frac{255 - h}{l} \tag{3.7}$$

where h is the number of hops to the loop and l is the length of the loop in question. The result ranges between 0 and 255 as it is limited by the possible values of the hop limit field within an IP packet. It could be used in combination with the number of imperiled nets per loop to compute an impact metric. The loop length is the most important factor for the amplification factor. The best case for an attacker are loops with a low loop length and a high number of imperiled targets involved. The actual distance to the loop is not important. An attacker can compensate for this by choosing a vantage point near the chosen loops.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

3.5.6 Yarrp Toolkit

yarrp_toolkit is the main program used to analyze the large amount of data collected by yarrp. For the full scan in IPv6, we collected around 2.7 TB of data, which we need to parse and extract information about routing loops. A fast solution was necessary. We started off with simple python scripts but realized soon that the ever-growing data set and the time to parse and analyze was too high.

The sequential performance sometimes increased up to 30x the performance of the existing python script. Further, we could focus on the actual implementation of analysis instead of search for more performance optimizations or even multithreaded approaches in python. The toolkit was focused on taking all input files at once, parsing them and leaving a single output directory with all information necessary for further steps, which worked perfectly for the IPv4 data set.

In IPv6, however, it was not possible to parse all files at once as the RAM consumption was not manageable. Therefore, we implemented the analysis process to first check and try to read an existing project before parsing the given files. This worked well, and we could read each file on its own while still writing to a single project. The full analysis process still took around two days for the IPv6 full scan to analyze. Thus, we started four analysis processes in parallel, each with its own working directory. After all analysis processes were finished, we merged the four projects and were left with a single project containing all analyzed data about the full scan, ready for further aggregated analysis.

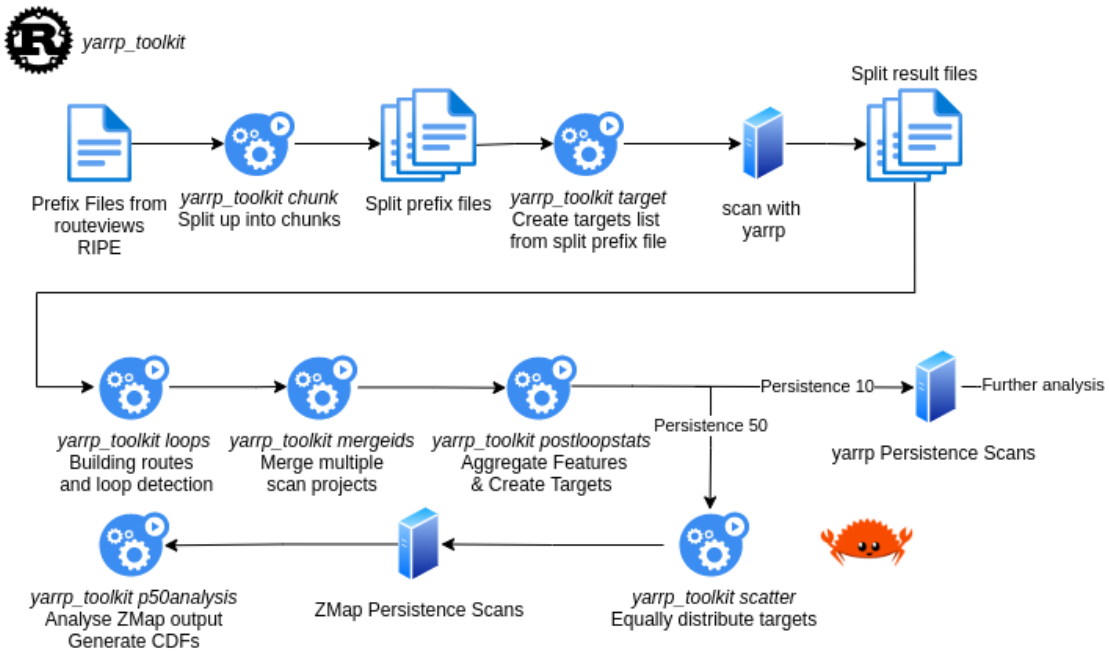

Figure 3.14: Flow of data between different Modes of the Toolkit

yarrp_toolkit is split into multiple different modules for different stages in the measure-

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

ment. Figure 3.14 demonstrates the preparation and analysis of a given scan and when the individual modules come into place.

The **chunk** mode was the preparation step for the initial target generation. We needed to parse the RouteViews BGP announcements, sort, filter and split the found prefixes into equal chunks so we could distribute the load over all prefixes.

The full target lists were still large in size. To save on disk space, we only kept the prefix lists used to generate the target lists. The **target** mode is then used to generate a target list from a target prefix list. By setting the seed for the target generator, a reproducible target generation can be ensured.

After tracing the targets with yarrp, we end up with a large amount of data to parse. The **loops** module reads the yarrp files either in a single batch during IPv4 or persistent scans or file by file for a large-scale IPv6 measurement. Multiple files containing the unique loops with its members, the shadowed destinations per loop and many other features are extracted.

If multiple projects need to be merged, e.g., after a file by file IPv6 scan analysis, the **mergeid** mode handles this. All files are read and merged accordingly, e.g., unique loop lists will not contain any duplicates and the router to loop dictionary will be updated if any new loops have been merged into a project.

The **postloopstats** module generates additional stats, like the unique loop lengths, the density calculations, the ASN attribution and many more. Further, it features the target generation for the persistence scans.

If creating a target list for the Persistence 50 scan for ZMap, the **scatter** mode sorts the target addresses into prefix buckets to spread the prefixes over the entire duration of the scan.

The resulting ZMap output files can be analyzed by the **p50analysis** mode to extract the number of response types for each probed prefix.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

CHAPTER 4

Results

In this section, we discuss the results found in the measurements. They are sorted in the order we performed them, starting with the basics of the full scan and reaching various persistence scans. Where possible, we will make a comparison with the data from Xia et al.[16] as we tried to match their methodology as closely as possible.

4.1 Full Scan

First up is the data for the full scan. The IPv6 scan ran from 2021-02-16 to 2021-03-12 with 4.463.363.543 traceroutes performed in to total. The IPv4 scan was done within a few hours on the day of 2021-05-09 with a total 11.851.437 traces scanned. We will explore the basic features of this scan and aggregate the data with the already used ASN dataset as well as the persistence scans.

4.1.1 Basic Features, Loop Characteristics

For both scans, we extracted all the features as described in our methodology. An overview of the raw numbers is featured in Table 4.1. In both IPv4 and IPv6, we found a large number of total loops. Out of the roughly 12 million traces of IPv4, 476.866 of them contained a loop, shadowing the target destination. In IPv6, we found around 125 million loops within the 4.5 billion traces done. Percentage-wise this is roughly around the same value with 4.03% of routes containing a loop for IPv4 and 2.79% for IPv6. This is in line with Xia et al.. Their initial measurement showed 207.891 candidate prefixes, or 3.77% of all routes, containing a loop on tehir path.

These numbers, however, do not represent unique loops. Rather, they are a count of total target destinations unreachable due to a loop on the way. There are large differences between loops encountered and actual unique loops, a visual representation of this can be found in Figure 4.1. Total loops is the number of targets with a loop on the path,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Feature	IPv4	IPv6
Routes	11.844.517	4.463.363.543
Total Loops	476.866	124.676.281
Total Loops in %	4,03%	2,79%
Unique Loops	37.835	204.530
Unique Routers	64.793	146.543
Load Balancers	0	637.260
Spammers	219.594	7.671.554
ASN Involved in Loop	9.675	4.777

Table 4.1: Basic features of both IPv4 and IPv6

while the number of unique loops is the count of unique Loop IDs found within the entire scan. To recall, we take the hops within each of these loops and create the unique Loop ID by hashing the individual router hops as described in section 3.4.1.

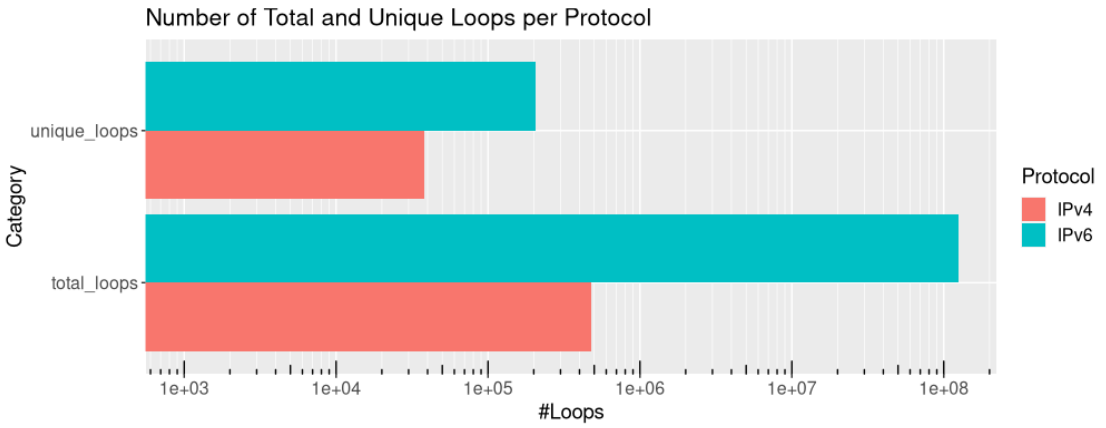

Figure 4.1: Total vs Unique Loops found for both IPv4 and IPv6

Overall there are significantly more loops in IPv6 than in IPv4. This was expected as the target space is much larger, and the node density lower. Of the 467.866 total loops in IPv4, around 8%, or 37.835, remain as unique loops. The decrease in IPv6 is higher. Of the 124 million total loops, only 0.16%, or 204.530, unique loops remain. Each loop in IPv4 has, on average, around 12 appearances, while in IPv6, each loop appears around 609 times on average. This is most likely due to each operator owning a much larger portion of address space in IPv6 compared to IPv4, where every IP address is looked at twice before assigning it.

The same trend is seen with the routers involved in loops. There are over twice as many

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

routers involved in IPv6, with 146543 routers over 64793 routers in IPv4. However, the number of loops a router is involved in is similar. Around 90% of the routers are only involved in a single loop. This is depicted in the CDF in Figure 4.2. In both protocols, there is a small number of routers involved in more than ten loops. For IPv4, there are only 23 routers that are involved in more than ten networks. In comparison with IPv6, that is rather low as there are 744 routers seen in more than ten loops. There are 86 routers with even more than 100 loops, 19 of those are seen in over 200 unique loops.

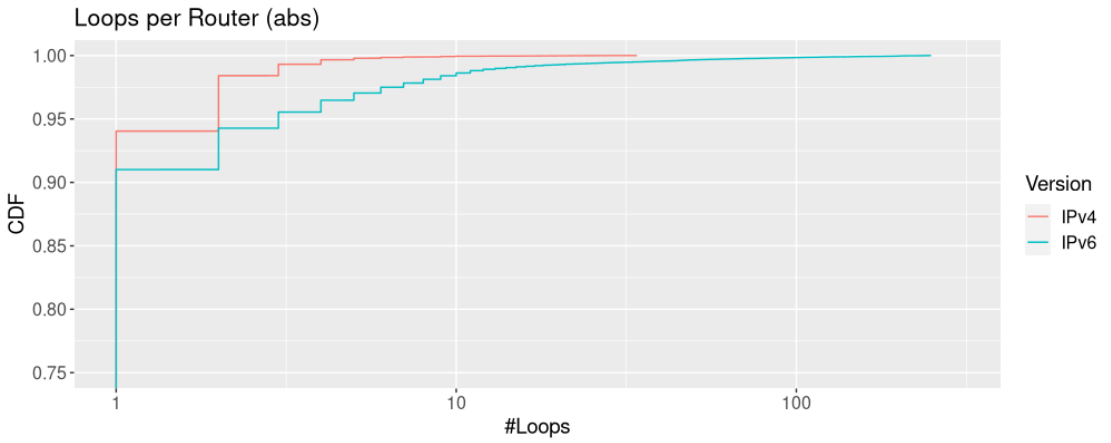

Figure 4.2: CDF of Amount of Loops a Router is Member of, CDF from 75% to 100%

Further, we evaluate the number of spammers and load balancers on all explored paths to the destinations. Spammers are counted on targets with or without loops on the given route, while load balancers are only counted on paths without a loop. We found 219.594 spammers on all routes to the IPv4 targets while we detected 7.671.554 spammers on IPv6. These are absolute numbers of spammers encountered on routes and do not count unique spammers. The most likely explanation for the existence of these spammers is the paper on Weaponizing Middleboxes[27] from Bock et al. It is not clear why we did not detect any load balancers on IPv4. We need to investigate this further and improve the detection of those entities. For IPv6, the detection clearly works. We found 637.260 load balancers on paths to reachable hosts.

Loop Lengths

Another basic but important feature is the length of each detected loop. One would assume that this metric starts with a minimum loop length of two and carries the most loops found. After that point, the number of loops only decreases the higher the loop length.

However, for both IPv4 and IPv6, we found a high amount of 1-hop loops, which is not fully explainable. There are multiple possible reasons: Again, the underlying cause could be middleboxes[27]. Alternatively load balancers could be the reason. However, this does not explain the high amount of 1-hop loops. For IPv6, the possibility of misconfigured

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

transition mechanisms is also a point worth considering. Even though public systems like 6to4 are not passing a lot of traffic anymore, ISPs still use similar techniques like 6rd in their private infrastructure.

Other than that, the loop lengths are as expected. The highest number of loops is observed at loop length two and the number of loops decreases the longer the loop is. Looking at the length for IPv6, we again compare the total number of loops against the number of unique loops, this is depicted in Figure 4.3 By a far margin, most loops are of length two, about 80% of them, second in place are 1-hop loops, with further decreasing numbers the longer a loop is. The numbers dwindle fast when looking at unique loops. 1-hop loops are down in numbers and even below loops with lengths of three and four. Still, there are more loops of length one than anticipated in the beginning. There is a noticeable spike at loop length 12 for total loops, which is absent when looking at the unique loops. This datapoint needs further investigation. At this point in time, we do not have an explanation for this.

For IPv4, the data is displayed in Figure 4.4. In IPv4, most loops occur as 1-hop loops and make up for about 80% of all found loops. Loops with two hops are in second place and the number of loops decreases the longer the loops become. As with IPv6, the numbers heavily decrease when comparing the total number of loops to the unique loops. Loops with length one are now below 2-hop loops but are still represented strong.

In contrast to Xia et al., 89,4% of the found loops in their data were of length two, with 10,4% being of length from three to nine. However, they stated, *"We found that a few traces contained the same address appearing continuously, which could be caused by a firewall. We also filtered out these traces in our study."* in section 4[16]. It appears they have encountered a low number of 1-hop loops, and discarded these data points. This is in contrast to our findings with a high number of 1-hop loops found in both protocols, which raises further questions.

4.1.2 Persistent Scans

The ten persistence scans allow to investigate for the persistence of routers, loops and promote the candidate prefixes to shadowed prefixes. With the newly acquired set of persistent loops, the original data is checked for non-candidate prefixes that have at least one router included in a loop along the way. These prefixes are called imperiled prefixes and form the set of prefixes that are vulnerable to overload attacks of routers included in a loop. First, the number of persistent entities is determined.

We have already taken a look at the overlap in section 3.5.2 and highlighted the low number of matching and the high number of new loops in comparison to the full scan.

Selection of persistence

We explored two different approaches to select persistent entities. The first one was the strict usage of loops only visible in all eleven scans like Xia et al. The second approach

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

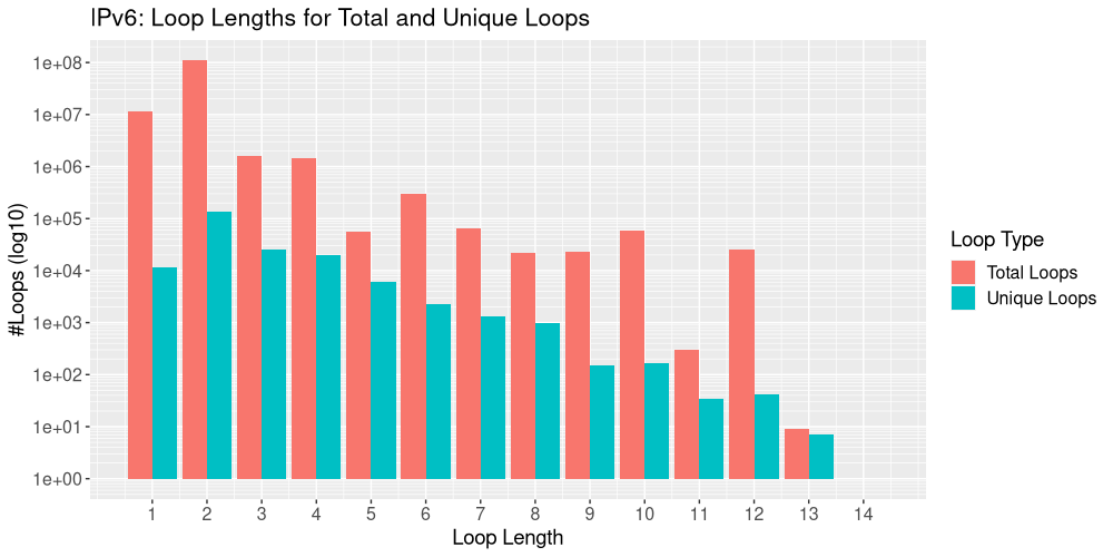

Figure 4.3: Number of Total and unique Loops per Length for IPv6

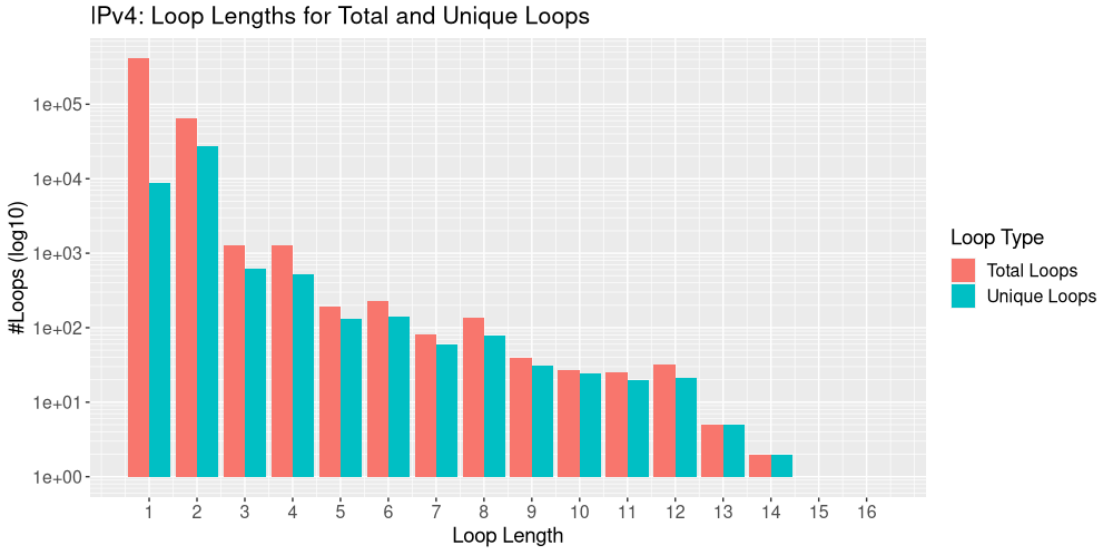

Figure 4.4: Number of Total and Unique Loops per Length for IPv4

was to take every loop or router that also appeared in the full scan, disregarding how many persistence scans the entity appeared in. This built upon the fact that more time passed between the full scan and the persistence scans than between the first and last persistence scans. Thus, a greater possibility of change exists between the full scan and the first persistence scan.

In the end, we choose the method of Xia et al. to compare the data between both scans. Our method would have increased the number of selected routers for IPv6 from 53.666

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

	Xia et al.	In Full Scan
IPv4 Loops	25.687	31.699
IPv6 Loops	21.227	32.930
IPv4 Routers	45.841	53.380
IPv6 Routers	53.666	69.028

Table 4.2: Numbers of Routers and Loops by using different Persistence Selection Approaches

to 69.028 and increased the number of loops from 21.227 to 32.930. Figure 4.5 shows the selection of routers per occurrence in scans for both IPv4 and IPv6. Most of the elements we gained by using all entities appearing in the full scan were only missing in one scan. None of the loops or routers only seen once were part of the full scan. Thus, we think the approach using all loops and routers seen in the full scan is the better fit and will be explored in future work. Further, Table 4.2 shows the difference in numbers between the two approaches.

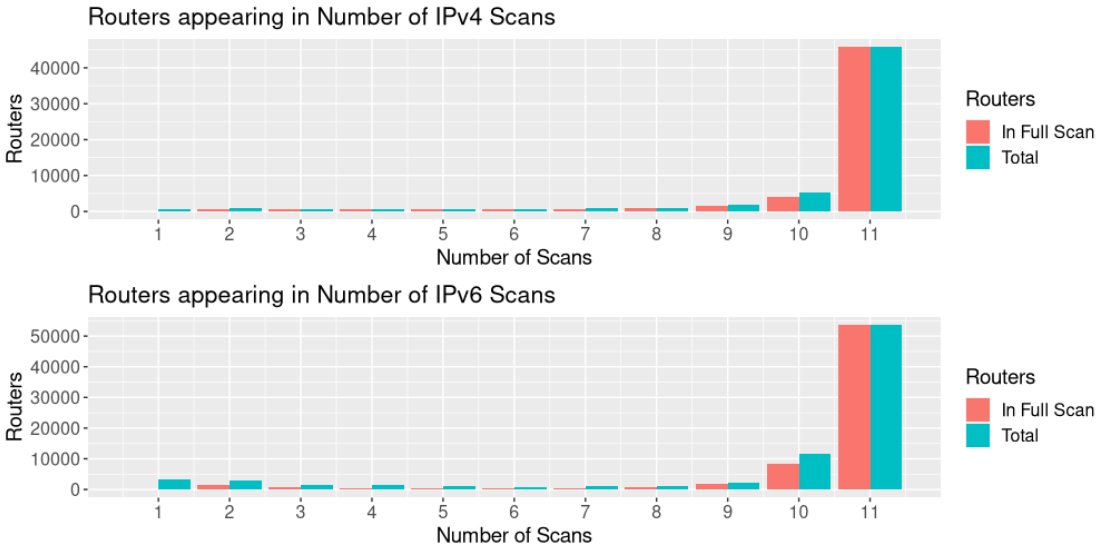

Figure 4.5: Routers seen in n Persistence Scans and the Full Scan

Table 4.3 showcases these final numbers of persistent loops and routers after the selection process. Of the 37.835 unique loops in IPv4, only 25.687 remained in the persistent set. For IPv6, the decrease is even worse due to the already discussed possible dynamic nature of the allocation of residential prefixes and, therefore, fluctuation seen in our graphs. The number of unique loops reduced from 204.530 to 21.227 persistent loops, only about 10% remain as loops. In the case of routers in IPv4, the numbers reduce in the same fashion as the number of loops does. Of 64.793 routers seen in loops, only 45.841 have been seen again. For IPv6, the number of routers has not decreased as hard as the number of loops. From 146.543 previously seen routers, 53.666 routers are still members of loops.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

	IPv4	IPv6
Unique Loops	37.835	204.530
Persistent Loops	25.687	21.227
Unique Routers	64.793	146.543
Persistent Routers	45.841	53.666

Table 4.3: Numbers of Data Points from the Persistence Scans

The histogram of loop lengths of persistent loops is not different for the loop lengths of unique loops, see Figure 4.6. Again most loops are in the 1-hop or 2-hop category and the number of loops decreases the longer the loops become. There are no loops with a length over nine in IPv6, which might be an indication of an error on our side. At the time of the full scan for IPv6, we did not yet fully understand all parameters of yarrp. Instead of using the fill rate correctly, we set it to the same number as the maximum hop limit and were rescanning targets that were still looping after 18 hops. This introduced data holes, especially at around 18 hops which might cause an issue here. This was noticed after the IPv6 full scan finished and was adjusted in every scan afterward. The IPv4 full scan had the correct parameters set and was not impacted in the same way.

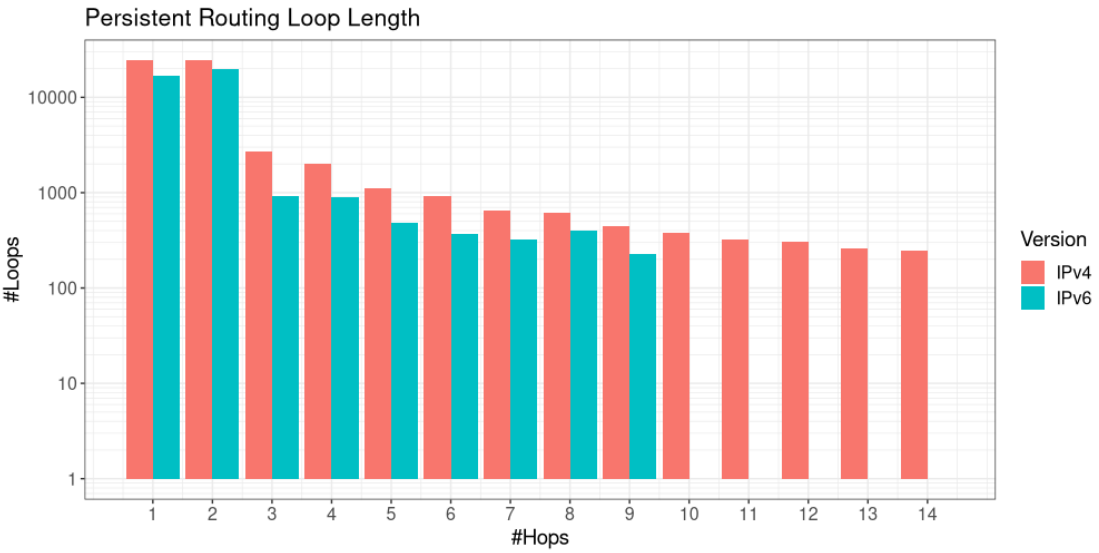

Figure 4.6: Number of Persistent Routing Loops per Length for both IPv4 and IPv6

4.1.3 AS involvement

As previously mentioned, there are a lot more ASes involved in IPv4 than in IPv6. There are 9.675 ASes with loops in IPv4, while there are only about half of that with 4.777 involved ASes for IPv6. First, let us take a look at ASes and their unique loops as well as the unique routers from the full scan without the persistence checks. We looked up the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

ASN for each router IP found in all unique loops, which we can collect for all members of a loop to attribute the loop to one or multiple ASes. With the loop attribution, we can count how many ASes are involved in every loop and aggregate those numbers.

There are loops and routers for both IPv4 and IPv6 for which we were not able to attribute an ASN. For IPv4, there are 808 loops and 992 routers from the full scan. Of those are 482 persistent loops and 643 persistent routers that could not be attributed to an AS. In IPv6, the numbers are 3866 loops and 1513 routers from the full scan, as well as 304 persistent loops and 363 persistent routers for which we could not find an AS. We marked them as *undefined* in our data set but stripped them from the following graphs as they might not be controlled by the same organization.

The following four graphs show the Top-10 ASes with their attributed loops and routers, sorted descending by the number of loops. In Figure 4.7, we can see the number of unique loops and unique routers from the full scan alone for each AS. AS3320 has the most loops, followed by AS174 and after that, the number of loops per AS decreases fast. A similar number of loops and routers might mean there are a lot of loops shared with other AS, like AS3320 and AS15557. On the other hand, ASes with a much higher number of routers than loops might indicate that the loops are mostly contained within the AS itself.

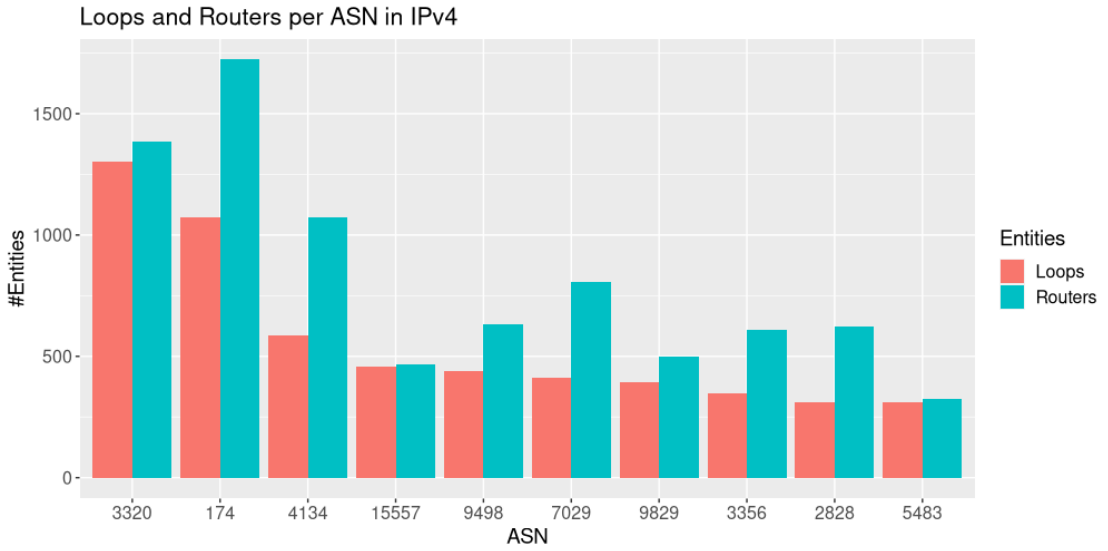

Figure 4.7: IPv4: Number of Unique Loops and Routers per ASN, sorted by Number of Loops

Comparing these numbers with the same chart, but this time using only the loops and routers that have been confirmed as persistent, we can see much of the same. Figure 4.8 shows that the Top-10 AS stayed relatively stable, with nine out of ten ASes stayed in the Top-10. AS9498 went out of the chart, AS8220 took its place. Otherwise, the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

positions shifted around a bit, but the ratio of loops to routers for the previous members stayed the same.

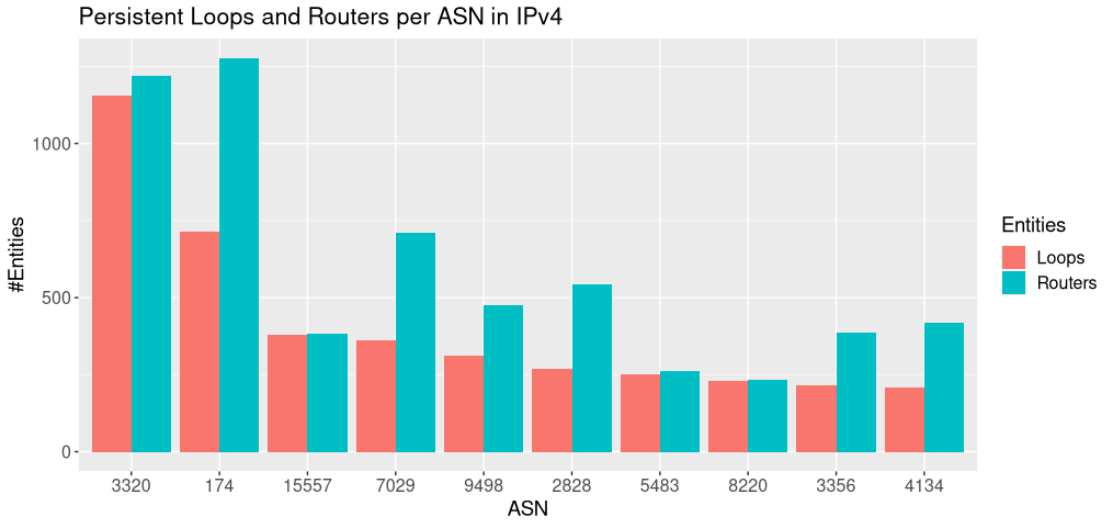

Figure 4.8: IPv4: Number of Persistent Loops and Routers per ASN, sorted by Number of Loops

With IPv6, the story is different, which can be seen in Figure 4.9. Again we are looking at the unique loops from the full scan. A lot of ASes show a high number of Loops with a low count of routers to them. These ASes are probably not ISPs, but rather Content Delivery Networks or Transit Providers. Instead of serving residential or small business customers, they focus on transmitting traffic between larger organizations. They have a low amount of routers, but each of these is part of a bunch of loops with other AS. AS13030 is the main contributor to this chart with 53.541 loops but only 2.917 routers. On average, every router is involved in 18 different, unique loops. Still, there are three ASes with at least as many routers as loops. They are the exact opposit and probably the reason for the spike of loops on the other AS. AS3303 is one of these ASes and is involved in 46.677 loops with 46.690 routers.

Again, we are comparing the previous chart containing unique loops of the full scan with Figure 4.10 containing only the persistent loops and routers per AS. In contrast to IPv4, the chart remains stable, with nine out of ten ASes remaining in the chart. The situation looks completely different for IPv6. Five out of ten ASes are not in the Top-10 anymore. Further, the chart does not resemble the previous figure in any shape or form. The large number of loops for AS13030 and AS6939 are gone, and AS12956, previously on place three, did not make it to the persistence chart at all. AS3303 is in complete contrast and it carried a large number of routers into its persistence set while losing a large portion of its loops. This might be an indication of fluctuation. A lot of router IPs stay the same but are now carried by another residential customer in different geographical regions and served by different intermediate routers. Another possible cause for the generally

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

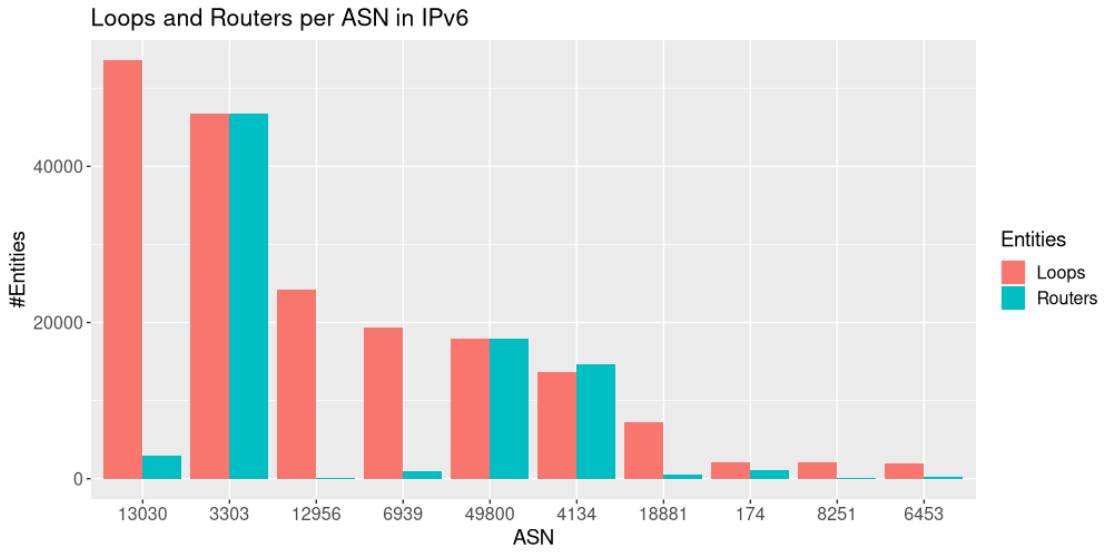

Figure 4.9: IPv6: Number of Unique Loops and Routers per ASN, sorted by Number of Loops

high number of unique loops could be the long duration of the IPv6 full scan. As the measurement lasted over 20 days, a lot of routes, as well as dynamic addresses used in residential connections, could have changed during the scan.

Further, AS3303 is no new name to us. It also was featured in the AS analysis during our long-term measurement in section 3.1. As can be seen in Figure 3.3, AS3303 took first place there as well. For further analysis, it would be interesting to confirm if this AS in particular has a high amount of fluctuation, both for the long-term measurement as well as in the full scan with the persistence checks.

Last, we take a look at the number of ASes involved in each persistent loop, which can be seen in Figure 4.11. Most loops are within a single AS, and these might be simple misconfigurations between an ISP and their residential or small business customers. All the customers receive already pre-configured equipment from the ISP. Thus, a small misconfiguration in the image of the equipment is likely to propagate to all customers, creating unique loops. Loops with two ASes involved are in second place, e.g., routes between two peering AS, one is announcing a larger prefix but forgets to set up their pull-up route, thus returning packets towards it via its configured default gateway. There is only a handful of loops with a higher number of involved AS. These are possible large-scale misconfiguration with static routes involved. Some loops exist with a higher loop length that could span multiple AS.

Not all loops are on this chart. IPv4 has one loop with five ASes and two loops with six AS, while IPv6 has one loop with seven ASes involved. These four loops are either not visible or have been removed from the graph as it would have been more confusing than helpful on the logarithmic scale.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

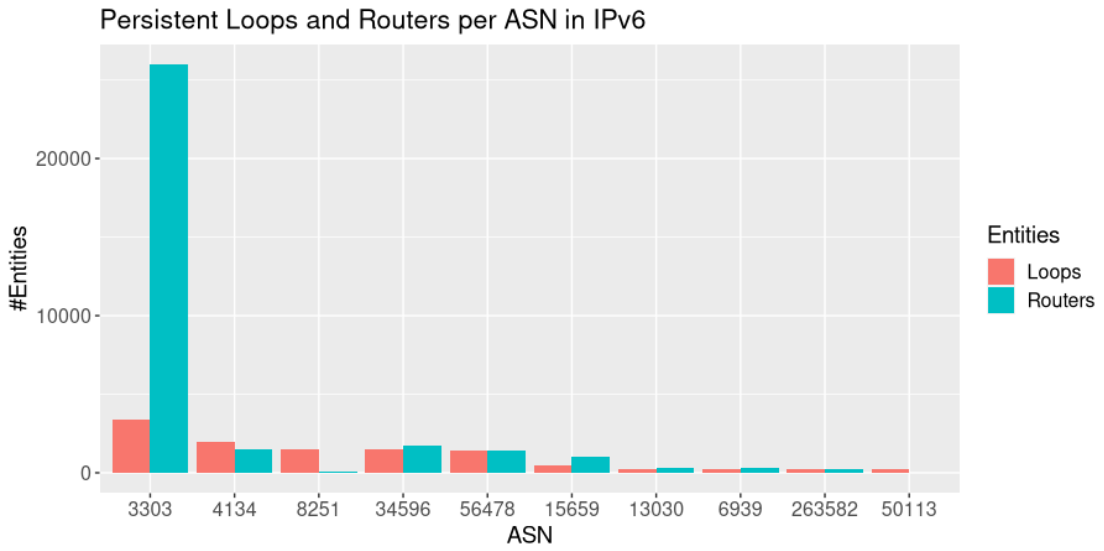

Figure 4.10: IPv6: Number of Persistent Loops and Routers per ASN, sorted by Number of Loops

4.1.4 Shadowed and Imperiled Prefixes

After collecting the persistent loops and routers and looking at the different ASes involved, shadowed and imperiled prefixes are investigated. A shadowed prefix contains a persistent loop on the way to it, thus being shadowed by a loop and cannot be reached by any internet-connected host, even though it is announced as such. An imperiled prefix, on the other hand, is a prefix that is announced and can be reached by any internet-connected host. However, on the path to the imperiled prefix is a router that is a member of a persistent loop and thus is threatening the connectivity of the prefix. Further, each shadowed prefix that indirectly endangers an imperiled prefix is called a dark prefix.

With the selected set of persistent loops and routers, we can now start the analysis of the full scan again. Each route in our data set is checked if it contains a router that is a member of a persistent loop. If it does, we mark it as an imperiled route so we can count them at the end. For each loop and router, we already count the number of shadowed and imperiled nets to be able to create aggregations based on shadowed nets per loop, for example. However, we cannot simply sum up the number of imperiled nets of all routers or loops as each imperiled net could be imperiled by multiple loops or routers. In Table 4.4, the number of candidate, shadowed, dark as well as imperiled prefixes are summed up.

Starting with IPv4, we generated and traced 11.844.517 target destinations. Of these 11 million destinations, 476.866 contained a loop and were thus marked as candidate prefix. Further, after the persistence scans, about 72,4%, a total of 345419 prefixes, of these candidate prefixes remained shadowed and thus were marked as shadowed prefixes. Of these shadowed prefixes, 80,4% were shadowed by a loop that imperiled at least one

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

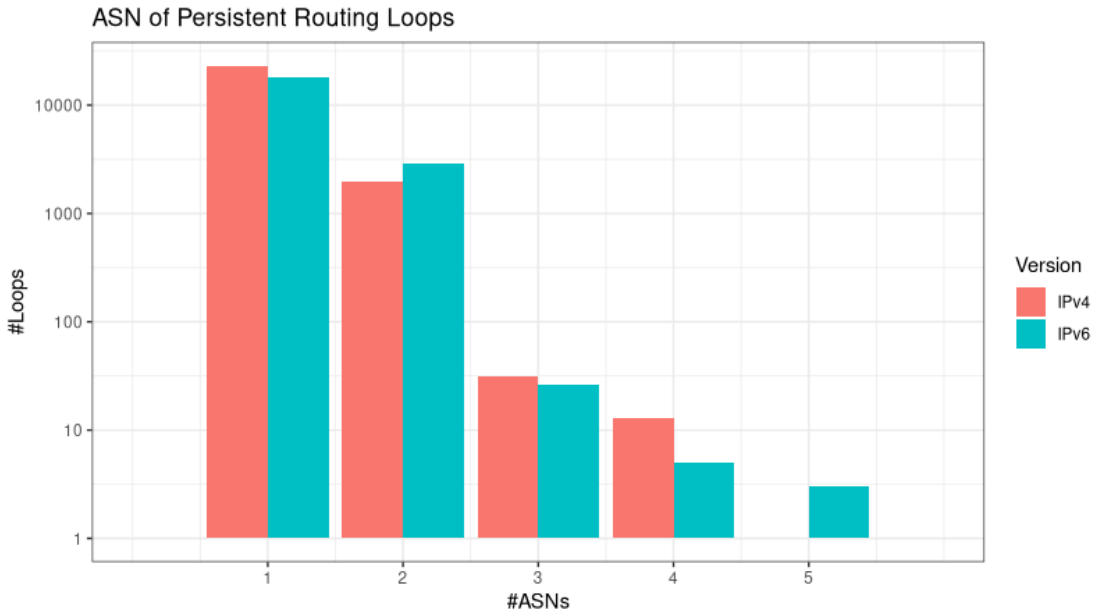

Figure 4.11: Number of Loops over Number of involved AS

other prefix. Thus we marked 277.646 prefixes as dark prefixes. A grand total of 9,6%, or 1.144.766 /24 prefixes, of the traced IPv4 internet are imperiled prefixes and are therefore vulnerable to potential overloads of routing loops.

For IPv6, over a time period of roughly 24 days, we traced 4.463.363.543 target destinations. Of these 4 billion addresses, 124.676.281 contained a loop and were again marked as candidate prefix. Again after finishing the persistence scans, 66,6%, or 83.015.076 prefixes, of the candidate prefixes were marked as shadowed prefixes. 69,3% of these Shadowed Prefixes were shadowed by a loop that imperiled at least another prefixes, resulting in 57.528.874 prefixes marked as dark prefixes. The entire analysis resulted in 1.390.853.262 imperiled prefixes, which represent around 31,2% of the traced IPv6 internet.

In comparison with Xia et al., out of their 207.891 candidate prefixes, they found 135.973 to be shadowed prefixes. This is around 65% which matches our numbers. However, they only found 0,78% of their traced prefixes, or 42.886 prefixes, to be imperiled prefixes. This is in contrast to our 9,6% or 1.144.766 prefixes we encountered.

It has to be kept in mind that these numbers are from our vantage point and might change depending on the location of the routing loops.

Next, we take a look at the distribution of the shadowed networks over the detected loops and routers. Figure 4.12 shows the number of persistent shadowed networks per routing loops for both IPv4 and IPv6. Most routing loops in IPv4 shadow a limited number of networks. Around 87% of the loops only shadow up to 10 /24 prefixes. Beyond that

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

	IPv4	IPv6
Total Targets	11.844.517	4.463.363.543
Candidate Prefixes	476.866	124.676.281
Shadowed Prefixes	345.419	83.015.076
in %	72,4%	66,6%
Dark Prefixes	277.646	57.528.874
in %	80,4%	69,3%
Imperiled Prefixes	1.144.766	1.390.853.262
% of all traces	9,6%	31,2%

Table 4.4: Candidate, Shadowed, Dark and Imperiled Prefixes for IPv4 and IPv6

point, we got a few loops shadowing a large number of networks, with 34 loops shadowing over 1000 prefixes and one loop shadowing more than 87.000 prefixes. In IPv6, about 75% of the loops shadow below 10 /48 prefixes. However, the loops above this 75% mark gain number of shadowed prefixes fast. The top 10% of the routers all shadow over 1000 /48 prefixes, with the number of networks further increasing. One hundred fifty-six loops shadow over 100.000 prefixes, with the Top-4 reaching over 1 million shadowed prefixes each.

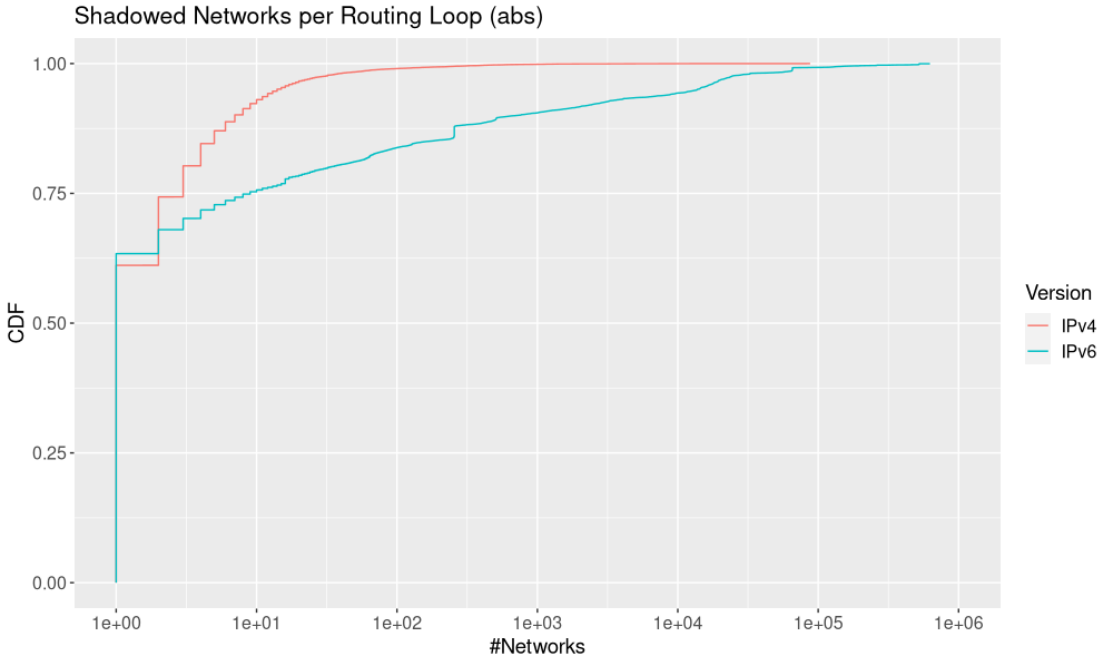

Figure 4.12: CDF showing the Amount of Persistent Shadowed Prefixes per Loop

These numbers per loop can also be split up per router. Basically, each member of the loop is attributed with the number of shadowed nets. This way, we can see the involvement of different routers within multiple loops, thus more shadowed nets per

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

individual router. We need to keep in mind that the number of shadowed prefixes here is getting inflated, as each shadowed net can be counted multiple times, once for each router involved in the shadowing loop.

The graph seen in Figure 4.13 for IPv4 looks rather similar to the previous one. Again we got around 87% of routers only shadowing under ten prefixes. The number of routers shadowing larger numbers of prefixes decreases fast and we found 36 routers with more than 1000 shadowed prefixes. This means, most of the loops with over 1000 shadowed nets per loop were 1-hop loops as we did not get any increase of router numbers over loops. In fact, the upper 65 loops sorted by shadowing nets are all of loop length one. Further, most routers are only in a single loop. Only one of the Top-10 routers has been seen in multiple loops.

Continuing with IPv6, the graph again looks similar. This time, around 80% of the routers shadow below ten prefixes. But again, the numbers of shadowed nets per router increase at a similar rate as the numbers of prefixes shadowed by loops. Again, 87% of all router shadow below 100 prefixes, and the top 10% shadow over 1000 nets. Over 100.000 prefixes are shadowed each by the top 326 routers found, and the Top-8 routers shadow over 1 million prefixes each. In contrast to IPv4, most loops in IPv6 have more than one hop at the top. Further, most routers are a member of multiple loops.

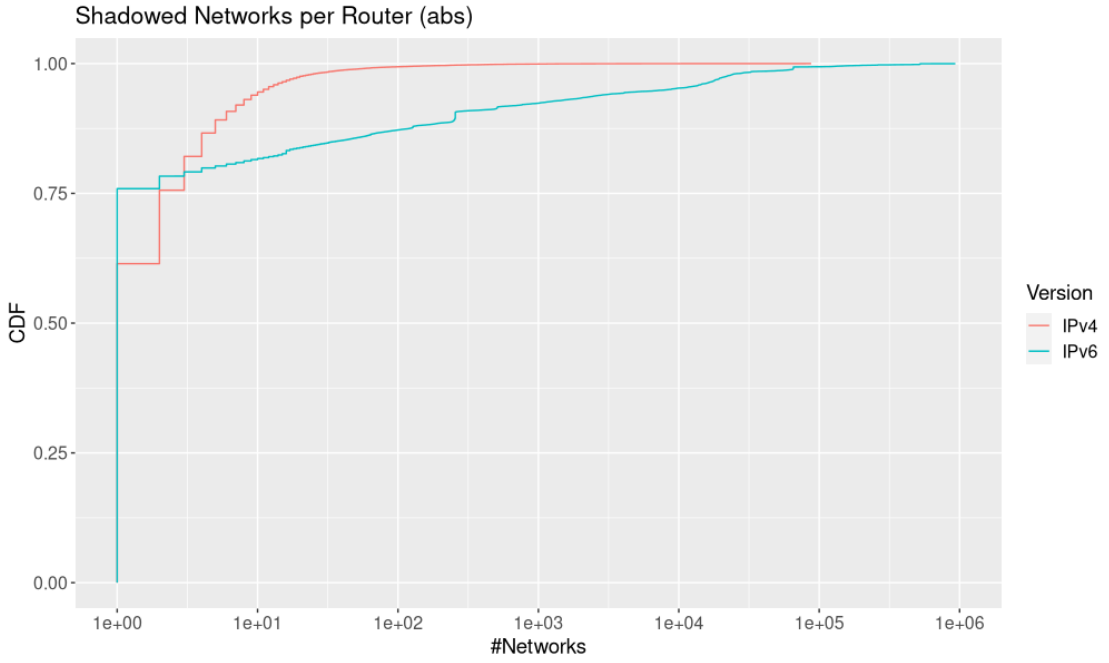

Figure 4.13: CDF showing the Amount of Persistent Shadowed Prefixes per Router

The same analysis can be done for imperiled networks. This time the relationship of prefixes to loops is different as one prefix can be imperiled by more than one router or loop. The data is depicted in Figure 4.14. For IPv4, the graph looks similar to the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

previous graphs for the shadowed prefixes. About 87% of loops imperil below ten prefixes. The top 84 loops imperil over 1000 prefixes each, with the top 13 gaining over 10.000 imperiled prefixes each. Coming in on top of the list is a rather unexpected entry. One loop imperils around 360.000 prefixes but shadows only a single prefix. We will look at this in more detail later.

For IPv6, most of the loops imperil at least one address. Still, around 50% of the detected loops imperil less than 10 entries. About 80% of all loops imperil under 100 prefixes each. Below that point, the number of imperiled prefixes per loop increases fast. The top 10% of loops have more than 1000 imperiled prefixes, and the top 540 loops are in front of 100.000 imperiled prefixes each. This trend only continues, as the top 23 loops are involved with over 10 million prefixes each. Despite the high number of imperiled, in both IPv4 and IPv6, the number of shadowed nets for these loops is quite low in comparison to what we have seen in the graphs about shadowed prefixes.

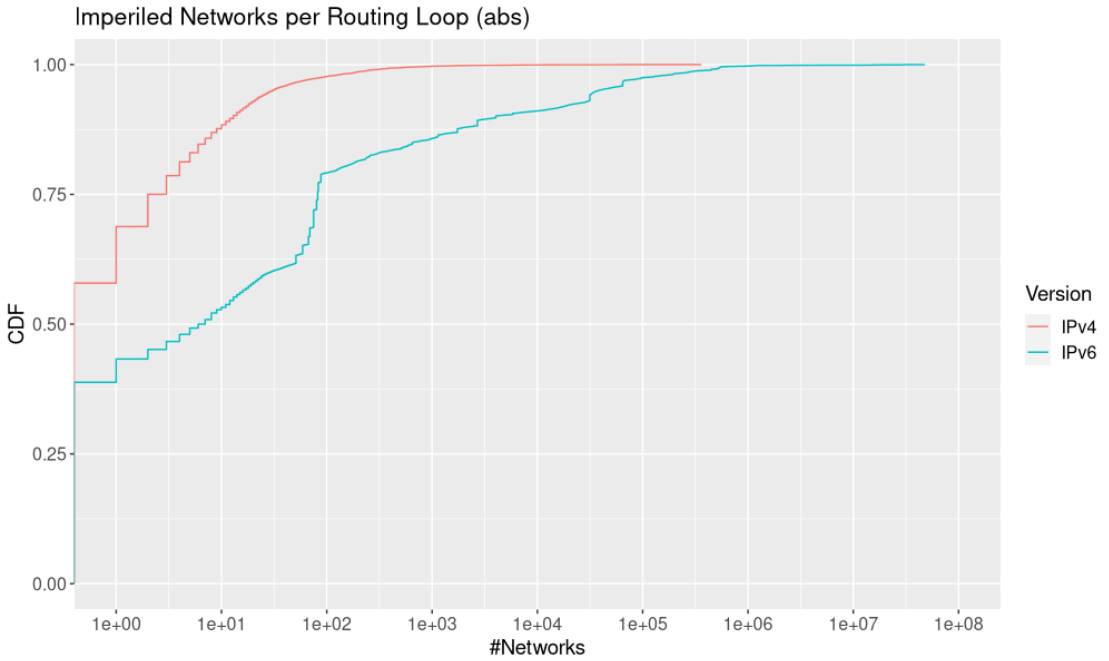

Figure 4.14: CDF of the Number of Imperiled Prefixes per Routing Loop

Looking at the routers in Figure 4.15 that are part of persistent loops and are actually the entity to imperil prefixes, we immediately notice far fewer routers imperil prefixes in contrast to the containing loops. In IPv4, around 73% of routers do not imperil any prefix and over 90% of the routers imperil less than ten prefixes. Only a small portion of routers is on the path to more than 100 imperiled prefixes. The top 76 routers have more than 1000 imperiled prefixes, whereas the top router tops the list with over 360.000 imperiled nets. This router is part of three loops and shadows in total three prefixes. We can only assume the router is somewhere on a high-value transit path. The IP address is

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

owned by Cogent. Further, all routers with a lot of imperiled prefixes do not shadow a lot of prefixes.

In IPv6, around 85% of all routers do not imperil a single prefix, while around 90% have below imperiled prefixes. Beyond this point, the number of imperiled nets per router increase drastically. 95% of all routers imperil below 1000 prefixes. The top 425 routers imperil at least 100.000 prefixes each, and the top 85 have above 1 million imperiled prefixes each. Over 10 million prefixes are imperiled each by a router in the Top-10. Eight out of ten routers are part of more than 10 loops, and three routers are part of over 100 loops each. But again, the number of shadowed addresses is relatively low compared to previous numbers.

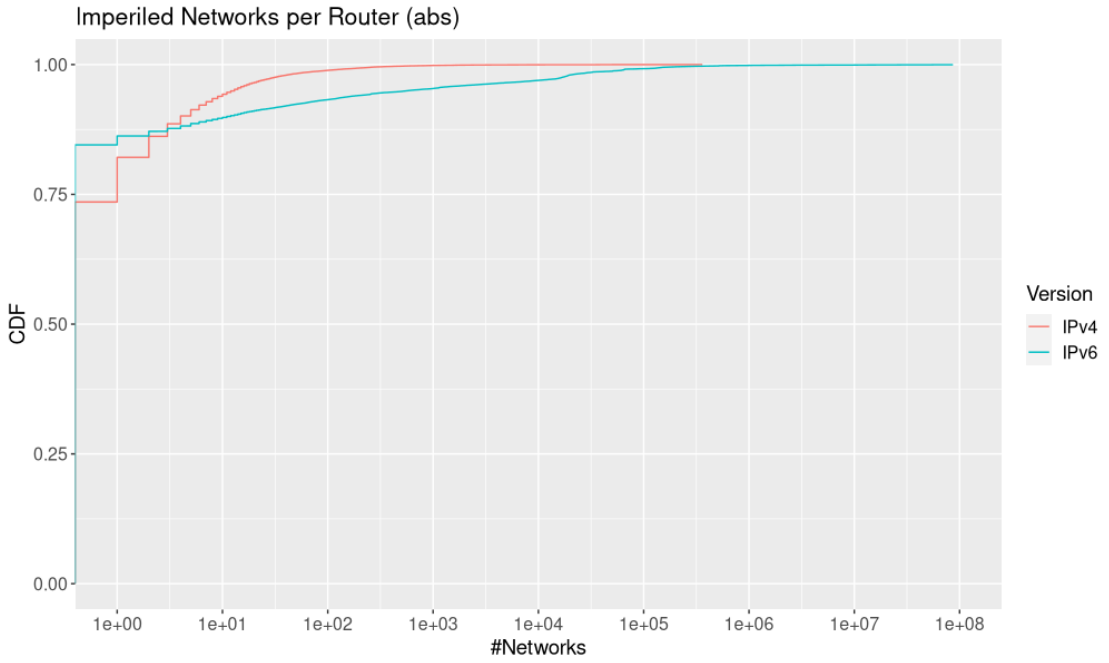

Figure 4.15: CDF of the Number of Imperiled Prefixes per Router

4.1.5 Amplification Factor

The Amplification factor is a metric that describes how much data or packets need to be handled by an attacked server for one packet sent to an amplifying entity. In our case, it describes how often a packet is handled by one router within a loop when sending a packet to a dark or shadowed prefix. In combination with the data of imperiled networks, this can be used to further measure the impact of each loop. Further, it is the main measurement to compare different loops in how much effort is needed to overload a single loop. Figure 4.16 displays a CDF of the percentage of loops with a certain amplification factor.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

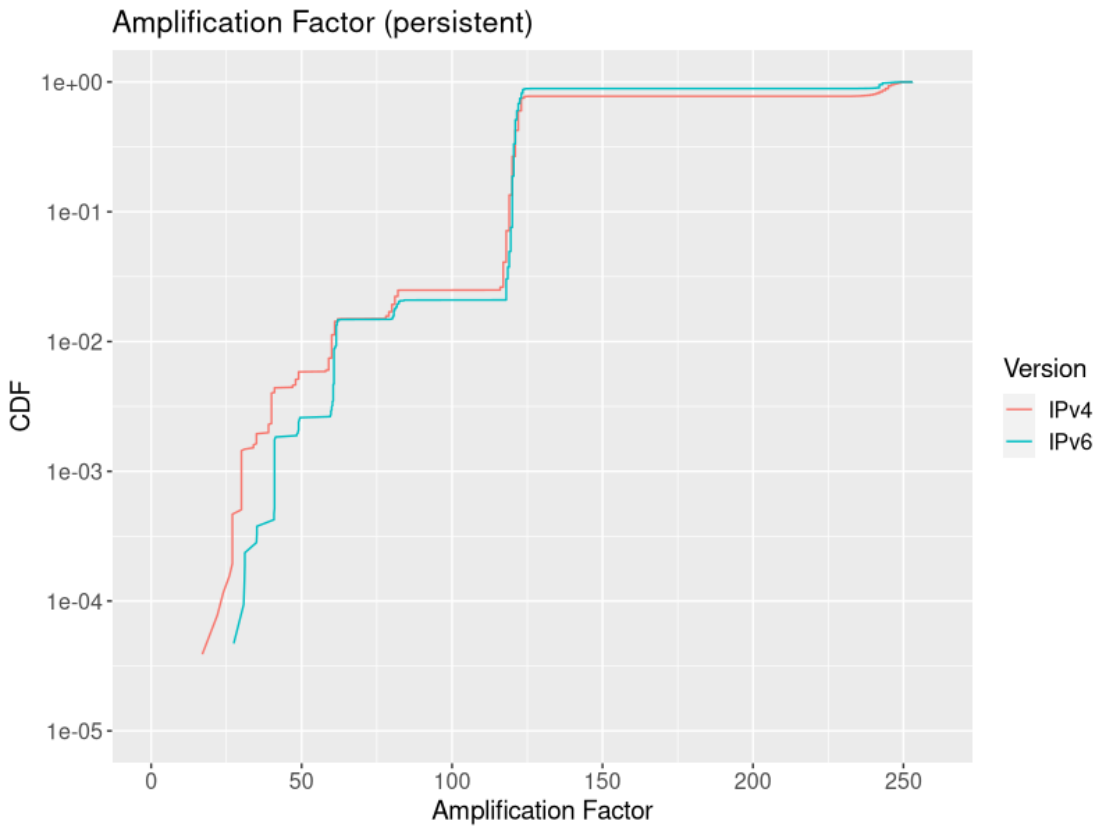

Figure 4.16: Amplification Factor of Persisting Routing Loops for both IPv4 and IPv6

In this Figure, we can see that loops with a low amplification factor represent a small portion of the entire set of persistent loops. The amplification factor of both IPv4 and IPv6 are rather similar, following the same ascends on the same amplification factors. This is to be expected as we are bound to the loop length of the loops, which has a similar distribution in both IPv4 and IPv6. Below 10% of the overall persistent loops have an amplification factor below 100 for both protocols. The most prevalent increase in the percent of loops is as expected for 2-hop loops around the 130 mark for amplification factor, followed by 1-hop loops around 240.

Figure 4.17 shows the number of imperiled nets in relation to the amplification factor. Each point on the figure is a persistent loop with its amplification factor on the y-axis and the number of prefixes that can be attacked by this loop on the x-axis. As already seen in the CDF before, there are not a lot of loops with an amplification factor below 100. This can be observed here as well. Most data points for both IPv4 and IPv6 are well above this value.

For potential exploitation of the found loops, the maximization of both the amplification factor and the number of imperiled nets is beneficial. This maximization corresponds with

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

the upper right section of the chart in Figure 4.17. Both IPv4 and IPv6 have quite a lot of loops with a near-maximum amplification factor at around 250. There are quite some loops that also carry a large number of imperiled prefixes, providing an attractive target set. However, if a specific prefix is targeted, these loops might not feature a particular prefix. A higher selection of imperiled prefixes is available around the amplification factor of 120. While the amplification factor is only half of the maximum possible value, the higher number of imperiled prefixes make up for that.

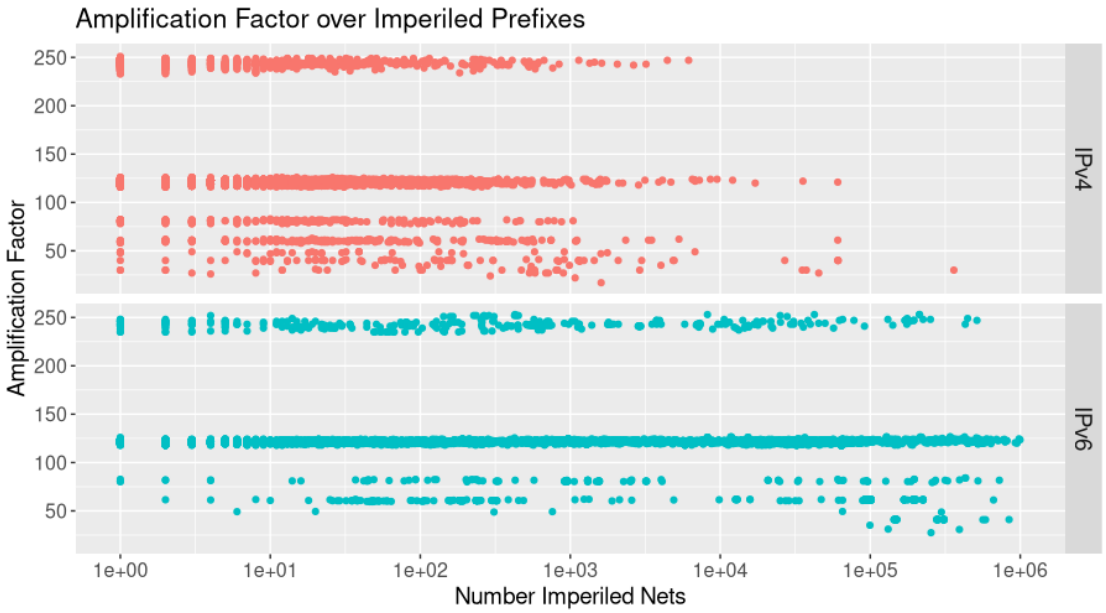

Figure 4.17: Amplification Factor over Number of Imperiled Prefixes per Loop

4.1.6 Density

The density function gives an overview of the localized property of the given shadowed or imperiled prefixes per persistent loop. It calculates the percentage of occupied addresses within a given prefix of the size of the bit overlap between the captured prefixes. Let us take three densities from the IPv6 graph of Figure 4.18 as an example.

Within the graph, we got three entries with almost the same density of about 16%, all three with a different number of bit overlap. For this example, we selected one entry with 41 bits overlapping and 21 addresses, one entry with 36 bits overlap with 670 addresses and one entry with 32 bits overlap and 10.691 addresses. All these numbers occupy around 16% of the maximum number of sub prefixes with length /48 within the respective super prefix of the length of the bit overlap of the shadowed prefixes. For example in a prefix of size /41 can fit 2_7, or 128, subnets of size /48. Twenty-one of these are shadowed by our loop. Thus we can calculate a density of 16%.

On the graph for IPv4, we can see that the density for 75% of all shadowed prefixes is

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

below -15 (or below 0,003%), which indicates a low localization of prefixes per loop. A look at the data shows that loops with a higher number of shadowed prefixes often have a bit overlap of zero, which is represented in the graph. Most prefixes with a higher density have a low number of prefixes with a high number of the same bits.

For IPv6, we see a much higher localized behavior. Only around 20% of the loops have a density lower than 0.1% (-10), and over 50% have a great density above 18% (-2,5). This means, around 18% ($2^{-2.5}$) of the prefix with the size of the number of same bits is occupied by shadowed prefixes. There is a hard spike at the end at zero. In the data, we can see a lot of prefixes of different sizes ranging from /29 to /48, where all traced destinations are shadowed by the same loop. These result in about 15% of loops shadowing all the /48 prefixes within the shared super prefix.

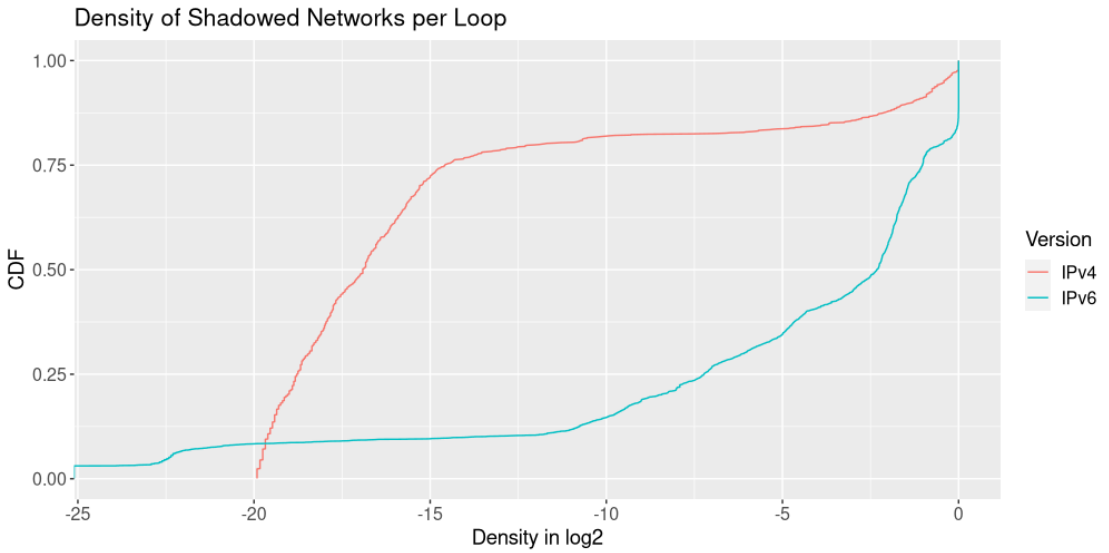

Figure 4.18: CDF of the Number of Shadowed Addresses over a given Density

Following up is the density of imperiled prefixes. This can be seen in Figure 4.19. The density of IPv4 for imperiled prefixes looks a lot like the density for shadowed prefixes. Around 75% of all loops have a density lower than -15, which means little to no localization between imperiled prefixes. As with the shadowed prefixes, loops with a higher count of imperiled prefixes have a low number of bit overlap, in most cases even zero bits.

In IPv6, we immediately notice the large jump of loops at a density of 0.05% (-11). For some reason, there are a lot of prefixes with a bit overlap of /31 and /32 that also have a similar number, between 60 and 90, of imperiled prefixes. Further, we can spot a second jump near zero on the density. These are a high number of prefixes that occupy over 98% of the prefix of the size of the bit overlap. These are no small prefixes sizes either, as there are a lot of /40, /39, /33 and /32 that are almost completely imperiled by a number of loops. Still, the density is not as high as with the shadowed prefixes. In comparison,

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

only around 15% of all loops have a density above 18% (-2.5), whereas, with shadowed prefixes, this number was at 50%.

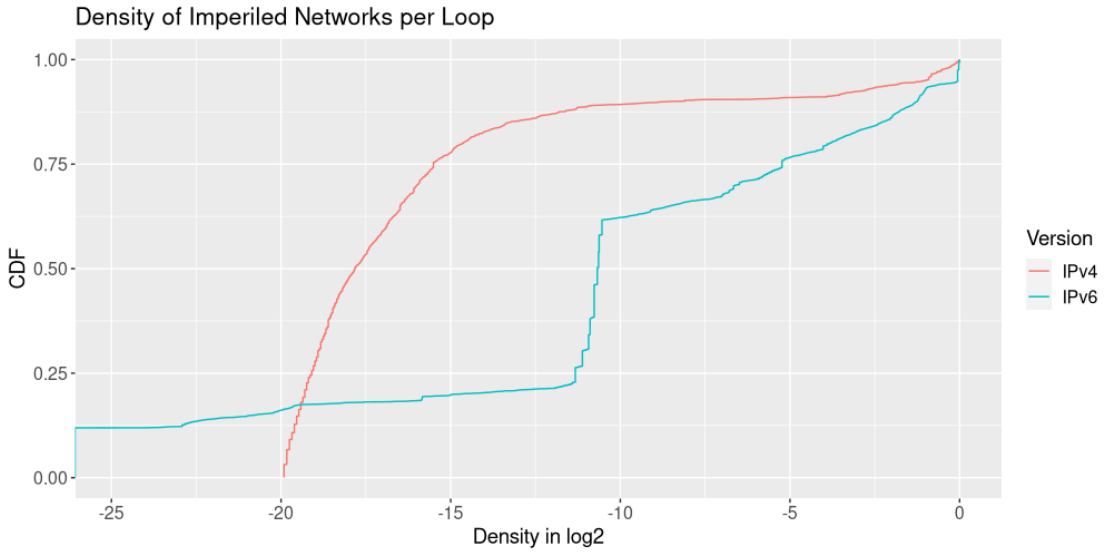

Figure 4.19: CDF of the Number of Imperiled Addresses over a given Density

4.1.7 Multiple Vantage Points

At the same time as we did the persistence scan to confirm the persistent loops, two additional vantage points conducted the persistence scans from two different continents. The persistence scans were thus done from our scan server in Vienna AT, from a VPS in Virginia US and from a VPS in Sydney AUS.

We traced 70.343 targets for IPv4 and 351.416 targets for IPv6 from each vantage point, ten times over the duration of five days. The server in Vienna scanned at UTC 0:00 and 12:00, Virginia followed at UTC 1:00 and 13:00 and Sydney was last at UTC 2:00 and 14:00. This was done to distribute the load and prevent hitting a single destination from all vantage points simultaneously.

First, we take a look at the overlap between routing loops within each vantage point, ignoring the full scan for the time being. Each graph shows the unique loops of each scan, the overlap with the first scan on this vantage point (Overlap n=1) and the overlap with the previously done scan on this vantage point (Overlap i-1). The x-axis displays the day and hour the scan was done. For the sake of a cleaner graph, we rounded the time of day to the closest 12th hour. On the Figures, both overlaps equal the unique loop number in the first data point, as no previous scan is available for comparison.

Figure 4.20 displays the overlap between multiple scans for each vantage point for IPv4. All lines stay stable, and no downtrend can be seen. The numbers differ quite a bit, especially between Sydney and the other two vantage points, which was expected as each

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

vantage point was on another continent. Vienna and Virginia stayed close to each other on the numbers. The routes between Europe and the US seem to be quite optimized.

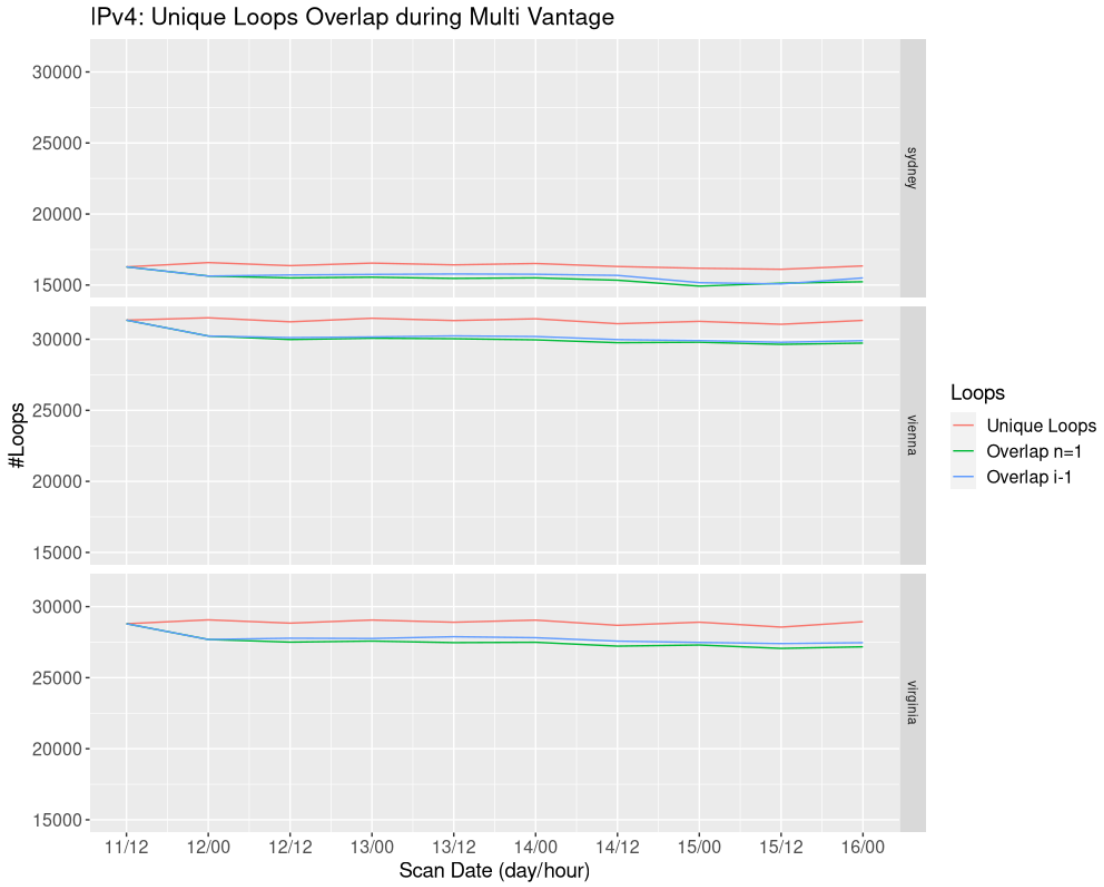

Figure 4.20: IPv4: Unique Loops with Overlap between all Scan

In IPv6, as can be seen in Figure 4.21, the number of unique loops per scan and the overlap with the previous scan on each vantage point is again stable. However, the number of unique loops that overlap with the first persistence scan decreases fast. Additionally, the number of loops are the lowest at the vantage point in Vienna, with just over 60.000 unique loops. The vantage point in Virginia stays stable at around 75.000 unique loops, while Sydney starts above 70.000 at the beginning, stabilizing around 67.000 loops. This difference in the number of routing loops might be a result of route aggregation on transit routes. After all ten scans, the overlap with the first persistence scan drops down to around 45.000 for both Sydney and Vienna. Virginia also sees a big drop in overlap and stays around 50.000 unique loops.

The following Tables 4.5 for IPv4 and 4.6 for IPv6 describe the numbers of loop and router overlap for the individual persistence scans for each vantage point. *Loop Overlap* describes the overlap of loops between all scans of each individual vantage point, which

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

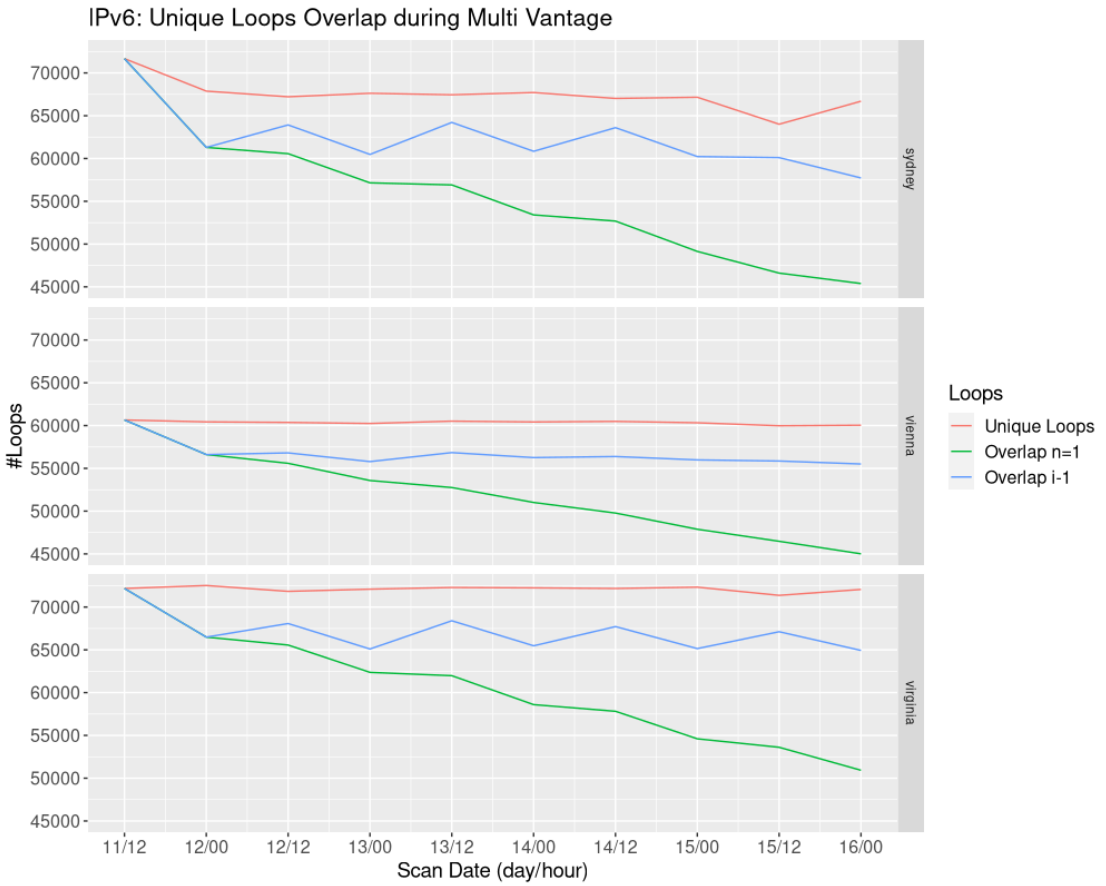

Figure 4.21: IPv6: Unique Loops with Overlap between all Scan

means only the number of unique loops that exist in all ten scans. *Router Overlap* is exactly the same thing but for individual, unique routers. *Persistent Loops* takes the set of loop overlaps and further compares it to the full scan. Thus, only the unique loops in all ten scans as well as in the full scan are within this set. Again, *Persistent Routers* does the exact same thing for the set of unique routers. *Routers of Persistent Loops* also is a count of unique routers but takes the set of persistent loops and gathers all routers that are members within these loops. This way, we can check how many routers are persistent without actually being in a persistent loop.

For IPv4 in Table 4.5, we can see the difference between Loop Overlap and the actual number of Persistent Loops is low. Further, the difference between Router Overlap, Persistent Routers and Routers of Persistent Loops is also low. The numbers of routing loops and the routers seem overall stable in IPv4.

The same data for IPv6 in Table 4.5 paints a rather different story. Of the high number of loop overlap, only a fraction remains when combining it with the full scan. Thinking back to Figure 4.21 with the ever-decreasing overlap, it does not surprise to see the

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

IPv4	Vienna	Virginia	Syndey
Loop Overlap	26.648	24.438	13.317
Router Overlap	46.987	43.210	24.184
Persistent Loops	25.687	22.229	11.915
Persistent Routers	45.841	40.649	22.635
Routers of Persistent Loops	44.541	38.692	21.407

Table 4.5: IPv4: Overlap of Loops and Routers within ten Persistence Scans

IPv6	Vienna	Virginia	Syndey
Loop Overlap	38735	43242	36192
Router Overlap	56881	67083	55852
Persistent Loops	21227	19666	15952
Persistent Routers	53666	63691	53131
Routers of Persistent Loops	28311	26390	20499

Table 4.6: IPv6: Overlap of Loops and Routers within ten Persistence Scans

number of persistent loops further decline. More surprising is the high difference in the number of routers between the three counting methods. The difference between Router Overlap and Persistent Routers is not too big and seems in line with IPv4. However, if we only take the number of routers that are members of persistent loops, only half the routers remain in the set. That means there is a high number of new loops that contain a high number of routers that we have already seen in the full scan. Thus, they appear in the persistent routers set but not in the recounted one. The ever-decreasing overlap in unique loops and this high difference in router overlap could be another indication for the fluctuation of dynamic assigned IPv6 prefixes.

Xia et al. traced a selection of IPv4 prefixes shadowed by persistent routing loops after confirming their persistence. We differed a bit in our approach, as we conducted the persistence scans from multiple vantage points. Still, the data should be comparable. They found around 90% of their persistent routing loops from four different vantage points. This matches the number from our vantage point in Virginia, with around 87% of persistent loops seen. However, the vantage point in Sydney only confirms a low number of 46% of persistent loops.

4.1.8 Persistence 50

For the final persistence check, we took the entire set of candidate shadowed prefixes and probed 50 random addresses for each prefix. The probing was done with ZMap, as stated in section 3.5.4. The ZMap results were aggregated and collected in a single file, noting every probed prefix with the total amount of responses per response type. To recite, we were probing 124.676.281 prefixes for IPv6 and 476.866 Prefixes for IPv4.

In Figure 4.22, we can see the CDF of collected *Hop Limit Exceeded* responses for both

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

IPv4 and IPv6. Further, in Figure 4.23, we can see the CDF for the returned *Echo Replies* for both protocols. In both figures, we look at the number of responses of the given type, either *Hop Limit Exceeded* or *Echo Replies*. However, we filtered for shadowed prefixes with a confirmed persistent loop. This made the most sense as most non-persistent shadowed prefixes did not answer or answered with a different error code, heavily shifting the graphs.

Most of the probed IPv4 targets were not responding with an *Hop Limit Exceeded* error, but we received some hosts within a number of prefixes responding with *Echo Replies*. 55% of the prefixes return 0 responses with *Hop Limit Exceeded*, 40% of them return 0 *Echo Replies*. The number of *Hop Limit Exceeded* responses per prefix increase after that point to about 25% of prefixes returning with at least 15 *Hop Limit Exceeded* errors. Finally, around 87% of the shadowed prefixes return at least 40 *Hop Limit Exceeded* errors. The number of prefixes responding with a low number of *Echo Replies* increases. Around 70% of prefixes respond with less than ten responses and around 85% respond with less than 20 *Echo Replies*. Only a low number of prefixes, around 10%, send more than 30 responses.

For IPv6, the numbers for *Echo Replies* are rather straightforward. Almost no prefix responded with any *Echo Replies* within the probed shadowed prefixes. However, of the prefixes responding with *Echo Replies*, 99,6% responded with at least 45 responses of this type. In regards to *Hop Limit Exceeded*, below 2% of all prefixes responded with no errors of this kind. The number of responses per prefix increases rather fast. Over 80% of all shadowed prefixes probed respond with at least 20 *Hop Limit Exceeded*. Further, around 60% of the shadowed prefixes respond with at least 40 errors, with 37% of prefixes return the maximum amount of 50 errors.

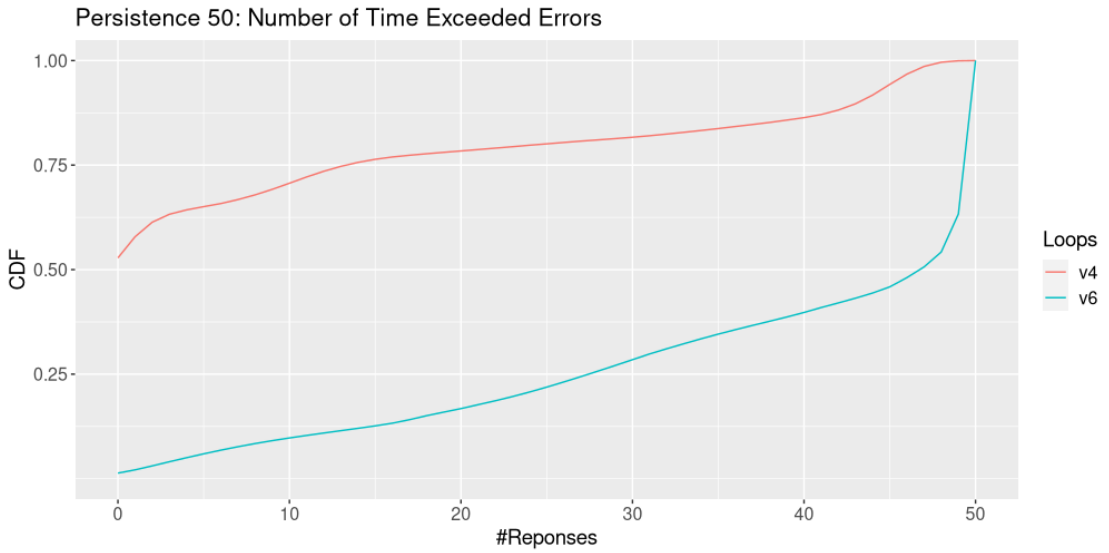

Figure 4.22: Number of Prefixes responding with *Hop Limit Exceeded*

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

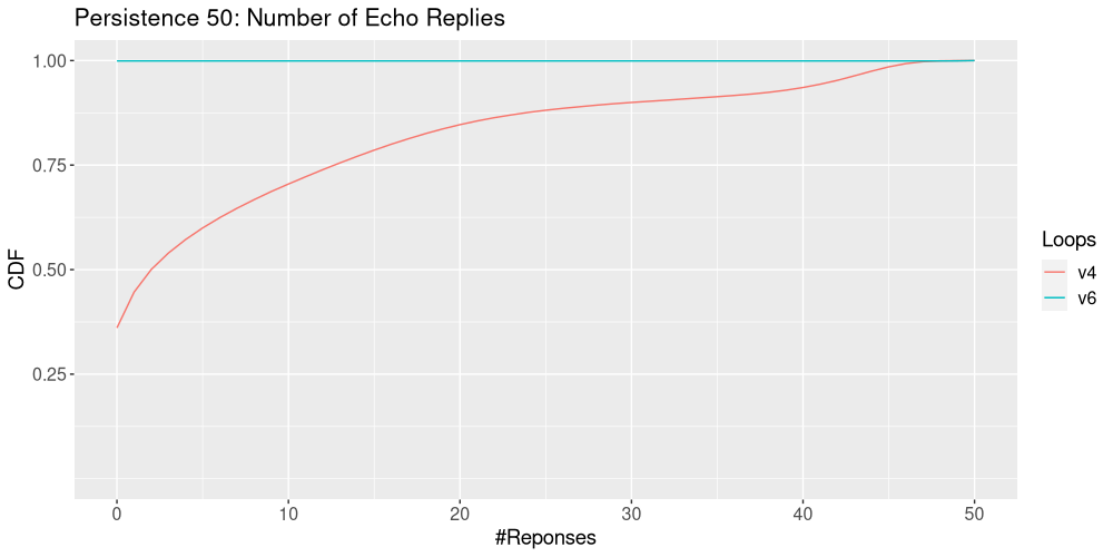

Figure 4.23: Number of Prefixes responding with *Echo Reply*

As suspected, for IPv6, if a prefix seems shadowed, it is likely that the entire prefix is shadowed. For IPv4, on the other hand, the ongoing segmentation is reflected in the response CDF, as only a low number of addresses per prefix respond with the *Hop Limit Exceeded* error type.

In comparison with Xia et al., they showed that 68,96% of their probed IPv4 prefixes shadowed by a persistent routing loops responded with 50 *Hop Limit Exceeded* errors. Further, only 1,38% show no errors of this type at all. This is in strong contrast to our findings and further highlights the segmentation of prefixes.

However, for IPv4, our conducted process was not optimal. This was due to an oversight with the randomly generated IP addresses in the small address space per shadowed prefix. For each shadowed prefix, the address space can fit exactly 254 addresses. By generating 50 randomized IP addresses, the chance to generate one IP address twice is moderately high. This has been fixed for further scans. In the future, IPv4 prefixes will be spread across the available space.

4.1.9 Unplanned Events

Bringing down an ISP: During our first test measurements in preparation for the full scan, we received an abuse mail from a European ISP mainly providing for residential users and smaller business customers. Even though we sent a rather low number of packets per second to their prefix, we managed to crash parts of their backbone infrastructure. They provided us with a generous amount of information about their deployed infrastructure to help us understand the issue they faced due to our scan.

To speed up their deployment, they used the transition mechanism 6rd, which enables

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

them to tunnel IPv6 traffic over their existing IPv4 infrastructure and provide their customers with IPv6 connectivity. This 6rd deployment did not like our sent probes, as we somehow managed to craft packets that triggered an unwanted outcome. The packets were valid IPv6 packets containing ICMPv6 echo requests with a valid IPv6 destination address within their prefix. We did not specifically target any 6rd deployment, and this address was randomly chosen. The 6rd deployment translated the IPv6 address to an IPv4 address and tried to forward the content via the tunnel. However, the address translated to an IPv4 address on the 127.0.0.0/8 loopback prefix, which confused the router itself.

This packet then managed to sneak through the routing process within the router without ever decreasing the hop limit, thus looping forever in the router. After a high enough number of packets reached and occupied the router, it would be unable to further process any packets, effectively rendering it useless without a reboot. The ISP was naturally prepared for a failing router, a failover took its place, but as they shared the same configuration, it was doomed to fail in the same fashion. All ICMPv6 Echo Requests from our location were blocked by them as soon as they realized what was happening and they started a conversation about the issue with us.

We cannot say for sure what the actual issue is here. It could be either a faulty implementation of 6rd in the used equipment or a misconfiguration of the transition mechanism on the operator side. To not further cause any problems for the ISP, we ended up adding their entire prefix to our blocklist, thus not sending any further packets to them. This problem needs further investigation. It was sadly way out of the scope of this thesis but will be handled in the future.

Invalid Router answering: During the analysis of all found routing loops and their corresponding routers, we noticed a number of invalid or internal IP addresses on the IPv6 side. Eleven routers in invalid, private or not yet allocated prefixes responded to our traceroutes with *Hop Limit Exceeded* errors and are part of a loop. The IPs are listed in Table 4.7. They are rather useless on their own, as they cannot be pinged due to non-existing routes to these prefixes.

We need to figure out where exactly they are located. The ASes of the given loops is a strong indication of the owner of the IP addresses. These IPs probably do not belong to edge routers between two AS, as these must have a valid and routable IP address for the data transfer. We did some research into the found prefixes, but most of them are not noted anywhere. The prefix 9009::/16 is used in the documentation of Juniper on how to configure OSPFv3[39]. This might be an inspiration for the usage of these IP addresses. Juniper as a network vendor, should lead by example and update their documentation to use the dedicated documentation prefix[40] 2001:db8::/32 for these purposes. For the prefixes starting with *0fe0* or *0fec* we did not find anything useful. They look similar to link local (fe80::/10), site local (fec0::/10) and private ranges (fc00::/7). However, they are not part of these prefixes as they start with a leading *0* instead of the leading *f*. These addresses most likely are part of an internal or testing network that went into production without changing the invalid addresses to correctly allocated ones.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

72f:1000:f::b
9009:3568:1::1
9009:3568:2::1
951:0:0:30::2
951:0:0:30::3
fe0:172:16:4::2
fec:10:10:10::2
fec:10:241::1
fec:10:74:210::1
fec:100:105:0:57::1
fec:200:233:233::2

Table 4.7: Invalid Routers found during traceroute

We noticed the issue while looking at the numbers of shadowed and imperiled nets per router. The router 951:0:0:30::3 is a member of 19 unique and persistent loops. It ranked number 56 in the list of routers with the most imperiled nets with over 2.6 million prefixes while still shadowing 589.140 prefixes.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

CHAPTER 5

Conclusion

Stability, uptime and fast data transfers have manifested themselves as important cornerstones of our modern internet. We encountered, identified and researched one possible and overlooked aspect of our internet that has the potential to disrupt and threaten the stability of the internet as we know it.

In this thesis, we planned and conducted multiple measurements to find routing loops in the wild and further check their persistence in both the IPv4 and IPv6 internet. We investigated their characteristics, the number of shadowed as well as imperiled networks per loop and the involvement of different AS. During the analysis, we found large numbers of loops in both protocols alike and were able to attribute the routers to various AS. With the additional persistence scans, we were able to find concrete evidence of the number of persistent loops and the number of impacted prefixes. We found 25.687 persistent loops across 9.675 AS in IPv4 and 21.227 persistent loops with 4.777 involved AS in IPv6. These loops imperil 9.6% and 31.2% of the traced address space, presenting a large attack surface. In comparison with Xia et al., we identified new loop properties and new results of the ongoing IPv4 segmentation. While their number of loops is comparable to ours, the number of imperiled prefixes increased drastically from 0.78% to 9.6% of traced prefixes in IPv4.

For future work, we identified multiple open questions. We would like to explore the existence of 1-hop loops in both IPv4 and IPv6 as there was an unexpected amount of these observable in our analysis. Further, the fluctuations of routing loops in IPv6, as well as better detection and detailed analysis of both load balancers and spammers sparks our interest. Additionally, the incident of the crashing 6rd infrastructure showcases the impact a small number of packets might have. Thus a detailed analysis of common transition mechanisms could provide valuable insights. In any case, the work done in this thesis presents a strong baseline and a detailed analysis toolkit to build upon for upcoming measurements.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Figures

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

List of Tables

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.

Bibliography

[1] Google, "IPv6 adoption statistics by Google." https://www.google.com/intl/en/ipv6/statistics.html. Accessed: 2021-09-25.

[2] F. Holzbauer, "IPv6-Reconnaissance - Internet-weite Analyse von ICMPv6." https://phaidra.fhstp.ac.at/detail/o:4272, 2020. Master's Thesis, St. Pölten University of Applied Sciences.

[3] A. Conta, S. Deering, and M. Gupta, "RFC 4443, Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, IETF, March 2006." https://datatracker.ietf.org/doc/html/rfc4443, 2006.

[4] Juniper Networks, "Configuring the JUNOS software ICMPv6 rate limit for ICMPv6 routing engine messages." https://www.juniper.net/documentation/software/topics/task/configuration/icmpv6-rate-limit-configuring.html. Accessed: 2021-05-10.

[5] PaloAlto, "ICMPv6 rate limiting." https://docs.paloaltonetworks.com/pan-os/8-1/pan-os-admin/networking/session-settings-and-timeouts/icmp/icmpv6-rate-limiting.html. Accessed: 2021-05-09.

[6] Cisco, "IOS IPv6 command reference - ICMPv6 rate limiting." https://www.cisco.com/c/en/us/td/docs/ios/ipv6/command/reference/ipv6_book/ipv6_06.html. Accessed: 2021-05-10.

[7] HPEnterprise, "Configuring the rate limit for ICMPv6 error messages." https://techhub.hpe.com/eginfolib/networking/docs/switches/5130ei/5200-3942_l3-ip-svcs_cg/content/483572521.htm. Accessed: 2021-07-23.

[8] Cisco Press, "Introduction to Routing Dynamically." https://www.ciscopress.com/articles/article.asp?p=2180210. Accessed: 2021-10-06.

[9] Cisco Press, "BGP Fundamentals." https://www.ciscopress.com/articles/article.asp?p=2756480&seqNum=17. Accessed: 2021-10-04.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[10] RIPE Network Coordination Centre, "What is IPv4 run out?." https://www.ripe.net/manage-ips-and-asns/ipv4/ipv4-run-out. Accessed: 2021-09-21.

[11] RIPE Network Coordination Center, "IPv4 pool." https://www.ripe.net/manage-ips-and-asns/ipv4/ipv4-pool. Accessed: 2021-09-21.

[12] IPv4 Market Group, "IPv4 Transfer Pricing." https://ipv4marketgroup.com/ipv4-pricing/. Accessed: 2021-09-21.

[13] IPv4.Global, "2021 Q1 IPv4 Address Market Roundup." https://www.ipv4.global/blog/2021-q1-ipv4-address-market-roundup/. Accessed: 2021-09-21.

[14] L. Prehn, F. Lichtblau, and A. Feldmann, "When wells run dry: the 2020 IPv4 address market," in *Proceedings of the 16th International Conference on emerging Networking EXperiments and Technologies*, pp. 46–54, ACM.

[15] S. Jia, M. Luckie, B. Huffaker, A. Elmokashfi, E. Aben, K. Claffy, and A. Dhamdhere, "Tracking the deployment of ipv6: Topology, routing and performance," *Computer Networks*, vol. 165, p. 106947, 2019.

[16] J. Xia, L. Gao, and T. Fei, "A measurement study of persistent forwarding loops on the internet," vol. 51, no. 17, pp. 4780–4796.

[17] R. Teixeira, A. Shaikh, T. Labs, F. Park, and T. Griffin, "Dynamics of hot-potato routing in IP networks," p. 12.

[18] B. Donnet, P. Raoult, T. Friedman, and M. Crovella, "Efficient algorithms for large-scale topology discovery,"

[19] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira, "Avoiding traceroute anomalies with paris traceroute," in *Proceedings of the 6th ACM SIGCOMM on Internet measurement - IMC '06*, p. 153, ACM Press.

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

[21] K. Vermeulen, S. D. Strowes, O. Fourmaux, and T. Friedman, "Multilevel MDA-lite paris traceroute," p. 14.

[22] M. Luckie, "Scamper: a scalable and extensible packet prober for active measurement of the internet," in *Proceedings of the 10th annual conference on Internet measurement - IMC '10*, p. 239, ACM Press.

[23] R. Beverly, "Yarrp'ing the internet: Randomized high-speed active topology discovery," in *Proceedings of the 2016 Internet Measurement Conference*, IMC '16, pp. 413–420, Association for Computing Machinery.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[24] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," p. 15.

[25] O. Gasser, Q. Scheitle, S. Gebhard, and G. Carle, "Scanning the IPv6 internet: Towards a comprehensive hitlist,"

[26] J. Rüth, T. Zimmermann, and O. Hohlfeld, "Hidden treasures – recycling large-scale internet measurements to study the internet's control plane," in *Passive and Active Measurement* (D. Choffnes and M. Barcellos, eds.), Lecture Notes in Computer Science, pp. 51–67, Springer International Publishing.

[27] K. Bock, A. Alaraj, Y. Fax, K. Hurley, E. Wustrow, and D. Levin, "Weaponizing middleboxes for TCP reflected amplification," p. 17.

[28] G. Nakibly and M. Arov, "Routing loop attacks using IPv6 tunnels," in *3 USENIX Conf. Offensive Technol.*, pp. 1–7, 2009.

[29] G. Nakibly and F. Templin, "RFC6324: Routing Loop Attack Using IPv6 Automatic Tunnels: Problem Statement and Proposed Mitigations."

[30] CAIDA, "Ark ipv6 topology dataset." `https://catalog.caida.org/details/dataset/ipv6_allpref_topology`. Accessed: 2021-9-30.

[31] CAIDA, "RouteViews IPv6 Prefix to AS mappings." `https://catalog.caida.org/details/dataset/routeviews_ipv6_prefix2as`. Accessed: 2021-9-30.

[32] CAIDA, "RouteViews IPv4 Prefix to AS mappings." `https://catalog.caida.org/details/dataset/routeviews_ipv4_prefix2as`. Accessed: 2021-9-30.

[33] P. Smith, Cisco, and Network Startup Resource Center, "BGP - best current practices." `https://nsrc.org/workshops/2021/apricot2021/bgp/presentations/BGP-BCP.pdf`. Accessed: 2021-05-09.

[34] RIPE Network Coordination Centre, "Routing working group recommendations on IPv6 route aggregation." `https://www.ripe.net/publications/docs/ripe-532`. Accessed: 2021-10-03.

[35] S. Strowes and RIPE Labs, "Visibility of IPv4 and IPv6 prefix lengths in 2019." `http://labs.ripe.net/author/stephen_strowes/visibility-of-ipv4-and-ipv6-prefix-lengths-in-2019/`. Accessed: 2021-05-09.

[36] University of Oregon, "RouteViews Project." `http://www.routeviews.org`, 2013. Accessed: 2021-09-11.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN

[37] RIPE Network Coordination Centre, "Best current operational practice for operators: IPv6 prefix assignment for end-users." https://www.ripe.net/publications/docs/ripe-690. Accessed: 2021-10-07.

[38] D. Allan and The Broadband Forum, "IPv6 in the context of TR-101." https://www.broadband-forum.org/technical/download/TR-177_Corrigendum-1.pdf.

[39] Juniper Networks, "Example: Configuring OSPFv3 stub and totally stubby areas." https://www.juniper.net/documentation/en_US/junos/topics/topic-map/ospf3-stubby-areas.html. Accessed: 2021-10-08.

[40] IETF, "RFC3849: IPv6 address prefix reserved for documentation." https://www.ietf.org/rfc/rfc3849.txt. Accessed: 2021-10-03.

Die approbierte gedruckte Originalversion dieser Diplomarbeit ist an der TU Wien Bibliothek verfügbar
The approved original version of this thesis is available in print at TU Wien Bibliothek.
TU Bibliothek Your knowledge hub WIEN