A Global Measurement of Routing Loops on the Internet

Abdulrahman Alaraj[1,3(✉)], Kevin Bock[2], Dave Levin[2], and Eric Wustrow[1]

[1] University of Colorado, Boulder, USA
abdulrahman.alaraj@colorado.edu
[2] University of Maryland, College Park, USA
[3] Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

Abstract. Persistent routing loops on the Internet are a common misconfiguration that can lead to packet loss, reliability issues, and can even exacerbate denial of service attacks. Unfortunately, obtaining a global view of routing loops is difficult. Distributed traceroute datasets from many vantage points can be used to find instances of routing loops, but they are typically sparse in the number of destinations they probe.

In this paper, we perform high-TTL traceroutes to the entire IPv4 Internet from a vantage point in order to enumerate routing loops and validate our results from a different vantage point. Our datasets contain traceroutes to two orders of magnitude more destinations than prior approaches that traceroute one IP per /24. Our results reveal over 24 million IP addresses with persistent routing loops on path, or approximately 0.6% of the IPv4 address space. We analyze the root causes of these loops and uncover new types of them that were unknown before. We also shed new light on their potential impact on the Internet.

We find over 320k /24 subnets with at least one routing loop present. In contrast, sending traceroutes only to the .1 address in each /24 (as prior approaches have done) finds only 26.5% of these looping subnets.

Our findings complement prior, more distributed approaches by providing a more complete view of routing loops in the Internet. To further assist in future work, we made our data publicly available.

1 Introduction

Routing loops* are the phenomenon in which packets never reach their destination because they loop among a sequence of routers. They are the result of network misconfigurations, inconsistencies, and errors in routing protocol implementations. In addition to being a pernicious threat to Internet reliability and reachability [30,13], routing loops can even enable or exacerbate denial of service attacks [42,26,3].

*The literature is split between referring to these as "routing loops" [26,13] or "forwarding loops" [42]. We use "routing loops" to differentiate them from loops that arise from application-level redirects [8].

Surprisingly little is known about the true global prevalence of routing loops. Although there have been several large, longitudinal, or distributed Internet measurements to detect routing loops [30,43,2,7], they tend to make several simplifying assumptions. For instance, one of the largest studies of routing loops of which we are aware [42] tracerouted only two IP addresses (`.1` and a random one) in each of about 5.5M /24 subnets—the implicit assumption being that addresses within a /24 will largely experience the same routing behavior. Unfortunately, we are unaware of any prior work validating such assumptions.

In this paper, we perform a straightforward yet illuminating experiment: we traceroute the entire IPv4 address space from two vantage points. We discover over 24 million IP addresses with routing loops: over 21× more than two concurrent, distributed traceroute scans [34,7] put together.

What allows us to scan many more addresses than prior work is that we do not perform *full* traceroutes of all destination IP addresses. Our insight is that we only need to use higher TTL values to discover routing loops; lower TTL values can largely be avoided. We use Yarrp [2], a large-scale network traceroute tool, to traceroute all 3.7 billion routable IPv4 addresses for a range of 10 TTLs per IP. To parameterize our scanning rate, we performed experiments to evaluate routers' maximum ICMP response rate, to avoid missing routing loops due to router response rate limits. By using fewer TTLs and more addresses, we are able to perform, to our knowledge, the most comprehensive study of routing loops to date.

We analyze our resulting dataset to better understand the nature of routing loops on today's Internet. In particular, we explore routing loops' root causes, locations, size, and potential impact.

Our results justify full-Internet scanning to discover routing loops: 35% of the /24 subnets in which we discovered routing loops have at most *10 IP addresses* experiencing routing loops. In addition, scanning just the `.1` address in each /24 would only discover 26.5% of the 320k unique routing-loop containing /24 subnets that we find when we scan every IP. Moreover, we discover that routing loops are not evenly distributed: within a /24 subnet, routing loops occur more often at higher last-octet values than low ones.

Collectively, our results demonstrate that the common strategy of scanning only one or two IP addresses [2,42,15,16,21,37] per /24 subnet is likely to miss many routing loops. In fact, we find that the common addresses that prior approaches sample—such as gateways (typically the `.1` address of a /24)—have the *least* routing loops.

Contributions We make the following contributions:

- We perform the largest traceroute study of the Internet to date from two vantage points, tracerouting over 3.7 billion IPv4 addresses.
- We analyze this dataset to understand the prevalence (§4) and structure (§5) of routing loops in today's Internet.
- We discover that sampling at the /24 subnet granularity is often insufficient to capture routing loops within the subnet.

- We compare our results with public distributed traceroute measurements, showing that we are able to detect 21× more looping destinations than prior efforts combined.
- We uncover a new type of routing loops, namely the transport-state dependent loops that can be abused for the TCP reflected amplification attacks.

To assist in future efforts, we made our tools and data publicly available at `https://github.com/RoutingLoops`.

Roadmap The rest of this paper is structured as follows: In §2, we offer a concrete definition of persistent routing loops and analyze the rate at which one can traceroute networks without getting blocked. In §3, we describe our measurement methodology. In §4, we analyze the prevalence of routing loops and validate our dataset. Then, in §5, we analyze the resulting dataset to learn about the structure of routing loops, their causes and potential impact. We review related work in §6, describe ethical considerations in §7, and conclude in §8.

2 Experiment Design

Our general goal and approach are straightforward: identifying IPv4 addresses with persistent routing loops by running partial traceroutes to every IPv4 address on the Internet. Although conceptually simple, there are several complications to doing this experiment in practice. In this section, we will describe our methodology and our experiments we used to design this approach.

To issue traceroutes, we use a modified version of Yarrp [2], a traceroute tool designed to scan large networks. To our knowledge, we are the first to use Yarrp to scan every IPv4 address; we will describe our minor modifications in Section 3 that enable us to use Yarrp in this way.

2.1 What constitutes a persistent routing loop?

We define a destination d as having a routing loop if our probes do not reach d (even with TTL=255) and we observe a router that appears at two different hops in the traceroute, separated by at least two hops. In other words, we must receive a response from the same router IP for two TTL values $t_1 < t_2$, such that $\Delta t = t_2 - t_1 > 2$. We perform further analysis of this definition in §4.

Requiring a small gap between repeated hops helps insulate us against router aliasing [36]: we observed traceroutes that had apparent repeated IPs in two sequential hops, but ultimately reached other routers or their endpoint in subsequent hops, indicating no routing loop.

Finally, to detect and eliminate transient routing loops [40,13] (e.g. an apparent routing loop caused by a network change or instability during our scans) from our dataset, we perform two consecutive Internet-wide scans 5 days apart. We apply the above routing loop definition, and take the intersection of destinations that have routing loops in both scans. We mark these destinations as having *persistent* routing loops on path.

2.2 Which TTLs should we scan?

Traditional traceroutes start by issuing TTL-limited probes with increasing TTL values (starting at TTL=1) until either the end destination is reached or a maximum TTL value is hit. To identify routing loops, however, we do not need to send so many probes. Instead, we will issue probes with ten TTL values 246–255 (inclusive).

Probing an endpoint d with such high TTL values and receiving ICMP Time Exceeded Messages indicates a routing loop on path to d. It could also indicate that d is 246–255 hops away from our scanning machine, or that there exist routers on path to d that respond with ICMP Time Exceeded Messages even if TTL $\neq 0$; however, we believe both to be rare events.

Note that like the traceroute tool itself, this approach cannot detect true infinite routing loops [3]: loops amongst routers that do not decrement the TTL (as this will never generate ICMP Time Exceeded messages).

2.3 How fast can we probe?

Many routers have *rate limits* on how fast they will respond with ICMP Time Exceeded messages. For tracerouting the Internet from a single vantage point, this could result in missing data: if we probe hop-sharing routes faster than their router's rate limit, those routers may only respond to a subset of our probes, which would cause our scan to miss hops and undercount routing loops.

To determine rate limits of routers, we wrote a Golang traceroute utility to send ICMP-eliciting probes to a sample of routers repeatedly at a specific rate, and observe the speed with which they respond. We selected a sample of 10,000 random IP addresses and sent probes with a TTL limit of 15 hops. We sent each IP address approximately 50,000 probes per second (about 22 mbps) for 1 second, and recorded the response rate we received. Our goal is to momentarily saturate the ICMP response rate limit of each router, allowing us to approximate its maximum response rate. To minimize interference between responses from routers, we scanned only 5 routers in parallel during our test.

Of the 10,000 target IPs we probed, 2803 (28%) had an on-path router that responded with an ICMP Time Exceeded message. In total, we observed 5361 unique router IPs[1]. The CDF of the maximum rate we received responses from these routers is shown in Figure 2. We observe a median rate limit of about 2,300*pps* (about 1.7 mbps), with small but clear steps around likely popular configuration such as 1,000, 2,500 or 15,000*pps*. Over 99% of routers that respond at all will respond to at least 3*pps*.

To determine our send rate for scanning the Internet, we start with a goal of averaging 3*pps* to any individual router (the 99th percentile response rate) to maximize the chance we receive a response. We observed from preliminary scans that there are over 35,000 unique router IPs at each hop we scan between 10-25

[1]Many targets had multiple routers that responded, due to load-balancing, ECMP, or the varying paths different probe packets take to reach their destination

hops from us. This means that for the 3.7 billion IPs we scan, we expect to hit each router on average 108,000 times each. If we want to send a probe to each one no more than 3 times a second, our scan should take approximately 36,000 seconds, which equates to a scan rate of 105*kpps*. We note that it is likely possible to scan much faster than this, as many routers will not respond to our probes at all, and higher TTLs will hit even fewer routers. Nonetheless, we conservatively scan at 100*kpps* for our full IPv4 scans, which takes approximately 10 hours per TTL value.

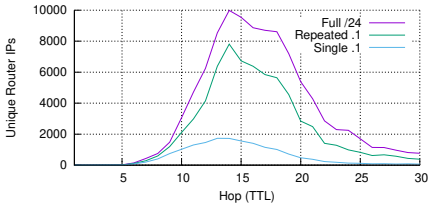


Fig. 1: **Unique Router IPs per hop** — We scanned 10,000 random /24 subnets for hops 1-30. For each /24, we scanned the full subnet (all 256 IPs) once (Full), as well as each gateway (`.1` address) of these subnets 256 times (Repeated .1). For each hop, we count the number of unique router IPs we observe. We find that traceroutes to only the gateway (`.1` address) in a given /24 fail to observe over 82% of router IPs beyond 14 hops (Single .1). This shows that scanning only a single address in a particular /24 subnet may not fully reveal all paths to addresses in that subnet. Note that the Repeated .1 scan used random destination ports.

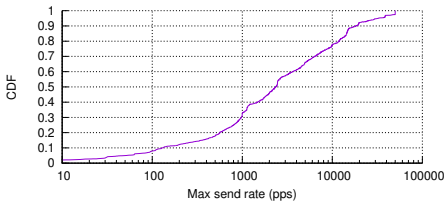


Fig. 2: **Maximum ICMP response rate** — We scanned 10,000 random IPv4 addresses at a rate of 50,000*pps* for 1 second each and observed the maximum response rate of ICMP Time Exceeded messages from the routers. The median rate was just over 2,300*pps*.

2.4 Can we sample subnets?

We initially hypothesized that we could significantly shorten our traceroute efforts by only scanning the gateway or a random IP per subnet. Many existing studies [5,2,15] adopt this hypothesis to only scan around 15 million IPs (one per /24) instead of 3.7 billion for a full IPv4 scan. However, we find that this hypothesis is incorrect, and that subnet level scans do not necessarily accurately represent a full IPv4 scan.

To evaluate this, we performed a short experiment using Yarrp: scanning a sample of 10,000 random /24 subnets in three different ways for TTLs (1–30) inclusive, one TTL value at a time. The goal of this experiment is to see if the gateway (`.1` address) for each /24 returns a representative topology for the entire subnet.

Figure 1 shows the number of unique routers we discovered in both full /24 scans (2.56 million IPs total), and the scans for the common gateway address (`.1`) for each /24. In our full /24 scan, we discovered 9,990 unique router IPs at hop 14, but scanning the gateway (Single .1) address returned only 1,723 unique router IPs (17%).

We hypothesized this difference may be caused by small path variations due to different flows being load-balanced to different paths. To account for this possibility, we performed a repeated gateway scan, where we sent 256 probes to the gateway (`.1`) address for each /24 (Repeated .1). For this repeated scan, we disabled Paris tracerouting [1] by using different destination ports to maximize the paths our traceroutes would take. This increased the number of unique router IPs discovered (e.g. 7,812 at hop 14), but still less than the number found in our full /24 scan.

These results motivate us to scan the entire IPv4 address space to obtain an accurate view of routing loops. We further evaluate this motivation using our results in Section 4.

3 Scanning Methodology

Informed by the experiments in the previous section, here we detail our scanning methodology. We will issue TCP probes to every IPv4 address with the `ACK` flag set to port 80, with TTL values from 246 to 255 (inclusive).

3.1 Using Yarrp

While Yarrp [2] is designed to traceroute large networks, we had to make several small modifications to allow it to scan *every* IPv4 address. Internally, Yarrp has an "Entire Internet" mode, designed to handle scanning large networks. Unfortunately, the number of bits available in this mode limits the possible permutation size for scanning to just 1 probe per /24 subnet; outside this mode, Yarrp's total permutation size is limited to a domain $(IPs \times max\_TTL)$ of size 2^{32}, still too small to scan the entire IPv4 Internet. We also discovered and fixed an integer overflow bug that prevented Yarrp from sending probes with $TTL > 127$.

To address these limitations, we provided Yarrp with a shuffled list of all 3.7 billion routable IPv4 addresses in a 50 GB file, and had it send probes to a single TTL at a time. We repeated this 10 times, for each TTL value in (246–255) inclusive. We limited Yarrp's sending rate to $100kpps$. As Yarrp validates and logs all of the ICMP Time Exceeded messages it receives, we can construct partial traceroutes for all IP addresses for the TTL hops we scanned.

Yarrp implements Paris traceroute [1] technique by calculating the source port as a function of the destination IP address. This technique maintains the 5-tuple for every destination probed regardless of the used TTL value. This leads to a more stable routing path that is less affected by per-flow load balancers.

3.2 Vantage Points

We performed two full IPv4 scans during the month of April 2022. The first started on the 14th and completed on the 18th. The second started on the 19th and completed on the 23rd. Both scans are from our vantage point[1] in the University of Colorado Boulder (AS104).

To validate our results, we used a secondary vantage point[2] in the University of Michigan (AS36375) to run two additional consecutive scans, running during the same period as the scans from VP 1.

4 The Prevalence of Routing Loops

We begin our analysis by investigating the number and persistence of routing loops in the Internet. We also validate our use of Yarrp by performing `traceroute` on each of the destinations we identified as routing loops.

4.1 How many routing loops are there?

In total, we find over 24 M looping IP addresses that persisted for two consecutive scans from VP 1.

For each scan from VP 1, we received ICMP Time Exceeded messages from over 500K unique router IPs, for over 29 million unique destinations from the 3.7 billion destinations we probed. In other words, our probes did not trigger ICMP Time Exceeded messages from the on-path routers for the majority of the 3.7 billion destinations we probed, and only triggered ICMP Time Exceeded messages from at least an on-path router for 29 million unique destinations. Of these 29 million destinations, only 24 million qualified as experiencing persistent routing loops based on our definition in §2.1. Over two thirds of the non-qualifying routing loops contain a single, non-repeating router IP; and only around 23,500 contain 10 unique router IPs, suggesting that the TTL range we scan (10 hops) is sufficient to discover most routing loops on the Internet.

We also investigate how tweaking the definition of routing loops changes the number that we find. Recall that our existing definition requires that a router IP appears at (at least) two distinct hops that are more than two hops apart. We explore the impact that "two hops apart" has in Table 1. We do not find a significant difference in the number of loops found as we vary to fewer or more hops apart, suggesting that the definition of routing loops is generally robust to slight changes.

[1] We refer to this vantage point as VP 1 for the rest of the paper.

[2] We refer to this vantage point as VP 2 for the rest of the paper.

Loop definition	1^{st} Scan	2^{nd} Scan	Persistent routing loops
$\Delta t > 0$	27.07 M	27.22 M	25.66 M
$\Delta t > 1$	27.02 M	27.15 M	25.59 M
$\mathbf{\Delta t > 2}$	26.26 M	26.36 M	**24.78 M**
$\Delta t > 3$	26.17 M	26.25 M	24.66 M
$\Delta t > 4$	24.99 M	25.09 M	23.45 M

Table 1: **Results by applying different routing loop definitions** — We apply different routing loop definitions on the reconstructed traceroutes for each scan from VP 1 to find persistent routing loops. Note that our scans spanned 10 hops.

4.2 Are these really loops?

To verify that the results we find from the Yarrp scans are not artifacts of our modified version of Yarrp, we used the `traceroute` utility to run TCP ACK traceroutes. This utility, however, is not designed for Internet-wide scanning: it has long timeouts, is not designed for scaling to scanning IPs simultaneously, and generally limited in speed. However, it does provide a useful check against our results.

To deal with traceroute's performance limitations, we limited its use to validating the list of 24 million looping IPs we found from VP 1 using Yarrp. We similarly limited it to the TTL range 246–255, a sending rate of 500 pps, and a timeout of 500ms. These parameters allow us to scan the 24 million IPs in a reasonable time, but note the aggressive timeouts may result in missed hops.

Using our same routing loop definition in §2.1, traceroute confirms 18 million IPs (out of 24 million) appear to be loops. We ran follow-up traceroutes with longer timeouts on a random sample of IPs found by Yarrp but not by traceroute, and found about 75% of them do have loops, suggesting that overall traceroute confirms over 93% of the loops we find. We believe the remaining difference is largely due to network churn (the traceroute scans happened a month after our original Yarrp scans).

Overall, we believe that while there may be loops that are transient (only present temporarily or for short periods of time), the vast majority of the loops that we identify are indeed persistent routing loops.

4.3 Do these loops persist across time / other vantage points?

Perspective from a second vantage point We performed two consecutive, full-Internet scans (TTLs 246–255) using Yarrp from VP 1. Simultaneously, we performed two additional scans from VP 2 which is in a different AS in the United States (located over 1,000 miles away).

Despite different locations, we find a significant overlap between the persistent routing loops found by our two vantage points. Table 2 shows the number of loops found in each scan. We apply the same routing loop definition in §2.1

Dataset	1^{st} Scan	2^{nd} Scan	**1^{st} Scan $\cap$ 2^{nd} Scan**	(Jaccard index)
VP 1	26.26 M	26.36 M	**24.78 M**	(0.8898)
VP 2	25.97 M	25.95 M	**24.59 M**	(0.8995)
VP 1 $\cap$ VP 2	24.46 M	24.47 M	**23.28 M**	(0.9076)

Table 2: **Scans Results** — We count the number of IP addresses with routing loops along their path. We consider a routing loop to be persistent from a vantage point if it exists in two consecutive scans from the respective vantage point. Note that 1^{st} Scan and 2^{nd} Scan are back-to-back consecutive scans run from each vantage point during the same period of time.

to find persistent loops, and find over 23 million persistent loops in common between both vantage points. Each vantage point also finds on the order of 5% of looping IPs that the other does not: VP 1 found 1.5 M loops that VP 2 did not, while VP 2 found 1.3 M not seen by VP 1.

We find that these relatively small differences are partly due to noise: for each of the 1.5 M (1.3 M) IPs discovered only by VP 1 (VP 2), we re-ran a follow-up Yarrp scan from VP 2 (VP 1). Each vantage point found over 400 K loops that it originally missed, suggesting that network noise (e.g. packet drops or ICMP rate limits) prevented us from finding these loops originally.

Perspective from previous scans In addition to our two consecutive scans in April 2022, we also compare our results from two preliminary scans in October 2021 from VP 1. At the time, these scans find similar results as our later scans (roughly 24 million IPs), and we observe 19 million destination IPs that have routing loops along their paths both in the October 2021 and April 2022 scans, suggesting that these loops may be persistent over multiple months (though without longitudinal scans, we cannot rule out the possibility that these loops were fixed and later reappeared).

4.4 How do existing datasets compare?

We analyzed routing loops present in the RIPE and CAIDA Ark datasets using the same routing loop definition in §2.1. This entails that we enforce the following requirement: for a destination in the RIPE dataset to be considered experiencing a routing loop, it must exhibit a looping behavior in two consecutive scanning windows, where each window is 5 consecutive days long. We enforce this requirement to exclude transient loops and maintain consistency with our routing loop definition in §2.1. Note that we confirmed that over 98% of destinations probed in the RIPE dataset are probed in both scanning windows at least once; however, these probes are not necessarily from the same source nor necessarily traversing the same path. We do not enforce this requirement on CAIDA Ark dataset since its random nature of scanning does not permit this kind of validation during the same period of time. This means that some of the routing loops we find in the CAIDA Ark dataset could potentially be transient loops. Therefore we do

Dataset	Traceroutes	Unique Destinations	Unique Router IPs	Looping Destinations (%)
This paper	7,400,351,422	3,700,175,711	512,785	24,783,989 (0.66%)
RIPE Atlas [34]	977,156,702	848,348	709,675	41,196 (4.85%)
CAIDA Ark [7]	213,333,652	206,007,571	2,645,943	1,137,830 (0.55%)

Table 3: **Traceroute Results** — We perform partial traceroute scans (TTLs 246–255) to the entire IPv4 Internet and compare the routing loops we discover from VP 1 to other distributed traceroute datasets.

not compare the persistent routing loops from our Internet-wide scans with it. Table 3 compares our dataset with these three datasets. Note that the number of unique router IPs in the table excludes IPs in the RFC 1918. Note that the RIPE Atlas probes do not provide any insight on the types or codes of the ICMP messages in their traceroutes. Therefore we assume that all ICMP messages in the RIPE Atlas dataset are ICMP Time Exceeded messages, unless we manually observe otherwise. We find that perspective plays an important role in finding routing loops: For the scans dataset from VP 1, we only observe around 43% of the 41,196 routing loops in the RIPE Atlas dataset.

We manually analyze a small random sample of the persistent routing loops that are found in the RIPE Atlas dataset but not in our Internet-wide scans for the same period of time. We performed manual traceroutes from VP 1 and did not observe routing loops for almost the entire sample, suggesting these are either loops that our scans did not originally identify and yet they resolved after our scanning window, or merely not visible from VP 1. For the few ones in the random sample that looped with manual traceroutes but did not exhibit a looping behavior in our dataset, we find that their neighboring IPs in their /24s do in fact exist in our dataset, suggesting that we potentially missed these loops due to ICMP rate limiting or packet loss.

4.5 How many unique routing loops are there?

We find over 24 million distinct IPv4 destinations with routing loops from VP 1, but many of these loops may be caused by a single misconfigured router or subnet, and could be considered as part of the same loop. For instance, if every IP address in a /24 subnet has a routing loop involving the same routers, it is natural to cluster these IPs together into a single loop, and say this single loop affects 256 addresses, rather than to consider it as 256 unique loops.

In this section, we investigate *clustering* the loops we found from VP 1 by grouping destination IP addresses into subnets, and comparing the routers involved in the loops.

Grouping destination IPs into subnets While ideally a subnet (e.g. a /24) that has a single loop would show loops for all (e.g. 256) addresses, we must account for missed packets in our dataset, due to rate limits, packet drops, or other errors. For instance, we may find that in a fully-looping /24, we only observed 230 IPs

that looped, but would nonetheless want to cluster these IPs into a single /24 (and not 230 distinct /32s, or some other fragmentation of the subnet).

To do this, we heuristically group looping IPs into the largest-containing (CIDR prefix) subnet that has more than a threshold fraction of IP addresses exhibiting a loop. For instance, if we set a threshold of 50%, we would label a /24 as a loop if that subnet contained more than 128 addresses that exhibit looping behavior.

Clustering based on routers involved Another way to group individual loops together is to look at the router IPs involved in the loops. For instance if two IPs have loops involving the same pattern of routers, they may be caused by the same misconfiguration and clustered together. A naive approach to clustering would be to look for exact matches of routers, but this could easily miss clusters due to router aliasing or load balancing. While Paris-style traceroutes help keep a route stable to an individual IP address, it cannot help when comparing traceroutes to two distinct destinations. For instance, one IP address may see a loop between router A and router B, while a second IP address in the same subnet sees a loop between router A and router C due to load balancing, despite this being caused by the same misconfiguration.

To account for this, we cluster two loops together if their looping traceroute for TTLs 246–255 share a single router. Note that this can cause loops to cluster transitively: loop A-B would cluster with loop A-C, which could cluster with loop C-D. This can help handle cases where one traceroute in a subnet did not receive a response from one router due to packet loss or rate limiting. Because we only look at routers that responded in high TTLs (i.e. likely involved in the loop), we avoid clustering all traceroutes together due to routers that simply appear in many traceroutes from our vantage point (e.g. our own immediate upstream routers, or core routers in well-connected transit ASes)

Using our subnet grouping and clustering criteria, we group our 24 million looping IPs into **270,450** clusters, which contain a total of 1.8 million subnets (at a 50% threshold). Figure 4 shows the distribution of number of subnets and hops in these clusters. Most of the clusters we find have only a single subnet in them, and involve only 1 or 2 router hops in the loop.

For instance, one of the largest clusters we find comprises 65425 destination IPs, all in 159.193.0.0/16 (so we observe over 99.8% of this subnet looping). These loops involve just two hops (78.77.181.70 and 78.77.181.71).

Figure 3 shows the distribution of largest subnets we can group into, based on two thresholds of the fraction of IPs in a subnet that must be a loop to consider the subnet a loop as a whole ($> 50\%$ and $> 75\%$). It also shows the distribution of subnets when we also cluster based on routers involved. The similarity of results for each of these methods suggests that it is a robust way to coalesce loops into clusters.

We find subnet groupings of nearly every size smaller than /16, suggesting that loops are not always a given size (e.g. /24). We also find many loops that are not near or similar to others, affecting only a *single IP address* (/32). If we

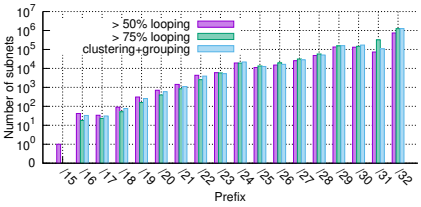


Fig. 3: **Coalesced subnets** — We examine the largest encompassing subnets of the looping IP addresses we find from VP 1. We group a set of looping IPs into their subnet if larger than a threshold fraction of the subnet's IPs exhibit looping behavior (we present 50% and 75% thresholds here). We also cluster based on the routers involved, and find similar results. We find a range of subnets that loop, from entire /16s to individual IP addresses (/32).

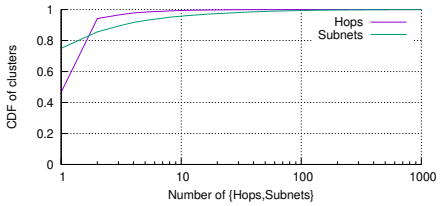


Fig. 4: **Cluster sizes** — Using clustering of loops by routers (hops) involved, and grouping by subnets, we discover 270K clusters. The majority of these clusters comprise only a single subnet, and have 1 or 2 hops involved. This suggests our clustering is accurate, and can help identify specific network misconfigurations that result in large subnets with looping IPs.

were to traceroute only at the subnet level, we would likely miss these loops, as discuss in Section 5.5.

4.6 Are we under counting loops?

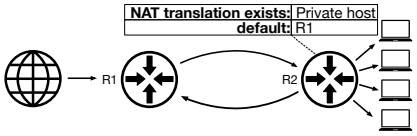
We analyze the scans results from VP 1 and find that we receive at least one ICMP Time Exceeded message for 29.7 M and 30 M unique destination IPs probed for the first and second scans, respectively. For the first scan, we label 3.4M reconstructed traceroutes as non-qualifying routing loops (based on our routing loop definition in §2.1), of which 2.5M have a single non-repeating router IP and 3.9K have 10 unique and non-repeating router IPs. For the second scan, we label 3.7M destination IPs based on their reconstructed traceroutes as non-qualifying routing loops, of which 2.6M have a single non-repeating router IP and 19.6K have 10 unique and non-repeating router IPs. These small numbers (3.9K and 19.6K) of routing loops with a size of more than 10 router IPs suggest that scanning the Internet for more than 10 TTL hops has diminishing returns on finding more routing loops.

5 The Structure of Routing Loops

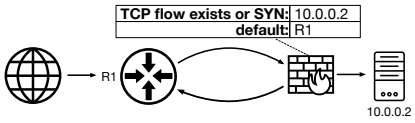
In this section, we investigate the nature of the routing loops themselves by asking: where do we observe them, which addresses are affected, what are the root causes, and what impact might they have?

5.1 What causes these loops?

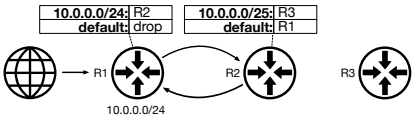
We adopt the general classification proposed by Xia et al. [42] and further expand on it. We classify the routing loops we find from VP 1 into two main categories based on the involvement of the destination AS in the loop. We argue that this classification helps attribute the root-cause of these loops, because once packets reach the destination AS, they are subject to the routing policies and (mis)configurations of the destination AS. We characterize the involvement of an AS in a routing loop as follows: Given a traceroute for a destination IP for hops 246–255 that experiences a routing loop, if there exists at least one router IP address that responds with an ICMP Time Exceeded message, then the AS of that IP is involved in the loop.



(a) We find some routing loops are caused by misconfigured NAT devices, which send traffic back upstream if there is no NAT translation for the incoming packet.



(b) We find some routing loops are caused by middleboxes that only allow network traffic for active TCP flows, and forward any inactive flows back upstream.



(c) The common misconfiguration of network announcemnts, causing packets destined for certain subnets to be errouneously forwarded back upstream.

Fig. 5: Examples of three types of routing loops in our dataset. We discovered the root cause for some routing loops by directly working with network operators of affected networks.

Loops involving the destination AS We observe over 79% (19.7M) of the routing loops we find involve the destination AS, indicating that these loops occur at the edge of the network and closer to their destination IPs. We reached out to several network operators for networks in which we discovered loops to ask about their root cause. Between these discussions, and additional follow-up experiments, we find several types of misconfigurations at the root of the looping behavior.

First, we find a common misconfiguration at the destination AS in which ingress network traffic destined to non-allocated subnets (where the looping IP is in) is not null-routed, but rather forwarded back to an upstream provider or another router at the destination AS, causing the looping behavior. This common misconfiguration has been studied [24,42] in the past and has been shown to cause routing loops on the Internet. Figure 5 (c) shows an illustration of this kind of loops.

Traceroute to
212.106.173.213 (AS15744):

hop	hop ip
246	212.106.159.26 (AS15744)
247	212.106.159.25 (AS15744)
248	212.106.159.26
249	212.106.159.25
250	212.106.159.26
251	212.106.159.25
252	212.106.159.26
253	212.106.159.25
254	212.106.159.26
255	212.106.159.25

(a) – Within dst AS

Traceroute to
144.208.106.43 (AS394994):

hop	hop ip
246	38.110.25.62 (AS17077)
247	38.88.11.58 (AS174)
248	38.110.25.62
249	38.88.11.58
250	38.110.25.62
251	38.88.11.58
252	38.110.25.62
253	38.88.11.58
254	38.110.25.62
255	38.88.11.58

(c) – No dst AS. Involves provider AS

Traceroute to
200.205.174.82 (AS10429):

hop	hop ip
246	81.173.106.94 (AS12956)
247	*
248	*
249	152.255.191.213 (AS26599)
250	187.100.39.54 (AS27699)
251	189.44.183.251 (AS10429)
252	72.246.185.2 (AS12222)
253	187.120.7.53 (AS22381)
254	*
255	81.173.106.94 (AS12956)

(e) Involves 6 Ases

Traceroute to
180.178.163.59 (AS141361):

hop	hop ip
246	221.120.208.150 (AS9557)
247	*
248	221.120.208.150
249	*
250	221.120.208.150
251	*
252	221.120.208.150
253	*
254	*
255	*

(g) No relationships can be found

Traceroute to
45.162.174.92 (AS268527):

hop	hop ip
246	45.162.172.106 (AS268527)
247	187.103.120.213 (AS14840)
248	187.103.120.178 (AS14840)
249	45.162.172.106
250	187.103.120.213
251	187.103.120.178
252	45.162.172.106
253	187.103.120.213
254	187.103.120.178
255	45.162.172.106

(b) – Between dst AS and provider AS

Traceroute to
45.224.118.229 (AS264836):

hop	hop ip
246	84.16.11.125 (AS12956)
247	*
248	84.16.11.125
249	*
250	84.16.11.125
251	*
252	84.16.11.125
253	*
254	84.16.11.125
255	*

(d) – Dst AS in customer-cone of AS12956

Traceroute to
156.240.40.21:

hop	hop ip
246	156.240.40.21
247	*
248	156.240.40.21
249	*
250	*
251	156.240.40.21
252	156.240.40.21
253	156.240.40.21
254	*
255	*

(f) Involves the end-point IP

Traceroute to
198.188.178.137 (AS2152):

hop	hop ip
246	209.147.58.38 (AS2920)
247	*
248	209.147.58.38
249	*
250	209.147.58.38
251	209.147.58.37
252	209.147.58.38
253	*
254	209.147.58.38
255	*

(h) Loops at the customer AS

Fig. 6: **Traceroute examples** — We show a sample of traceroutes for the persistent routing loops we find from VP 1.

Second, we find misconfigured NAT devices in which ingress network traffic that has no translation on the NAT device is forwarded based on a configured rule instead of being dropped. The traffic, after being forwarded back upstream, eventually encounters the same NAT policy that forwards it again, creating the looping behavior. We discovered this root cause by speaking to network operators of a university network in which we discovered over 1,000 routing loops. From a network scanning perspective, there are no distinct characteristics that distinguish these loops from loops caused by the previous misconfiguration, even though they have different underlying causes. Figure 5 (a) shows an illustration of this kind of loops.

Third, transport-state dependant. We find that misconfigured middleboxes that filter traffic for TCP applications can also cause looping behavior. These middleboxes, instead of dropping outstanding TCP packets that are not part of any active TCP flow or do not initiate a new one, forward the packets based on a configured rule. These forwarded packets are then sent back towards their destination ultimately creating the looping behavior. A distinguishing feature of these routing loops is that they are dependent on the *transport state*, and that some destinations that exhibit a looping behavior for TCP `ACK` packets are

reachable or do not exhibit any looping behavior for TCP `SYN` packets. To find the transport-state dependent loops, we run a full Internet scan using Yarrp with TCP `SYN` packets. We then calculated the set difference between the persistent loops in our original `ACK` scan and this `SYN` scan. We find over 6M looping IP addresses that did not exhibit routing loops with `SYN` packets. These types of routing loops can only be discovered by `ACK` scans (or any outstanding TCP packet that is not a `SYN`). Figure 5 (b) shows an illustration of this kind of loops.

Fourth, a misconfigured destination in which packets reach the destination IP but are not delivered due to some unclear misconfiguration at the destination. We observe 27K destinations in our dataset that exhibit this misconfiguration. Figure 6 (f) shows an example of this kind of loops.

Loops not involving the destination AS Over 21% (5M) of the persistent routing loops we discover from VP 1 do not involve the destination AS. Using CAIDA's dataset of AS classification [6], we classify the relationship between the AS(es) in the loop and the AS of the destination. We do not investigate the root-cause of these loops and leave this to future work.

No apparent relationship For 5.81% (291K), we cannot find any relationships between the destination AS and the ASes involved in the loop based on the traceroutes in our dataset. Figure 6 (g) shows an example of this kind of loops.

Customer relationship We find 79.3% (3.97M) have at least one router IP belonging to an AS to which the AS of the destination is a customer. For 4.86% (244K), the destination AS is in the customer cone of one of the ASes involved. Figure 6 (d) shows an example of this kind of loops. After analyzing the relationships in that loop, we conjecture that our odd TTL probes expire at an AS (AS267699) to which the destination AS is a customer. Whereas our even TTL probes expire at an AS (AS12956) to which AS267699 (the provider AS to the destination AS) is a customer.

Provider relationship For 4.75% (238K), we find that the destination AS is a provider to one of the ASes involved in the loop. Figure 6 (h) shows an example of this kind of loops.

Peering relationship For 1.51% (76K) of loops, we find that the destination AS is in a peering relationship with at least one AS in the loop. And for 0.07% (3842), we find one of the ASes in the loop to be in the customer cone of the destination AS. For instance, the loop to 83.234.188.233 (AS20485) involves the router IP 109.196.208.113 (AS50439) which is in the customer cone of the destination AS.

No AS announced Finally, for around 3.66% (183K), we find that the router IPs in the loop are not announced by any AS, therefore we cannot infer any relationships

We note that it is possible that the destination AS of a given looping IP is, in fact, involved in its routing loop, but the destination AS's routers limit their

ICMP Time Exceeded responses, therefore obscuring its involvement based on the traceroutes in our dataset. It is also possible that the probes to the looping destination IPs reach their destination ASes, but are then forwarded at the layer two level (e.g., via managed switches) to an upstream provider before reaching a border router at the destination AS. In such a case, we cannot infer the involvement of the destination AS. Figure 6 (c) shows a potential example of this kind of loop, where probes never reach a router at the destination AS and rather loop between the AS's provider and another on-path AS.

5.2 Where are these routing loops?

Topologically Using the April 17, 2022 RouteViews dataset [29], we analyze the Autonomous System Number (ASN) of each of the destination IPs as well as the routers involved in the persistent routing loops we discovered from VP 1. Although other router ownership inference approaches [23] provide a better accuracy, we used the IP-to-ASN mapping approach for simplicity. We exclude router IPs that are not announced by any AS (which amount to only 0.74% of router IPs in our dataset).

Figure 7 shows the distribution of how many ASes are involved in each routing loop. While over 91% of loops contain routers within a single AS, we find many loops that include multiple ASes. For instance, we find 3 routing loops with as many as 6 distinct ASes involved in the loop. Figure 6 (e) shows one such example, involving routers from Akamai (in Miami, FL, USA), multiple ASes of Telefonica Brazil (in Sao Paulo, BR), and intermediate routers from GlobeNet.

We find that over 25% of all routing loops have at least one of the involved routers in a different AS than the destination of our probe. Meanwhile, 0.11% of routing loops *involve the destination IP itself* in the routing loop, as shown in Figure 6 (f). We performed follow up traceroutes on a small sample of these addresses (e.g. 46.8.120.192 and 90.83.38.250) and confirmed that the round trip time for each probing packet increases with higher TTL values for these peculiar routing loop cases, which suggests a looping behavior.

Geographically Using the April 2022 MaxMind [25] dataset, we analyze the geolocation of the destination IPs in the persistent routing loops we found from VP 1. We find 19% are in the US, followed by 6% in each of India, Brazil and Japan, followed by 5% in China.

We also find that 5% of the loops have a destination IP address in a different country than at least one of the routers in the loop. However, we note that this may simply be due to geolocation error: prior work has shown that geolocating router IPs can be complex and leads to inaccuracies [11]. We leave correcting this problem to future work, such as potentially using router hostnames to infer location [20].

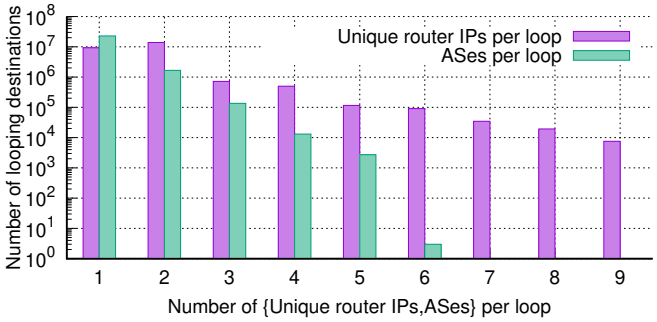


Fig. 7: **Routing loops sizes and the number of ASes they span** — For each persistent routing loop, we count the number of unique router IPs, and the number of unique Autonomous System Numbers by resolving the routers' IP addresses to Autonomous System Numbers using recent datasets.

5.3 How large are routing loops?

Routing loops can range in size from one repeated router to containing many routers in a repeating loop. However, we measure the size of the routing loops in our dataset based on the number of unique router IPs, meaning that a routing loop could have multiple router IPs in it, yet belonging to a lesser number of physical routers. We do not perform router dealiasing [36] techniques on our dataset.

The majority (57%) of routing loops we find contain two unique router IP addresses, a result corroborated by prior work [42]. We also find larger loops—up to 9 unique IP addresses—in a small percentage of cases (0.03%). Figure 7 shows the distribution of loop sizes observed.

5.4 How many loops do /24 subnets experience?

We find that routing-loop containing /24 subnets are limited to a small number of looping IPs in them. For each /24 subnet, we counted the number of target IP addresses that contained a routing loop in the corresponding traceroute. We expected that if a /24 subnet contained a routing loop to one IP address, it would also contain routing loops to many other IP addresses in the same subnet. Figure 8 shows the distribution of looping IP addresses per routing-loop containing /24 subnet, ranging from 1-256 loops. Over half of the /24 subnets that contained a routing loop had fewer than 25 looping IP addresses (out of 256). While it is possible that some subnets only responded to a subset of our probes, our scanning rate should have had each /24 receive a packet from us every 2 minutes on average, well under the ICMP rate limits for most routers (see Figure 2).

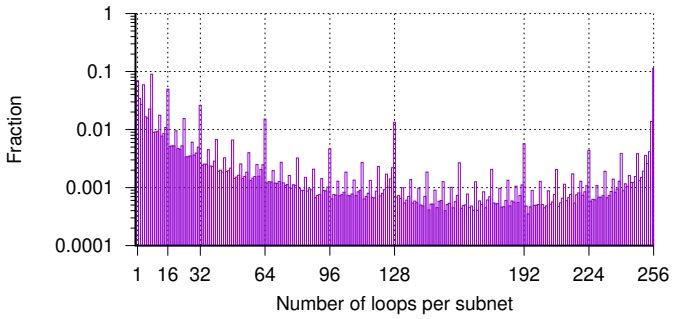


Fig. 8: **Loops per /24 subnet** — For each /24 subnet, we count the number of destination IP addresses whose path showed a routing loop. 35% of subnets had at most 10 IP addresses that had a loop along their path, and only about 19% had over 200 IP addresses. This demonstrates that subnet sampling is likely to miss routing loops.

5.5 Which addresses have routing loops?

A natural hypothesis is that scanning well-known addresses within a /24 subnet, such as the `.1` address (e.g. 10.0.0.1), is as likely to discover routing loops as scanning other addresses. Indeed, many other studies rely on probing or scanning the `.1` address of each /24, and assuming that routing topologies are largely consistent at the /24 level [2,42,16,21,37].

However, we find that this is not the case in our analysis of routing loops. Figure 9 shows the distribution of routing loops broken down by the last octet of the destination IP address. The `.1` addresses reveal the *least* number of routing loops, 21% less than the maximum last octet (`.255`). This suggests that the `.1` last octet does not act as a good *sample* for finding routing loops. This result is corroborated by prior work [10] where the authors showed that the `.1` last octet is the most *responsive* last octet in an Internet census, suggesting that it is less likely to find routing loops at such addresses.

We find a general trend that higher octets are more likely to contain routing loops than lower octets, with noticeable exceptional dips after (sums of) powers of two (e.g. 64, 128, 192). The `.0` octet also has, unexpectedly, a high number of routing loops.

We do not know for certain why this trend exists, but hypothesize that it may be due to the common pattern of IP allocation [10]. IP addresses are often allocated from the bottom of a subnet incrementally, and it is possible that router misconfigurations that default-route unallocated addresses may lead to more loops on higher last-octet addresses compared to the bottom of a subnet. In addition, the last address (e.g. `.255` in a /24) may correspond to the broadcast address, either receiving special treatment with routers or getting sent to more devices that might retransmit the packet in a loop.

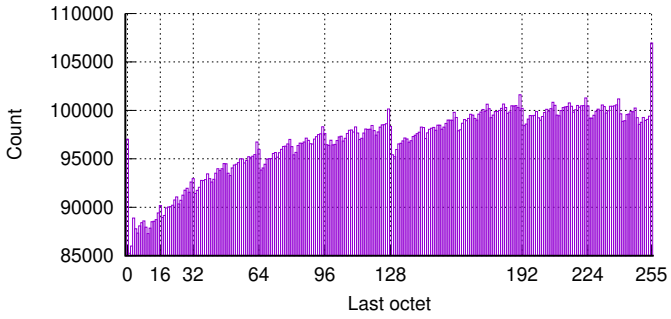


Fig. 9: **Last octet distribution** — We count the number of looping IP addresses for each last octet (e.g. 1.2.3.x). The .1 address contains the least number of routing loops (85,254), while the `.255` contains the most (106,950), likely due to being a common broadcast address. (Note the y-axis does not start at 0)

How many IPs per subnet are needed to find looping subnets? A natural follow-up question is if there is any sample (short of scanning all 256 addresses) of a /24 that can be probed to find most loops?

Our dataset identifies over 320k unique /24 subnets that contain routing loops. Sending probes to only the `.255` last octet in all /24 networks would have identified just over 33% of these subnets-containing loops. Adding additional last octets would help find more, but as Figure 10 shows, it would take over 45 IPs probed per /24 to discover over 90% of the routing-loop containing /24 subnets that we find when scanning all IPs. We believe this result justifies scanning all addresses, and not sampling a handful of IPs per subnet.

5.6 Do these loops matter?

Prior work has already shown that routing loops can have an impact on network attacks. In 2021, researchers discovered that middleboxes could be weaponized to launch reflected amplification attacks, and that routing loops could be abused to make the attack more damaging [3]. An attacker launches the attack by spoofing their source IP address to that of their victim and sending a packet sequence that contains a request for some forbidden resource. When the middlebox responds to the seemingly forbidden request (such as by sending a block page), it will send this response to the victim, effectively amplifying the attacker's traffic. The authors found that if a vulnerable middlebox is within a *routing loop*, the middlebox can be re-triggered each time the packets circle the routing loop, significantly improving the amplification factor for the attacker (up to *infinite* amplification for infinite routing loops). In this regard alone, we believe that trying to identify routing loops is an important goal. Note that [3] examined the top 1 million amplifying hosts (by number of packets sent) from their `SYN`; `PSH+ACK` scan to identify which ones have routing loops along their path. We, however, approach this differently.

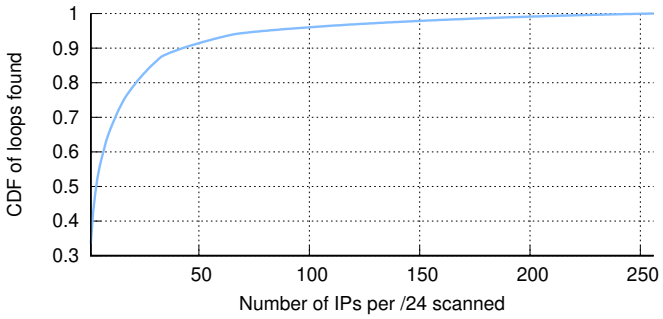


Fig. 10: **Finding Loops by sampling** — We measured how many unique /24 subnets that contain loops can be found as a function of the number of last-octets needed to be scanned. For instance, by only scanning the `.255`, `.127`, `.0`, and `.191` last octets (the top 4 in our dataset), one could identify over 51% of the /24 subnets that contained routing loops (compared to scanning all last octets). However, it would take scanning over 45/256 IPs per /24 to discover over 90% of unique /24s containing loops.

After we scanned the entire IPv4 Internet from VP 1 using `ACK` packets and found over 24 M destinations experiencing persistent routing loops, we now ask the question of how many of these routing loops can be weaponized in a TCP reflected amplification attack. We discussed in §5.1 that some routing loops are transport-state dependant and that scanning the IPv4 Internet with `ACK` packets reveals additional 6 M routing loops that cannot be discovered with a `SYN` scan.

We performed two *forbidden_scans*[1] on the 24 M looping destinations using two different packet sequences, namely a single `PSH+ACK`, and a `SYN` followed by `PSH+ACK` (`SYN`; `PSH+ACK`).

We first performed the `PSH+ACK` single packet scan. We found 273,201 looping destinations to be true amplifiers with an average amplification rate of 386.73, a median of 2.46 and a maximum of 1,414,267. We note that routing loops with an amplification rate of ≥ 100 are concentrated at 6601 destinations from 119 different ASes in 13 different countries.

Second, we performed the `SYN`; `PSH+ACK` scan. We found 1,089,457 looping destinations to be true amplifiers with an average amplification rate of 4.44, a median of 1.02 and a maximum of 1204.37.

This shows that finding and exploiting the transport-state dependant loops renders TCP reflected amplification attacks more effective.

In addition to this, we explore an avenue of the potential impact of routing loops that, to our knowledge, has not been studied before: are there any (named) services behind the destinations that experience routing loops? To answer this question, we reached out to a few AS operators for some of the looping IPs in our dataset to draw their attention to these loops and obtain ground-truth infor-

[1] https://github.com/breakerspace/weaponizing-censors

mation about the root-causes of them. We received a response from a university network engineering team detailing the cause of the loops in their network. We elaborated on the root-cause of these loops in §5.1. Based on the nature of these NAT-related loops, the looping IPs in this case do not have services that otherwise could have been reachable, since the IPs are used to translate internal addresses to external, public ones. We do not know how prevalent this kind of loops is, as they do not have distinguishing characteristics. For our correspondence with other AS operators, we unfortunately have not received any response.

In an attempt to find whether any domain name resolves to a looping IP in our dataset, we used ZDNS[1] to resolve the top 5 M domain names in the tranco list [32]. We found 1256 domain names that resolve to 719 looping IPs in the dataset from our `ACK` scan from VP 1. Then using ZMap, we ran a `SYN` scan on these 719 IPs and found that 400 of them respond with `SYN;ACK` indicating that they do not exhibit any looping behavior when scanned with `SYN` packets. We also manually browsed a sample of the domains behind these 400 IPs and were able to reach and view their web content as if no loops existed along their paths. After further examination, we found that these 400 IPs experience the transport-state dependant kind of loops that we discussed in §5.1. Since the web content for the domains behind the 400 transport-state dependant looping IPs is reachable and browse-able, the impact of their loops can be dismissed; however, this kind of loop consumes their network resources unnecessarily and can be used against these domains to disrupt their service.

6 Related Work

Our work extends prior work primarily by probing many more destination IP addresses, uncovering many more routing loops than previously found, uncovering new types of routing loops that to our knowledge have not previously been known, and discovering that IP addresses within the same /24 can experience different routing loop behavior. We compare to prior work in terms of how we find routing loops, and what our findings reveal.

Identifying routing loops Our overall approach to detecting routing loops—looking for repeated entries in a traceroute—is well-established in prior literature. To name a few: Paxson [30] ran periodic traceroutes between 27 sites and looked for IP addresses repeated at least three times to infer the presence of a routing loop. Paxson further differentiated routing loops as being *persistent* (they never reached the destination during the traceroute) and *temporary* (they did resolve during the traceroute, and ultimately reached the destination). Xia et al. [42] also look for repeats in traceroutes to identify persistent routing loops, and Lone et al. [19] similarly use them to infer the absence of ingress filtering.

Other studies have investigated how routing loops can be predicted by changes in BGP [46,39], how the presence of loops can indicate route leaks [18], and the dynamics of transient loops [31]. However, all of these works rely on small-scale

[1] https://github.com/zmap/zdns

traceroutes of samples of the Internet, and use these to infer trends on the larger Internet.

In addition to these active techniques, routing loops have also been detected passively. Hengartner et al. [13] used packet traces from a tier-1 ISP to detect routing loops. By comparison, this method does not have as broad a view of routing loops as ours (they find only 4318 routing loops in total), but is able to detect transient routing loops—they find that most routing loops last less than 10 seconds.

Performing massive scans Where we primarily differ from the above is the sheer scale at which we actively probe for routing loops. Prior work has used dozens [30], hundreds of thousands [43], and as many as 11M [42] destination IP addresses to discover routing loops. More recently, FlashRoute [15] presented a tool capable of tracerouting one IP per /24 subnet in a matter of minutes. Using this technique, they also discover over 16K prefixes that contain routing loops. Rüth et al. found 439K prefixes containing routing loops in 2019 using follow-up traceroutes to ZMap scans, performing 27M traceroutes [35].

In contrast to prior work, our study scans the *entire* public[1] IPv4 address space: over 3.7 billion IP addresses in total, and we find over 24 million IPs that contain a routing loop, comprising over 320K /24 subnets.

Prior work also assumed that sampling one or two IP addresses within each routable /24 would provide representative samples [42,2,21,16,37]. Our work challenges this assumption by showing that routing loops are not uniform within /24 boundaries—rather, to obtain a global view of routing loops, far more comprehensive scans are necessary.

Prior findings about routing loops Paxson [30] observed that persistent routing loops existed as early as 1996. To estimate the scale of persistent routing loops, Xia et al. [42] issued traceroutes to two IP addresses (`.1` and a random one) in each of about 5.5M /24s. From this, they identified 207,891 /24s with at least one temporary routing loop; of those, they repeatedly probed the two IP addresses and found that 135,973 of the /24s had loops for each traceroute. Xia et al. assumed that if the two candidate addresses experienced routing loops, then *all* addresses in the /24 would, as well, resulting in their estimate of 135,973 * $2^8 \approx$ 35M total IP addresses with routing loops. Our work draws this assumption into question by empirically demonstrating that different IP addresses in the same /24 can exhibit different looping behavior.

Representative scanning Heidemann et al. [10] scanned the entire IPv4 Internet and proposed a method for generating responsive, complete and stable *hitlist*, a list of representative, alive IPv4 addresses that should suffice to represent their respective /24s during an Internet scan. We differ in this matter–we scan the Internet not to discover live hosts, but rather routing loops. We also show that the /24 granularity is not sufficient to discover all routing loops on the Internet.

[1] We adopt ZMap's blacklist, which excludes reserved addresses, private addresses, the loopback prefix, and the addresses reported to us to opt-out from being scanned.

Preventing loops There is also work on mitigating routing loops and the harm caused by them, using reconfigurable networks [38], static analysis of the data plane [22], and real-time detection of transient loops in network traffic [17,14]. These works primarily focus on fixing or preventing loops in the first place, rather than measuring them comprehensively across the Internet as we do.

ICMP rate-limiting studies To parameterize our tool, we evaluated the maximum rate at which we could probe routers without hitting a rate limit (§2). We are not the first to do such a study. Ravaioli et al. [33] sent TTL-limited ICMP echo requests from 180 PlanetLab hosts at a rate of 1–4,000*pps*, with TTLs ranging from one to five. They reported that 60% of the routers exhibited rate limits. By comparison, we explored larger send rates (up to 50,000*pps*) and larger TTL values (1–15), but from only a single vantage point. Guo and Heidemann [12] performed a different experiment, measuring the rate-limiting of pings (not ICMP Time Exceeded responses), and observed only six out of approximately 40,000 subnets rate-limiting them on the forward path. In contrast, our study focuses on ICMP Time Exceeded messages.

Broad implications of loops Persistent routing loops can be used to perform DoS attacks against the involved networks directly, by exploiting the simple fact that a single packet will get relayed multiple times [41]. But more recent work has shown how routing loops can be used to enable or exacerbate attacks indirectly as well. For instance, Bock et al. [3] detail how middleboxes can be used for DoS amplification attacks, and how routing loops around vulnerable middleboxes can worsen these attacks. Nosyk et al. [28] detail how routing loops can exacerbate DNS-based DoS attacks, finding 115 routing loops that enable high-amplification attacks. Attackers can also create or induce their own routing loops in order to amplify DoS attacks, by misconfiguring content delivery networks [8], leveraging IPv6 tunnels [27], or ARP spoofing on a wireless network [4]. Finally, Marder et al. [24] detail how loops complicate inferring outbound addresses in traceroutes, which is useful for inferring router ownership or organizations.

BGP AS path looping behavior (BAPL) Loops can also be observed at the BGP level, by looking for loops in the AS paths of BGP update messages [44]. These so-called BGP AS path looping (BAPL) behavior can result in multi-AS routing loops [45].

7 Ethics

We designed our experiments to have a minimal impact on other hosts. Our IPv4 Internet-wide scans were designed such that each intermediate router would receive a packet from us at a rate of 3 packets per second, which should have a negligible impact on end hosts. To avoid overwhelming destination networks, our experiments probed addresses randomly, which spreads out traffic to any given

destination network across the length of the scan. Our scanning rate should have had each /24 receive a packet from us every 2 minutes on average.

We follow the best practices for high speed scanning laid out by [9]. Our scanning machines we used for these experiments hosted a simple webpage on port 80 to explain the nature of our scans and provides a contact email address to request exclusion from future scans.

For our highest throughput experiments, we took additional precautions. We planned our experiment to saturate the ICMP rate limit of specific routers, we limited the experiment to a relatively small number of IP addresses (10,000). For each IP address, the experiment was limited to 1 second long and only 22mbps.

8 Conclusion

Routing loops have long been known to exist, but their prevalence and nature have long been shrouded behind common but untested assumptions, like that all destination addresses within a /24 are likely to experience the same routing loops. In this paper, we perform a straightforward but illuminating experiment: we look for routing loops to *all* IPv4 addresses by tracerouting to a limited range of TTLs to examine the true global prevalence of routing loops. We discover over 24 million IPs with routing loops—over 21× more than three concurrent datasets combined—comprising over 500K routers. And we discuss their structure and root causes. Also, we uncover new types of routing loops that can be abused for TCP reflected amplification attacks.

Our resulting datasets confirm some prior results, but also expose unidentified biases in prior measurement efforts that may inform future studies. In particular, we find that scanning only the `.1` address per /24 misses 73.5% of the routing loops we were able to find. Indeed, for 35% of the /24s in our dataset, fewer than 11 of their 256 destination addresses result in loops.

Ultimately, our results motivate full-Internet traceroutes. To assist in future efforts, we made our code and data publicly available at `https://github.com/RoutingLoops`.

Acknowledgments

We thank Jack Wampler for the insightful discussions. We also thank the anonymous reviewers for their helpful feedback. This work was supported in part by NSF award CNS-1943240 and NSF award CNS-1954063.

References

1. Augustin, B., Cuvellier, X., Orgogozo, B., Viger, F., Friedman, T., Latapy, M., Magnien, C., Teixeira, R.: Avoiding traceroute anomalies with Paris traceroute. In: ACM Internet Measurement Conference (IMC) (2006)
2. Beverly, R.: Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In: ACM Internet Measurement Conference (IMC) (2016)

3. Bock, K., Alaraj, A., Fax, Y., Hurley, K., Wustrow, E., Levin, D.: Weaponizing Middleboxes for TCP Reflected Amplification. In: USENIX Security Symposium (2021)
4. Brown, J.D., Willink, T.J.: A New Look at an Old Attack: ARP Spoofing to Create Routing Loops in Ad Hoc Networks. In: Ad Hoc Networks, pp. 47–59 (2018)
5. CAIDA: IPv4 Prefix-Probing Traceroute Dataset. `https://www.caida.org/data/active/ipv4_prefix_probing_dataset.xml` (April 2022)
6. CAIDA: The CAIDA UCSD AS Classification Dataset. https://www.caida.org/catalog/datasets/as-classification (April 2022)
7. CAIDA: The IPv4 Routed /24 Topology Dataset. `https://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml` (April 2022)
8. Chen, J., Jiang, J., Zheng, X., Duan, H., Liang, J., Li, K., Wan, T., Paxson, V.: Forwarding-Loop Attacks in Content Delivery Networks. In: Network and Distributed System Security Symposium (NDSS) (2016)
9. Durumeric, Z., Wustrow, E., Halderman, J.A.: ZMap: Fast Internet-wide Scanning and Its Security Applications. In: USENIX Security Symposium (2013)
10. Fan, X., Heidemann, J.: Selecting Representative IP Addresses for Internet Topology Studies. In: ACM Internet Measurement Conference (IMC) (2010)
11. Gharaibeh, M., Shah, A., Huffaker, B., Zhang, H., Ensafi, R., Papadopoulos, C.: A Look at Router Geolocation in Public and Commercial Databases. In: ACM Internet Measurement Conference (IMC) (2017)
12. Guo, H., Heidemann, J.: Detecting ICMP Rate Limiting in the Internet. In: Passive and Active Network Measurement Workshop (PAM) (2018)
13. Hengartner, U., Moon, S., Mortier, R., Diot, C.: Detection and Analysis of Routing Loops in Packet Traces. In: ACM Internet Measurement Workshop (IMW) (2002)
14. Holterbach, T., Molero, E.C., Apostolaki, M., Dainotti, A., Vissicchio, S., Vanbever, L.: Blink: Fast Connectivity Recovery Entirely in the Data Plane. In: Symposium on Networked Systems Design and Implementation (NSDI) (2019)
15. Huang, Y., Rabinovich, M., Al-Dalky, R.: FlashRoute: Efficient Traceroute on a Massive Scale. In: ACM Internet Measurement Conference (IMC) (2020)
16. Katz-Bassett, E., Madhyastha, H.V., John, J.P., Wetherall, D., homas Anderson: Studying Black Holes in the Internet with Hubble. In: Symposium on Networked Systems Design and Implementation (NSDI). USENIX Association, San Francisco, CA (Apr 2008), `https://www.usenix.org/conference/nsdi-08/studying-black-holes-internet-hubble`
17. Kučera, J., Basat, R.B., Kuka, M., Antichi, G., Yu, M., Mitzenmacher, M.: Detecting Routing Loops in the Data Plane. In: ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT) (2020)
18. Li, S., Duan, H., Wang, Z., Li, X.: Route leaks identification by detecting routing loops. In: International Conference on Security and Privacy in Communication Systems. pp. 313–329. Springer (2015)
19. Lone, Q., Luckie, M., Korczyński, M., van Eeten, M.: Using Loops Observed in Traceroute to Infer the Ability to Spoof. In: Passive and Active Network Measurement Workshop (PAM) (2017)
20. Luckie, M., Huffaker, B., Marder, A., Bischof, Z., Fletcher, M., Claffy, K.: Learning to Extract Geographic Information from Internet Router Hostnames. In: ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT) (2021)
21. Madhyastha, H., Isdal, T., Piatek, M., Dixon, C., Anderson, T., Krishnamurthy, A., Venkataramani, A.: iPlane: An Information Plane for Distributed Services. In: Symposium on Operating Systems Design and Implementation (OSDI) (2006)

22. Mai, H., Khurshid, A., Agarwal, R., Caesar, M., Godfrey, P.B., King, S.T.: Debugging the Data Plane with Anteater. ACM SIGCOMM (2011)
23. Marder, A., Luckie, M., Dhamdhere, A., Huffaker, B., claffy, k., Smith, J.M.: Pushing the boundaries with bdrmapit: Mapping router ownership at internet scale. In: ACM Internet Measurement Conference (IMC) (2018)
24. Marder, A., Luckie, M., Huffaker, B., Claffy, K.: Vrfinder: Finding Outbound Addresses in Traceroute. Proceedings of the ACM on Measurement and Analysis of Computing Systems **4**(2) (2020)
25. MaxMind: GeoLite2. `https://dev.maxmind.com/geoip/geoip2/geolite2` (October 2021)
26. Nakibly, G., Arov, M.: Routing Loop Attacks using IPv6 Tunnels. In: USENIX Workshop on Offensive Technologies (WOOT) (2009)
27. Nakibly, G., Arov, M.: Routing Loop Attacks using IPv6 Tunnels. In: USENIX Workshop on Offensive Technologies (WOOT) (2009)
28. Nosyk, Y., Korczyński, M., Duda, A.: Routing Loops as Mega Amplifiers for DNS-Based DDoS Attacks. In: Passive and Active Network Measurement Workshop (PAM) (2022)
29. of Oregon, U.: Route Views Archive Project (October 2021), `http://archive.routeviews.org/bgpdata`
30. Paxson, V.: End-to-End Routing Behavior in the Internet. In: ACM SIGCOMM (1996)
31. Pei, D., Zhao, X., Massey, D., Zhang, L.: A Study of BGP Path Vector Route Looping Behavior. In: IEEE International Conference on Distributed Computing Systems (ICDCS) (2004)
32. Pochat, V.L., Goethem, T.V., Tajalizadehkhoob, S., Korczyński, M., Joosen, W.: Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In: Network and Distributed System Security Symposium (NDSS) (2019)
33. Ravaioli, R., Urvoy-Keller, G., Barakat, C.: Characterizing ICMP Rate Limitation on Routers. In: IEEE International Conference on Communications (ICC) (2015)
34. RIPE NCC Staff: RIPE Atlas: A global internet measurement network. Internet Protocol Journal **18**(3) (2015)
35. Rüth, J., Zimmermann, T., Hohlfeld, O.: Hidden Treasures – Recycling Large-Scale Internet Measurements to Study the Internet's Control Plane. In: Passive and Active Network Measurement Workshop (PAM) (2019)
36. Sherry, J., Katz-Bassett, E., Pimenova, M., Madhyastha, H.V., Anderson, T., Krishnamurthy, A.: Resolving ip aliases with prespecified timestamps. In: ACM Internet Measurement Conference (IMC) (2010)
37. Sherwood, R., Bender, A., Spring, N.: Discarte: A Disjunctive Internet Cartographer. In: ACM SIGCOMM (2008)
38. Shukla, A., Foerster, K.T.: Shortcutting Fast Failover Routes in the Data Plane. In: Symposium on Architectures for Networking and Communications Systems (ANCS) (2021)
39. Sridharan, A., Moon, S.B., Diot, C.: On the correlation between route dynamics and routing loops. In: ACM Internet Measurement Conference (IMC) (2003)
40. Wang, F., Mao, Z.M., Wang, J., Gao, L., Bush, R.: A Measurement Study on the Impact of Routing Events on End-to-End Internet Path Performance. In: ACM SIGCOMM (2006)
41. Xia, J., Gao, L., Fei, T.: Flooding Attacks by Exploiting Persistent Forwarding Loops. In: ACM Internet Measurement Conference (IMC) (2005)
42. Xia, J., Gao, L., Fei, T.: A measurement study of persistent forwarding loops on the Internet. Computer Networks **51**(17), 4780–4796 (2007)

43. Zhang, M., Zhang, C., Pai, V., Peterson, L., Wang, R.: PlanetSeer: Internet Path Failure Monitoring and Characterization in Wide-Area Services. In: Symposium on Operating Systems Design and Implementation (OSDI) (2004)
44. Zhang, S., Liu, Y., Pei, D.: A Measurement Study on BGP AS Path Looping (BAPL) Behavior. In: International Conference on Computer Communication and Networks (ICCCN) (2014)
45. Zhang, S., Liu, Y., Pei, D., Liu, B.: Measuring BGP AS path looping (BAPL) and Private AS Number Leaking (PANL). Tsinghua Science and Technology **23**(1), 22–34 (2018)
46. Zhang, Y., Mao, Z.M., Wang, J.: A Framework for Measuring and Predicting the Impact of Routing Changes. In: IEEE Conference on Computer Communications (INFOCOM) (2007)