University of Glasgow | School of Computing Science

Measuring NAT64 Characteristics using RIPE Atlas

Elizabeth Boswell (2398840b)

13th April 2023

ABSTRACT

NAT64 is an IPv6 transition mechanism which translates IPv6 to IPv4, enabling IPv6 hosts to access the IPv4 Internet. There have been no large scale measurement studies on the impact NAT64 has on different path characteristics. I perform such a study using RIPE Atlas, comparing NAT64 to native IPv4 in terms of latency, path similarity, and impact on traceroute. The NAT64 paths have a slightly larger path length and latency, and there is a possible interaction between NAT64 and the functionality of traceroute.

1. INTRODUCTION

With the depletion of the IPv4 address space, widespread adoption of IPv6 becomes increasingly important[22]. IPv4 and IPv6 cannot interoperate directly, though, so special measures need to be put in place to allow IPv6 hosts to connect to the IPv4 Internet [22]. One such measure is NAT64 [17] together with DNS64 [18]. DNS64 translates IPv4 DNS responses by encoding the IPv4 address in an IPv6 address. The host sends packets to that IPv6 address, and they are received by the NAT64, which extracts the IPv4 address and translates the packet into IPv4. This allows IPv4 and IPv6 hosts to communicate without being aware that they are using two different address families.

It is not clear what impact NAT64 has on network performance, i.e. whether using NAT64 increases latency compared to native IPv4. The NAT64 might be a performance bottleneck, or packets might take longer and slower paths through the network when using NAT64. If using NAT64 leads to significantly worse performance this might discourage people from switching to IPv6-only deployments, slowing down IPv6 adoption. Similarly, it is not clear if using NAT64 has an effect on the functioning of traceroute, a tool that is widely used to determine the hops on the path between a host and a destination. If NAT64 becomes widely used but it interferes with traceroute, then this will make network troubleshooting and measurements more difficult.

In this paper, I use RIPE Atlas[1] to analyse the behaviour of NAT64. RIPE Atlas is a measurement network consisting of over 11,600 probes in a variety of networks around the world. I determine which probes use NAT64, and perform traceroutes with these probes using native IPv4 and NAT64. I compare the latency and path length of the IPv4 and NAT64 paths, analyse their similarity, and investigate the impact NAT64 has on traceroute itself. I show that, on average, the paths with NAT64 are 25.65% longer and have a 15.23% higher RTT, and that there is a possible interaction between NAT64 and traceroute.

[1]https://atlas.ripe.net/

There have been several studies of NAT64 performance in small test networks (e.g. [16], [14], [25]). Such studies are important, but their external validity is limited because the test networks might not behave like the Internet. There have been no large-scale studies of the behaviour of real-world NAT64 deployments. This paper seeks to fill this gap.

I structure the remainder of this paper as follows. Section 2 gives background information on NAT64 and RIPE Atlas. Section 3 introduces the research problem. In Section 4 I describe how I searched the RIPE Atlas network for probes that use NAT64, and the results of these measurements. I used these probes to perform traceroutes, the methodology and results of this are described in Section 5. Related work is discussed in Section 6, and the paper concludes with suggestions for future work in Section 7.

2. BACKGROUND

This section provides background information on NAT64 (Section 2.1) and RIPE Atlas (Section 2.2).

2.1 NAT64

There are currently two versions of the Internet protocol (IP) in use, IPv4 [21] and IPv6 [12]. IPv4 is the older version, standardised in 1981, but it is still widely used. For example, on 5 April 2023, 60.83% of users accessed Google over IPv4[2]. However, IPv4 has some design features that make it unsuitable for today's Internet, the most notable of which is its small address space [22]. IPv4 addresses are 32-bits long, which leads to an address space of 4,294,967,296 addresses, but not all of these can be used in practice [23].

One way of coping with the small IPv4 address space is with Network Address Port Translation (NAPT), a kind of Network Address Translation [8]. NAPT allows several hosts to share an IPv4 address by multiplexing on the port number. However, NAPT creates other problems. For example, two hosts behind two different NAPTs can't connect to each other directly [13]; they need to use NAT traversal (e.g. ICE [13]), which is slow and error-prone.

IPv6, standardised in 1998, addresses IPv4's design issues. It offers a substantially larger address space (2^{128} addresses) and other new features. However, IPv4 and IPv6 can't interoperate: a host that only supports IPv4 can't directly communicate with an IPv6-only host [22]. As a result, many hosts today are dual-stack hosts, they support both IPv4 and IPv6 [9]. IPv6-only hosts might become more common, though, as IPv4 addresses become increasingly scarce [17]. These hosts will still need to access the IPv4 Internet for the foreseeable future [22]. This can done using NAT64.

[2]https://www.google.com/intl/en/ipv6/

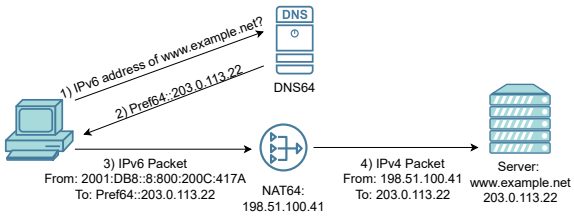

Figure 1: Diagram visualising how NAT64 works.

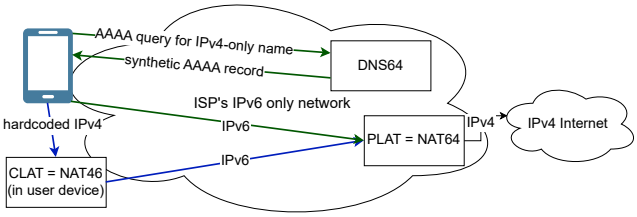

Figure 2: 464XLAT: IPv4 is accessed through NAT64/DNS64 (green arrows). If software uses hardcoded IPv4/won't use IPv6 the CLAT translates it to IPv6 locally (blue arrows).

NAT64 [17] is a kind of Network Address Translation that translates IPv6 into IPv4. It is normally used together with DNS64 [18]. It works as follows: an IPv6 host sends a DNS AAAA query (a query for an IPv6 address) for a domain name that only has an IPv4 address to the DNS64 resolver. The DNS64 replies with a AAAA record, which contains an IPv6 address that encodes the IPv4 address of the target host. The IPv6 address is made up of a prefix, which is either the standard NAT64 prefix 64:ff9b::/96 or a custom prefix, followed by the IPv4 address, starting at a bit position defined in [15] (often the last 32 bits). Packets sent to that address by the IPv6 host are routed to the NAT64. The NAT64 extracts the IPv4 address from the IPv6 address, translates the packet to IPv4, and sends the packet to the IPv4 host. Replies from the IPv4 host are similarly translated back into IPv6. This process is depicted in Figure 1. Listing 1 shows a traceroute through a NAT64 (from RIPE Atlas probe 2589 to an IPv4 NTP server). After hop three, all addresses start with the NAT64 prefix and encode IPv4 addresses in the last 32 bits.

Listing 1: Annotated traceroute through a NAT64

```
1  2a01:568:4000:810::1 = Probe 2589
2  2a01:568:4000:768::1
3  2a01:568:4000:525::2
4  64:ff9b::4f62:20f5 = 79.98.32.245
5  64:ff9b::4f62:20fe = 79.98.32.254
6  64:ff9b::b96a:86ee = 185.106.134.238
7  * * 64:ff9b::b868:cd49 = 184.104.205.73
8  64:ff9b::b869:50ae = 184.105.80.174
9  64:ff9b::b869:506a = 184.105.80.106
10 * * *
11 64:ff9b::66de:6699 = 102.222.102.153
```

NAT64s can be set up by ISPs [5], or by individual users.[3] There are also public NAT64/DNS64 providers.[4] Hosts send DNS queries to a public DNS64 resolver, and their packets are sent to a public NAT64 which translates them and sends them to the destination.

A special kind of NAT64 that was created by T-Mobile US is 464XLAT [19]. As described in a NANOG presentation,[5] user devices are connected to an IPv6-only network, and use NAT64 and DNS64 to access IPv4. The devices also contain a NAT46, which translates hardcoded IPv4 addresses to IPv6 - otherwise some applications fail. This is shown in Figure 2. As discussed in Section 5.3.1, a significant number of probes analysed in this paper use 464XLAT.

[3]https://ripe72.ripe.net/archives/video/228/
[4]https://nat64.xyz

Figure 3: Location of RIPE Atlas probes. Green dots: connected probes, yellow dots: disconnected probes. Source: https://atlas.ripe.net/results/maps/network-coverage/ (captured 5 April 2023)

2.2 RIPE Atlas

RIPE Atlas is a measurement network created by the RIPE NCC, consisting of over 11,600 probes deployed around the world. It can be used to perform large scale performance and connectivity measurements. RIPE Atlas probes are hosted by volunteers, and are located in residential, educational, and business networks in 172 countries. Probes are usually hardware devices provided by RIPE Atlas, but there are also software probes, running on the volunteer's own hardware. Apart from small probes in homes or offices, there are also anchor probes, which are more powerful probes running in data centres and other networks with high availability. Figure 3 shows the location of probes across the world. RIPE Atlas continuously performs pre-defined traceroute, ping, DNS, HTTP, and TLS measurements to various targets. Users can also start user-defined measurements in exchange for credits, which can be earned by hosting a probe.

As of October 2021, RIPE Atlas has been used in over 80 studies [11]. For example, it has been used to improve the IPv6 hitlist [27], to investigate a new kind of DNS vulnerability [20], and to study the impact that COVID-19 had on network latency and packet loss in Europe [4].

3. CHARACTERISING NAT64 BEHAVIOUR

There have been several small-scale studies measuring the performance impact of NAT64, using custom setups of various NAT64 implementations [14] [25] [16]. They were able to get meaningful results concerning the translation overhead, performance of different types of NAT, and performance under load, but they might not reflect the kinds of setups that are used in practice, and also can't determine how NAT64 interacts with other network components, because they are not measuring NAT64s on the Internet.

To overcome the limitations of other studies, I use RIPE

[5]https://archive.nanog.org/meetings/abstract?id=2359

Atlas to study the behaviour of real-world deployments of NAT64. I use dual-stack probes that use NAT64 to compare the latency of NAT64 and native IPv4, and to determine whether NAT64 affects the functioning of traceroute. I answer the following research questions:

RQ1: How many probes use NAT64 (Section 4)?

RQ2: Does using NAT64 affect whether traceroutes reach the destination (Section 5.2.1)?

RQ3: Does using NAT64 increase the number of missing hops in traceroute (Section 5.2.2)?

RQ4: Does using NAT64 increase the path length and Round Trip Time (RTT) (Section 5.3)?

RQ5: Do network paths differ when using NAT64 compared to native IPv4 (Section 5.3.3)?

RQ1 approximates how commonly NAT64 is used. While the RIPE Atlas probes are not representative of all hosts on the Internet [3], they are present in many ASs and can detect NAT64s deployed by many network operators.

RQ2 and 3 are traceroute-specific, but they also show how NAT64s handle ICMP (Internet Control Message Protocol) packets. There are two versions of ICMP, ICMPv4 [1] (for IPv4) and ICMPv6 [10] (for IPv6), so the NAT64 needs to translate ICMP packets as well. Additionally, traceroute is widely used for diagnostics and measurements, so it is important to study whether NAT64 has an impact on it. These research questions also provide context for the following questions, as they need to use the traceroute data.

RQ4 and 5 study the effect that NAT64 has on latency. RQ4 shows whether NAT64 paths have a higher latency, and RQ5 shows if NAT64 operators handle NAT64 traffic differently from native IPv4 traffic. If the paths are very similar, then any latency differences found in RQ4 are more likely to be due to the influence of the NAT64.

4. FINDING NAT64 PROBES

In order to study NAT64 performance it is first necessary to find the set of RIPE Atlas probes that use NAT64. These probes are used in the following measurements. RIPE Atlas does not specify whether a probe is using NAT64, so various tests had to be used, as described in Section 4.1. This is followed by a characterisation of the probes in Section 4.2.

4.1 Methodology

In order to find NAT64 probes, I first checked if any probes that use IPv6 use DNS64, by testing if they resolve domain names for IPv4-only hosts to IPv6-mapped addresses. Then I tested if their NAT64 is working, by checking if they can use these IPv6-mapped addresses to reach IPv4 hosts. I also searched for probes that can use NAT64 but don't use DNS64. All this was done using four different tests.

DNS Test 1 uses the NAT64 prefix discovery procedure from RFC 7050 [24]. Probes send a DNS AAAA query for the `ipv4only.arpa.` domain, a special-use domain name that only resolves to IPv4. Probes that use DNS64 will receive a synthetic AAAA record from the DNS64. Probes without DNS64 will get a NXDOMAIN error because this domain doesn't have a AAAA record.

DNS Test 2 is similar to DNS Test 1, but involves resolving another IPv4-only name, `time-c-b.nist.gov`. This test was added because some probes were found to pass DNS Test 1 but fail resolve other IPv4-only names (see Section 4.1.1).

The **Standard Prefix `ping` test** involves pinging the IPv6 address 64:ff9b::5bc9:7f3. This is the standard NAT64 prefix 64:ff9b::/96 with the address of an IPv4-only RIPE Atlas anchor probe (probe 6771, 91.201.7.243) encoded in the last 32 bits. An anchor probe was chosen because they respond to `ping`. This test was used to confirm that probes with a DNS64 that returns the standard prefix can ping addresses with that prefix. Additionally, it was used check whether any probes can ping addresses with the standard NAT64 prefix, even if they fail the DNS tests.

Table 1: Number of probes that passed/failed the DNS tests and the standard prefix ping test

Test name	Outcome	# of probes	Percent
DNS test 1	Failed	5938	96.49
	Passed	201	3.27
	Inconclusive	15	0.24
DNS test 2	Failed	6105	99.20
	Passed	44	0.71
	Inconclusive	5	0.08
std prefix ping test	Failed	6080	98.80
	Passed	66	1.07
	Inconclusive	8	0.13

The **Custom Prefix `ping` Test** is similar to the Standard Prefix `ping` Test, but uses the custom prefixes discovered through the DNS tests. It was used to check that probes with a DNS64 that uses a custom prefix have a functioning NAT64. If a probe received a response with a non-standard prefix (not 64:ff9b::/96) in DNS Test 1, I checked if it has a functioning NAT64 by encoding the address of RIPE Atlas anchor probe 6771 (91.201.7.243, as above) into an IPv6 address with the non-standard prefix, and pinging the address.

Additionally, I used the Custom Prefix `ping` Test to find more NAT64 probes: for each probe that passed DNS Test 1 and discovered a custom prefix, all other probes in the same AS performed the Custom Prefix `ping` Test using that prefix. This made it possible to find probes that are able to ping addresses with a non-standard NAT64 prefix, but are not configured to use the DNS64.

To maximise the set of available probes, DNS Test 1, DNS Test 2, and the Standard Prefix `ping` Test were run repeatedly across the set of available probes for several weeks, to account for probes that were not available in the initial tests. In total, 6154 probes performed these three tests. Only 765 of these probes performed the Custom Prefix `ping` Test, so this set of probes might not include all probes that would theoretically be able to ping a NAT64, as they also need to be in the same AS as a probe that passed DNS Test 1.

Table 1 shows the results of DNS Test 1, DNS Test 2, and the Standard Prefix `ping` Test. "Inconclusive" means that the probes had different results on different runs of the test, e.g. failed one time but succeeded another time. Due to the way the tests were run some probes were tested several times; all results were recorded. This means that some probes are more likely to have inconclusive results. Most RIPE Atlas probes don't use NAT64 and don't pass any of the tests.

Table 2 shows how many probes passed the different combinations of DNS Test 1, DNS Test 2, and the Standard Prefix `ping` Test (inconclusive results are not counted here). As in table 1, most probes did not pass any of the tests, with

Table 2: Number of probes that passed/failed different combinations of tests (F:Fail, P:Pass)

DNS test 1	P	P	P	P	F	F	F	F
DNS test 2	P	P	F	F	P	P	F	F
std prefix ping	P	F	P	F	P	F	P	F
# of probes	14	18	2	160	0	3	44	5889

only 241 probes conclusively passing any of the three tests.

4.1.1 Limitations of DNS Test 1

Of the 241 probes that conclusively passed any of the three tests, 160 passed DNS Test 1 but not DNS Test 2 or the Standard Prefix `ping` Test. Including the results from the Custom Prefix `ping` Test, 165 probes passed DNS Test 1 but not DNS Test 2 or the `ping` test for the prefix that they got from DNS Test 1 (5 probes that only passed DNS Test 1 received a custom prefix). These probes can resolve `ipv4only.arpa.` to an IPv6-mapped address but can't resolve other IPv4-only names or use NAT64.

Some DNS64 resolvers only create a synthetic AAAA record for `ipv4only.arpa.`, the host needs to extract the prefix from the result and create its own synthetic IPv6 addresses [6]. It can thus be possible for a probe to only pass DNS Test 1 but still have a functioning NAT64, but then it would also be able to ping addresses with the prefix that it got from the DNS64. Since these probes can't ping addresses with this prefix their DNS resolvers are most likely just misconfigured. The vast majority of these probes are in AS 12322, so this is mostly a localised problem.

Thus, it is not enough to rely DNS Test 1 to find NAT64 probes. This leads to false positives (probes that pass DNS Test 1 but don't have a functioning NAT64) and false negatives (probes that don't pass the test but can ping addresses with a NAT64 prefix). DNS Test 2 and the `ping` tests were used to check that the probes can actually use NAT64.

4.1.2 DNS Behaviour for NAT64 Probes

I distinguish two sets of probes that can be considered to have working NAT64: **NAT64+DNS64** and **NAT64-only**. Figure 4 shows the tests leading to each categorisation. The set of NAT64+DNS64 probes contains probes that passed DNS Test 1, DNS Test 2, and the `ping` test for the prefix that was returned by the DNS tests. The set of NAT64-only probes contains probes which failed one or both DNS tests, but passed the Standard or Custom Prefix `ping` Test (unless the prefix belongs to a public NAT64). Note that a probe is only considered to have passed a test if it passed it conclusively (i.e. passed every run of the test).

The purpose of this grouping is to separate probes with a fully functional NAT64+DNS64 setup from probes that can only ping addresses with a NAT64 prefix, and possibly are not meant to use the NAT64 at all.

To find the probes in the set NAT64+DNS64 I checked which probes passed DNS Test 1 and DNS Test 2. I found 36 such probes. Table 3 shows the result of them pinging an address with the prefix obtained from DNS Test 1 (the results are the same with the prefix from DNS Test 2).[6] As before, "inconclusive" means that the probes received different results on different runs of the test. Only 18 probes were able to consistently ping an address with the prefix returned by the DNS64. These 18 probes make up the set NAT64+DNS64. The other 18 probes passed DNS Test 1 and DNS Test 2 but couldn't consistently ping addresses with the prefix that they received from DNS Test 1.

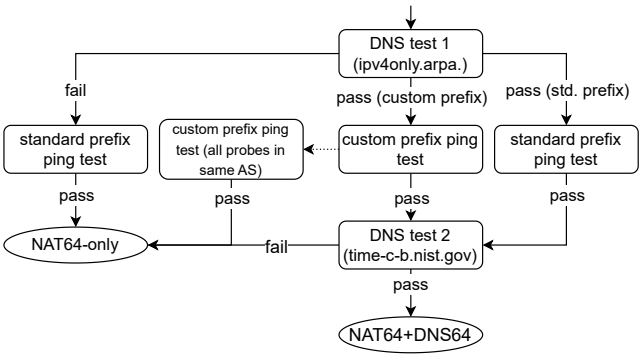

Figure 4: Sequence of tests and the resulting grouping.

Table 3: Result of the probes that passed DNS test 1 and 2 pinging the returned prefix

Test type	Result	# of probes
Standard prefix ping test	Passed	13
	Failed	10
	Inconclusive	4
Custom prefix ping test	Passed	5
	Failed	3
	Invalid	1

To find the probes in the set NAT64-only I checked which of the 6154 probes that performed DNS Test 1, DNS Test 2 and the Standard Prefix `ping` Test failed at least one of the DNS tests and passed one of the `ping` tests. Once I found this set of probes I checked which DNS test, if any, they did pass, and tested how many of these probes are likely to be able to use a NAT64 that they are not meant to use.

Of the 6118 probes that failed DNS Test 1 or 2 and performed one or both `ping` tests, 326 passed one or both `ping` tests, while 5792 were also not able to ping addresses with a NAT64 prefix. Table 4 shows the addresses that were pinged successfully (the encoded IPv4 address is always the address of the anchor probe 91.201.7.243). The addresses 2a0a:e5c0:2:10::5bc9:7f3[7] and 2001:67c:2960:6464::5bc9:7f3[8] use a public NAT64; these results were excluded because any host can ping these addresses. Excluding these addresses, there are 206 NAT64-only probes, of which 160 were able to ping custom prefixes and 52 were able to ping the standard prefix (6 were able to ping both).

203 of the 206 probes in NAT64-only failed both DNS tests. One probe passed DNS Test 1 but not DNS Test 2 (i.e. the prefix that it was able to ping was returned by DNS Test 1). It could be that this probe's DNS64 is misconfigured, or the probe *is* set up to use NAT64, but has to synthesise the address locally. I still count it as a NAT64-only probe because this kind of NAT64 and DNS64 is different from the setups used by the NAT64+DNS64 probes, which are fully

[6]The "invalid" result is due to a DNS64 returning a prefix which could not be pinged.

[7]https://redmine.ungleich.ch

[8]https://nat64.xyz/

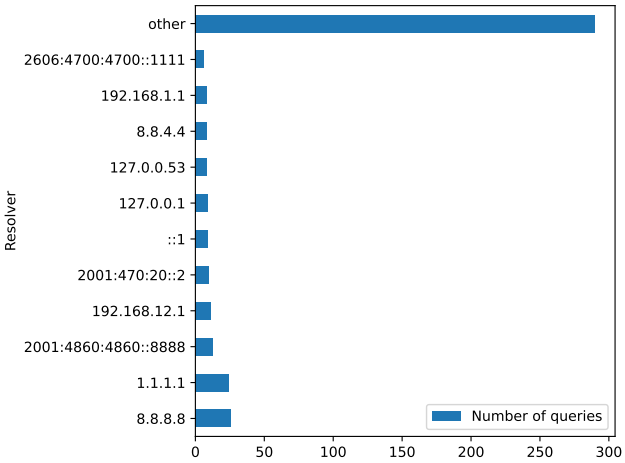

Figure 5: DNS resolvers used by NAT64-only probes (other: resolvers with < 5 queries)

transparent to the probe.

Two probes passed DNS Test 2 but failed DNS test 1. These probes passed DNS Test 1 most of the time; they failed once out of 13 and 23 runs of the test, respectively. Their DNS Test 1 results were thus counted as inconclusive (this is why they are not shown in table 2). It is possible that they are actually NAT64+DNS64 probes for which DNS Test 1 failed once due to external factors, but for consistency I still consider them to be NAT64-only probes.

A probe can either be in NAT64-only because it is not really supposed to use the NAT64, or because its configuration prevents it from using the DNS64. A probe that uses a public DNS resolver but is able to ping a NAT64 prefix might have been categorised as a NAT64+DNS64 probe if it used its default resolver instead. I checked if any of the resolvers used by the probes in NAT64-only for the DNS tests are in a list of public resolvers[9]. Many probes use several resolvers, I counted a probe as using a public resolver if any of the resolvers are in the list. Figure 5 shows the resolvers used by the NAT64-only probes. Out of the 206 probes in NAT64-only, 47 probes use a public DNS resolver from the list of resolvers. Excluding these probes and the NAT64-only probes that passed *any* DNS tests, 147 probes are probably not meant to use the NAT64. This difference is important when analysing the behaviour of these probes.

To summarise, 18 probes have a fully functional NAT64 and DNS64 setup: they pass both DNS tests and can ping addresses with the prefix returned by the DNS tests. 206 probes failed one or both DNS tests, but they are able to ping addresses with a NAT64 prefix. Of those probes, 147 are likely only able to access the NAT64 by accident. There are fewer probes in NAT64+DNS64 and more NAT64-only probes than expected. Generally, NAT64 is not widely used.

4.2 NAT64 probe characteristics

In this section, I describe basic characteristics of the NAT64 probes: their ASs, their physical location, and the prefixes that they use. This shows how many countries and ASs my measurements cover, and whether any countries or ASs have a particularly high concentration of NAT64 probes. If the probes cover a large number of countries and ASs then they

[9]https://github.com/trickest/resolvers/blob/main/resolvers-trusted.txt

Table 4: Number of NAT64-only probes that were able to ping various synthetic IPv6 addresses

Ping target	Number of probes
2001:470:703e:acfb:1:0:5bc9:7f3	146
2001:67c:2960:6464::5bc9:7f3	119
2607:7700:0:18:0:1:5bc9:7f3	8
2607:7700:0:27:0:1:5bc9:7f3	9
2607:7700:0:2d:0:1:5bc9:7f3	7
2607:7700:0:4:0:2:5bc9:7f3	5
2607:7700:0:9:0:2:5bc9:7f3	7
2a0a:e5c0:2:10::5bc9:7f3	1
64:ff9b::5bc9:7f3	52

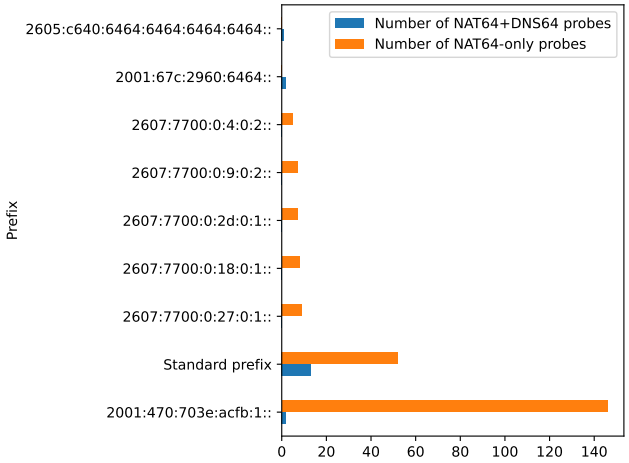

Figure 6: Prefixes used by the NAT64 probes

are more representative of all NAT64 hosts on the Internet.

Figure 6 shows which prefixes the two groups use (some NAT64-only probes are counted several times because they can ping several prefixes). For NAT64+DNS64 probes, the most common prefix is the standard NAT64 prefix, followed by 2001:67c:2960:6464::/96 (belonging to a public NAT64 provided by Level66) and 2001:470:703e:acfb:1::/40. This prefix is by far the most common prefix for NAT64-only, followed by the standard prefix. According to RIPEStat, 2001:470::/32 is announced by AS 6939 (Hurricane Electric)[10]. I expected many probes to use the standard prefix, as it is not specific to any AS. However this Hurricane Electric prefix is far more common for the NAT64-only probes.

Figure 7 and Figure 8 shows the ASs of NAT64+DNS64 and NAT64-only probes, respectively. Many ASs only contain only one NAT64-only probe; in order to make Figure 8 more readable all these ASs are represented by "Other AS". The most common AS for NAT64-only probes is AS 6939 (Hurricane Electric), which explains why so many NAT64-only probes can ping a NAT64 prefix in that AS. Many ASs only contain one NAT64+DNS64 probe as well. Thus, while the number of NAT64 probes found is comparatively small, they cover many ASs. Altogether, the probes are in 145 IPv4 ASs, and 44 IPv6 ASs. The larger number of IPv4 ASs is to be expected since NAT64 is relatively uncommon.

The set of NAT64+DNS64 probes contains a large number of IPv6-only probes. This makes sense, as NAT64 is meant to be used by IPv6-only hosts. However, surprisingly most

[10]https://stat.ripe.net

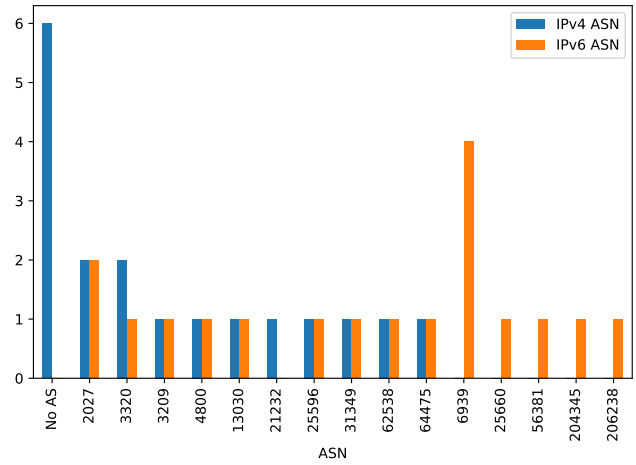

Figure 7: ASNs of the probes in NAT64+DNS64

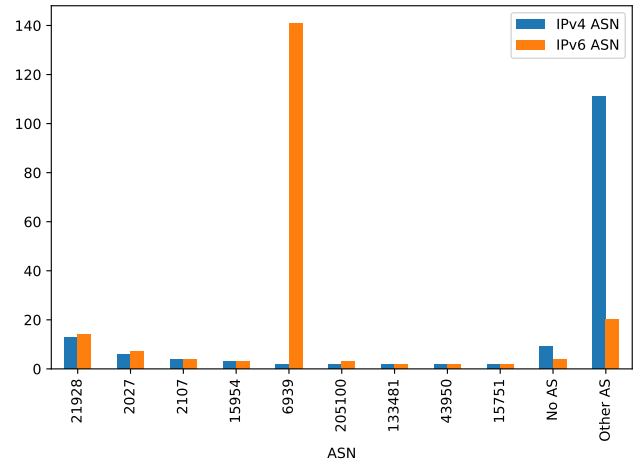

Figure 8: ASNs of the probes in NAT64-only

NAT64+DNS64 probes *are* dual-stack probes: they use an IPv6 transition mechanism but also use native IPv4. Most NAT64-only probes are dual-stack probes, the number of IPv6-only probes is comparatively low. This is expected, as these probes don't have a fully functioning NAT64+DNS64 setup (with a few possible exceptions, see section 4.1.2). Interestingly, this set also contains some IPv4-only probes.

Figure 9 shows which countries the probes are in, and Figure 10 shows their locations on the map. The probes in the set NAT64+DNS64 are mainly located in Europe, with a few probes in the US and one in Indonesia. The NAT64-only probes are more spread out: while most probes are in North America and Europe there are also some in Asia, Oceania, and South America. The country with the most NAT64-only probes is the US, followed by Germany, France and Russia. The US is also the country with the most RIPE Atlas probes, followed by Germany and France; Russia has the 6th most probes. Thus, while the NAT64 probes are not distributed evenly across the world, they do roughly follow the global distribution of RIPE Atlas probes.

4.3 Summary

Probes that use NAT64 are rare on RIPE Atlas. Out of 6154 probes, 18 probes have a fully functional NAT64 and DNS64 setup. A further 206 probes can reach a NAT64 but

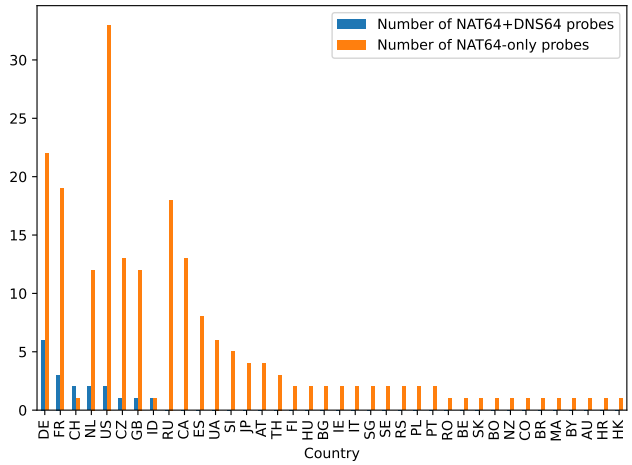

Figure 9: Countries of the NAT64 probes

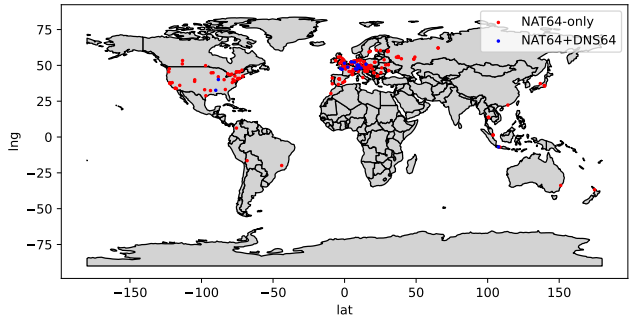

Figure 10: Locations of the NAT64 probes

don't use DNS64, of which 147 probes appear to just be able to reach the NAT64 by accident. It is not enough to rely on the standard NAT64 prefix discovery (DNS Test 1) to identify NAT64 probes, as some probes passed the test but did not have a working NAT64, and some probes failed but could still access a NAT64.

The probes are present in 145 IPv4 ASs and 44 IPv6 ASs, and roughly follow the geographical distribution of RIPE Atlas probes (which is skewed towards certain countries). The most commonly used NAT64 prefix is a custom prefix in the Hurricane Electric AS, followed by the standard prefix.

5. TRACEROUTE MEASUREMENTS

I performed traceroutes to 18 IPv4-only targets, using the dual-stack probes in the set of NAT64+DNS64 and NAT64-only probes. This allowed me to see how NAT64 changes path characteristics, compared to native IPv4. Using dual-stack probes enabled me to more directly compare NAT64 and native IPv4. Section 5.1 describes the methodology.

I studied how NAT64 affects traceroute itself, i.e. whether it has an effect on the number of missing hops or whether the traceroute reaches the destination (Section 5.2). I also investigated how NAT64 affects latency: I studied how the path length and RTT differ between native IPv4 and NAT64 paths, and how similar the paths are (Section 5.3).

5.1 Methodology

There are 193 dual-stack NAT64-only probes, and 12 dual-stack NAT64+DNS64 probes, so 205 probes were selected to perform these measurements (however only 183 probes ac-

tually participated in the measurements). The targets are seven IPv4-only NTP servers (dodo.mcc.ac.uk, d.st1.ntp.br, time-c-b.nist.gov, ntp1.nog.net.za, ntp1.st.keio.ac.jp, time-b-g.nist.gov, ntp2.urz.uni-heidelberg.de) and 11 IPv4-only RIPE Atlas anchor probes (probe IDs 6771, 6994, 6678, 6827, 6688, 6356, 6366, 6138, 6712, 6299, 6711). The NTP servers were chosen because they are present in a variety of countries and I assume they use a simple hosting setup without CDNs or other techniques that could lead to inconsistent results between probes. The anchor probes were chosen because they are likely to respond to traceroutes. Some of the targets don't respond to traceroutes at all, though, this made it possible to see differences between IPv4 and NAT64 in paths that don't reach the destination.

The IPv6 addresses of the targets were synthesised locally using the prefixes obtained during the NAT64 search experiments (see Section 4). I did not perform separate DNS lookups for the traceroutes to ensure that the same prefixes that were discovered in the NAT64 search are used here.

The traceroutes were performed using Paris Traceroute [2]. Three UDP probing packets were sent for each hop, but in order to reduce the complexity of the data I only used the first (non-missing) address for each hop in the analysis. I ran enough traceroutes to have one IPv4 path and one IPv6 path from every probe and NAT64 prefix that the probe was able to ping, to every target.[11] However the probes were not always able to perform all traceroutes so the set is missing some routes.

When RIPE Atlas encounters more than 5 missing hops in a row, it ends the traceroute[12]. Many of the traceroutes, particularly in IPv6, have many missing hops (see Section 5.2.2). In order to have a complete route I had to run several traceroutes, the first starting with TTL=1, the second starting with TTL=1+(last TTL of the previous path) etc, and concatenate the results. This was done several weeks after the initial measurements, as this feature is undocumented and I wasn't aware of it initially. As a result, the IPv4 and IPv6 traceroutes weren't always performed at the same time, and some traceroutes ran over the course of several hours.

5.1.1 Path overview and excluded paths

The 183 probes that participated in the traceroutes produced 3565 pairs of paths. Most probes performed one traceroute for each target and address family; some probes can ping several prefixes and thus performed several traceroutes for each target and address family. I expected the structure of the paths to follow that of Listing 1: it starts with regular IPv6 hops, and after a certain point all hops start with the NAT64 prefix. The addresses with the NAT64 prefix encode an IPv4 address, usually in the last 32 bits. I expected some paths to not reach the destination, ending in a sequence of missing hops until reaching the maximum length (32 hops).

Some paths had to be excluded from the measurement set because they deviated from this expectation too much to be useful for my analysis. Firstly, 2320 of the traceroutes to an IPv6-mapped address don't contain a hop starting with the NAT64 prefix, they have a similar structure to the path shown in Listing 2. It is likely that these NAT64s don't respond to traceroutes, and block ICMP time exceeded and port unreachable packets from hosts beyond the NAT64.

The vast majority of these paths (2291 paths) got to addresses with the prefix 2001:470:703e:acfb:1::/40, which is the most common NAT64 prefix found during the NAT64 search (see Figure 6). For this prefix, only the paths starting at the NAT64+DNS64 probe that uses this prefix contain hops with this prefix[13]. It is not clear why the probes were able to ping targets with these NAT64 prefixes, but the NAT64s appear to block traceroutes. I have excluded these paths from further analysis, as they don't include the NAT64 prefix and are thus not "true" NAT64 paths.

Listing 2: Traceroute to an IPv6-mapped address that doesn't contain a hop with the NAT64 prefix 2001:470:703e:acfb:1::/40.

```
Traceroute to
2001:470:703e:acfb:1:0:66de:6699
1  2001:470:0:6f7::1
2, 3, 4  *
5  2001:470:0:4b6::2
6, 7 *
8  2001:470:0:43f::1
9, 10, 11  *
12 2001:470:0:3ea::2
13, 14, 15 *
16 2001:470:0:63d::1
17 2001:470:0:69::2
18 2a01:4f8:c2c:b754::1
19 *
20 2001:470:703e:acff::2
<the rest of the path is missing hops>
```

I also excluded the NAT64 probe in AS 34779. It appears to be able to successfully traceroute to any target: the destination address appears somewhere in every traceroute, and for most targets it appears several times, with different TTL values. This is even the case for bogon addresses (I tested it with a traceroute to 198.51.100.1, from the IPv4 TEST-NET-2). It is possible that some hops on these paths set the source address of the ICMP time exceeded packets to the destination address of the traceroute. This probe is excluded because this makes it impossible to know the real addresses of the hops and whether the path reaches the destination.

Excluding these paths, the measurement set contains 1230 paths, starting at 54 probes. 45 of these probes are in the set NAT64-only and nine are NAT64+DNS64 probes. 44 probes use one NAT64 prefix, four probes use two, three use three and three use four prefixes. I have a full set of 18 measurements (one for each target) for 51 {probe, NAT64 prefix} combinations, 22 {probe, NAT64 prefix} combinations lack paths to some targets (e.g. because the probe was not available at the time).

5.1.2 NAT64 locations

One way of grouping the paths is by the AS that the NAT64 is in, relative to the AS(s) of the probe. This grouping will be used in the following sections.

I determined the AS of the NAT64 in the following way: if the NAT64 uses a non-standard prefix, then the AS is the AS of that prefix. If that prefix is not announced, or the standard prefix is used, then the AS is the AS of the last address in the path that doesn't use the NAT64 prefix and that is announced. If no AS can be determined via this

[11] I will refer to the traceroutes through the NAT64 as IPv6 traceroutes.

[12] https://www.ripe.net/ripe/mail/archives/ripe-atlas/2023-February/005393.html

[13] The NAT64 appears to be a custom setup by the owner of the NAT64+DNS64 probe

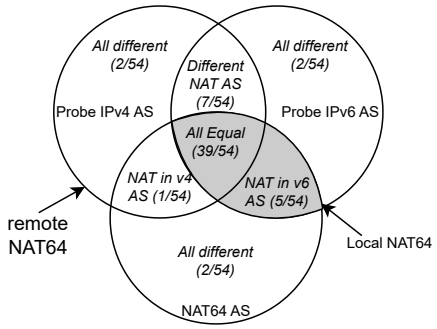

Figure 11: Different NAT64 locations and their categorisation

method (neither the NAT64 prefix nor any of the hops before it are announced) then I assume that the NAT64 is in the same AS as the IPv6 AS of the probe. The reasoning is that the path will most likely enter the AS of the NAT64 a few hops before the NAT64, the NAT64 is unlikely to be in a different AS than the hop before. If none of the hops before the NAT64 are announced the path is unlikely to have left the probe's AS. This is just an approximation, especially if the path contains many missing hops.

Figure 11 shows the possible combinations of ASs and the number of probes with that configuration. In most cases, the probe's IPv4 and IPv6 ASs and the AS of the NAT64 are the same. The NAT64s in the All Equal and NAT in v6 AS categories can be considered to be **local NAT64s**, they are most likely provided by the ISPs or set up in the probes local network. All other categories are considered **remote NAT64s** - the path has to leave the local AS to go to the NAT64. These are public NAT64 services, and NAT64s that are accessible from outside the AS. To summarise, if the NAT64 is in the probe's IPv6 AS then it is a local NAT64, otherwise it is a remote NAT64.

Altogether, there are 44 probes with a local NAT64, and 10 probes with a remote NAT64 (probes that can use several NAT64s can be in both groups at the same time). Of the probes with a local NAT64, 81.82% are in NAT64-only, compared to 90.00% of probes with a remote NAT.

5.1.3 Public non-public NAT64s

Out of the nine probes with a remote NAT64 that are in the set NAT64-only, five are probably not meant to use the NAT64 - they did not pass any DNS tests and don't use a public DNS resolver. Interestingly, 100.0% of those probes use the standard NAT64 prefix. They can ping addresses with the standard NAT64 prefix, even though they don't use DNS64 and aren't in an AS with a NAT64 that uses the standard prefix. The packets are routed to another AS with such a NAT64. It is possible that this AS is the default destination for all packets, and it happens to have a NAT64 which doesn't check the source address of packets before translating them. This is a possible security risk, as it can be used to hide the source address of the packets.

5.1.4 Summary

In total, 183 dual-stack probes performed 3565 IPv4 and IPv6 traceroutes to 18 different IPv4-only targets. Only 1230 of these pairs are usable for this analysis, 2335 pairs had to be excluded: 2320 paths going through NAT64s contain a NAT64 that appears to block traceroutes. One probe appears to be able to traceroute to invalid targets and its traceroutes contain the destination address several times.

I classify the paths based on where the NAT64 is in relation to the probe: in paths with a local NAT64 the NAT64 is in the IPv6 AS of the probe, in paths with a remote NAT64 it's in a different AS. Some probes are able to use a remote NAT64 that they are probably not meant to use, this could be a security risk.

Table 5: Percent success rate for groups of probes/NAT64s

Set	IPv4 success	IPv6 success
NAT64-only	68.07	62.09
NAT64+DNS64	64.78	59.75
Local NAT64	67.64	61.60
Remote NAT64	67.65	62.94

5.2 Impact of NAT64 on traceroute

First, I discuss whether the presence of a NAT64 has an effect on the functioning of traceroute itself. I study whether traceroutes through a NAT64 are less likely to reach the destination, and whether they have more missing hops than the corresponding IPv4 paths. This provides context for the following sections, as they involve analysing the traceroute data. This analysis is also important on its own, though - traceroute is widely used for troubleshooting and research. If NAT64s have a negative impact on the functionality of traceroute, and NAT64s become more widely used, this will make using traceroute more difficult in general. Finally, it also shows how NAT64s handle ICMP packets.

5.2.1 Does NAT64 affect traceroute's success rate?

I consider a traceroute to be successful if it contains the destination address (it does not need to be the last address on the path). The targets were not explicitly chosen to all respond to traceroutes, so I did not expect all traceroutes to succeed. However, I did expect the percentage of traceroutes that reach the destination to be very similar for the IPv4 and IPv6 paths and for the different groups of probes.

The overall success rate across all IPv4 paths is 67.64%, the IPv6 (i.e. via the NAT64) success rate is 61.79%: IPv6 is somewhat less successful than IPv4 (5.85% difference in success rates). There are 759 pairs of IPv4 and IPv6 paths from the same probe to the same target that both reached the destination (61.71% of pairs). Table 5 shows the success rate for the groups defined in Section 4 and Section 5.1.2. The success rates don't differ much between the groups, and are similar to the overall success rates.

Figure 12 shows the success rates split by NAT64 prefix (the standard prefix is further split by the probe's IPv6 AS). This approximates a splitting by NAT64, but note that some probes in different ASs use the same NAT64 (see Section 5.1.3). I still decided to split based on prefix and origin AS, as the paths to the NAT64 will be different for those probes even if the NAT64 is the same. This grouping is also used in following sections. IPv4 and IPv6 have similar success rates, though IPv6 is somewhat less successful. For unknown reasons the probe using the standard prefix in AS 213318 has an especially high success rate.

Figure 13 shows the success rate for each target, and also the percentage of paths that didn't reach the target, but that did reach the AS of the target. 10.73% of unsuccessful

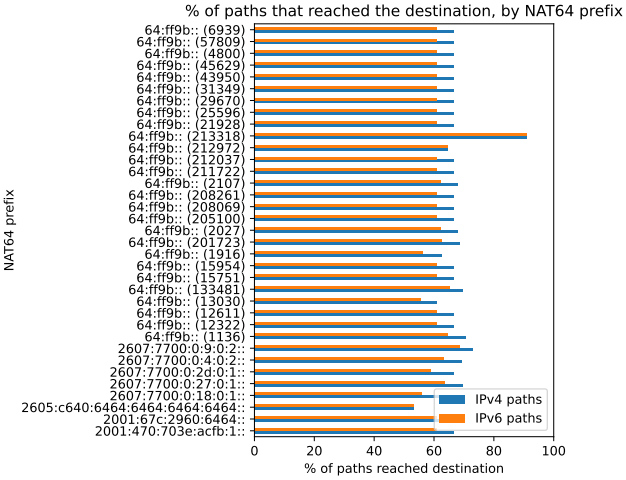

Figure 12: % of paths that reached the destination, by NAT64

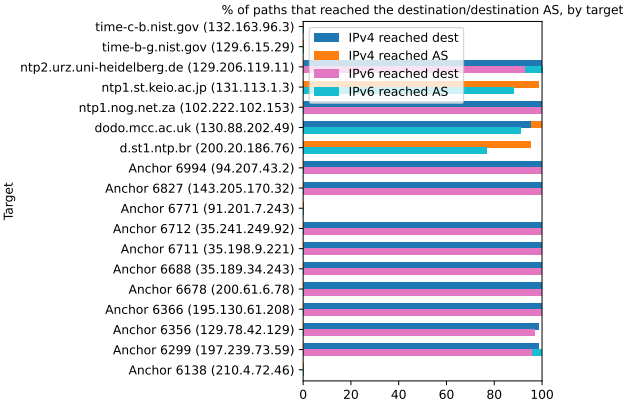

Figure 13: % of paths that reached the destination or destination AS, by target

IPv4 paths and 14.55% of unsuccessful IPv6 paths reached the target AS. Six of the targets have a 0% success rate for all probes, though for two of these targets most IPv4 and IPv6 paths do reach the target AS. It appears that these targets do not respond to traceroute.

There is one unexplained case: one target, the NTP server dodo.mcc.ac.uk, can only be reached in IPv4, though most of the IPv6 paths do reach its AS. The server should not be able to distinguish between IPv4 that has been translated from IPv6, and native IPv4, and only respond to the latter. Excluding this target, the success rates are 66.04% for IPv4 and 65.35% for IPv6, so the difference in success rates is primarily because of this target. One of the probes uses a public NAT64 prefix (2001:67c:2960:6464::), so I tried to reproduce this behaviour from my home network, varying different parameters (source port, destination port, protocol) on Paris Traceroute. In every case either the traceroutes would not succeed for any destinations, or they also succeeded for this target. The source code for RIPE Atlas also shows that it chooses the source and destination ports for IPv4 and IPv6 in the same way. It's possible that this behaviour is due to a subtle difference in how RIPE Atlas runs IPv4 and IPv6 traceroutes, but it is not clear what it is.

5.2.2 *Does NAT64 increase the number of missing hops?*

A missing hops is a hop that does not respond to traceroute probing packets with an ICMP time exceeded or port unreachable message. A high number of missing hops impacts the accuracy of the other metrics, especially the path similarity metrics (Section 5.3.3). I expected the number of missing hops to be similar in IPv4 and IPv6.

Figure 14 shows the distribution of percentages of missing hops. I only considered paths where the IPv4 path and the equivalent IPv6 path reached the destination (paths that don't reach the destination end in a string of missing hops, which skews the results). Paths with no or few missing hops are more common in IPv4. The mean percentage of missing hops for the IPv4 paths is 14.86% (SD 11.94, median 14.29%), in IPv6 it is 35.50% (SD 17.18, median 33.33%).

A plausible explanation for the greater amount of missing hops in IPv6 would be that the NAT64s stop the ICMP Time Exceeded packets from getting back to the probe. However this does not seem to be the case. There are more missing hops following the NAT64 than before, but if the NAT64 was stopping the traceroutes there would be many paths without any IPv6-mapped addresses. This is not the case - all paths considered here contain at least one hop starting with the NAT64 prefix, paths with no such hop were excluded (Section 5.1.1 - it is possible that these excluded paths do have a NAT64 that filters out traceroutes). In fact, most paths contain several hops starting with the NAT64 prefix. If the NAT64s filtered out traceroute packets there would likely also be probes that are only able to reach destinations in IPv4, which is not the case (see Section 5.2.1). A more likely explanation for the higher number of missing hops after the NAT is hops in the probe's local network being less likely to drop packets than hops close to the target (e.g. the target's firewall).

Another possible explanation for the higher number of missing hops in IPv6 is that they are caused by the same phenomenon that causes the IPv6 traceroutes to dodo.mcc.ac.uk to fail (Section 5.2.1). To test whether the missing hops are caused by the behaviour of particular routers, I repeated the IPv6 traceroutes to the two targets that the most probes could reach successfully (197.239.73.59 and 195.130.61.208), to find how often the same hops are missing in both traceroutes. It is impossible to definitively say that a missing hop in the first traceroute corresponds to a missing hop in the second, as the address of the missing hop is unknown. Instead, I considered all the runs of missing hops in the first traceroute which are preceded and followed by hops that are also in the second traceroute. For example, if traceroute one contains [address 1, missing hop, missing hop, address 2] I check if the second traceroute also contains address 1 and address 2, and then check if it also contains this exact sequence of hops (address 1 followed by two missing hops, followed by address 2). I only consider cases where addresses 1 and 2 are in both traceroutes in order to exclude cases where the second traceroute took a different path. On average, 23.42% of runs of missing hops were considered.

On average, 85.61% of runs of missing hops also occured in the second traceroute. (median 100.00%, SD 23.61%). Figure 15 shows the distribution of different percentages of recurring runs of missing hops. This provides some evidence that the missing hops are not due to random failure, but rather caused by specific hops not responding to traceroutes. Future work could involve investigating this further.

5.2.3 *Summary*

For most targets, IPv4 and IPv6/NAT64 traceroutes are

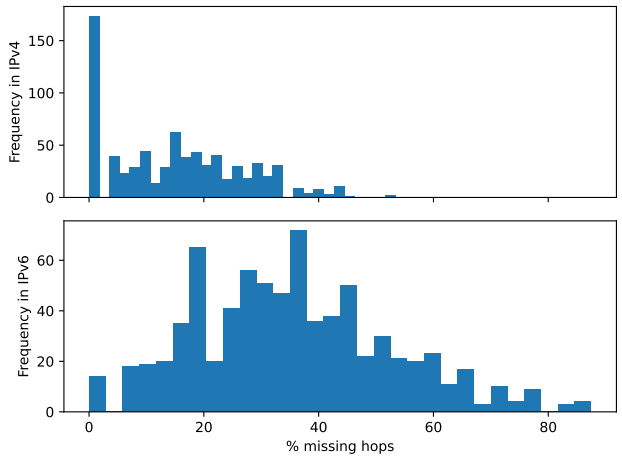

Figure 14: Distribution of the percentage of missing hops across all successful pairs of IPv4 and IPv6 paths

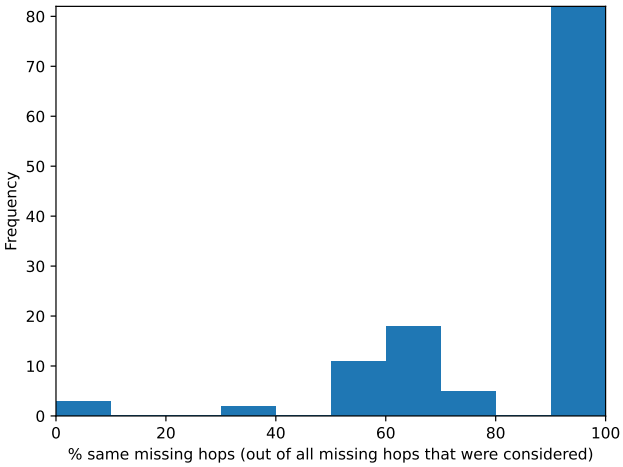

Figure 15: Distribution of the percentage of missing hops that probably appear in both traceroutes

almost equally likely to reach the destination. The overall success rate is 67.64% for IPv4 and 61.79% for IPv6, the difference is due to a target which for unknown reasons is only reachable via IPv4 traceroutes. 10.73% of IPv4 paths and 14.55% of IPv6 paths did not reach the destination but did reach the AS of the target. None of the NAT64s on the paths considered here block traceroute traffic.

The IPv6/NAT64 traceroutes have significantly more missing hops than the IPv4 paths. It is possible that the issue is caused by the same problem that makes one of the targets only reachable in IPv4. This affects the other metrics discussed in the paper, especially the path similarity metrics (Section 5.3.3), as less information is available about the contents of the IPv6 paths. If this problem is not caused by an issue with RIPE Atlas this suggests that NAT64 does interfere with traceroutes in some way.

5.3 Impact of NAT64 on Latency

In this section, I consider how the use of NAT64 affects latency, compared to native IPv4. If NAT64 substantially increases the latency then it is not a suitable replacement for native IPv4. I consider differences in path length (Section 5.3.1) and RTT (Section 5.3.2). I also analyse the similarity

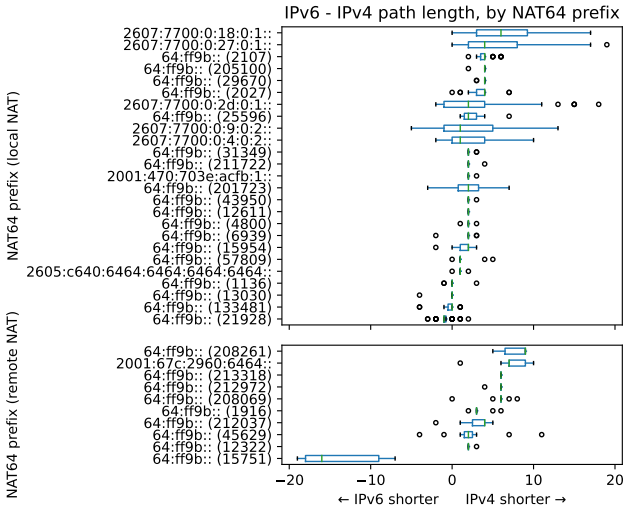

Figure 16: IPv6-IPv4 path length, by NAT64 prefix

Table 6: Basic statistics on the length difference, by NAT64 location

	All paths	Remote NAT64	Local NAT64
mean	2.96	2.95	2.96
std	4.12	6.47	3.60
median	2.00	5.00	2.00

of the paths (Section 5.3.3) by measuring how many hops and ASs the paths have in common, and where the paths diverge and converge. This provides context for the path length and RTT metrics: if the paths are similar, then differences in latency are more likely to be due to the NAT64 and not other path characteristics.

In the sections concerning length difference (Section 5.3.1) and RTT (Section 5.3.2) I only consider paths where both the IPv4 path and the corresponding IPv6 path reached the destination. These are 759 pairs of paths (61.71% of pairs).

5.3.1 Does NAT64 affect the path length?

I expected the IPv6 paths to be slightly longer than the IPv4 paths on average, as some paths, especially those with a remote NAT64, will need to take a detour to get to the NAT64. I also expected the paths with a remote NAT64 to have a higher length difference than the local NAT64 paths.

The average path length across all IPv4 paths is 15.16 hops (SD 6.18, median 14.00), the average IPv6 path length is 18.13 (SD 6.77, median 17.00). Figure 16 shows the difference in path length (IPv6 length - IPv4 length), split by NAT64 prefix and NAT64 location (local and remote). Table 6 shows the mean and median length difference and standard deviation for all paths and the two NAT64 locations.

As expected, the IPv6 paths are about 3-5 hops longer than the IPv4 paths. The paths with a remote NAT64 have a higher median length difference than the local NAT64 paths, but the means are almost the same, which is unexpected. There are also some outliers, which I discuss in the following.

The probe in AS 15751 using the standard prefix has IPv6 paths that are much shorter than the corresponding IPv4 paths (see bottom of figure 16). Manual inspection of the paths for this probe shows that the IPv4 paths and the unsuccessful IPv6 paths are a “normal” length, but the IPv6 paths that do reach the destination all consist of only two

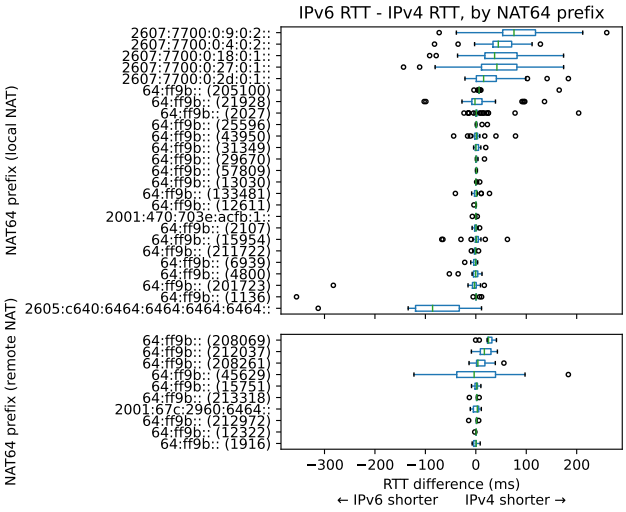

Figure 17: IPv6-IPv4 RTT, by NAT64 prefix, in ms

Table 7: Basic statistics on the RTT difference, by NAT64 location, in ms

	All paths	Remote NAT64	Local NAT64
mean	17.30	6.78	19.02
std	49.09	28.38	51.51
median	2.17	2.04	2.80

hops. This is also the case for the probe's native IPv6 paths, and is not due to the RIPE Atlas feature discussed in Section 5.1. I repeated one traceroute with this probe with an initial TTL of 3, and the path had the same length as the corresponding IPv4 traceroute. It appears that hop two is manipulating the TTL in some way, possibly changing it to a high values so the traceroute reaches the destination right away (perhaps to reduce traffic on the network). It then re-sets the TTL in the reply to the original value. If the destination doesn't reply this fails because there is no response from the target, so those traceroutes continue on as normal.

Some of the prefixes with the largest length difference are of the form 2607:7700:0:x:0:y (where x and y vary). These appear to be 464XLAT prefixes (see Section 2.1): The prefixes belong to T-Mobile US (based on WHOIS information from RIPE Stat), which first introduced 464XLAT, some of the probes are tagged 464XLAT, and the traceroutes show that two address translations are performed on the IPv4 path. For five of these probes, the IPv6 addresses are translated several times: they go from regular IPv6 to IPv6 link local addresses, to IPv4-mapped IPv6, back to IPv6 link local, and then finally to addresses starting with the NAT64 prefix. Some of the regular IPv6 traceroutes for these probes also contain IPv4-mapped IPv6 addresses, so this is probably part of the network's IPv6 configuration. The other five 464XLAT probes go straight from the first link-local addresses to the NAT64 addresses. Either way, the IPv6 paths are made longer by these additional translations.

5.3.2 *Does NAT64 affect the RTT?*

I used the RTT measured by traceroute for the first hop that is equal to the target address, averaging across the RTT for all three traceroute probe packets (provided they reached the destination). I expected the RTT to be strongly correlated with the path length. I also expected the IPv6 RTT to

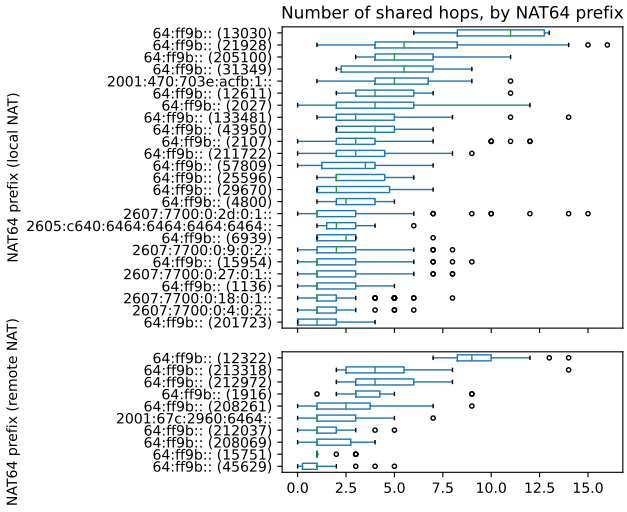

Figure 18: Number of shared hops between IPv4 and IPv6, by NAT64 prefix

Table 8: Basic statistics on the number of shared hops, by NAT64 location

	All paths	Remote NAT64	Local NAT64
mean	3.27	3.31	3.26
std	2.87	3.12	2.83
median	3.00	2.00	3.00

be larger than the IPv4 RTT, but not much larger; and I expected the paths with a remote NAT64s to have a larger RTT difference (IPv6-IPv4) than the local NAT64 paths.

The mean RTT across all IPv4 paths is 178.04 (SD 97.72, median 189.96), the mean IPv6 RTT is 195.34 (SD 102.32, median 210.90). Figure 17 shows the difference in RTT (IPv6 - IPv4), grouped by NAT64 prefix and NAT64 location. The prefixes are sorted by mean RTT difference. Table 7 shows the mean and median length difference and the standard deviation for all paths and the two NAT64 locations.

As expected, the IPv6 paths have a larger average RTT than the IPv4 paths, but the difference is close to zero. However, as can be seen in table 7, the local NAT64 paths have a higher average RTT difference than the remote NAT64 paths. This is because most of the prefixes with the highest average RTT difference are the 464XLAT prefixes, which are all in the local NAT64 set. Excluding those prefixes, the local NAT64 paths have a average RTT difference of -0.78ms.

Comparing Figure 16 and Figure 17, some of the paths with shorter IPv4 paths tend to also have shorter IPv4 RTTs, though there is only a moderate correlation between RTT difference and path length difference (Pearson correlation coefficient 0.35). The prefix with the smallest path length difference (standard prefix, AS 15751) has a mean RTT difference of 1.14, which further confirms that the length of the IPv6 traceroutes for that prefix is inaccurate.

5.3.3 *How much do NAT64 and IPv4 paths differ?*

In this section I look at the similarities between the IPv4 paths and the corresponding IPv6 paths. I consider how many hops they have in common, and where they diverge and converge. This provides additional context for the findings in Sections 5.3.1 and 5.3.2. As described in those sections, the paths with NAT64 have a higher RTT and more hops than the native IPv4 paths. Looking at the contents of

the paths can help determine the cause of these differences: if the IPv4 and IPv6 paths are very dissimilar the differences might also be due to factors other than the NAT64.

For each pair of IPv4 and IPv6 paths from the same probe to the same target I counted the number of shared hops between the paths. This was done for all paths, regardless of whether they reached the destination. I consider a hop to be a shared hop if the IPv6 address can be translated to the corresponding IPv4 address[14]. Each missing hop is considered to be unique, I don't assume that any missing hop in the IPv6 path corresponds to any missing hop in the IPv4 path. The number of shared hops shown here is thus a lower bound on the actual number. I expected the local NAT64 paths to have a large number of shared hops following the NAT64, as both the native and translated paths start in the same AS. By the same reasoning I expected the remote NAT64 paths to have fewer shared hops.

Table 8 shows basic statistics on the number of shared hops, split by NAT64 location. Figure 18 shows the number of shared hops, split by NAT64 prefixes and sorted by mean number of shared hops. The average number of shared hops is similar for paths with a remote and local NAT64, which is not what I expected. It is possible that some of remote NAT64 paths are similar to the IPv4 paths because the NAT64 is in an AS that the probe uses for IPv4 transit. It might also be due to the fact that the local NAT64s are, on average, *further* away from the probe than the remote NAT64s. For the local NAT64s the mean distance is 7.50 hops (SD 3.84, median 7.00), while the mean distance for the remote NAT64 paths is 5.79 (SD 2.31, median 5.00). Hops that precede the NAT64 can't be counted as shared hops: before the NAT64 the IPv6 path still uses native IPv6 addresses, so it is not possible to tell if they correspond to the same physical hop as an IPv4 hop. If more of the path comes before the NAT64 then there are fewer potential shared hops. It is not clear why the remote NAT64s are closer to the probe than the local NAT64s.

Figure 18 also shows that there are many paths with a small amount of shared hops. For many prefixes, the median number of shared hops is around 1. This is also not what I expected for the local NAT64 paths. Since this affects both the local and remote NAT64 paths this could be caused by the large number of missing hops.

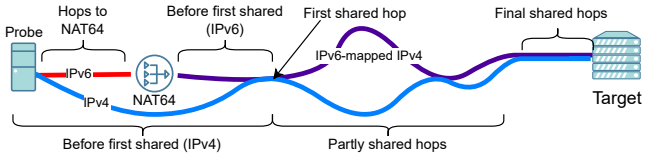

Figure 19: The different parts of the IPv4 and IPv6 paths

Figure 20 shows the average length, in hops, of the different parts of the IPv4 and IPv6 paths (regardless of whether they reached the destination), grouped by NAT64 prefix and NAT64 location, and sorted by the average distance to the NAT64 (in hops). The different parts are depicted in Figure 19. Hops to NAT64 is the number of hops needed to reach the first address starting with the NAT64 prefix. In IPv4, Before 1st shared is the number of hops from the probe to the first shared hop. In IPv6, it is the number of hops after the NAT64 before the first shared hop (as the hops to the

[14]Note that IPv4-mapped addresses are also translated here

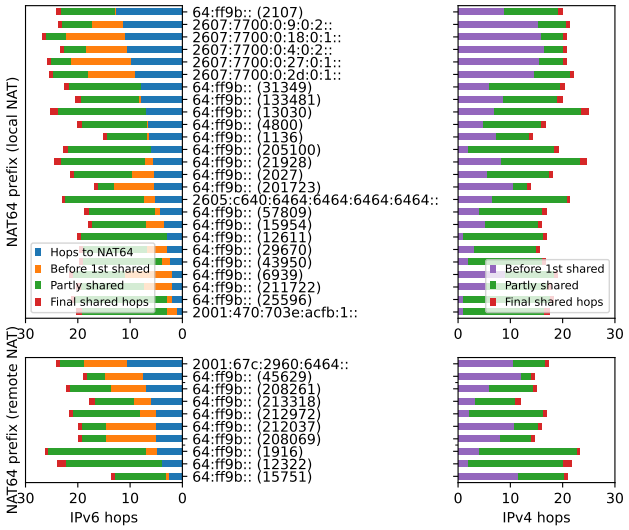

Figure 20: Average length in hops of different parts of the IPv4 and IPv6 paths, by NAT64 prefix

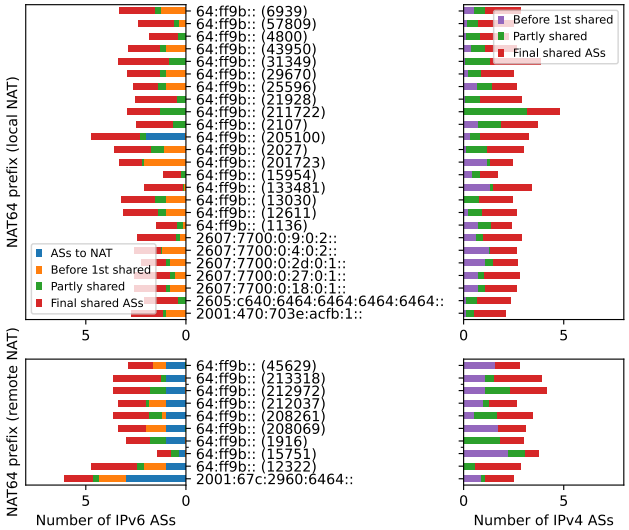

Figure 21: Average length in ASs of different parts of the IPv4 and IPv6 paths, by NAT64 prefix

NAT64 are also before the first shared hop). Final shared hops is the number of contiguous shared hops at the end of the IPv4 and IPv6 paths. Partly shared is the number of hops after the first shared hop, before the final shared hops.

In both groups, there is a lot of variation in the length of the different parts of the paths, with no clear patterns visible either within or across the two NAT64 locations. Generally, the number of final shared hops is low compared to the rest of the path, but the number of partly shared hops is high. The low average number of final shared hops is partly because not all paths considered here reached the destination. Some pairs encounter the first shared hop very soon after the beginning, while other paths are mostly not shared. The number of partly shared hops does not decrease as the distance to the NAT64 increases. This is counter-intuitive, one would expect that the further away the probe is from the NAT64, the further away it is also from the IPv4 path.

Figure 21 is similar to figure 20, but shows the number of

ASs on the path instead (not including the initial AS). Note that the probes using the standard prefix in AS 205100 have to traverse another AS to get to the NAT, even though it is a local NAT64: the path starts in one AS, then enters other ASs, and then returns to the original AS to get to the NAT.

The number of ASs traversed in IPv4 and IPv6 is similar: the mean difference in number of ASs (IPv6 ASs - IPv4 ASs, across all paths) is -0.10 (SD 1.20, median 0.00). The mean number of ASs traversed for the local NAT64 paths is 2.71 (SD 1.27, median 3.00), for the remote NAT64 paths it is 3.62 (SD 1.62, median 3.00). The remote NAT64 paths traverse about one AS more on average, which makes sense considering that the NAT64 is in another AS.

The number of shared ASs (between the IPv4 and IPv6 paths) is also similar for the local and remote NAT64 paths. The average number of shared ASs across all local NAT64 paths is 1.99 (SD 1.16, median 2.00), for the remote NAT64 paths is the average is 2.09 (SD 1.34, median 2.00). The AS-level analysis partly removes the uncertainty caused by the large number of missing hops (a similar approach was taken in [7]). This shows that the greater than expected similarity of the remote NAT64 paths is not just due to the missing hops skewing the results.

5.3.4 Summary

NAT64 has a moderate impact on path length and RTT, increasing the average number of hops by 25.65% (2.96 hops), and increasing the average RTT by 15.23% (17.30ms). There is a moderate correlation between differences in path length and RTT (Pearson correlation coefficient 0.35). Paths with a local NAT64 have a larger average RTT difference. This is because the 464XLAT paths all have comparatively high IPv6 RTTs. One probe's paths have a hop that appears to modify TTLs.

The IPv4 and IPv6 paths are less similar than expected, though most paths have a relatively large section of "partly shared" hops. The paths with a remote NAT64 are about as similar to their IPv4 equivalents as the paths with a local NAT64, which is surprising. This similarity is also apparent at the AS level. The remote NAT64 probes are also closer to the NAT64 than expected.

This means that the differences in RTT and path length found in sections 5.3.1 and 5.3.2 might not be caused solely by the NAT64, but rather by other differences in the paths.

6. RELATED WORK

Small scale studies have been done to evaluate the performance of NAT64 implementations. Lencse and Répás [14] compared the performance of the NAT64 implementations TAYGA and PF under load on a custom test network (consisting of the NAT64, eight IPv6 hosts, and one IPv4 host). Both TAYGA and PF degrade gracefully under load, but PF has better performance under load. Llanto and Yu [16] compare the performance of NAT44, NAT64 (TAYGA) and native IPv6 on a small test network and compare NAT44 and NAT64 on a larger university network. NAT64 and NAT44 had similar performance, the performance of IPv6 was better than NAT64. Tsetse et al. [25] used a small test deployment to measure the translation overhead of the IVI translator, a translator similar to NAT64 used by CERTNET [26]. Most of the translation time was used to translate the header, and translation was faster going from IPv6 to IPv4 than from IPv4 to IPv6. These studies are very different from the measurements done in this paper, as they are small-scale, fine grained measurements, focussing on specific characteristics of particular NAT64 implementations. My work does not distinguish between NAT64 implementations, and studies the broader impact of a larger number of NAT64s. I also compared the NAT64s with native IPv4, and not IPv6.

De Vries et al. [7] use RIPE Atlas investigate how much the forward and reverse paths in traceroute differ. This is similar to this work because it is also a large-scale traceroute measurement study investigating path similarities. The researchers primarily compare the paths using a similarity metric based on the Levenshtein string distance (on a per-AS basis). This approach (on a per-hop basis) was considered in this work as well, but it didn't produce useful results, probably due to the large number of missing hops.

7. CONCLUSIONS AND FUTURE WORK

This study provides a first insight into how NAT64 affects traceroute and various path characteristics. Future work will involve increasing the number of NAT64s and characteristics studied, and investigating the unexpected behaviours observed during the study.

While RIPE Atlas contains a large number of probes, only a small fraction uses NAT64. Future work could use a different measurement network to study the behaviour of a larger set of NAT64s. There are also other aspects of NAT64 that can be measured, such as the translation overhead, or the latency of DNS64.

Some of the NAT64s and DNS64 resolvers behaved in unexpected ways. Some probes receive a response to a query for the special-use name `ipv4only.arpa.` even though they do not appear to use NAT64 (Section 4.1.1). As described in Section 5.1.3, there are some probes that can access a (non-public) NAT64 from outside of their AS. Future work could investigate how common these behaviours are, and which issues (e.g. security risks) they create.

In Section 5.2.2, I found that the IPv6 paths have far more missing hops than the IPv4 paths, and in Section 5.2.1 I found one target that is only reachable via IPv4 traceroutes. Future work could find the cause of this. If these issues can be mitigated future work could repeat the path similarity analysis (Section 5.3.3), as the large number of missing hops might have skewed the results.

In this study I used DNS and ping measurements to search the RIPE Atlas network for NAT64 probes. The dual-stack NAT64 probes found performed traceroutes to 18 IPv4 targets, using IPv4 and NAT64. On average, the IPv6/NAT64 paths were 25.65% longer, had a 15.23% higher RTT, and included more missing hops. I also analysed how similar the IPv6 and IPv4 paths are, and found that the number of shared hops is generally low (about 3 shared hops on average) with no clear pattern to the location of shared hops. The large number of missing hops in the IPv6 paths has made this analysis more difficult. However the data still shows that the impact that NAT64 has on latency is only moderate, and most of the NAT64s studied did not prevent traceroutes. Thus, NAT64 is a workable substitute for native IPv4.

Acknowledgments. Thanks to my supervisor Colin Perkins for the support throughout the project. Thanks also to S3lph and Stephen Strowes for helping me understand the more subtle features of RIPE Atlas.

8. REFERENCES

[1] Internet Control Message Protocol. Request for Comments RFC 792, Internet Engineering Task Force, Sept. 1981. Num Pages: 21.

[2] B. Augustin, X. Cuvellier, B. Orgogozo, F. Viger, T. Friedman, M. Latapy, C. Magnien, and R. Teixeira. Avoiding traceroute anomalies with Paris traceroute. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, IMC '06, pages 153–158, New York, NY, USA, Oct. 2006. Association for Computing Machinery.

[3] V. Bajpai, S. J. Eravuchira, and J. Schönwälder. Lessons Learned From Using the RIPE Atlas Platform for Measurement Research. *ACM SIGCOMM Computer Communication Review*, 45(3):35–42, July 2015.

[4] M. Candela, V. Luconi, and A. Vecchio. Impact of the COVID-19 pandemic on the Internet latency: A large-scale study. *Computer Networks*, 182:107495, Dec. 2020.

[5] G. Chen, Z. Cao, C. Xie, and D. Binet. NAT64 Deployment Options and Experience. Request for Comments RFC 7269, Internet Engineering Task Force, June 2014. Num Pages: 22.

[6] S. Cheshire and D. Schinazi. Special Use Domain Name 'ipv4only.arpa'. Request for Comments RFC 8880, Internet Engineering Task Force, Aug. 2020. Num Pages: 17.

[7] W. de Vries, J. J. Santanna, A. Sperotto, and A. Pras. How Asymmetric Is the Internet? In S. Latré, M. Charalambides, J. François, C. Schmitt, and B. Stiller, editors, *Intelligent Mechanisms for Network Configuration and Security*, Lecture Notes in Computer Science, pages 113–125, Cham, 2015. Springer International Publishing.

[8] K. B. Egevang and P. Srisuresh. Traditional IP Network Address Translator (Traditional NAT). Request for Comments RFC 3022, Internet Engineering Task Force, Jan. 2001. Num Pages: 16.

[9] R. E. Gilligan and E. Nordmark. Basic Transition Mechanisms for IPv6 Hosts and Routers. Request for Comments RFC 4213, Internet Engineering Task Force, Oct. 2005. Num Pages: 27.

[10] M. Gupta and A. Conta. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. Request for Comments RFC 4443, Internet Engineering Task Force, Mar. 2006. Num Pages: 24.

[11] L. Hestina. Help Build a Bigger, Better RIPE Atlas, Oct. 2021.

[12] B. Hinden and S. E. Deering. Internet Protocol, Version 6 (IPv6) Specification. Request for Comments RFC 2460, Internet Engineering Task Force, Dec. 1998. Num Pages: 39.

[13] A. Keränen, C. Holmberg, and J. Rosenberg. Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal. Request for Comments RFC 8445, Internet Engineering Task Force, July 2018. Num Pages: 100.

[14] G. Lencse and S. Répás. Performance analysis and comparison of the TAYGA and of the PF NAT64 implementations. In *2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, pages 71–76, July 2013.

[15] X. Li, M. Boucadair, C. Huitema, M. Bagnulo, and C. Bao. IPv6 Addressing of IPv4/IPv6 Translators. Request for Comments RFC 6052, Internet Engineering Task Force, Oct. 2010. Num Pages: 18.

[16] K. J. O. Llanto and W. E. S. Yu. Performance of NAT64 versus NAT44 in the Context of IPv6 Migration. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, page 8, 2012.

[17] P. Matthews, I. v. Beijnum, and M. Bagnulo. Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. Request for Comments RFC 6146, Internet Engineering Task Force, Apr. 2011. Num Pages: 45.

[18] P. Matthews, A. Sullivan, I. v. Beijnum, and M. Bagnulo. DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers. Request for Comments RFC 6147, Internet Engineering Task Force, Apr. 2011. Num Pages: 32.

[19] M. Mawatari, M. Kawashima, and C. Byrne. 464XLAT: Combination of Stateful and Stateless Translation. Request for Comments RFC 6877, Internet Engineering Task Force, Apr. 2013. Num Pages: 14.

[20] G. C. M. Moura, S. Castro, J. Heidemann, and W. Hardaker. TsuNAME: exploiting misconfiguration and vulnerability to DDoS DNS. In *Proceedings of the 21st ACM Internet Measurement Conference*, IMC '21, pages 398–418, New York, NY, USA, Nov. 2021. Association for Computing Machinery.

[21] J. Postel. Internet Protocol. Request for Comments RFC 791, Internet Engineering Task Force, Sept. 1981. Num Pages: 51.

[22] P. Richter, M. Allman, R. Bush, and V. Paxson. A Primer on IPv4 Scarcity. *ACM SIGCOMM Computer Communication Review*, 45(2):21–31, Apr. 2015.

[23] RIPE NCC. Understanding IP Addressing and CIDR Charts, 2019.

[24] T. Savolainen, J. Korhonen, and D. Wing. Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis. Request for Comments RFC 7050, Internet Engineering Task Force, Nov. 2013. Num Pages: 22.

[25] A. K. Tsetse, A. L. Wijesinha, R. K. Karne, and A. Loukili. Measuring the IPv4-IPV6 IVI translation overhead. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, RACS '12, pages 186–190, New York, NY, USA, Oct. 2012. Association for Computing Machinery.

[26] J. Wu, H. Zhang, X. Li, M. Chen, and C. Bao. The China Education and Research Network (CERNET) IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition. Request for Comments RFC 6219, Internet Engineering Task Force, May 2011. Num Pages: 22.

[27] J. Zirngibl, L. Steger, P. Sattler, O. Gasser, and G. Carle. Rusty clusters? dusting an IPv6 research foundation. In *Proceedings of the 22nd ACM Internet Measurement Conference*, IMC '22, pages 395–409, New York, NY, USA, Oct. 2022. Association for Computing Machinery.