TTL Jumps: Unexpected TTL Rewrites Impacting Inferences from Traceroutes

Sebastian Kappes
sebastian.kappes@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Anja Feldmann
anja@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Tobias Fiebig
tobias@internet.wien
TU Wien
Vienna, Austria

Johannes Zirngibl
johannes.zirngibl@mpi-inf.mpg.de
Max Planck Institute for Informatics
Saarbrücken, Germany

Abstract

Traceroute is an important Internet measurement tool used to infer the Internet's topology and to identify middleboxes. It relies on network devices decrementing the Time to Live (TTL) in IP packet headers by one at each hop and sending Internet Control Message Protocol (ICMP) Time Exceeded error messages if the TTL reaches zero. However, we show that *TTL jumps* exist on the Internet: some devices *rewrite* the TTL, often to larger value of up to 255. These rewrites hide the remaining path from traceroute and can lead to incorrect inferences like spurious router and Autonomous System (AS) links. Based on controlled experiments and public data from RIPE Atlas and CAIDA Ark, we show that at least 47 ASes are impacted by path-impairing devices that rewrite the TTL. A prominent example is AT&T where more than 90 % of outgoing IPv6 paths from CAIDA Ark nodes are affected since 2023.

ACM Reference Format:
Sebastian Kappes, Anja Feldmann, Tobias Fiebig, and Johannes Zirngibl. 2026. TTL Jumps: Unexpected TTL Rewrites Impacting Inferences from Traceroutes. In *Proceedings of the 2026 ACM Internet Measurement Conference (IMC '26), October 12–16, 2026, Karlsruhe, Germany*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3777912.3809149

1 Introduction

The Internet is commonly referred to as *"a network of networks."* Exactly *how* these different networks are interconnected is often opaque. Hence, researchers regularly work on identifying the Internet's physical [5, 20], logical, [15], and overlay [14, 39] topology.

The de facto standard for this type of research are *traceroute measurements*, which allow researchers to trace a one-directional path between a source and a destination IP address. Traceroute relies on routers decreasing the Time to Live (TTL)[1] by one at each hop and sending Internet Control Message Protocol (ICMP) Time Exceeded error messages when the TTL reaches zero. Traceroutes are not only useful for network operations and topology-related measurements: they are also used to identify and study middle boxes [13, 19, 46] or to continuously check network connectivity of experimental setups [16]. To perform such large-scale traceroute measurements, researchers often rely on the distributed measurement platforms CAIDA Ark [8] and RIPE Atlas [40], which allow them to set up and run both one-time and regular traceroute measurements, while also publicly publishing all collected data.

However, TTLs do not necessarily decrease monotonically along the path. Indeed, some devices can *rewrite* them on-path. The most well-known examples of TTL rewrites are due to *tunneling*: in 1994, RFC1702—Generic Router Encapsulation over IPv4 [18]—already discussed that tunneled packets need to experience *some* form of TTL decrement upon decapsulation. The exact form of TTL decrement is usually a configurable option. Common options include decrementing by one and decrementing by the number of hops between the tunnel endpoints.

Traceroute presumes that the devices along the path generally *decrement* the TTL. If a rewriting device, however, *increases* the TTL or sets the TTL to an *absolute* value (instead of performing a relative change), this implicit assumption is no longer valid. For example, if an on-path Device A increases the TTL by X, the next X routers after A will be invisible, even if they themselves reduce the TTL and send error messages as expected. Furthermore, if A sets the TTL to value greater than the remaining number of hops to the destination, the packet will reach the destination without the TTL expiring. To show a real-world example of this happening, Table 1 summarizes three traceroutes conducted from a CAIDA Ark node within AT&T (AS7018). The first three hops—all within AT&T—are identical between the three traces. For each traceroute, the fourth hop is the target. Based on the traceroutes, it looks as though hop 3 and the traceroute's target are directly adjacent. However, all three targets are in different, distant countries[2] in Autonomous Systems (ASes) that, according to routing information, have no direct connectivity with AT&T. As we will show later in the paper, this is because a TTL rewrite happens after hop three, which obscures the remaining path to the destination.

In a recent paper, Fiebig and Feldmann [16] claim that TTL rewrites are rather common, affecting close to $\frac{1}{8}$ of all IPv6 traceroutes. However, as traceroute measurements are not the main focus

[1] To improve readability, we use the term TTL to refer to both the IPv4 TTL field and its IPv6 counterpart, the "Hop Limit" field.

[2] We use IPInfo to check the location.

This work is licensed under a Creative Commons Attribution 4.0 International License.
IMC '26, Karlsruhe, Germany
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2327-8/2026/10
https://doi.org/10.1145/3777912.3809149

Table 1: Three traceroutes from CAIDA Ark node igx2-us on 01.10.2025 with the same first three hops. The three targets—located in different ASes and continents—already appear at the fourth hop.

Hop	Address	Org	Location
1	2600:1700:6b0:f200:da3a:ddff:fe48:395f	AT&T	US
2	2001:506:6000:116:76:229:56:171	AT&T	US
3	2001:506:6000:116:76:229:56:130	AT&T	US
↪ 4	2804:3c78:f000::1	Util Tec.	BR
↪ 4	2a03:e40:46::1	Skroutz	GR
↪ 4	2a03:bf01::1	Vidanet	HU

of their work, they did not conduct an in-depth investigation, Nevertheless, their findings already imply that TTL rewrites, at least for IPv6, challenge the reliability of a core instrument in network measurement research.

Research Questions: In this paper, we address this challenge by asking: *"Do TTL rewrites on the Internet impact the reliability of traceroute measurements?"*. Specifically, we tackle the following research questions:

- **RQ1:** How can we identify TTL rewrites in traceroutes?
- **RQ2:** How prevalent are TTL rewrites on the Internet per IP address family?
- **RQ3:** Which ASes are likely performing these TTL rewrites?
- **RQ4:** Is the prevalence of TTL rewrites a new phenomenon, and if so, since when has it gained prevalence?

To investigate this issue, we (1) perform dedicated RIPE Atlas [40] traces towards NLNOG RING Nodes and (2) analyze historical datasets from CAIDA Ark [8] and RIPE Atlas. The former allows us to use a set of hosts in a wide range of ASes across the Internet and to also capture arriving packets. The latter gives us a historical perspective on TTL rewrites.

Contributions: We make the following contributions:

- We are the first to study the issue of TTL rewrites and their impact on traceroute measurements using historical (CAIDA Ark and RIPE Atlas) as well as dedicated measurements (using RIPE Atlas and NLNOG RING [32]).
- We find that TTL rewrites are common in both IPv4 and IPv6, where 47 ASes seem to rewrite TTLs to most commonly 255 (43 ASes), and sometimes 64 (4 ASes).
- Our data shows that major ASes, including AT&T (AS7018) and Orange (AS5511), perform TTL rewrites, which significantly limits their value as vantage points for traceroute measurements. More than 95 % of IPv6 topology measurements from a CAIDA Ark node within AT&T are impacted by TTL rewrites.
- We show that TTL rewrites are visible in public data sources since at least 2018. However, we can show that for some ASes like AT&T the issue of rewrites is comparatively new. We find a notable uptake of TTL rewriting since 2023, and also discuss possible root causes of this issue.

Structure: The remainder of the paper is structured as follows. In Section 2, we describe the problem in detail and provide preliminary background knowledge. In Section 3 we introduce the data sources used in our study. Section 4 describes our method to infer TTL rewrites in traceroutes. We then evaluate a set of active traceroute scans as ground truth in Section 5 and study rewrites in existing topology data in Section 6. We perform a case study of a Tier-1 provider, namely AT&T, in Section 7. Section 8 covers related work, before we discuss our findings in Section 9 and conclude in Section 10.

2 Background

In this section, we provide the necessary background knowledge.

2.1 Time-to-Live (TTL)

The IPv4 TTL [36] and the IPv6 "Hop Limit" [11] are 8 bit fields with values up to 255. Their goal is to prevent packets from traveling 'forever' if they enter a routing loop. Each node forwarding a packet *should* decrement this field. The node where the TTL field reaches zero (i.e., where the TTL *expires*) should drop the packet and notify the sender by sending an ICMP Time Exceeded [37] or corresponding ICMPv6 [10] message. Thus, a packet should only be able to travel along up to 254 hops before being discarded.

2.2 Traceroute

Traceroute leverages the TTL mechanism to discover the forward path to a target. It sends probe packets with increasing TTLs, starting at one. In response to the packets, each node on the path should receive a packet where it decreases the TTL to zero and hence has to respond with an ICMP/ICMPv6 message. Thus, a common—but not necessarily valid—assumption is that nodes responding to successive TTLs are adjacent on the Internet. This is the basis of most Internet topology studies.

If a probe triggers an error message at the target (e.g., Port Unreachable) or the last reachable hop (e.g., Host Unreachable or Network Unreachable), the target or hop usually responds with an ICMP error message that *quotes* the original probe packet, including its TTL when it reached the target [10, 37].

Equal Cost Multi Path (ECMP) Routing and Paris Traceroute: One well-known complication for traceroute is that traffic can use multiple different paths between a source and a target. One mechanism for this is ECMP routing [21, 45]. With ECMP, a router spreads traffic across multiple paths with the same Interior Gateway Protocol (IGP) cost, for example based on a hash value calculated over specific header fields. However, as traceroute relies on independent probes that may yield different hash values, it is unable to infer if two packets with sequential TTLs traveled different paths and may therefore falsely infer that nodes from different paths are adjacent.

Several traceroute derivatives have been proposed to identify (a) consistent paths for ECMP [2], (b) all paths for ECMP [3, 49], (c) service-specific paths [31], or (d) reverse paths [23, 48]. While these variants no longer rely on independent probes, they still use the traditional traceroute mechanism: sending probes with increasing TTLs to trigger ICMP Time Exceeded messages.

2.3 Challenges for Traceroute

Among the well-known challenges for using traceroute data are non-responding nodes, ICMP rate-limiting, nodes responding using unexpected IP addresses, as well as tunnels.

Non-responding nodes and rate limiting: If there is a specific TTL for which traceroute does not receive a response from a node,

it will timeout (commonly denoted by an asterisk "*"); anyone using traceroute data has to handle this.

Unexpected IP addresses: Typically, routers have multiple interfaces with different IPv4 and IPv6 addresses. Most Internet backbone routers also have an additional 'loopback' address, i.e., the address of the node itself. When sending ICMP/ICMPv6 error messages, most routers use either the IP address of the interface where original packet causing the error was received, or the IP address of the interface through which the sender is reachable [4] as the packets' source address. In case a router has no address of the required IP address family on either interface, e.g., when using IPv4-with-IPv6 nexthop [26], hosts also use the loopback address. If this is unavailable, they fall back to the IPv4 dummy address, 192.0.0.8, as the last resort [12].

Tunneling technologies: Tunneling technologies like Multiprotocol Label Switching (MPLS) or Generic Router Encapsulation (GRE) are also a challenge. Initially GRE [17, 18] and IPIP tunnels [44] were common; later new requirements made MPLS [42] (often integrated as L2/L3 VPN services [1, 30]) and SRv6 [25] more prevalent.

Depending on the configuration, the TTL of the encapsulated packet and the outer tunnel can be *independent* or *synchronized*. If the TTL is independent, the tunnel is not visible in the traceroute—the traceroute continues after the tunnel with the decapsulated probe. If the TTL is synchronized, the TTL is decremented by the decapsulating router; thus, the length of the tunnel is visible in the traceroute. If nodes in the tunnel also support ICMP error messages, they are visible as well.

MPLS uses label switching via Label Switching Routers (LSRs). It adds one or multiple labels, so-called Label Stack Entries (LSEs), with enough information for intra-AS forwarding. LSRs are special routers that forward packets based on LSEs between an ingress and an egress router. Each LSE has its own TTL field, which the LSRs decrement. As with GRE [17, 18], the IP-TTL of the payload packet can be copied to the LSE-TTL field on ingress (and back from the LSE to the IP packet on egress), which would preserve the overall path length information. However, operators can also use static LSE TTL values, e.g., 255, which makes the LSRs invisible in the path.[3] In that case, the outer TTL should *not* be copied into the IP packet upon egress.

Implications: The above challenges complicate many of traceroutes' use cases like correctly mapping hops to ASes, inferring links/topology, and identifying routers. Unexpected IP addresses lead to the problem of 'de-aliasing' addresses in traceroutes, which has been extensively studied: for example, existing proposals suggest to use deprecated options [27], path length [28], or AS relationships [29] to achieve this. Tunnels make it hard to evaluate path lengths, traversed hops, and adjacencies. Therefore, their detection is an extensively studied field to improve topology inferences [14, 22, 39, 47].

3 Datasets

In this section, we introduce the existing datasets we use in the course of our analysis.

[3]Note that RFC4950 [7] provides an extension to ICMP that allows LSRs to add MPLS information to ICMP errors. However, this needs to be supported by the corresponding LSRs.

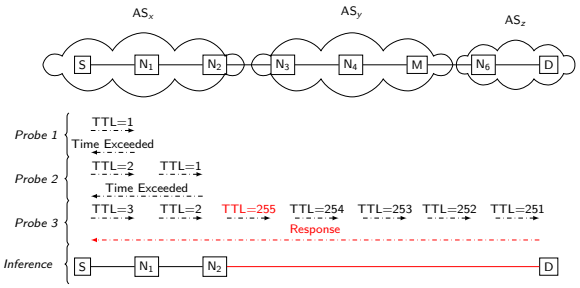


Figure 1: Example traceroute from a source S to a destination D, normally traversing six nodes in three ASes. However, N_2 rewrites the TTL to 255. Thus, probe 3 reaches the target, hiding four nodes and one AS.

3.1 Traceroute Datasets

We use existing traceroute datasets from CAIDA Ark [8] and RIPE Atlas [40], which are often also used for topology inference. Both platforms operate distributed nodes that offer various measurement capabilities.

CAIDA Ark performs daily topology measurements that are publicly available and include historic data. In our analysis, we use two different CAIDA Ark measurements for IPv4 and IPv6. For IPv4, we use the measurement which targets a random address within each routed /24 from one source node on a daily basis (team-probing). As each target is randomly assigned to a source, the same target might not be scanned by the same source on consecutive days. For IPv6, a subset of sources scan one random address within each routed /48, respectively. Thus, each source probes each prefix on a daily basis, but not necessarily the same address.

RIPE Atlas also conducts a set of continuous measurements. However, the majority of measurements are triggered by users of the platform. Therefore, the sources and targets are more heterogeneous. RIPE Atlas publishes results from all measurements on a daily basis, but only offers an archive for the last 30 days. However, we received access to a third party mirror with data going back to February 2018.

3.2 IP-to-AS Mapping

To map IP addresses from the traceroutes to ASes, we use Border Gateway Protocol (BGP) data from RouteViews [33]. Furthermore, we use Internet Routing Registry (IRR) data from Regional Internet Registries (RIRs) to identify owners of prefixes not announced and visible in BGP. We do not use existing tools, e.g., bdrmapIT [29], to identify aliases of routers or AS mappings as these tools rely on traceroute data, whose reliability we evaluate. We leave the evaluation of the impact of TTL rewrites on them to future work.

4 Identifying TTL Rewrites

TTL rewriting—setting the TTL to a constant value or increasing the TTL—impacts traceroutes by preventing the TTLs initially set by traceroute from expiring further down the path. In this section, we first discuss the potential impact of TTL rewrites and then how one can identify them.

4.1 Impact of TTL Rewriting on Traceroute

We start with an example of a traceroute along a path where a rewrite occurs, see Figure 1. Assume that node N_2 rewrites the TTL of the traceroute packets to 255. Even if N_3 decreases the TTL, the resulting value is only 254—meaning the TTL does not expire. N_3, as well as the remaining nodes along the path, therefore will not send a TTL expired message and instead continue forwarding the packet until it reaches the destination. Thus, none of these nodes will be visible in the traceroute.

Once the probe reaches the target, there are four possibilities: (a) The target responds to the traceroute probe and, thus, appears as the next hop after N_2. (b) The target or last reachable node responds with an error message (e.g., Port Unreachable, Host Unreachable, or Network Unreachable) that quotes the original probe with the TTL of the probe [10, 37]. In case of a rewrite, the quoted initial packet would hence show an unusually high TTL. (c) The target does not respond, and no error message is sent. (d) A Time Exceeded error message is sent if the packet enters an unexpected routing loop. In all cases, the intermediate nodes are hidden, which can result in incorrectly inferring that nodes N_2 and D are adjacent, which can also further lead to inferring incorrect AS relations (i.e., AS_x and AS_z are directly connected, recall Table 1). Depending on the position of the TTL rewriting node in the path and whether the source and/or target are in the same AS, the problem may not be easily identifiable from traceroute data alone.

4.2 Review of Traceroute Measurements (CAIDA Ark and RIPE Atlas):

We use data from RIPE Atlas and CAIDA Ark based on Paris traceroute. The traceroutes stop probing once

(i) the target is reached,
(ii) an error message indicating that the target is unreachable is received (except for TTL expired errors),
(iii) a routing loop is detected, based on receiving responses for different TTLs from the same IP address,
(iv) 5 consecutive hops are unresponsive, upon which the source may send a final probe with TTL 255 to check the liveness of the target[4], or
(v) a maximum TTL, usually 32, is reached.

The output format of both platforms changed over time. RIPE Atlas, for example, added fields for ICMP error messages and quoted TTL [41]. Since these fields are optional, not all probes support or report them reliably. Nevertheless, since they provide valuable information, we use these values to provide a lower bound of impacted traces.

4.3 Identifying TTL Rewrites

Identifying TTL rewrites from traceroutes is not trivial as traceroute measurements are uni-directional, i.e., they do not include any data from the target's perspective. In this subsection, we discuss indicators helpful to identify TTL rewrites.

***(i)* Packet captures at the target:** One way to identify TTL rewrites is to rely on a combination of traceroute data and packet captures at the target. If the TTL values captured at the destination are larger than any of the TTL values sent, one can conclude that a hop along the path must have rewritten the TTL. In this case, the TTL rewriting was most likely caused by the AS of the last hop in the trace. We leverage this method in Section 5.

***(ii)* Quoted TTL:** If the impacted traceroute triggers an ICMP error message other than TTL expired, the response contains the quoted probe packet and thus also its TTL. The quoted TTL corresponds to the TTL the packet had at the point along the path where it triggered the error. Therefore, if the quoted TTL is larger than the one initially set in the probe, we consider this as an indication of TTL *rewriting*, again likely at the last observed hop seen in the traceroute. Both platforms (CAIDA Ark and RIPE Atlas) record the quoted TTL. We use quoted TTLs to study TTL rewrites in Section 6.

***(iii)* Path length:** While traceroute traces the forward path, the trace also contains information about the reverse path: *responses* to probes also contain a TTL that is decremented on the way back to the source, which can be used to estimate the length of the reverse path. Nodes typically use fixed TTL values when sending IP packets, e.g., 32, 64, 128, or 255. When one receives an answer from the destination, one can hence estimate the reverse path length by taking the next-largest value from this list and subtracting the received TTL value. While forward and reverse paths often differ, researchers have used large differences to identify tunnels [22, 47]. If no tunnel can be identified, the difference might imply TTL rewriting. While the effect is not deterministic enough for a quantitative evaluation, we still leverage it for a case study in Section 7.

***(iv)* Routing loops with one hop:** If the same IP address occurs twice in a traceroute for different TTLs, it is an indication of a routing loop. If a TTL rewrite along the path to the loop sets the TTL to a static value, the same IP address will answer two probes with directly adjacent TTLs. In the traceroute this appears as a one hop loop, leading to the termination of the traceroute by CAIDA Ark and RIPE Atlas (cf. Section 4.2). Thus, single hop loops may indicate TTL rewriting. We see this effect in our case study in Section 7.

***(v)* Topology information:** TTL rewrites can mislead topology inference strategies. Thus, if one knows about the link level Internet topology or can gather topology information by other means than traceroute, one can use this knowledge as ground truth to infer potential TTL rewrites in traces. However, because reliable ground truth on the physical topology is scarce, this is difficult in practice. At the AS level, additional datasets are available that one can use to check AS level topology inferences. If the target and the rewriting node are in different ASes, a direct AS link between the two ASes will appear in the trace, recall Figure 1. Using public BGP data and peering information from PeeringDB [34], it is possible to crosscheck these AS links. For example, the three traceroute examples discussed in Table 1 all contain unexpected AS links. Still, routing data is not complete and traceroute does not always see the "best" (from a topology detection perspective) source IP address of a node (cf. Section 2.3). Thus, direct peering between ASes cannot be excluded.

No information: If the target is unresponsive, no error message is received, and no loop is visible, the traceroute might end at the

[4]While the traceroute used by RIPE Atlas does this, it is optional for CAIDA Ark and turned off by default.

hop that rewrites the TTL. However, since no further information is available to distinguish such cases, it is not possible to infer this.

5 Evaluation: Active Scans

We start out with a set of controlled experiments to check (1) if we can identify TTL rewrites and (2) which ASes are likely performing them. For this we use our first indicator from Section 4.3: we check if the received trace packets have a TTL higher than the one initially set. Unfortunately, CAIDA Ark and RIPE Atlas do not allow users to capture packets. Thus, we cannot use their nodes as targets for these experiments. Rather, we run traces both (1) towards a target under our control and (2) to nodes from NLNOG RING—a measurement platform operated by network operators for network operators.[5] NLNOG RING allows us to capture traceroute packets at different destinations, but we choose to only do this at a limited number of nodes to not negatively impact the platform.

5.1 Data Gathering

We divide our controlled experiment into two phases. First, we conduct traceroute measurements from all active RIPE Atlas probes (11 375 IPv4, 5722 IPv6) to a target under our control, where we capture the incoming probing packets (Section 5.2).

Next, we increase the number of targets to 100 using NLNOG RING nodes. For this measurement, we limit our sources to reduce load on NLNOG RING and to work within the permissible rate limits of RIPE Atlas. We focus on RIPE Atlas probes in ASes that we previously saw performing TTL rewrites or that contained sources affected by TTL rewrites. This gives us a better perspective on rewriting behavior in general, and it should also allow us to better pinpoint on-path ASes performing rewrites. On the target side, we selected NLNOG RING nodes such that we cover all 52 available countries, while also avoiding ASes in which we use RIPE Atlas probes. For two countries, we can only select ASes that are also among the selected source ASes. For each source–target AS pair, we run two traces four days apart. Note that the specific RIPE Atlas probes within a source AS may change during the experiments.

We configure our RIPE Atlas traceroute measurements to stop after no more than 32 hops. Consequently, our RIPE Atlas probes typically send a maximum TTL of 32. However, as explained in Section 4.2, RIPE Atlas occasionally performs liveness checks with a TTL of 255. In these cases, we cannot unambiguously identify TTL rewrites.

5.2 All Probes/Single Target

We start by identifying TTL rewriting in our initial measurements, see Table 2 for an overview. At our target, we were able to capture and identify the probing packets for 86% of 19 826 traceroutes (86% 11 375/13 168 IPv4, 86% 5722/6658 IPv6).[6] Among those, TTL rewrites affected 365 probes (138 IPv4, 236 IPv6) from 63 ASes (39 IPv4, 29 IPv6) *somewhere* on the way to our destination. Rewrites mostly occur in the probe's source AS (180 affected probes, 78 IPv4, 103 IPv6) or in unknown ASes (172 probes, 24 IPv4, 17 IPv6). Here, "unknown AS" means that the last hop either did not send a TTL exceeded message ("⋆" in trace), or responded with an IP address that is neither announced in BGP nor related to an operator according to RIR delegation files, preventing us from attributing it to an AS. Only 18 probes (7 IPv4, 11 IPv6) experience TTL rewrites by an on-path AS.

5.3 Affected Probes/Multiple Targets

Next, we look at the results of our second experiment, see Table 3. We use 532 source ASes from which we received a packet with a TTL≥32 or at which we saw an on-path rewrite in our first experiment.

In total, we find traces from 950 probes (800 IPv4, 538 IPv6) in 471 ASes (446 IPv4, 263 IPv6) to be affected by TTL rewrites. Compared to Table 2, we only find source AS rewrites in 17 ASes (9 IPv4, 9 IPv6). However, performing traces to different vantage points increases the share of probes that are affected by on-path TTL rewrites to 184, with 58 for IPv4 and 135 for IPv6. Also, note the increase in the number of probes affected by TTL rewrites where the rewriting AS is unknown (860, with 739 IPv4 and 408 IPv6). This is mostly related to a target in AS59943. This target alone observed TTL rewrites by one or more unknown ASes for 469 distinct source ASes. According to BGP data, AS59943 has only a single upstream provider; by manually inspecting a sample of the affected traces, we find that the last known hops tend to be in ASes that peer with AS59943's upstream AS. We therefore suspect that this upstream AS is responsible for the rewrites.

Table 4 shows updated statistics that exclude the heavy-hitter destination in AS59943. Rewrites in source ASes do not change. However, on-path and "unknown" rewrites reduce to 25 % and 30 %. Still, traceroutes from 335 probes (159 IPv4, 203 IPv6) in 140 ASes (90 IPv4, 67 IPv6) are impacted.

5.4 Observed Rewriting Behavior

In the following subsection, we analyze the behavior of rewriting ASes based on the combined data from the initial and the multi-target scans, see Table 5. The combined dataset includes 225 078 traces (120 229 IPv4, 104 849 IPv6) between 93 287 source–destination pairs (82 795 IPv4, 71 438 IPv6), with 5966 pairs (3776 IPv4, 2913) experiencing TTL rewriting. The table shows the number of impacted ASes and probes, the observed TTL, the resulting path length, and whether the rewrite is in the source AS or in transit. We note: rewrites occur in many ASes, including Tier-1 ASes (e.g., AT&T, AS7018), but also smaller networks (e.g., Junet, AS59702).

Next, we look at the distribution of TTL values as captured by the traceroute targets, see Figure 2. We separate them into three classes: (1) those where we identify TTL rewrites, (2) likely liveness checks,[7] and (3) those where we observe no rewrites. Over 95 % (for both IPv4 and IPv6) of the traces experiencing TTL rewrites arrive with a TTL above 214. In these cases, the rewriting node most likely changes the TTLs in the trace packets to 255, which is

[5] We did not receive privileged access to the NLNOG Ring. Instead, we contribute to the infrastructure ourselves and have the same access as any other participant. Before starting our measurements, we additionally conferred with the RING admins, who saw no issue in us running these specific measurements.

[6] Some RIPE Atlas probes sent their probing packets from a different source address than the one they record in the trace data. In this case, we cannot map the trace packets arriving at the destination to a specific measurement.

[7] We can infer if Atlas performed a liveness check from our trace data. RIPE Atlas records the TTLs of the trace packets it sent—if it sent a TTL of 255, we know that it performed a liveness check and classify the trace accordingly.

Table 2: *Active Scans:* Observed rewrites for all RIPE Atlas probes to a single target. The unknown AS category includes traces with unresponsive hops ("*") or when the IP address to AS mapping fails.

	Total probes		Rewrite in source AS		Rewrite on path		Rewrite in unknown AS		Total rewrites	
Protocol	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*
IPv4	11375	3898	78 (0.69%)	12 (0.31%)	7 (0.06%)	6 (0.15%)	53 (0.47%)	24 (0.62%)	138 (1.21%)	39 (1.0%)
IPv6	5722	1929	103 (1.8%)	9 (0.47%)	11 (0.19%)	7 (0.36%)	122 (2.13%)	17 (0.88%)	236 (4.12%)	29 (1.5%)
∑	12047	4100	180 (1.49%)	20 (0.49%)	18 (0.15%)	13 (0.32%)	172 (1.43%)	39 (0.95%)	365 (3.03%)	63 (1.54%)

Table 3: *Active Scans:* Observed rewrites for RIPE Atlas probes to NLNOG Ring targets.

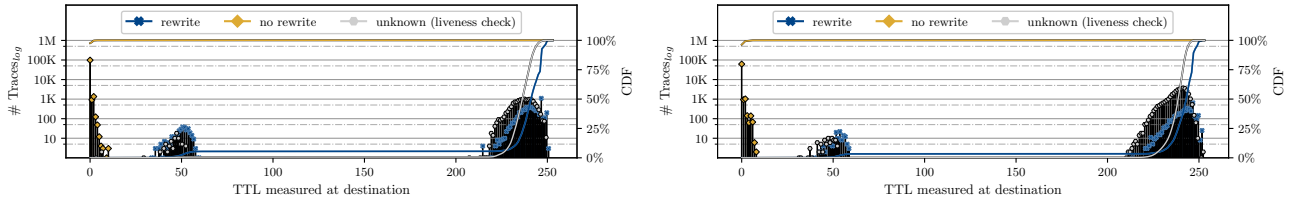
	Total probes		Rewrite in source AS		Rewrite on path		Rewrite in unknown AS		Total rewrites	
Protocol	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*
IPv4	1510	486	39 (2.58%)	9 (1.85%)	58 (3.84%)	42 (8.64%)	739 (48.94%)	435 (89.51%)	800 (52.98%)	446 (91.77%)
IPv6	1392	449	51 (3.66%)	9 (2.0%)	135 (9.7%)	84 (18.71%)	408 (29.31%)	224 (49.89%)	538 (38.65%)	263 (58.57%)
∑	1634	532	87 (5.32%)	17 (3.2%)	184 (11.26%)	116 (21.8%)	860 (52.63%)	462 (86.84%)	950 (58.14%)	471 (88.53%)

Table 4: *Active Scans:* Observed rewrites for RIPE Atlas probes to NLNOG Ring targets excluding the destination in AS59943, as this AS alone observed TTL rewrites by one or more unknown ASes for 469 distinct source ASes.

	Total probes		Rewrite in source AS		Rewrite on path		Rewrite in unknown AS		Total rewrites	
Protocol	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*	*Probes*	*Src. ASes*
IPv4	1510	486	39 (2.58%)	9 (1.85%)	27 (1.79%)	22 (4.53%)	111 (7.35%)	72 (14.81%)	159 (10.53%)	90 (18.52%)
IPv6	1392	449	51 (3.66%)	9 (2.0%)	40 (2.87%)	26 (5.79%)	147 (10.56%)	56 (12.47%)	203 (14.58%)	67 (14.92%)
∑	1634	532	87 (5.32%)	17 (3.2%)	66 (4.04%)	47 (8.83%)	249 (15.24%)	118 (22.18%)	335 (20.5%)	140 (26.32%)

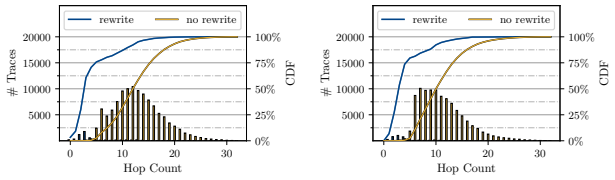
Table 5: Active Scans: TTL rewriting ASes

| | ***IPv4*** | | |



(a) IPv4 (b) IPv6

Figure 2: *Active Scans:* Captured TTLs at the target. Traceroutes with a TTL rewrite are identified based on the TTL of incoming packets at the target. RIPE Atlas traceroutes with TTL 255 are liveness checks and labeled as unknown.



(a) IPv4 (b) IPv6

Figure 3: *Active Scans:* Hop count for each traceroute in our combined active scans. Traceroutes with a TTL rewrite are identified based on the TTL of incoming packets at the targets.

Here, the rewriting ASes are transit ASes and the traces originate from a single source AS, whereby the distance between source and rewrite is comparably large. Thus, the median path length is already large *before* the rewrite.

TTL rewriting can occur in transit ASes en-route to the destination or within the probe's source AS, see column "in transit" in Table 5. Most of the rewriting ASes also change the TTL of packets from at least one outside probe. This shows that TTL jumps are not limited to measurements from within specific networks, but also impact transit traffic.

We further investigate if ASes *always* rewrite the TTL in the trace packets from our measurements. For this, we collect all AS paths from our traces that contain a rewriting AS. If there is at least one path where the rewriting AS is part of the AS path, but is later followed by a different AS (other than the destination AS), we conclude that the AS did not rewrite the TTL. We mark such rewriting ASes that also let packets pass without interfering with a "×" in Table 5. If we do not find an identifiable AS further down the AS path, but a non-responsive hop (i.e., a "*") or a hop with an unannounced address, we cannot tell if the trace has at that point left the AS without a TTL rewrite ("○" in Table 5). We find that the majority of rewriting ASes also appear in traces without changing the TTL (column "always last?" in Table 5). The ASes for which we do not find an instance of a non-affected trace solely appear in traces from one distinct vantage point. Therefore, traceroutes cannot be pre-classified as impacted if they traverse a rewriting AS. Rather, each path has to be checked individually.

We conducted our two sets traceroutes to the NLNOG Ring nodes four days apart. During this time some prominent ASes changed their behavior. In our initial measurement, AS7018—the rewriting AS with the most affected source probes—rewrote the TTL at a hop with an IP address attributable to the AS. Four days later, during the second NLNOG Ring measurement, the probes in this AS still experienced TTL rewrites, but the rewrites now occurred at a non-responsive hop ("*"). Still, there are no other ASes along the AS path except the source and target AS, hence it is likely that the rewrite still occurs in this AS, albeit at a non-responsive hop. Nevertheless, to stick to our classification we now count the traces from the 20 affected probes as rewrites in "unknown" ASes. For an in-depth investigation of AS7018 (AT&T), see Section 7.

5.5 MPLS Tunnels

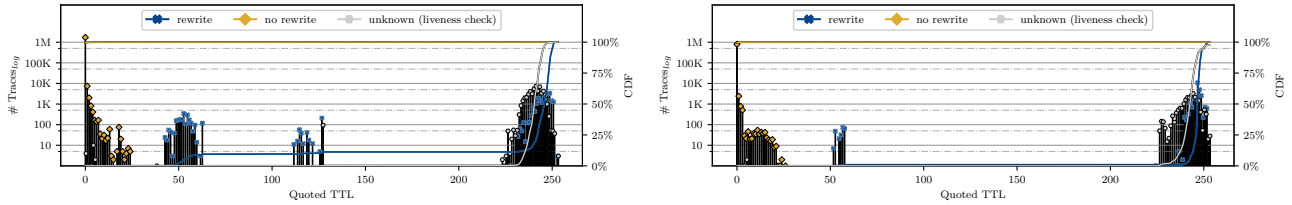
One possible reason for TTL rewrites can be misconfigurations in the context of tunnels (cf. Section 2). Thus, we check for the presence of tunnels, in particular MPLS tunnels. Out of 5.9 k traces with a TTL rewrite, only 906 (507 IPv4, 399 IPv6) contain some MPLS information [7]. Interestingly, the last visible hop before the target, potentially rewriting the TTL, adds MPLS information to the ICMP error message in 107 cases (12 IPv4, 95 IPv6). For 55 traceroutes, the last hop is an IPv6 address from an announced prefix owned by Orange (`2a01:c000::/20`). The remaining cases are in AS1299 (Arelion, 36 traces), AS2914 (NTT, 6 traces), and 10 ASes with 1 to 3 traces. For one trace, the last hop is a bogon address and no AS is identifiable. A potential reason for the TTL rewrite may be incorrect synchronization between the inner and outer TTL as discussed in Section 9.1.

Key Takeaways: *We use active measurements from RIPE Atlas towards controlled targets to find TTL rewriting ASes. Using controlled targets simplifies rewrite detection as we can inspect the TTL received at the target. We show that TTL rewrites can be seen in 47 ASes, including Tier-1 providers and eyeballs. Rewrites mostly set the TTL to 255 and typically occur within a few hops of the source. However, not only source ASes rewrite TTLs but also some ASes on the path.*

6 Re-Evaluation: Historic Data Sets

Next, we check to what extent TTL rewrites are visible in the archived long-term traceroute data from RIPE Atlas. More specifically, we analyze the daily traces on the first day of each week from February 2018 to November 2025.

For these traces we unfortunately do not have packet captures at the target to extract the TTL values of the arriving probes. Hence, we use the "Quoted TTL" indicator from Section 4.3, which relies



(a) IPv4 **(b) IPv6**

Figure 4: *Revisiting Scans—RIPE Atlas:* Quoted TTLs from ICMP error message. Traceroutes with a TTL rewrite are identified based on the quoted TTL. RIPE Atlas traceroutes with TTL 255 are liveness checks and labeled as unknown. See Appendix C, Figure 8 for an explanatory decision chart illustrating our classification.

on traces that result in an ICMP error packet from the target. These error packets should contain the quoted TTL of the trace's probe packet. If the quoted TTL—the TTL at the target—is larger than any of the initial TTLs in the trace packets sent, we conclude that the trace was affected by a rewrite. Note that our results are unlikely to detect all cases of TTL rewrites as we only focus on ICMP error packets. Thus, the results are a conservative lower bound.

6.1 Passive vs. Active Scans

While our active scans focused on identifying rewriting ASes, our analysis of historical RIPE Atlas traces investigates the development of the phenomenon over time. We first explore how the TTL rewrites in daily RIPE Atlas traces compare to the results from our active scans (cf. Section 5). For this, we take a closer look at a recent snapshot, namely November 10th 2025.

The snapshot from RIPE Atlas contains 204.8 M traces of which 2.65 M received ICMP error messages (1.8 M IPv4, 849.3 k IPv6). Of these, 139.6 k include a quoted TTL (86.3 k IPv4, 53.4 k IPv6). For 49.6 k or 35.5% of the traceroutes with quoted TTL, we detect TTL rewrites (23.1 k IPv4, 26.4 k IPv6). Figure 4 shows the quoted TTL received in affected and unaffected traces. These results are similar to those from our active measurements (cf. Section 5.4, Figure 2). One exception is that we now also see 419 rewrites with quoted TTL values between 113 and 128, albeit only for IPv4 traces. Here, we suspect the packets encountered nodes that rewrote the TTL to 128. In 208 traces, the quoted TTL is *exactly* 128—all these traces target destinations in one of two Microsoft ASes (AS8069 and AS8075). AS8069 is used for the "Microsoft Routing Preference product" [35]. Furthermore, we find that out of the 47 rewriting ASes (35 IPv4, 24 IPv6) we identified in our active measurements (cf. Section 5, Table 5), 25 (16 IPv4, 9 IPv6) are also present in the affected traces on this day.

For comparison, the corresponding CAIDA Ark snapshot of the same day contains 150.4 M traces of which 6.9 M received ICMP error messages (1.9 M IPv4, 5.1 M IPv6). All of these include a quoted TTL: in contrast to RIPE Atlas, Scamper—the traceroute implementation used by CAIDA Ark—records the quoted TTL even if it is 1. For 268.4 k or 3.8% of these traceroutes, we detect TTL rewrites (137.0 k IPv4, 131.4 k IPv6). Figure 9 in Appendix C shows the quoted TTL received in affected versus unaffected CAIDA Ark traces. For IPv4, traces with a rewrite to 128 target destinations in Microsoft ASes, similar to the previous results based on RIPE Atlas. For IPv6, 71.6 k (54.4%) of the affected traces originate from a probe located in AT&T. We evaluate this in more detail in Section 7. Among the IPv6 traces where the quoted TTL is larger than 200, there is a small subset where scamper actually sent probing packets with higher TTLs up to 255. Hence, we do not classify these as rewrites.

6.2 TTL Rewrites over Time

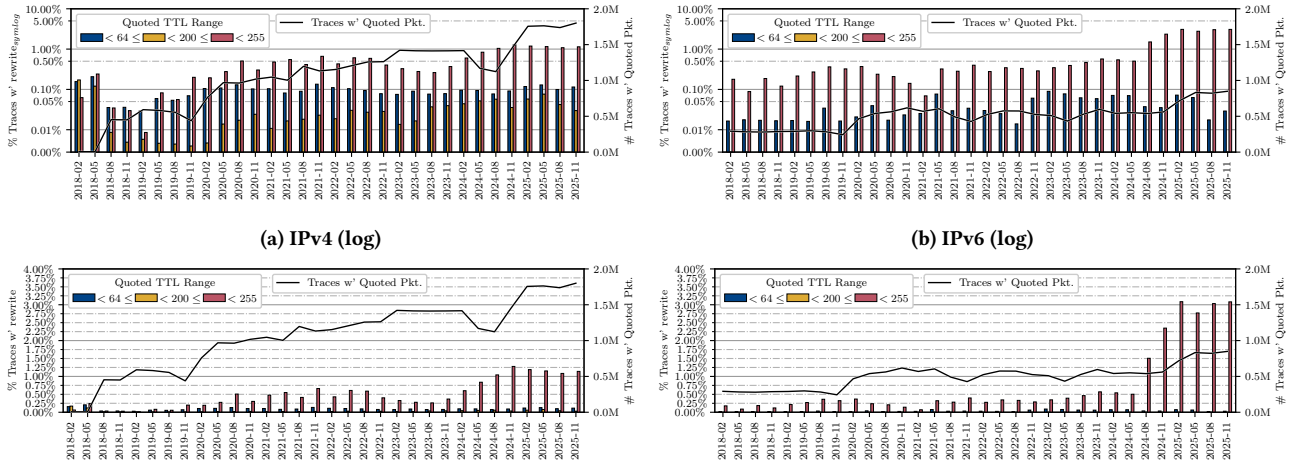
Next, we take advantage of the historic RIPE Atlas traces. Figure 5 shows the percentage of daily scans affected by rewrites according to the "Quoted TTL" from ICMP error packets sent by the target. Our baseline is the number of traces that received an ICMP error messages other than TTL expired from either the target or a hop on the path. Given that the error message and the quoted TTL are optional, this gives us a lower bound on the number of affected traces.

Amazingly, we find indicators of TTL rewriting as far back as 2018. Moreover, over the last 7.5 years, the number of traces receiving ICMP error messages increased from 298k per day in February 2018 to 2.6M in November 2025. Since February 2018, 0.1 %–1.1 % of IPv4 traces and 0.1 %–3.1 % of IPv6 traces with quoted request packets are affected by rewrites, with the share of affected traces increasing by two orders of magnitude. However, especially IPv6 sees a dramatic increase from 2024-08 onwards, where it rises from 0.5 % to over 3.0 %.

Throughout our study, the quoted TTL of most of the affected traces is above 200, followed by traces with quoted TTLs below 64 (0.02 %–0.1 % per day for IPv4, 0.01 %–0.05 % for IPv6). Quoted TTLs in between are mostly absent for IPv6, and rare for IPv4 (<0.01 %–0.05 % per day).

Next, we focus on the 47 ASes that we saw rewriting TTLs in our active scans. Looking at the historical data, we can trace back since when these ASes have been tampering with TTL values. For this we use our daily snapshots dating back to 2018. We again rely on the "Quoted TTL" to identify rewrites and focus on the subset of traceroute measurements that result in an ICMP error message by the target.

Figure 6 is a heatmap that shows the development of TTL rewrites in the ASes from Section 6.2 over time. For each analyzed day, we plot the percentage of all traceroutes traversing the AS that experience a TTL rewrite. Dark gray cells denote days on which no trace traversing the AS triggered a usable ICMP error message.



(a) IPv4 (log) (b) IPv6 (log)

(c) IPv4 (linear) (d) IPv6 (linear)

Figure 5: *Revisiting Scans—RIPE Atlas:* Percentage of affected traces in RIPE Atlas over time.

We find that TTL rewrites do not merely occur short-term. Several of the ASes we identified (AS20712, AS8764, etc.) have been rewriting TTLs for multiple years. Furthermore, we find that TTL rewrites in IPv4 are dominated by AS15557 (SFR SA, a French mobile carrier), which started rewriting TTLs in 2022 and later drives the trend in Figure 5c. For IPv6, the increase seen in Figure 5c is caused by multiple operators starting to apply TTL rewrites: AS57269 (Digi Spain Telecom S.L.) and AS5511 (Orange S.A.) are most dominant here, while AS42346 (NorthC), AS35000 (JSC "Severen-Telecom"), AS9136 (Wobocom), AS8426 (Claranet), and AS7018 (AT&T) also have notable contributions. Interestingly, except for AS42346 (NorthC), all of these operators are access ISPs; furthermore, most of them started rewriting TTLs in 2024.

Beyond that, we also observe several operators who *quit* TTL rewriting again. For example, AS29695 (Lyse Tele AS, from mid 2021 to late 2022) performed IPv6 TTL rewrites in the past, but then seemingly stopped doing so. This could have been caused either by a hardware change, or by rewrites being considered a misconfiguration which was then fixed.

Key Takeaways: *We use the quoted TTL from ICMP error messages as an indicator of TTL rewrites on existing traceroute data to evaluate the impact on historic data. We show that TTL rewrites are visible since 2018 and have increased over time. Our evaluation provides a lower bound on the number of impacted traces, i.e., only if the target responds with an ICMP error message and only if the RIPE Atlas records the error and quoted TTL. Still, our results highlight that TTL rewrites affected existing traceroute measurements and thus need to be considered in future studies.*

7 Case Study: AT&T

The ASes we observed doing TTL rewrites also include networks from well-known, large Tier-1 providers. One example of this is AT&T (AS7018). Given the size and significance of this AS, both RIPE Atlas and CAIDA Ark have a substantial presence within AT&T—6 CAIDA Ark and 162 RIPE Atlas probes[8] are currently connected as of November 2025. Therefore, TTL rewrites by AT&T (AS7018) potentially impact many traceroutes from these platforms.

Some CAIDA Ark nodes in AT&T are part of CAIDA's daily IPv6 prefix probing.[9] During this measurement, each node probes one random IPv6 address per announced /48 prefix. It is therefore an ideal dataset for us to evaluate whether the TTL rewrites within AT&T are limited to a small subset of destinations or widespread. While the IPv6 prefix probing is done every day, the set of involved CAIDA Ark nodes is not stable. However, one of the CAIDA Ark nodes located within AT&T, `igx2-us`, has been frequently involved in this measurement between 2020 and 2025 and therefore allows us to study the evolution of TTL rewrites by AT&T over time.

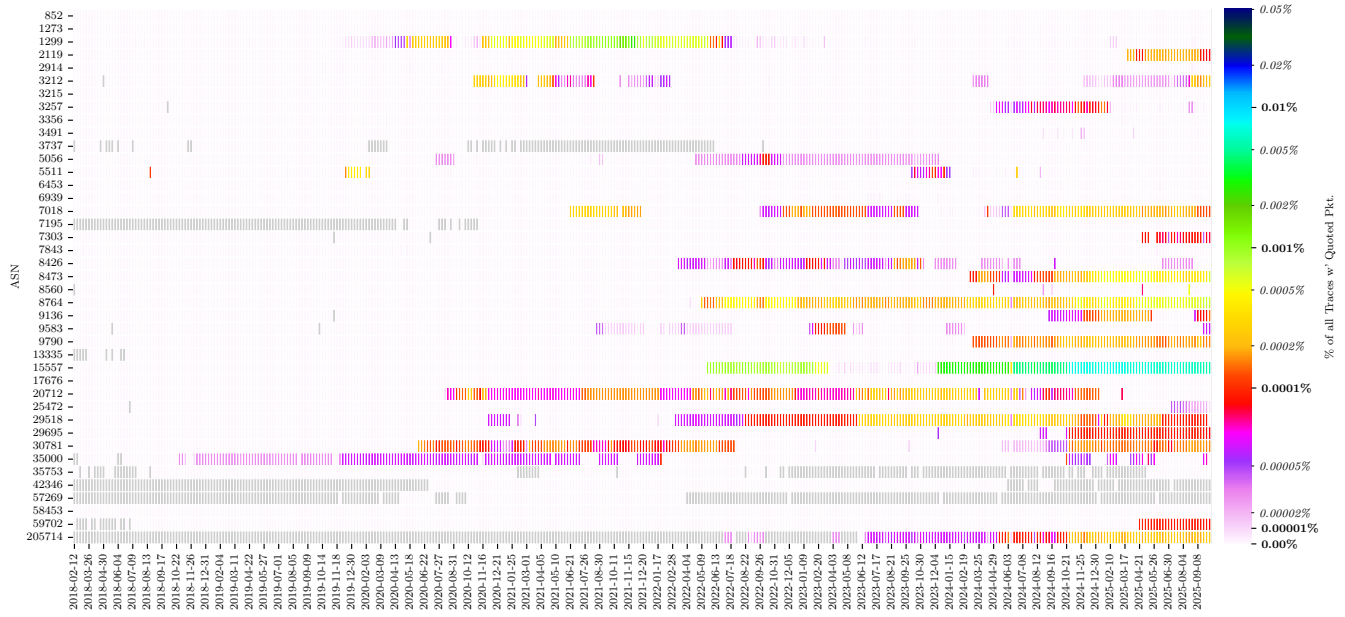
From the selected CAIDA Ark node, we first observe IPv6 TTL rewrites in the second half of 2023. To visualize the historic development of TTL rewrites in AT&T, we take a closer look at three dates: one before the TTL rewrites started (March 2023), one shortly after rewrites started (December 2023), and a recent date (October 2025).

Figure 7a shows the distribution of the hop counts of traces on each of the selected dates. In March 2023, `igx2-us` probes 351.0 k targets. As shown in the figure, the number of hops is close to a normal distribution with a mean of 14.8 hops and a standard deviation of 4.4. This is expected as some trace targets are close while others are farther away. Still, the majority of hosts have similar medium distances.

This behavior changes over the course of the year. In December 2023, `igx2-us` probes 432.6 k targets. The mean distance reduces to 11.2 hops and the standard deviation increases to 5.8. While the distance to half of the targets still forms a normal distribution, three outliers are visible at 4, 6, and 8 hops. A closer look at these peaks reveals that each corresponds to a set of traces with indications of TTL rewrites. For all three peaks, the first three hops are within the

[8] https://atlas.ripe.net/probes/public?sort=-id&country_code__in=All&status=1&toggle=all&page_size=100&search=AS7018&page=1

[9] https://www.caida.org/catalog/datasets/ipv6_allpref_topology_dataset/



(a) IPv4

(b) IPv6

Figure 6: *Revisiting Scans—RIPE Atlas:* Heatmap showing the percentage of daily affected traces in RIPE Atlas that the rewriting ASes from our active measurements account for. We sample the first day of each week. Gray cells denote days on which none of the traces we examine traversed the respective AS.

AT&T network. The peak at four hops comes from targets where the fourth hop is the target address of the trace. This is a clear case for the TTL rewriting indicator "path length", recall Section 4.3.

The peak at six hops consists of traceroutes that terminated because they identified a loop within the AT&T network after three hops. Interestingly, 37.7 % of all identified loops (9.3 k) and 99 % of loops with a length of 6 include the same IP address three times. While loops with a single device *can* happen, one would usually assume that loops involve *multiple* nodes. Thus, there should be other addresses visible between the two hops with the same IP

addresses that constitute the loop. For comparison, in 85.3 % (25.9 k) of the loops on March 2023, there is at least one address between the duplicate addresses that trigger the loop detection, and only 1.9 % of loops (563) contain the same address three times. This can be explained by TTL rewriting: under normal circumstances, the TTL value for each probe will only be increased by one each time traceroute sends a new packet. In case of a TTL rewrite, however, the TTL is set to a fixed value at some point in the network. If the packets with rewritten TTLs then enter a loop, they will each expire at the same hop, hiding other potential hops in the loop. This is therefore a clear case of the TTL rewriting indicator "routing loops with one hop" (cf. Section 4.3).

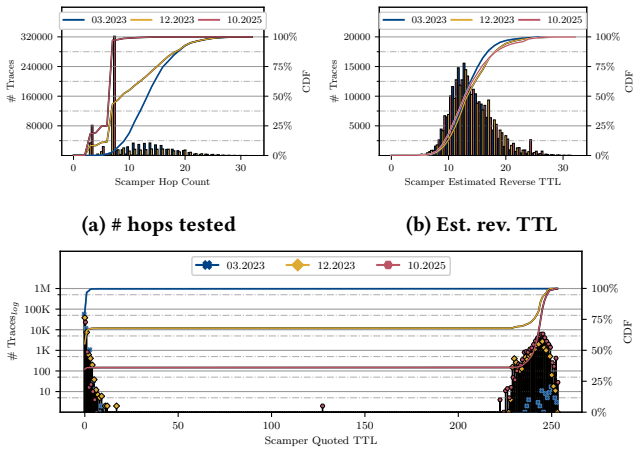
The peak at eight hops is from traceroutes that terminate because they reach the limit of 5 unresponsive hops. While there is no further indication that these traceroutes were impacted by TTL rewrites, they still fit the previous pattern: The only responsive hops are the first three hops, all in the AT&T network. If a TTL rewrite happens at hop three and the target is unresponsive, the probes cannot trigger any further ICMP Time Exceeded messages and will time out.

After 2023, the number of paths impacted by TTL rewrites increases further. In October 2025, the majority of paths experience TTL rewrites. In this snapshot, `igx2-us` probes 465.8 k targets with a mean hop count of only 7.1 (less than half the number of hops compared to March 2023) and a standard deviation of only 1.9. As shown in Figure 7a, there is no longer a normal distribution: the majority of traceroutes are now part of one of the three previously described peaks. By now, 93.4 % of the completed traceroutes (45.3 k) have a length of 4, and 95.9 % (22.2 k) of all identified loops end with three identical hops. The third hop of all completed traceroutes responds with a single IP address. The final fourth hops are 45.3 k distinct IP addresses in 12.3 k distinct ASes. Similar High Degree Nodes (HDNs) have previously been attributed to MPLS tunnels by related work [22, 47]. In this case, however, the HDN is caused by TTL rewrites, and we find no indications for MPLS tunnels.

The increasing impact of TTL rewrites is also visible in the quoted TTL from ICMP error messages, see Figure 7c. In 2023, the majority of quoted TTLs are 1 or below 10. The handful of large TTLs is most likely caused by the liveness test of Scamper after 5 unresponsive probes (cf. Section 4.2). The number of quoted TTLs larger than 200 rises in December 2023 and reaches its peak in 2025, mirroring the decrease in hops in Figure 7a.

Figure 7b shows the development of estimated reverse TTLs (cf. Section 4.3) for the same dates. Compared to Figure 7a, there is a clear difference: The estimated reverse TTL follows the expected normal distribution and does not change over time. This large discrepancy in the distributions can only be explained by TTL rewrites.

Key Takeaways: *TTL rewrites in the AT&T network started in 2023 and increased afterwards. While AT&T also impacts IPv4 traceroutes, the impact on IPv6 is more drastic: as of 2025, rewrites affect most IPv6 paths from our chosen CAIDA Ark node towards one IPv6 address in all announced /48 prefixes. For each target, only the first three hops in AT&T are visible, making the resulting data unusable for inferring network topology. We reached out to AT&T but did not receive a comment.*



(a) # hops tested **(b) Est. rev. TTL**

(c) TTL from quoted packets. Especially before early 2023, most quoted TTLs have the expected value of 1.

Figure 7: *Case Study - AT&T:* CAIDA Ark IPv6 traceroutes from probe igx2-us located in AT&T (AS7018).

8 Related Work

RIPE Atlas data quality: Fiebig and Feldmann [16] motivated this work by pointing out the phenomenon of TTL rewrites in traceroute measurements. They used traceroutes from RIPE Atlas to confirm that specific paths are used for their study. Given their knowledge of the upstream of their vantage point, they noticed that $\frac{1}{8}$ of their IPv6 traceroutes missed ASes. While they concluded that the traceroutes cannot be correct, they did not investigate this issue any further. Holterbach et al. [20] evaluated the quality of RIPE Atlas measurements. However, they focused on the interference of measurements by parallel measurements on the platform. We focus on network effects that may influence the results drawn from measurements and public data.

Long-haul links: Carisimo et al. [9] used data from CAIDA Ark to investigate *long-haul links*: connections between trace hops with an exceptionally high latency. Their goal was to study how submarine cables are reflected in the network-layer Internet topology; for this reason, they specifically examined intercontinental links. In their paper, Carisimo et al. also briefly touch on the topic of MPLS tunnels, which they find are used on a varying percentage of long-haul links depending on the AS. In contrast to our work, they do not investigate TTL rewrites and only focus on intercontinental links in IPv4.

Traceroute measurements: Traceroute is an important tool to identify IP paths and infer the topology of the Internet. Different derivatives have been proposed to identify consistent paths in case of load balancing [2], all paths in case of load balancing [3, 49], service-specific paths [31], or reverse paths [23, 48]. To identify specific features of the paths, these solutions rely on different factors that influence the forwarding process, e.g., IP Identifier field or transport layer ports for load balancing. Nevertheless, they are all based on the traditional traceroute mechanism: sending probes with increasing TTLs to trigger ICMP Time Exceeded messages.

The issues highlighted in this paper therefore also affect all of these tools.

(MPLS) tunnel detection: Different studies evaluated the presence of MPLS tunnels on the Internet and their impact [14, 22, 39, 47]. Vanaubel et al. [47] proposed a method for finding and measuring different types of tunnels. They manage to identify *implicit tunnels*, where the tunnel hops reveal themselves, *opaque tunnels*, where only the egress router reveals itself with an MPLS extension in the error message, and also *invisible tunnels*. They find the latter based on differences in TTL between the forward and reverse path, together with router fingerprinting. This paper was later reproduced by Huddleston et al. [22], who showed that similar numbers of tunnels are still prevalent in 2025. Both works focus on MPLS tunnels in IPv4, and not the potential effects of tunnel egress like TTL rewrites. They reason that routers with uncommonly high direct connections (HDNs) can be attributed to MPLS tunnels hiding actual connectivity. We show that TTL rewrites have a similar effect on topology data—further explaining HDNs.

9 Discussion

Next, we discuss potential reasons for TTL rewrites, their potential impact on existing and future research relying on traceroutes, and the limitations of our methodology.

9.1 Potential Reasons

Despite our best attempts, we are unable to identify a citable source that can attest to the underlying misconfiguration and/or bug that is causing the TTL rewrites. Moreover, our attempts to reproduce the scenario in a lab setup failed. Thus, we reached out to several affected operators to inquire about possible reasons for rewrites.

One operator using devices from a major vendor informed us that the problem in their network was related to a double-labeling of broadband access customers' packets at ingress. At egress, both label stacks are popped, and, owing to the implementation of that vendor, the devices write the inner label's TTL to the innermost IP payload rather than decreasing that packet's TTL by the decrease observed for the outer packet.

A second operator pointed us to an open bug with an open source Software Defined Networking (SDN) focused Network Operating System (NOS) they are using in their platform. This operator noted that the issue occurs on the Broadband Network Gateway (BNG) and that the NOS vendor is aware of it. According to that vendor, the issue is caused by an implementation bug related to how the relevant functionality is mapped in the ASIC producer's SDK for the platform.

In addition to the two anecdotal indications that TTL rewrites can be MPLS related, we received further input and conjectures from several operators. These mostly gravitate around implementation issues with disaggregated routing setups (likely similar to the two anecdotes above).

Some voices also argued that this might be done to hide business relationships: pinning the TTL effectively hides the networks from traceroute and thus obfuscates peering and/or transit relationships. However, given that such a practice also limits an operator's debugging capabilities, we consider this to be unlikely. Furthermore, to effectively hide topology, an operator would have to use TTL pinning on customer ingress as well as on the network border. In our work, we mostly observed TTL rewrites close to access customers, making it unlikely that this is intentional.

9.2 Impact

The TTL rewrites we identified impact various types of research that rely on traceroute data. Unfortunately, quantifying their impact on existing studies is almost impossible (the exact data used and the filter steps applied are not easily reproducible). Nevertheless, the clearly visible impact of TTL rewrites must be considered in future traceroute-based work.

Topology studies: Traceroute is often used to infer network topology, AS relationships, and links that are of special interest, for example submarine cables. Ramanathan and Abdu Jyothi [38] use traceroute data as input for their detection of submarine cables and connected ASes. One factor in their methodology is that two consecutive hops are located on different continents. Traceroutes subject to TTL rewrites often fulfill this property but only due to hidden nodes. Recall that in the examples shown in Table 1, all three targets (the fourth hop) are located on different continents while the third hop is located in the US. Two targets are located in countries with access to the sea, potentially connected to the US via submarine cables. Yet it is unlikely that there is a direct fiber link between Dallas (the last hop in the US) and Greece or Hungary (two of the target locations according to geolocation data by IPInfo).

Middleboxes: The identification of middleboxes and the evaluation of their impact also often relies on traceroute-based tools, for example Tracebox [13] or Yarrpbox [19]. These tools identify middleboxes based on differences in the quoted IP packet in ICMP Time Exceeded error messages before and after a middlebox. Considering the example shown in Figure 1, a middlebox M would be detected by comparing the quoted packet in the ICMP Time Exceeded error messages before, at, and after the middlebox. However, after a TTL rewrite, no further Time Exceeded packets will be visible, and path impairing middleboxes can no longer be identified.

Routing loops: TTL rewrites can also aggravate the problem of routing loops as increasing the TTL results in packets surviving longer in loops. Thus, if traceroute tools (e.g., stateless tools such as Yarrp [6]) do not detect loops early, they might send more probes than necessary that end up in routing loops. Furthermore, Koch et al. [24] recently pointed out that routing loops can result in large amplifications. While one should therefore always use conservative TTLs (between 32–64) in measurement studies, TTL rewrites can cause packets to travel on with a high TTL even if the initial TTL was chosen conservatively.

Overall, TTL rewrites challenge traceroute studies in two ways: they hide portions of the forward path, and they can cause false topology inferences. As a bare minimum, one should therefore always both (1) keep in mind that rewrites exist when making inferences about the network topology based on traceroute data and (2) check if one's traces received any quoted TTLs that suggest a rewrite. For researchers that want to err on the side of caution, the most reliable way to address rewrites is to limit traces to destinations where (1) they can capture arriving trace packets or (2)

the host is known to respond with ICMP error messages (containing the quoted TTL). In both cases, one can analyze the arrival TTL of trace packets to identify and discard traces that were affected by rewrites. Naturally, doing this will drastically limit the number of destinations researchers can use in their measurements. Researchers therefore need to carefully assess to what extent TTL rewrites could skew findings in their measurement studies and filter their traces if necessary.

9.3 Limitations

Vantage points: Our evaluation is mainly based on Internet measurement platforms (i.e., RIPE Atlas and CAIDA Ark). Sermpezis et al. [43] discuss how Internet measurements platforms are biased by the location of their probes. Indeed, our results only show a lower bound of impacted networks and paths. In particular, since many RIPE Atlas measurements are triggered by users, they rapidly change over time and do not consistently probe the same paths. While CAIDA Ark uses a fixed set of measurements, the chosen nodes and targets vary between measurements. This biases our evaluation across time. Still, together they are the best available source of historic traceroutes and lay a solid foundation for our study.

Limitations of traceroute data: Furthermore, the evaluation of available data is impacted by inaccuracies of the data itself. Without traffic captures at traceroute targets, the TTL rewrite can only be deduced using our indicators, which work well for TTL *increases* but less so for *decreases*. Yet, some ASes rewrite the TTL to 64, which can result in decreases if the source uses a larger TTL (e.g., during the liveness check; cf. Section 4). We are aware of these complications and prune inaccuracies as carefully as possible to err on the side of false negatives, which gives us a conservative estimate of the actual impact of TTL rewrites.

Attributing rewrites to ASes: Throughout the paper, we referred to the ASes of the last-observed IP addresses in the affected traces as the "rewriting ASes". However, it is not guaranteed that the rewrites happen in these ASes: some of these hops could—in theory—also be border routers. We can only infer in which AS rewrites most likely happen since the path of the packet after a rewrite is unknown to us. For the same reason, we also cannot leverage more sophisticated solutions for mapping AS boundaries like bdrmapIT [29] (cf. Section 3.2).

Accuracy of identification methods: In Section 4 we presented several methods to identify TTL rewrites. Two of these can decisively show that a packet was indeed subject to a rewrite: (1) capturing packets at the destination and (2) observing a quoted TTL packet through an ICMP error. The other three methods (finding rewrites through path length, routing loops, or topology information) are not as strong: they are neither necessary nor sufficient conditions for the existence of a rewrite. On their own, these indicators cannot be used to conclusively identify rewrites—they are only useful as a supplementary analysis to one of the two more definite methods.

Identifying MPLS tunnels: We mention in Sections 5.5 and 9.1 that some of the TTL rewrites may be because of MPLS tunnels. In Section 9.1, we suggested that TTL rewrites might happen because the egress LSR copies the static LSE-TTL back to the payload packet. In that case, we will not receive any further ICMP Time Exceeded messages once the packet has entered the tunnel. Consequently, we will also not receive any ICMP error from the egress LSR and thus cannot apply the tunnel identification strategies of Vanaubel et al. [47] (cf. Section 8).

9.4 Future Work

In our study, we only searched for TTL rewrites from a limited number of measurement nodes to a small set of destinations (cf. Section 9.3). Our observations therefore only pertain to a limited number of paths. It is likely that there are other networks practicing TTL rewriting that can still be discovered.

Also, we did not investigate *which* paths traversing a rewriting AS are affected by TTL rewrites. Our analysis showed that many rewriting ASes also appear in traces where they do not change the TTL (cf. Section 5.4), and we observed that packets on the return path are also not necessarily subject to rewrites (cf. Section 7). Furthermore, it is possible that different underlying protocols (ICMP/TCP/UDP) also influence rewriting behavior. Figuring out where and under which circumstances an AS tampers with TTL values therefore remains an open question for future research.

As mentioned in Section 9.3, there are two methods how one can decisively show that a packet was subject to a rewrite: (1) capturing packets at the destination and (2) observing a quoted TTL packet through an ICMP error. These two options, however, are not always available: it is hard to identify rewrites in situations where one cannot capture packets or receive ICMP errors. Future work could hence explore new ways to reliably recognize rewrites under such conditions, which would allow for a more complete view of TTL rewrites on the Internet.

10 Conclusion

Traceroutes are the de facto standard for operators and researchers to explore the Internet's physical, logical, and overlay topology. Traceroute relies on nodes decrementing the TTL along the path and sending ICMP Time Exceeded error messages. In this paper, we propose five indicators to detect TTL rewrites (RQ1). Using them, we show that TTL jumps—i.e., TTL rewrites to static, often larger values—occur frequently on the Internet. We show (RQ2, RQ3) that TTL rewrites are visible in at least 47 ASes, including Tier-1 providers, and in 49.6 K existing RIPE Atlas traceroutes. They impact traceroutes from vantage points within these networks, but also in transit for both IPv4 and IPv6. Rewrites have occurred since at least 2018 (RQ4) and are increasingly visible in public topology data from CAIDA Ark and RIPE Atlas. Hereby, specific ASes on specific paths can heavily bias the data, which is consistent with anecdotal indications that TTL rewrites are often related to implementation issues with disaggregated routing setups. Thus, future studies using traceroutes should check for TTL rewrites on a per-path basis to reduce and/or at least quantify the impact of TTL rewrites on their work.

References

[1] L. Andersson and E. Rosen. 2006. *Framework for Layer 2 Virtual Private Networks (L2VPNs).* RFC 4664. IETF. https://www.rfc-editor.org/rfc/rfc4664.txt

[2] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. 2006. Avoiding

traceroute anomalies with Paris traceroute. In *Proceedings of the ACM SIGCOMM Conference on Internet Measurement*. https://doi.org/10.1145/1177080.1177100
[3] Brice Augustin, Timur Friedman, and Renata Teixeira. 2007. Multipath tracing with Paris traceroute. In *Workshop on End-to-End Monitoring Techniques and Services*. https://doi.org/10.1109/E2EMON.2007.375313
[4] F. Baker. 1995. *Requirements for IP Version 4 Routers*. RFC 1812. IETF. https://www.rfc-editor.org/rfc/rfc1812.txt
[5] Khalid Bakhshaliyev, Muhammed Abdullah Canbaz, and Mehmet Hadi Gunes. 2019. Investigating Characteristics of Internet Paths. *ACM Trans. Model. Perform. Eval. Comput. Syst.* (2019). https://doi.org/10.1145/3342286
[6] Robert Beverly. 2016. Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/2987443.2987479
[7] R. Bonica, D. Gan, D. Tappan, and C. Pignataro. 2007. *ICMP Extensions for Multiprotocol Label Switching*. RFC 4950. IETF. https://www.rfc-editor.org/rfc/rfc4950.txt
[8] CAIDA. 2025. *Archipelago (Ark) Measurement Infrastructure*. Retrieved 2025-10-26 from https://www.caida.org/projects/ark/
[9] Esteban Carisimo, Caleb J. Wang, Mia Weaver, Fabián E. Bustamante, and Paul Barford. 2023. A Hop Away from Everywhere: A View of the Intercontinental Long-haul Infrastructure. *Proc. ACM Meas. Anal. Comput. Syst.* 7, 3, Article 47 (Dec. 2023), 26 pages. https://doi.org/10.1145/3626778
[10] A. Conta, S. Deering, and M. Gupta. 2006. *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*. RFC 4443. IETF. https://www.rfc-editor.org/rfc/rfc4443.txt
[11] S. Deering and R. Hinden. 2017. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 8200. IETF. https://www.rfc-editor.org/rfc/rfc8200.txt
[12] R. Despres, S. Jiang, R. Penno, Y. Lee, G. Chen, and M. Chen. 2015. *IPv4 Residual Deployment via IPv6 - A Stateless Solution (4rd)*. RFC 7600. IETF. https://www.rfc-editor.org/rfc/rfc7600.txt
[13] Gregory Detal, Benjamin Hesmans, Olivier Bonaventure, Yves Vanaubel, and Benoit Donnet. 2013. Revealing Middlebox Interference with Tracebox. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/2504730.2504757
[14] Benoit Donnet, Matthew Luckie, Pascal Mérindol, and Jean-Jacques Pansiot. 2012. Revealing MPLS tunnels obscured from traceroute. *ACM SIGCOMM Computer Communication Review* (2012). https://doi.org/10.1145/2185376.2185388
[15] Rodérick Fanou, Bradley Huffaker, Ricky Mok, and K. C. Claffy. 2020. Unintended Consequences: Effects of Submarine Cable Deployment on Internet Routing. In *Proc. Passive and Active Measurement (PAM)*. https://doi.org/10.1007/978-3-030-44081-7_13
[16] Tobias Fiebig and Anja Feldmann. 2025. How I learned to stop worrying and love IPv6: Measuring the Internet's Readiness for DNS over IPv6. In *Proc. ACM Internet Measurement Conference (IMC)*.
[17] S. Hanks, T. Li, D. Farinacci, and P. Traina. 1994. *Generic Routing Encapsulation (GRE)*. RFC 1701. IETF. https://www.rfc-editor.org/rfc/rfc1701.txt
[18] S. Hanks, T. Li, D. Farinacci, and P. Traina. 1994. *Generic Routing Encapsulation over IPv4 networks*. RFC 1702. IETF. https://www.rfc-editor.org/rfc/rfc1702.txt
[19] Fahad Hilal and Oliver Gasser. 2023. Yarrpbox: Detecting Middleboxes at Internet-Scale. *Proc. ACM Netw.* (2023). https://doi.org/10.1145/3595290
[20] Thomas Holterbach, Cristel Pelsser, Randy Bush, and Laurent Vanbever. 2015. Quantifying Interference between Measurements on the RIPE Atlas Platform. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/2815675.2815710
[21] C. Hopps. 2000. *Analysis of an Equal-Cost Multi-Path Algorithm*. RFC 2992. IETF. https://www.rfc-editor.org/rfc/rfc2992.txt
[22] Jarrett Huddleston, Matthew Luckie, and Alexander Marder. 2025. Replication: Characterizing MPLS Tunnels over Internet Paths. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/3730567.3764457
[23] Ethan Katz-Bassett, Harsha V Madhyastha, Vijay Kumar Adhikari, Colin Scott, Justine Sherry, Peter Van Wesep, Thomas E Anderson, and Arvind Krishnamurthy. 2010. Reverse traceroute. In *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.
[24] Maynard Koch, Raphael Hiesgen, Marcin Nawrocki, Thomas C. Schmidt, and Matthias Wählisch. 2025. Scanning the IPv6 Internet Using Subnet-Router Anycast Probing. *Proc. ACM Netw.* (2025). https://doi.org/10.1145/3768997
[25] S. Krishnan. 2024. *Segment Routing over IPv6 (SRv6) Segment Identifiers in the IPv6 Addressing Architecture*. RFC 9602. IETF. https://www.rfc-editor.org/rfc/rfc9602.txt
[26] S. Litkowski, S. Agrawal, K. Ananthamurthy, and K. Patel. 2020. *Advertising IPv4 Network Layer Reachability Information (NLRI) with an IPv6 Next Hop*. RFC 8950. IETF. https://www.rfc-editor.org/rfc/rfc8950.txt
[27] Pietro Marchetta, Walter de Donato, and Antonio Pescapé. 2013. Detecting Third-Party Addresses in Traceroute Traces with IP Timestamp Option. In *Proc. Passive and Active Measurement (PAM)*. https://doi.org/10.1007/978-3-642-36516-4_3
[28] Alexander Marder. 2020. APPLE: Alias Pruning by Path Length Estimation. In *Proc. Passive and Active Measurement (PAM)*. https://doi.org/doi.org/10.1007/978-3-030-44081-7_15
[29] Alexander Marder, Matthew Luckie, Amogh Dhamdhere, Bradley Huffaker, kc claffy, and Jonathan M. Smith. 2018. Pushing the Boundaries with bdrmapIT: Mapping Router Ownership at Internet Scale. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/3278532.3278538
[30] Y. El Mghazli, T. Nadeau, M. Boucadair, K. Chan, and A. Gonguet. 2005. *Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management*. RFC 4176. IETF. https://www.rfc-editor.org/rfc/rfc4176.txt
[31] Ivan Morandi, Francesco Bronzino, Renata Teixeira, and Srikanth Sundaresan. 2019. Service Traceroute: Tracing Paths of Application Flows. In *Proc. Passive and Active Measurement (PAM)*. https://doi.org/10.1007/978-3-030-15986-3_8
[32] NLNOG. 2025. *NLNOG RING*. Retrieved 2025-10-26 from https://ring.nlnog.net/
[33] University of Oregon. 2025. *RouteViews Project*. Retrieved 2025-10-26 from https://www.routeviews.org/routeviews/
[34] PeeringDB. 2025. *The Interconnection Database*. Retrieved 2025-10-26 from https://www.peeringdb.com/
[35] PeeringDB. 2025. *Microsoft AS8069*. Retrieved 2025-11-18 from https://www.peeringdb.com/asn/8069
[36] J. Postel. 1980. *DoD standard Internet Protocol*. RFC 760. IETF. https://www.rfc-editor.org/rfc/rfc760.txt
[37] J. Postel. 1981. *Internet Control Message Protocol*. RFC 777. IETF. https://www.rfc-editor.org/rfc/rfc777.txt
[38] Alagappan Ramanathan and Sangeetha Abdu Jyothi. 2023. Nautilus: A Framework for Cross-Layer Cartography of Submarine Cables and IP Links. *Proc. ACM Meas. Anal. Comput. Syst.* (2023). https://doi.org/10.1145/3626777
[39] Davila Revelo, Mauricio Anderson Ricci, Benoit Donnet, and José Ignacio Alvarez-Hamelin. 2016. Unveiling the MPLS structure on Internet topology. In *Proc. Network Traffic Measurement and Analysis Conference (TMA)*.
[40] RIPE. 2025. *RIPE Atlas*. Retrieved 2025-10-26 from https://atlas.ripe.net/
[41] RIPE Atlas Docs. 2025. *Measurement Result Format*. Retrieved 2025-11-18 from https://atlas.ripe.net/docs/apis/measurement-result-format
[42] E. Rosen, A. Viswanathan, and R. Callon. 2001. *Multiprotocol Label Switching Architecture*. RFC 3031. IETF. https://www.rfc-editor.org/rfc/rfc3031.txt
[43] Pavlos Sermpezis, Lars Prehn, Sofia Kostoglou, Marcel Flores, Athena Vakali, and Emile Aben. 2023. Bias in Internet Measurement Platforms. In *Proc. Network Traffic Measurement and Analysis Conference (TMA)*. https://doi.org/10.23919/TMA58422.2023.10198985
[44] W. Simpson. 1995. *IP in IP Tunneling*. RFC 1853. IETF. https://www.rfc-editor.org/rfc/rfc1853.txt
[45] D. Thaler and C. Hopps. 2000. *Multipath Issues in Unicast and Multicast Next-Hop Selection*. RFC 2991. IETF. https://www.rfc-editor.org/rfc/rfc2991.txt
[46] Bulut Ulukapi, Anna Sperotto, and Ralph Holz. 2025. Tracing Vendors: A Middlebox-Centric Study of Network Interference. In *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. https://doi.org/10.1109/EuroSPW67616.2025.00027
[47] Yves Vanaubel, Jean-Romain Luttringer, Pascal Mérindol, Jean-Jacques Pansiot, and Benoit Donnet. 2019. TNT, Watch me Explode: A Light in the Dark for Revealing MPLS Tunnels. In *Proc. Network Traffic Measurement and Analysis Conference (TMA)*. https://doi.org/10.23919/TMA.2019.8784525
[48] Kevin Vermeulen, Ege Gurmericliler, Italo Cunha, David Choffnes, and Ethan Katz-Bassett. 2022. Internet scale reverse traceroute. In *Proc. ACM Internet Measurement Conference (IMC)*. 694–715. https://doi.org/10.1145/3517745.3561422
[49] Kevin Vermeulen, Stephen D. Strowes, Olivier Fourmaux, and Timur Friedman. 2018. Multilevel MDA-Lite Paris Traceroute. In *Proc. ACM Internet Measurement Conference (IMC)*. https://doi.org/10.1145/3278532.3278536

A Ethics

This paper raises no ethical questions. We conduct active measurements from RIPE Atlas towards a small set of vantage points (one under our control, 100 operated by NLNOG RING). Our scans focus on likely paths with TTL rewrites to reduce the overall scan volume and are conducted with a low rate. No users are impacted by our studies. Before starting our measurements, we additionally conferred with the RING admins, who saw no issue in us running these specific measurements. Besides this small volume active scan, we limit ourselves to existing, open data sources, namely RIPE Atlas and CAIDA Ark.

B Open Science

The packet captures and trace data of both our single-destination measurement as well as our two measurements targeting NLNOG Ring nodes are available at https://doi.org/10.17617/3.D5GQFG.

C Supplementary Figures

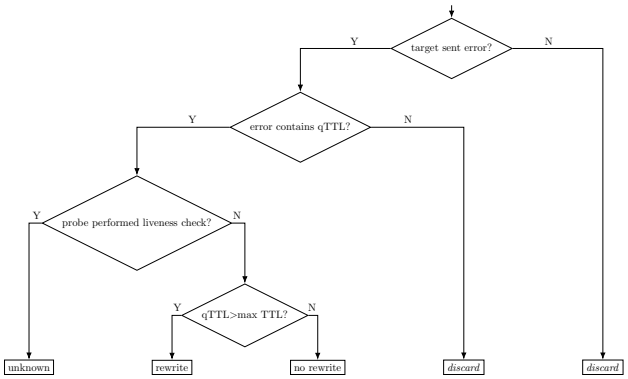
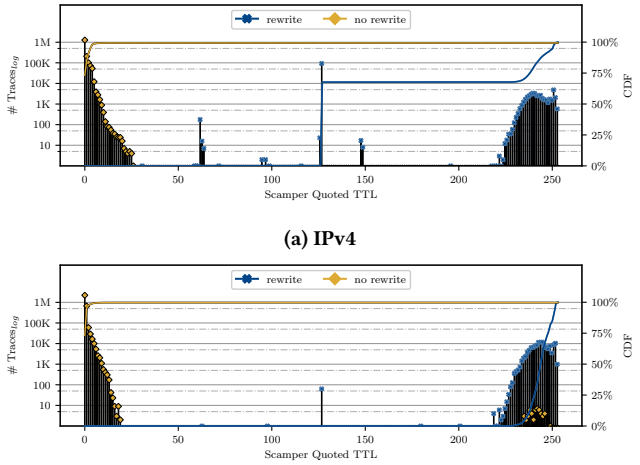


Figure 8: Decision chart illustrating how we label RIPE Atlas traces in Figure 4 (cf. Sections 4.3 and 6).

Figure 8 is a decision chart illustrating how we classify RIPE Atlas traces in Section 6: we first select all traces that received an error and check if the RIPE Atlas probe recorded a quoted TTL. If the probe performed a liveness check in a particular trace, we cannot tell if a rewrite occurred and label it as "unknown". Otherwise, if the quoted TTL is larger than the greatest TTL sent in the course of the trace, we know that the TTL must have been rewritten and label it accordingly.



(a) IPv4

(b) IPv6

Figure 9: *Revisiting Scans—CAIDA Ark:* Quoted TTLs from ICMP error message. Traceroutes with a TTL rewrite are identified based on the quoted TTL.

Figure 9 shows the affected traces from our analysis of a single-day CAIDA Ark snapshot in Section 6.1.