Destination Reachable: What ICMPv6 Error Messages Reveal About Their Sources

Florian Holzbauer
Faculty of Computer Science
Doctoral School Computer Science
University of Vienna
Vienna, Austria
florian.holzbauer@univie.ac.at

Markus Maier
SBA Research
Vienna, Austria
mmaier@sba-research.org

Johanna Ullrich
University of Vienna
Vienna, Austria
johanna.ullrich@univie.ac.at

Abstract

The probability of hitting an active IPv6 address by chance is virtually zero; instead, it appears more promising to analyze ICMPv6 error messages that are returned in case of an undeliverable packet. In this paper, we investigate the implementation of ICMPv6 error messages by different router vendors, whether a remote network's deployment status might be inferred from them, and analyze ICMPv6 error messaging behavior of routers in the IPv6 Internet. We find that Address Unreachable with a delay of more than a second indicates active networks, whereas Time Exceeded, Reject Route and Address Unreachable with short delays pinpoint inactive networks. Furthermore, we found that ICMPv6 rate-limiting implementations, used to protect routers, allow the fingerprinting of vendors and OS-versions. This enabled us to detect more than a million periphery routers relying on Linux kernels from 2018 (or before); these kernels have reached end of life (EOL) and no longer receive security updates.

CCS Concepts

• **Networks** → **Network protocols**; **Routers**; **Network measurement**; **Public Internet**.

Keywords

IPv6, ICMPv6 Error Messages, Network Activity Classification, Router Classification, Rate Limiting

ACM Reference Format:
Florian Holzbauer, Markus Maier, and Johanna Ullrich. 2024. Destination Reachable: What ICMPv6 Error Messages Reveal About Their Sources. In *Proceedings of the 2024 ACM Internet Measurement Conference (IMC '24), November 4–6, 2024, Madrid, Spain.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3646547.3688420

1 Introduction

An exhaustive scan of the IPv4 Internet takes less than an hour [1, 11], which remains infeasible with the successor protocol IPv6 due to the sheer size of the address space. Alternative approaches are needed. In fact, the probability of hitting an alive IPv6 host by chance is virtually zero [12, 18, 32] and it appears more promising to analyze the numerous ICMPv6 error messages that are returned in case of undeliverability as they provide insight into remote networks.

ICMP error messages have been collected before, both for IPv4 and IPv6. Best known are topology discovery [6, 9, 16], also known as tracerouting, and routing loop detection [22, 23]. Also, ICMPv6 messages have been intentionally triggered to extract source addresses, thus collecting millions of IPv6 addresses of periphery devices [22, 33].

We take a different stance and analyze ICMPv6 error messages beyond their source addresses. The main goal of this paper is to (I) examine the different error message types that are returned by active and inactive networks on the Internet and (II) classify routers within these networks. Based on the type, code, and timing of an error message, we infer routing scenarios other than routing loops such as active networks, ACL filtering, or null routes. Our analysis considers aspects such as the responsiveness of routers, given that most ICMPv6 error message types are sent voluntarily, the compliance of routers with the ICMPv6 specification [10], and whether variances in type usage and rate limiting implementations allow to classify router and OS versions.

We follow a threefold methodology for both aspects, namely (I) *network activity classification* and (II) *router classification*. First, we observe ICMPv6 error messaging behavior of eleven router vendors in the network simulator GNS3, providing full control over the setup. Second, we use labeled datasets [2, 14] to verify whether the behavior observed in the virtual setup is congruent with that of actual routers in the IPv6 Internet. Third, we conduct Internet measurements to gain insight into the current state of ICMPv6 error message implementations across routed networks and to enumerate ICMPv6 error message handling within a more diverse set of networks. Our research makes several key contributions, detailed in the following sections. The code for our measurements is publicly available [1].

Network Activity Classification (§4.1, §4.2). We associate ICMPv6 error message types with the activity status of a remote network. The receipt of message type *Address Unreachable* with long delays is found to indicate active networks with a probability of 95.1%, while *Time Exceeded*, *Reject Route* and *Address Unreachable* with short delays indicate inactive networks with a probability of 79.5%.

BValue Steps (§4.2). To validate our network activity detection, we develop the BValue step method to create datasets of addresses in

[1] https://github.com/sbaresearch/icmpv6-destination-reachable

IMC '24, November 4–6, 2024, Madrid, Spain

ACM ISBN 979-8-4007-0592-2/24/11
https://doi.org/10.1145/3646547.3688420

active and inactive networks from the IPv6 Hitlist Service [15]. This method is also useful to investigate the error message responding behavior of individual networks in case an active address is known.

Network Activity Scans(§4.3). We collect and classify ICMPv6 error messages across a wide portion of routed IPv6 Internet. In two measurements, we discovered 83M (of a total of 5Bn) /48s and 356M (of a total of 6Bn) /64s to be active. Our methodology is useful in order to guide host discovery toward more promising parts of the Internet.

Router Classification (§5.1, §5.2). In our GNS3 environment, we identify different ICMPv6 rate limiting behavior of routers and exploit it to remotely classify them (vendor/operating system). Relying on SNMPv3-vendor labels for ground truth [2], we were able to verify our approach on the Internet and to extend it with additional fingerprints. Our method fills a gap as previous work based on varying iTTL values [3, 36] is not applicable since the Hop Limit in IPv6 has been increasingly harmonized [8].

Linux-Based Routers at the End of Life (§5.3). In a large-scale measurement study, we classified 1.4M routers applying our method on the Internet and found 1M periphery routers relying on Linux kernels from 2018 or before. These kernel versions have reached end of life at latest by January 2023 and pose a potential security risk.

2 Terminology

Routed, Active, and Inactive Networks. If a network prefix is available in routing tables, packets towards its addresses can be forwarded. We consider these addresses to be *routed*. The mere routability is not sufficient for successful delivery. On the receiver's side, it also requires a last-hop router attached to the local network that forwards the packets to their final destinations. Therefore, the router conducts Neighbor Discovery [25] to resolve IP addresses into link-layer addresses. In this work, we denote such networks as *active* networks. If a last hop router is not prevalent or if it is discarding traffic towards the destination network, we refer to these networks as *inactive*. The distinction between *active* and *inactive* prefixes facilitates reconnaissance since responsive IPs can only exist in active networks.

Assigned, Unassigned and Responsive Addresses. In an active network, only a fraction of its addresses are assigned to individual hosts and might be used to communicate with others. We refer to these addresses as *assigned* addresses. If such an address is returning packets (e.g., *ICMPv6 Echo Replies*) upon request (e.g., *ICMPv6 Echo Requests*), it is considered to be a *responsive* address. An address that is not assigned to any host – and thus cannot be used in communication – is referred to as an *unassigned* address.

ICMPv6 Error Messages. RFC4443 [10] defines two informational message types – (*Echo Request* and *Echo Reply*), used for diagnosis (ping) – and four error message types. The message types and their subcodes are listed in Table 1. For readability, we use two-letter abbreviations instead of the messages' full name. If no response is received, we use the symbol ∅. The RFC defines processing of ICMPv6 messages as follows: (1) Only *TB* and *TX* messages are mandatory; the others are optional. (2) ICMPv6 error messages include the packet triggering the error as a payload. This allows the extraction of the initial request's destination. (3) Neighbor Discovery [25] uses the ICMPv6 message format. A router sends a *Neighbor Solicitation* to resolve an IPv6 into a link-layer address. Per address to be resolved, the sending of only one such message per second is allowed. If unresolved after three attempts, the router should return *AU*. (4) Rate limiting of ICMPv6 messages is mandatory, and a token bucket algorithm is proposed. For each message sent, a token is removed. If the bucket is empty, messages are discarded until a refill.

ICMPv6 Types and Codes	Abbr.
Destination Unreachable	
No route to destination	*NR*
Admin. prohibited	*AP*
Beyond scope of source address	*BS*
Address unreachable	*AU*
Port unreachable	*PU*
Ingress/egress policy	*FP*
Reject route to destination	*RR*
Time Exceeded	*TX*
Packet Too Big	*TB*
Parameter Problem	*PP*
Echo Request	*EQ*
Echo Reply	*ER*
Unresponsive	∅

Table 1: ICMPv6 error message types from RFC4443 and abbreviations used in the paper.

3 Methods Overview

In this paper, we rely on a three-step methodology for both of our goals, (I) the classification of a remote network's activity status based on the received ICMPv6 error message types in Section 4, and (II) the router classification based on ICMPv6 rate limiting behavior in Section 5. In particular, we

- (M1) investigate ICMPv6 error messaging behavior of eleven router vendors in a virtual GNS3 setup, facilitating full control of the router configurations[2].
- (M2) validate if routers on the IPv6 Internet behave in the same way as observed in our fully controlled laboratory environment. Our validations build upon labeled datasets [2, 14].
- (M3) perform measurements on the IPv6 Internet to show the extent of our findings and provide insights into the current state of the Internet's deployment.

Network Activity Classification. We analyze ICMPv6 error message types and classify them to draw conclusions about a remote network's activity status. (M1) (§4.1) In our virtual GNS3 setup, we tested 15 routers and firewalls from 11 vendors in six different routing scenarios – such as forwarding packets to unassigned IP addresses, lacking routing table entries, or null routes – to see whether they show coherent behavior among each other as well as with regard to RFC4443 [10]. Based on the results, we associate ICMPv6 error message types with the activity status (active, inactive, ambiguous) of the remote network that has returned this message. (M2) (§4.2) The virtual setup is by definition limited in variety. Consequently, we performed a measurement to verify whether our observations are congruent with the diverse routers on the Internet. Applying our BValue steps method, we therefore inferred

[2] https://github.com/sbaresearch/router-lab

data sets of (unassigned) addresses in active and inactive networks from the IPv6 Hitlist Service [14, 34, 38]; then, we compared them against our assumptions on the correlation between ICMPv6 error message type and network activity status. (M3) (§4.3) Finally, we conducted two measurements. Relying on yarrp [6], the first probes all BGP-announced prefixes at the granularity of /48, resulting in 5 Billion traces. In this measurement, shorter prefixes (e.g. a /32) are resolved into multiple /48 prefixes. Prefixes less specific than /24 are prescanned for promising /24s by scanning 2 targets per included /32 and take those for which we receive a response. The second measurement used ZMap [11] to exhaustively probe the 92,856 /48 prefixes that are announced in BGP (as of November 2023) at /64 granularity. In the latter measurement, less-specific prefixes announced in BGP were ignored.

Router Classification. We exploit ICMPv6 rate limiting behavior to identify a remote router's vendor and/or OS version. (M1) (§5.1) In our virtual setup, we measured the vendors' default settings for ICMPv6 rate limiting as a baseline for comparison with real-world behavior on the Internet. Many routers are based on Linux or BSD; thus, we additionally investigated different kernel versions to understand their default behavior. (M2) (§5.2) Previous work found that certain routers unintentionally reveal vendor and other information by responding to unsolicited and unauthenticated SNMPv3 requests [2]. We were able to measure 50K routers with SNMPv3 vendor labels available, allowing us to verify our results against ground truth and to additionally extend our database of fingerprints. (M3) (§5.3) After confirming the congruence of our router classification against ground truth, we measured and classified a total of 1.4M routers with regard to their vendors and/or operating systems. In addition, we were able to group them into Internet core- and periphery devices – depending on the number of paths they appeared in the previous measurement with yarrp – thus revealing different router populations.

Limitations. *(I) Router Coverage.* Our testbed is, by definition, limited to vendors and versions available in GNS3. This also includes configuration options as some were restricted, e.g., ACLs for two RUTs or Null Routes for another ones. We marked these scenarios with (-) in Table 9 in Appendix B. We countered this limitation by extending our validation to routers on the IPv6 Internet. *(II) Validation in the Internet.* We did not have access to ground truth other than virtual appliances in our laboratory setup and labeled datasets from related work [2, 14]. We used these data sets to validate if our findings are representative by comparing if routers on the Internet behave similarly to those in the laboratory setup. *(III) Network Coverage.* We could not fully cover the routed IPv6 address space in our prefix-seeded measurements. For routed /48 networks, we can cover the subnet space up to /64. For networks larger than /48, we sampled their subnet space. In the second measurement, we prioritized coverage of many networks instead of going in-depth into single networks.

4 Network Activity Classification

Our measurements in the virtual laboratory (§4.1) reveal that the router implementations of ICMPv6 error messages deviate from the specifications outlined in RFC4443 [10]. These discrepancies led to a reassessment of our classification of error messages, but also allow to fingerprint a router's vendor. In our validation (§4.2), we found the same distinctive behavior for routers in the IPv6 Internet. We performed two measurements to find active networks on today's Internet (§4.3), reducing the search space for host discovery to 1.7% and 12%, respectively.

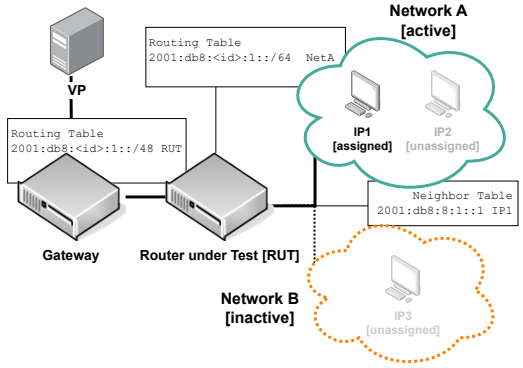


Figure 1: GNS3 laboratory setup. As common for IPv6 [22, 28], the RUT routes traffic to the active /64 network A, but not to inactive network B.

4.1 ICMPv6 Error Message Defaults

In a virtual setup, we analyzed the default ICMPv6 responding behavior of 15 routers and firewalls in six routing scenarios. Implementations show variance and deviations from RFC 4443 [10].

Router Laboratory. In the network emulator GNS3, we set up a test network, see Figure 1. The gateway forwards traffic towards a /48 prefix to the router-under-test (RUT), but the RUT is only configured as a last-hop router for a /64 subnetwork (network *A*). According to our terminology, the /48 prefix is routed, but only network *A* is active. In network *A*, address *IP1* is assigned to an alive host and responsive, while *IP2*, belonging to the same network range, remains unassigned. In contrast, network *B* is inactive due to the RUT not being configured to handle traffic for network *B* and, thus, lacking a last-hop router conducting Neighbor Discovery. *IP3* represents an address within the inactive network's range *B*.

Routing Scenarios. We configured six different routing scenarios, (S1) to (S6), which trigger ICMPv6 error messages based on the specification in RFC4443. We use 15 virtual images for the RUT to reveal the different routers' default ICMPv6 messaging behavior. If ICMPv6 error messages are not sent by default, we enable them for our experiments. We probe IP addresses using ICMPv6 Echo Requests, TCP SYNs, and UDP requests to verify protocol-specific response behavior. We list the expected response types based on the specification of RFC4443 for each scenario next to the scenario's name.

- (S1) **Active Network** - *AU*. Network *A* is directly configured on one of the RUT's interfaces. Requests towards IP2 reveal the ICMPv6 error message in case of an unassigned address in an active network.

	S1	Active Network	S2	Inactive Network	S3	Active Netw. ACL	S4	Inactive Netw. ACL	S5	Null Route	S6	Routing Loop
NR	○	0	●	14	●	1	●	2	●	2	○	0
AP	○	0	○	0	●	4	●	5	●	3	○	0
AU	●	14	○	0	○	0	○	0	●	1	○	0
PU	○	0	○	0	●	3	●	2	○	0	○	0
FP	○	0	●	1	●	1	●	2	○	0	○	0
RR	○	0	○	0	○	0	○	0	●	2	○	0
TX	○	0	○	0	○	0	○	0	○	0	●	15
∅	●	1	○	0	●	4	●	3	●	9	○	0

NOTE: Number = # of routers that return the error message type in a scenario; a single RUT can return multiple error message types if more than one configuration option is available.

Table 2: ICMPv6 error messages from 15 RUTs in 6 routing scenarios. The expected error message is indicated in gray. We list individual RUTs in Table 9 in Appendix B.

- (S2) **Inactive Network** - *NR*. The RUT receives a packet for which it has no entry in its routing table. Requests towards IP3 reveal the ICMPv6 error message in case of an inactive network.
- (S3) **Active Network with ACL** - *AP*,*FP*. We configure ACLs that either filter packets (I) towards network A (destination-based filtering) or (II) from our vantage point (source-based filtering). We probe IPs in network *A* to reveal the ICMPv6 error message for an active network with ACL.
- (S4) **Inactive Network with ACL** - *AP*,*FP*. An ACL for network *B* is configured to verify whether differences among active and inactive networks with ACLs are observable. Requests towards IP3 reveal the error message in case of an inactive network with ACL.
- (S5) **Null Routes** - *RR*. A null route is configured, discarding/rejecting all packets towards network B, and address IP3 is probed.
- (S6) **Routing Loops** - *TX*. The RUT maintains a default route towards the gateway. As network B is not routed, requests towards IP3 will be routed back via the incoming interface, forming a routing loop.

Results. Table 2 provides an overview on the received ICMPv6 error messages per routing scenario. Focusing on *implementation coherency*, we found congruent behavior among the different routers for (S1), (S2), and (S6) – with single exceptions for (S1) and (S2). For the remaining scenarios, we see five different message types each due to vendor-specific filtering implementations. We also found *differences between scenarios* (S3) and (S4). For routers that rely on forward chain filters, the routing decision is made before the filter is applied. This results in three RUTs that are more likely to be used in the Internet edge, returning the same error message type as in (S2). Regarding *response timings*, there is a peculiarity for *AU*. For (S1), we notice delays of 2, 3, and 18 seconds as the messages are only returned after the Neighbor Discovery's timeout. We discovered that a delay of 2s is unique to Juniper, while 18s to Cisco XRv, allowing to fingerprint these vendors based on the delays. The other routers show delays of 3s as proposed in the RFC. For (S5), *AU* is returned immediately. Also, all other message types are returned immediately. Comparing the *request protocols*, we only find differences in the presence of ACLs. Two RUTs try to mimic protocol-specific responses from the target host for TCP and UDP.

Status	*NR*	*AP*	$AU_{>1s}$	$AU_{<1s}$	*PU*	*FP*	*RR*	*TX*
active	○	○	●	○	○	○	○	○
inactive	○	○	○	●	○	○	●	●
ambig.	●	●	○	○	●	●	○	○

Table 3: Classification of ICMPv6 error message types indicating activity/inactivity of a remote network.

Network Activity Classification. Our goal is to differentiate ICMPv6 messages indicating active remote networks from those indicating an inactive ones. Based on the results in Table 2, we classify messages that have only been returned for active networks ((S1), (S3)) as *active*, those that have only been returned for inactive networks ((S2), (S4), (S5), (S6)) as *inactive*, and those appearing in both cases as *ambiguous*.

AU is consistently received for active networks ((S1)) but has also been returned by Juniper routers for (S5). Thus, we would have to consider *AU* to be ambiguous. Yet, the difference in timing – (S1) always causes a delay of multiple seconds that is longer than typical RTTs on the Internet – allows differentiation. For the remainder of the paper, we differentiate between $AU_{RTT>1s}$ indicating an active network, and $AU_{RTT<1s}$ indicating an inactive network. Table 3 summarizes our classification of ICMPv6 error message types.

Compliance with the RFC. Testing router vendor implementations, we found behavior that deviates from the specification in RFC4443 [10]. This has an impact on the diagnostic value of the error message types, as one cannot simply rely on the RFC, but needs to know the vendor-specific behavior. This leads to a different network activity classification than if we had based our classification solely on the RFC.

The affected types and scenarios are *FP* for inactive networks ((S2)), *NR* and *PU* for filtering ((S3) and (S4)) and *NR*, *AP* and *AU* for null routing ((S5)). Based on the RFC, *PU* and *AU* should be returned for active networks. In addition, *PU* should only be returned by destination nodes only, i.e., assigned IPs. In our measurement, however, we found one of the firewalls using *PU* to mimic responses from the target host. Next to reporting a failure in Neighbor Discovery for unassigned addresses, we found one RUT implementing *AU* instead of *RR* for null routing. Following the specification, *NR* should be used for inactive networks due to a lacking entry in the routing table. However, we also found one RUT to return *NR* for active networks with ACL in (S3) and null routes in (S5).

4.2 ICMPv6 Error Messages in the Internet

The virtual setup, as used in the previous section, is limited to the availability of router images and does not fully reflect the variety of routers on the Internet. For validation of our network activity classification in Table 3, we need unassigned addresses from networks on the Internet, categorized as active and inactive. Probing these addresses helps us collect error messages specific to each category. Since no datasets of addresses exist for these categories, we developed a method called BValue Steps to separate addresses in active and inactive networks.

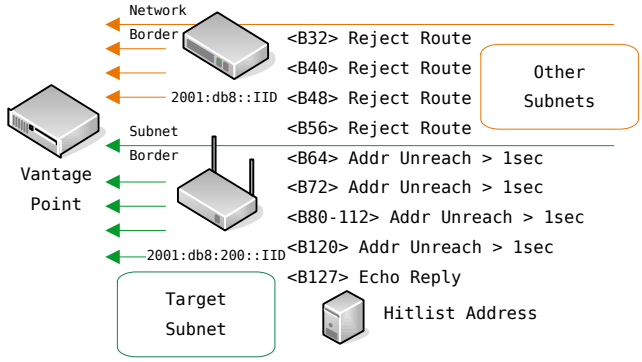


Figure 2: BValue Steps aim for a change in ICMPv6 error messages. Message types before the change represent active, those after the change inactive networks.

Data Set Generation. Based on our terminology, a responsive IP address, as those present in hitlists, resides in an active network. To collect ICMPv6 error messages for unassigned addresses in the same active network, we derived addresses from the responsive address by randomizing their lower bits. With more and more randomized bits, we eventually reached the network border and probed addresses outside the active network. This way, we can collect ICMPv6 error message for other, likely inactive parts of the BGP-announced prefix. We measured addresses from hitlists this way, and included those with a change in received ICMPv6 error message types into our analysis. For these networks, we label the error message type *before* the change to represent addresses in active networks, and the one *after* the change to represent addresses in inactive networks, see Figure 2.

BValue Steps. Assuming knowledge of an assigned IPv6 address and its respective routed network prefix length, we take the address and replace its lower bits – in multiples of eight bits – with random values. Figure 3 shows our approach with an example address. These addresses are referred to as BValue (Border value) addresses for bit 120, 112, 104, etc. (short B120, B112, B104, etc.). The number indicates the highest randomized bit. If the network border – in our example bit 32 – is reached, the process stops. In total, five addresses are generated for each BValue step. This allows to compensate the loss of individual responses or rare positive replies from hitting an assigned address/active network by chance. For higher BValue steps, there is a higher chance of targeting other assigned addresses close to the hitlist address. Therefore, for each step, a majority vote decides on the error message type, ignoring protocol-specific positive responses such as *ER*, *TCPACK*, *RST*. Additionally, we measure *B*127, an address congruent with the seed address, flipping only the last bit.

Measurement Setup. We apply our method of BValue Steps to the IPv6 Hitlist Service [14] which provides responsive addresses. For the network borders, we use RIPE RIS BGP looking glass [26]. Preventing bias from networks with many addresses in the hitlist, we only take a single address per BGP-announced prefix. In total, we probed 47,923 addresses with three protocols (ICMPv6, TCP - Port 443, and UDP - Port 53) from two vantage points on five successive days in March 2023 (2023-03-14 to 2023-03-18).

```
Original hitlist address:
          2001:db8:1234:abcd:1234:abcd:1234:0101
Generated addresses:
          <original bits>              <random bits>
B127      2001:db8:1234:abcd:1234:abcd:1234:0100
B120      2001:db8:1234:abcd:1234:abcd:1234:01e8
B112      2001:db8:1234:abcd:1234:abcd:1234:6aa1
B104      2001:db8:1234:abcd:1234:abcd:1221:f38d
...
B48       2001:db8:abcd:5276:d080:ccd6:7fc3:311c
B40       2001:db8:ab3e:3eb7:4c66:7f16:ade5:2b3d
B32       2001:db8:7438:221f:b244:476c:66bb:8da5
```

Figure 3: BValue Steps address generation. From an active address, more and more bits are randomized in steps of 8.

		Vantage 1 (σ)			Vantage 2 (σ)		
W. Ch.	ICMPv6	21,070	(79)	44.2%	20,847	(30)	44.1%
	TCP	18,393	(57)	38.6%	18,142	(25)	38.3%
	UDP	24,620	(108)	51.7%	24,287	(33)	51.3%
W/o. Ch.	ICMPv6	8,165	(41)	17.1%	8,014	(24)	16.9%
	TCP	6,808	(28)	14.3%	6,727	(26)	14.2%
	UDP	6,005	(25)	12.6%	5,879	(29)	12.4%
∅	ICMPv6	18,407	(62)	38.6%	18,461	(28)	39.0%
	TCP	22,441	(89)	47.1%	22,452	(53)	47.4%
	UDP	17,017	(59)	35.7%	17,156	(24)	36.3%

NOTE: # of Networks = mean and σ = standard deviation of five days.

Table 4: As a basis for validation, BValue differentiates networks (i) with a change in ICMPv6 error message, (ii) without such a change, and (iii) unresponsive networks.

Data Set. For 44% (ICMP), 38% (TCP), and 52% (UDP) of the seed addresses, we were able to differentiate active from inactive networks by observing at least one change in ICMPv6 error messages, see Table 4. Depending on the protocol, around 12% to 17% of prefixes show no change in error message types and 36% to 47% do not return error messages at all. The results are consistent across both vantage points.

Comparing source addresses at ICMPv6 error message type changes, in 86% of the cases, the change in response type aligns with a change in the source address (and consequently the responding router), further supporting our assumption on network borders. For the other cases a single router serves the target network and other network ranges.

Figure 4 shows the BValue after which the message type changes have been observed. 71.6% are found at B64+, reflecting well-known IPv6 address assignment strategies [30] and supporting our hypotheses. While the overall percentage is low, we also detect networks with multiple network borders. This does not directly impact our labeling, but verifies common network borders used in IPv6. 5% of the networks with a first change also show a second change at the /56 or /48 border. In addition, 0.1% show a third change at the /48 or /40 border. For the remainder of the analysis, we label the message types received for the higher BValues (from B127, i.e., before the first change) to represent active networks, and lower BValues (up to Bxxx, i.e., after the first change) to represent inactive networks.

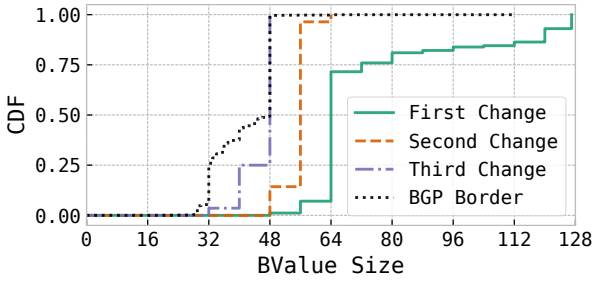
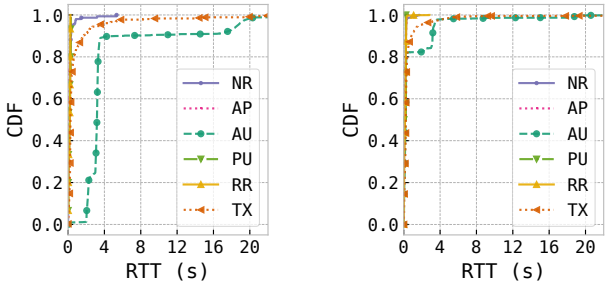


Figure 4: Inferred distribution of IPv6 suballocation sizes for 21,184 (44.2% of measured) IPv6 networks. Results for ICMPv6 on 2023-03-14.



(a) Active Networks **(b) Inactive Networks**

Figure 5: Also on the Internet, *AU* is delayed by multiple seconds for active networks and returned immediately for inactive ones.

Validation - Error Message Response Timings. In a first step, we analyze whether the delays of multiple seconds for *AU* – allowing to differentiate $AU_{RTT<1s}$ from $AU_{RTT>1s}$ – are also observed on the Internet. Figure 5 shows RTTs for the ICMPv6 error message types, separated for active and inactive networks. We see sharp increases at 2, 3, and 18 seconds for active networks (22.25% 2s, 68.5% 3s, 9.25% 18s), reflecting the same delays due to Neighbor Discovery timeout that were also observed for router appliances in the laboratory setup. This implies that also on the Internet it is feasible to distinguish *AU* for active networks from those for inactive networks.

Validation - Error Messages for Active Networks. In the left column of Table 5, the error message types for probing addresses in active networks are shown. With a probability of 95%, classification is successful as we received message types associated with active networks for ICMPv6. In only 2% of the cases, these networks were classified as ambiguous, i.e., no decision can be made. In 3% of the cases, the networks are incorrectly classified as inactive. We reach comparable classification rates for TCP. UDP performs worse with only 56% of networks labeled as active also classified as active. The reason is as follows: For UDP, we cannot verify if *PU* error messages came from the target itself or were caused by a filter. Thus, we categorize the remote network ambiguous instead of active. For many networks we target assigned IPs close to the hitlist address, resulting in *PU* being returned by assigned IPs close to the hitlist address. A difference of nearly 40 percentage points indicates a large share of networks is affected by this. While this negatively impacts our labeling for UDP, it supports our claim that host discovery in these networks is feasible. Still, we cannot classify *PU* as active due to its usage for firewalling. This renders ICMP the preferred protocol for the task of network activity classification.

Validation - Error Messages for Inactive Networks. In the right column of Table 5, the error message types for probing addresses in inactive networks are shown. For ICMPv6, the networks are correctly classified as inactive in 80% of the cases. No classification is feasible in 16% of the cases, and in 5% of the cases they are incorrect. Again TCP shows similar and UDP worse performance.

		labeled active			labeled inactive		
		Netw.	σ	%	Netw.	σ	%
active	ICMPv6	17,361	109	95.1%	471	11	4.6%
	TCP	14,522	112	93.7%	620	12	7.4%
	UDP	12,490	82	56.2%	3,687	35	32.0%
ambig.	ICMPv6	352	10	1.9%	1,645	12	15.9%
	TCP	566	10	3.7%	1,552	14	18.6%
	UDP	9,377	91	42.2%	1,455	7	12.6%
inactive	ICMPv6	537	13	2.9%	8,230	34	79.5%
	TCP	405	8	2.6%	6,191	26	74.0%
	UDP	337	12	1.5%	6,396	49	55.4%

NOTE: σ Standard deviation over five days.

Table 5: Network activity classification (active, ambiguous and inactive) for networks (active, inactive) labeled by BValue Steps.

4.3 Network Activity Scans

In a final step, we performed two prefix-seeded measurements, not including any ground truth, to collect and classify ICMPv6 error messages across a wide portion of the routed IPv6 Internet. In measurement M1, we probed a random address in each routed /48 prefix. Thereby, prefixes of shorter length (e.g., /32) are split in multiple /48 prefixes. In measurement M2, we took only those prefixes that are announced as a /48 in BGP, and exhaustively probe them at the granularity of /64. The first measurement prioritizes breadth over depth and targets more towards the Internet's core, the second measurement focuses on depth instead of breadth and the Internet's periphery.

M1 - Sampling the Internet at /48 Granularity. We take a total of 45,434 prefixes with a prefix length of /48 or shorter. We traceroute a random address within each /48 prefix using yarrp [6, 7], resulting in a total of 5Bn destinations measured from our vantage point 1 between 2023-03-16 and 2023-04-05. Figure 6 visualizes the distribution of active, inactive, ambiguous and unresponsive /48 prefixes. We received 616M responses, representing 12% of all destinations. Classifying the received error messages, we find 83M active and 341M inactive /48s. 192M remain ambiguous. For the detailed share of responses, we refer to Table 6. In comparison to previous measurements, the share of unresponsive destinations appears to be high; however, aggregating them to BGP prefixes, only 39% (17,580) of them do not respond at all. This number is comparable to the previous experiment.

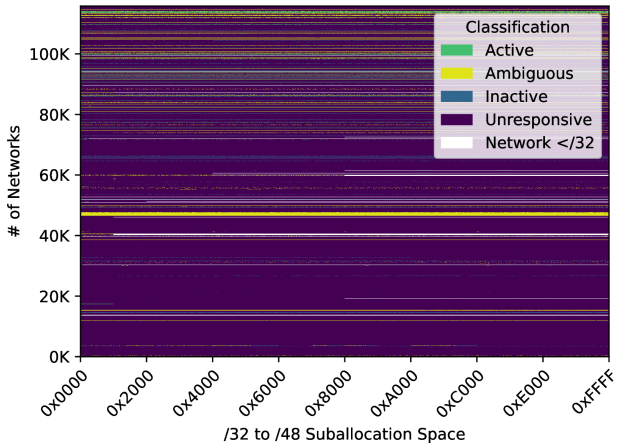


Figure 6: Sampling the Internet at /48 granularity. Each row represents a /32 network and each column one /48 network inside the /32.

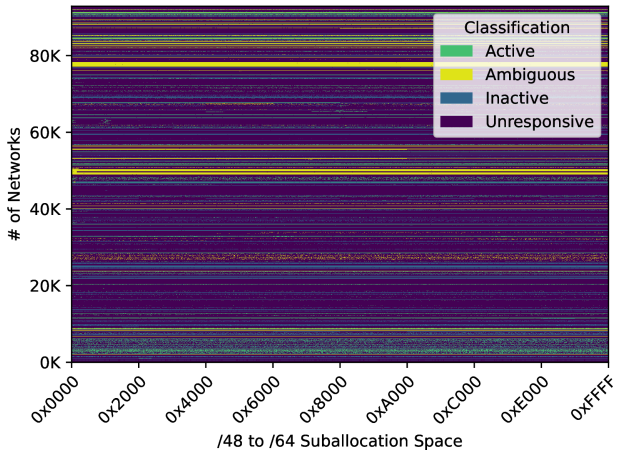


Figure 7: Exhaustive probing of BGP-announced /48 prefixes. Each row represents a /48 prefix and each column a /64 inside the /48.

Type	M1 - Core	M2 - Periphery
$AU_{RTT>1s}$	13.5%	26.0%
NR	20.3%	13.6%
AP	4.3%	1.6%
FP	0.0%	0.0%
PU	6.5%	0.0%
$AU_{RTT<1s}$	13.1%	16.7%
RR	33.3%	9.1%
TX	8.9%	32.8%
Total	616M	1368M

active, ambiguous and inactive networks.

Table 6: Share of ICMPv6 error message types received in measurements M1 and M2.

M2 - Exhaustive Probing of /48s. We focus on the 92,856 networks that were announced as a /48 prefix in BGP (2023-11-01) as we are able to exhaustively probe them at the granularity of /64 and investigate behavior of the Internet periphery. Using ZMap, we probe a random address in each encompassed /64 prefix, resulting in a total of 6Bn destinations. Figure 7 visualizes the distribution of message types that are classified as active, inactive, ambiguous and unresponsive. We received 1.4Bn responses, representing 23% of all destinations. We classified 356M as active, 802M as inactive and 210M remain ambiguous. In comparison to M1, we received a higher share of responses that are classified as active. We discovered 45.3M unique sources of error messages, with 14M periphery routers that perform Neighbor Discovery. Of those 4M rely on EUI-64 addresses, with the most represented vendors (>10K routers) being Huawei, ZTE, T3, Dasan, DZS, PPC Broadband, Taicang, Nokia and Netlink. Assigning the responses to the individual BGP-announced prefixes, we see again that 39% of the BGP prefixes do not respond at all, a number similar to the one in M1. It also shows that similar to results from related work, inactive address space is often not routed correctly, leading to routing loops in over 62.9% of prefixes that return error messages [22, 23].

Message Types. Table 6 outlines the contribution of the individual error message types to the classification of our network activity scans. In M1 - core we see a higher share of null routing through *RR* (33.3%) and $AU_{RTT<1s}$ (13.1%) while for M2 -periphery we see a higher share (32.8%) of routing loops (*TX*) and active networks indicated by $AU_{RTT>1s}$ (26%). For target networks classified as ambiguous that could both be active or inactive *NR* contributes the most with 20.3% in M1 and 13.6% in M2.

Network Activity. We found active networks to account for 1.7% of the IPv6 Internet at /48 granularity. We find a higher share (12%) of active networks for /64 periphery networks. This 12% of active networks are divided across 34,924 – equal to 61% – of responsive /48 prefixes. The respective error messages indicate that the request was forwarded and triggered Neighbor Discovery. This makes them a priority target for further reconnaissance efforts. Narrowing down the search space to these networks is however tricky as we cannot guarantee that these networks are the only active networks inside the target network range. Active networks with filters might discard our requests and remain silent, i.e., our results have to be considered to represent a lower bound for the number of active networks. We also find 22% of prefixes return error messages for inactive networks only. While these networks can be excluded for host discovery, periphery and subnet discovery does not depend on the returned response type [7]. However, we find that periphery discovery based on error messages is not a solution for every network [22]. For around 38% to 39% of prefixes in the IPv6 Hitlist Service, M1 and M2 do not return error messages. For the remainder of the paper, we focus on networks that return error messages. As response behavior varies between different network equipment vendors, this necessitates the identification of the router type used within a network.

5 Router Classification

Measuring ICMPv6 rate limiting behavior, we found varying behavior among vendors in our lab (§) that could be validated against SNMPv3 labels from routers in the IPv6 Internet (§). Our approach extends router classification to IPv6 routers that are not SNMPv3 responsive. In a final measurement (§), we fingerprinted routers on the Internet. For periphery routers, fingerprinting vendors is

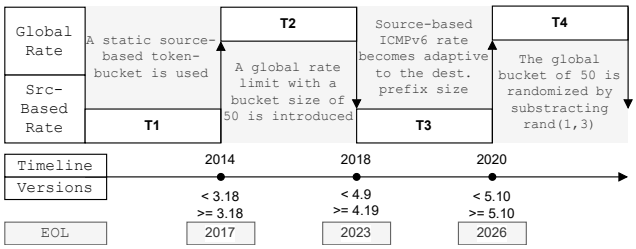


Figure 8: ICMPv6 rate-limiting behavior for different Linuxs kernel versions.

– in comparison to core routers – limited as they mainly show Linux default behavior. Yet, we are able to estimate Kernel versions, detecting many routers that have reached end of life.

5.1 Router Defaults

We rely again on our GNS3 setup, but instead of a single request we send ICMPv6 Echo requests at 200 pps for a time period of ten seconds to (I) unassigned addresses (see IP2 in Figure 1) in active network A triggering $AU_{RTT>1s}$ at the RUT, (II) addresses in inactive network B (IP 3 in Figure 1) triggering *NR*, or (III) with Hop Limits triggering *TX* as a response (as in Scenario (S6) in Section 4.1). The requests contain ascending sequence numbers as a payload and allow to check which requests remain unanswered. The responses are shaped by ICMPv6 rate-limiting behavior, typically implemented with a token bucket algorithm [10]; we infer its parameters as follows:

- *Bucket size:* Missing packets pinpoint the bucket's depletion. We determine the first missing response; its sequence number is equivalent to the bucket size.
- *Refill size:* The refill size is equivalent to the number of replies between two successive depletions. We count this number for all successive depletions and take the median of the collected values as refill size.
- *Refill interval:* The refill interval is equivalent to the time between two refills. Therefore, we infer the time spans between successive responses, remove the ones reflecting our measurement rate (5ms), and take the median. This value represents the pause between two bursts. Combining it with the duration of the previous burst, we infer the refill interval.
- *Number of error messages*: As a simplistic indicator, including the other parameters of a router's rate-limiting behavior, we count the total number of error messages received in a time span of ten seconds.

We conduct this measurement from a single source, and repeat it with two source addresses to see whether rate limits are configured globally or per source address.

Vendor Defaults. Table 8 shows our results in detail. Seven routers apply rate limiting per source address, another six only apply a global limit, and two do not limit ICMPv6 error messages at all. We observe differences among vendors – though not among all (e.g., the Linux-based Mikrotik, OpenWRT, VyOS, and Aruba) – but also between routers/versions of the same vendors (e.g., Cisco XRV9000 and Cisco IOS 15.9), and in some cases, even between the different error message types from the same routers (e.g., Juniper and Huawei). While Linux-based routers show a token-bucket rate limit algorithm, FreeBSD ones show generic rate limits, where the refill size equals the bucket size. Another peculiarity has been observed for Huawei; the bucket size is randomly chosen between 100 and 200. This appears to be a countermeasure against idle scanning or exploiting routers as remote vantage points for scanning [4, 28].

Prefix Size	Refill Interval (ms)			# Error Messages
Kernel HZ ->	100	250	1000	
0	60	60	62	165-167
1-32	120	124	125	85-86
33-64	248	248	250	45-46
65-96	500	500	500	25-26
97-128	1,000	1,000	1,000	15-16

Table 7: Since kernel 4.19, the refill interval depends on the IPv6 prefix length and the kernel tick rate.

Linux Kernel Defaults. For Mikrotik routers relying on the Linux kernel rate limiting, we observed a difference in behavior between version 6.48 and 7.7, and consequently investigated the limiting behavior of Linux kernels using Debian live images in more detail. This led to detecting a change in the peer-based rate limiting behavior (for completeness we list the values for all tested kernel versions in Table 12 in the Appendix), which is congruent with the change between Mikrotik version 6 and 7.

Linux changed its peer-based rate-limiting behavior between kernel 4.9 and 4.19 (between 2016 and 2018). Before the change, the rate limits behaved static; now, it is dependent on the router's assigned prefix size, see Table 7. Figure 8 shows the evolution of ICMPv6 rate limiting in the Linux kernel over time. The code for the prefix-based rate limit exists since kernel 2.1.111, but was not effective until 4.9/19. In this paper we focus on measuring the peer-based rate limit. Measuring the global rate limits is more invasive as it requires to bypass the peer-based rate limit by measuring with multiple source addresses in parallel. Pan et al. already showed that the global rate limit can be measured this way [28]. Also, hosts with global rate limits were exploited as remote vantage points for scanning [4, 28]. This led to a new behavior of the Linux kernel, similar to Huawei routers, by subtracting a random integer of up to 3 from the default bucket size of 50. Both the introduction and the randomization of the global rate limit provide additional steps to fingerprint the Linux kernel versions. In this paper we aim to separate routers, relying on the Linux kernel rate limiting, based on the peer-based rate limit into T2 or before and T3 and after.

5.2 Rate Limits in the Internet

As a next step, we aim to verify if we see the same fingerprints on the Internet. To do this, we use a dataset with responses to unauthenticated, unsolicited SNMPv3 requests revealing vendor information for 476K IPv6 addresses [2]. We chose to elicit *TX* messages at routers, as these are mandatory according to RFC4433. However, we need a suitable combination of destination address and Hop Limit to trigger *TX* at a certain router. We checked whether the SNMPv3 dataset addresses used as ground truth were also present in our M1 dataset collected by yarrp (see Section 4.3), enabling us to set the destination address and Hop Limit in our requests

	Router OS	iTTL	Delay	Bucket Size			Refill Interval (σ)			Refill Size			# Error Messages			Per
		All	AU	TX	NR	AU	TX	NR	AU	TX	NR	AU	TX	NR	AU	Src
Diff AU/NR/TX	CiscoXRV9000	64	18	10	10	10	1,000	1,000	1,000	1	1	1	19	19	0★	
	CiscoIOS 15.9	64	3	10	10	10	~100	~100	3,800★	1	1	10	~105	~105	22★	
	CiscoCSR1000 17.03	64	3	10	10	10	~100	~100	3,000★	1	1	10	~105	~105	22★	
	Juniper 17.1	64	2	52	12	12	~1,000	10,000	10,000	52	12	12	~520◇	12	12	
	HPE VSR1000	64	3	∞	∞	★	∞	∞	★	∞	∞	★	∞	∞	★	
	Huawei NE40	64	3	100-200	8	/	1,000	1,000	/	100	8	/	1,000-1,100	88	/	
No diff for AU/NR/TX	Arista 4.28	64	3		∞			∞			∞			∞		
	VyOS 1.3	64	3		6			250*			1			45*		✓
	Mikrotik 6.48	64,255	3		6			1000			1			15		✓
	Mikrotik 7.7	64	3		6			250*			1			45*		✓
	OpenWRT 19.07	64	3		6			250*			1			45*		✓
	OpenWRT 21.02	64	3		6			250*			1			45*		✓
	ArubaOS 10.09	64	3		6			250*			1			45*		✓
	Fortigate 7.2.0	255	3		6			10			1			1000		✓
	PfSense 2.6.0	64	3		100			1000			100			1000		

~ ... Refill interval is less stable / ... The response type is not returned by the RUT. ★ ...Affected by the Neighbor Discovery Process. ∗ ... /48 destination prefix; for other prefix sizes see Table 7 ∞ ... RUT is either not rate-limited or > $scanrate$ (tested up to 10K pps). ◇ ... Juniper's Neighbor Discovery for hop limit 0 packets causes a 2-second delay also for TX.

Kernels: Linux, Wind River Linux and FreeBSD.

Table 8: ICMPv6 rate limiting behavior of routers observed in GNS3 laboratory setup. Parameters vary among vendors, versions, and sometimes even for message types.

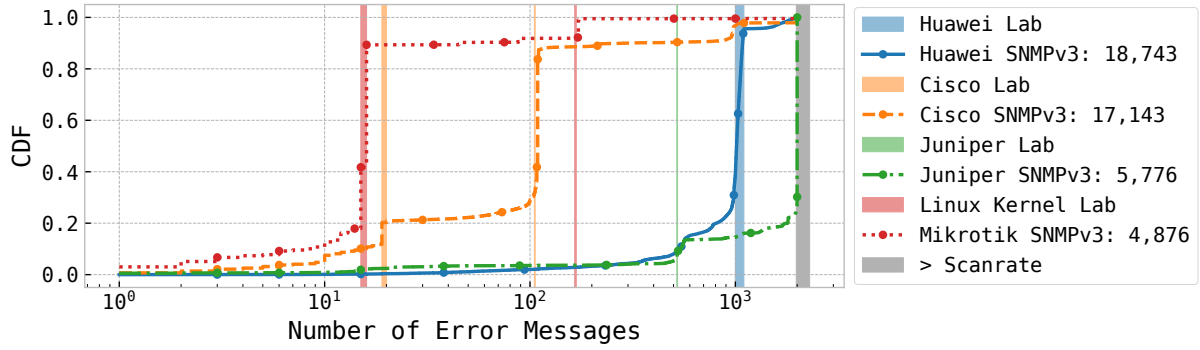


Figure 9: No. of error messages in 10 s for SNMPv3 routers matches the laboratory results (marked vertically).

accordingly. As before, we sent requests at a rate of 200 pps over a time period of 10 s. This way, we could validate the behavior of 50,952 IPv6 addresses against SNMPv3 labels.

Classification. To match router rate limits to recorded vendor fingerprints, we rely on a more elaborate approach than comparing the number of received error messages. In the first step, classification is based on one-dimensional vectors, each element describing the number of received ICMPv6 error messages per second. If the distance between a router's behavior and the collected labels lies within a predefined threshold, we assign the respective label. The threshold is adaptive based on the total number of error messages received ranging from 10 (<100 error messages) to 100 (<2,000 error messages). Only if labels from different routers overlap, we compare the token bucket algorithm's parameters of refill interval and refill size in a second step. From the fingerprints that match all these, the vendor fingerprint with the lowest distance from the one-dimensional vector is selected. If this is not the case, we classify it as *New Pattern*. Some routers in the Internet measurements appear to apply a dual double token bucket algorithm for rate limiting, including two refill intervals and sizes. With the refill interval being the median of refill intervals, we rely on the skewness measure $abs(1 - mean/median) > 0.5$ to check for a second refill interval and label these routers accordingly.

Comparison with Virtual Laboratory. Figure 9 compares the total number of ICMPv6 error messages returned by routers recorded in the lab compared to the number of error messages collected for SNMPv3-labeled routers in the IPv6 Internet. Each vertical line represents the number of error messages we saw for the specific vendor in our laboratory. We see overlapping behavior with our results from the virtual laboratory setup: Applying our classification, the rate-limit patterns observed in our laboratory account for 70% of the Cisco, 51% of the Huawei, and 91% of the Mikrotik routers in our Internet measurement. In contrast, the Juniper router (Junos 17.1) from the laboratory only accounts for 5% of the Juniper-labeled routers, and HPE label for 7%. This is, however, not surprising as our laboratory setup contains only a limited set of routers. Juniper routers' rate-limiting implementations seem to vary more across different versions than for other vendors [21], and we found that 82% of Juniper-labeled routers are rate-limited above our scanrate of 200 pps. However, we do not conduct scans with higher pps due to ethical considerations.

Additional Fingerprints. SNMPv3 labels allow to extend our fingerprints from the laboratory setup. We relied on clustering with varying k-values from 2 to 10 [19] on the one-dimensional vector to detect rate-limiting patterns for each vendor. We used the elbow method to detect the number of different error message

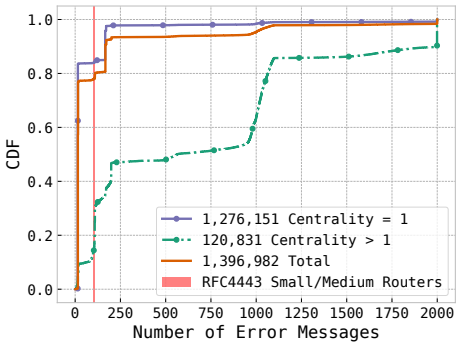


Figure 10: Routers on multiple paths (centrality>1) have higher rate limits than ones on one path (centrality=1).

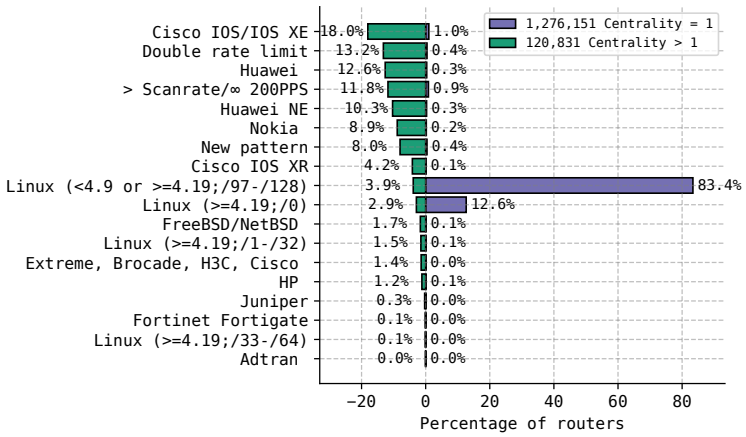


Figure 11: Router classification. Core routers (centrality > 1) are diverse, periphery routers (centrality=1) mostly Linux-based.

rates for each vendor. We found that vendors show a maximum of four different rate-limiting patterns. Based on the patterns, we manually inferred additional fingerprints for Nokia (Number of error messages over 10 seconds=100-200), HP (NR10=5), Adtran (NR10=42), and Huawei (NR10=1000-1100,550) routers. In addition to NR10 we also extract the bucket size, refill intervals and refill sizes.

Multi Vendor Fingerprints. We also detected overlapping behavior. The SNMPv3-label H3C reveals the same fingerprint as for Cisco IOS and IOS XE, but there remains a subtle difference – H3C is more likely to show 11 initial responses – facilitating their separation with improved classification. For sure, we will never be able to differentiate vendors if all the rate-limiting parameters are identical. This is the case for Extreme, Brocade, H3C, and Cisco sharing a common fingerprint of a random bucket size between 10 and 20, a refill interval of 100ms and a refill size of 10.

5.3 Router Classification on the Internet

Finally, we conducted a large-scale study of router types on the Internet. From the M1 dataset (see Section 4.3), we extracted all addresses responding with a *TX* as we consider them to be router addresses. We sent requests at a constant rate of 200 pps to the destination addresses with the respective router en route and set the Hop Limit accordingly; both were also inferred from the tracerouting data set. This way, we could infer a rate limit for 1,396,982 IP addresses.

Results. Figure 10 shows the total number of returned *TX* messages over a period of 10 seconds for all these routers and dominant behavior at 15 packets. We separate routers into two groups: those with $centrality = 1$, assumed to be on the Internet's periphery (appearing on a single path), and those with $centrality > 1$, located closer to the Internet's core (appearing on multiple paths). Figure 10 shows distinct results for these two groups and suggests that they consist of different router types.

Figure 11 shows our results: Core routers come from different vendors, including Cisco (multiple fingerprints combined: 22.2%), Huawei (multiple fingerprints combined: 22.9%), and Nokia (8.9%). Meanwhile, 83.3% of the periphery routers either rely on Linux kernel version 4.9 (or even older) or a current kernel version with an assigned prefix length of /97-/128. However, as such long prefix lengths are not very common on the Internet [27], this implies that up to 1,066,856 routers in our measurement reached the end of life in January 2023. Only 12.6% of the periphery routers run newer kernel versions.

Comparison with SNMPv3 and LFP. Our approach of ICMPv6 error message based router classification extends SNMPv3 router labels similarly to LFP for IPv4 [3] especially since only 476,000 IPv6 routers were found to be SNMPv3 responsive [2]. Unlike unsolicited SNMPv3 responses, rate limiting is unlikely to be disabled in the future, and routers must return *TX* error messages per RFC4443. However, we also faced drawbacks similar to those of LFP. Our classification does not fully cover every router vendor. Juniper is underrepresented in our dataset, as most Juniper routers on the Internet are rate-limited above our scan rate. In contrast to SNMPv3 engineIDs that uniquely identify router vendors, our classification also includes multi-vendor labels, and we cannot distinguish vendors that rely on the Linux kernel default. Notably, Linux kernel version fingerprinting is a new application not previously attempted.

6 Related Work

ICMP Error Messages. For both protocols, IPv4 and IPv6, the most prominent use of ICMP error messages has been topology discovery (tracerouting) [6, 9, 16] and analysis of error message types mainly focused on routing loop detection [23, 31]. For the IPv4 Internet, Bano et al. [5] and Rüth et al. [31] investigated ICMPv4 error messages that had been received as a byproduct of ZMap-based measurements. Depending on the protocol for probing, 0.7% (ICMPv4), 9.0% (TCP), resp. 81.3% (UDP) of all responses were ICMPv4 error messages [5]. In IPv6, the share of error messages tends to be higher: For addresses close to responsive addresses, ICMPv6 error messages have a share of 60% and this number rapidly increases to 99% for addresses with more random bits, see Table 10 in the Appendix.

IPv6 Reconnaissance. Initial approaches of IPv6 reconnaissance relied on the collection of addresses from public sources [12, 15] or algorithms generating target addresses based on a data set of known addresses [7, 13, 24, 35]. The most successful passive approach by Rye et al. relied on NTP servers to collect active addresses from 7.2M /48s [32]. Rye and Beverly performed active measurements by intentionally triggering error messages to collect 64 million IPv6 addresses at the Internet periphery [33], and Li et al. [22] scanned 15 IPv6 ISP network ranges to trigger *Destination Unreachable* messages at periphery routers collecting 52 million IPv6 addresses. Both approaches triggered ICMP error messages and extracted source addresses, but did not further classify the periphery as we did by interpreting distinct error types and timings. Our measurements also point out a limitation. Across all of our measurements, we find approx. 38% of IPv6 prefixes that do not return ICMPv6 error messages, rendering all above-mentioned approaches useless for these networks.

Router Classification. Vanauble et al. [36] derived router signatures from the initial *TTL* (iTTL) values of *Time Exceeded* resp. *Echo Reply* messages; this way, the authors could infer a router's vendor through remote measurements. Holland et al. [20] combined banner grabbing (SSH, SNMP, telnet) and active probing of routers to train an automated classifier. While the classifier was trained on diverse features, the iTTL remained the most distinctive feature to distinguish vendors. Meanwhile, iTTLs are harmonized among the majority of vendors (see Table 8 and [8]). Consequently, a new methodology is necessary to distinguish vendors. The most recent approach for router classification found 476,000 routers that reply to unauthenticated SNMPv3 requests with replies including vendor-specific engineIDs [2]. Albakour et al. extended their approach for non-SNMPv3 responsive IPv4 routers by including protocol header specific information such as the IPID [3]. However, this methodology cannot be applied for IPv6 due to the required fields missing in IPv6 and harmonized iTTL values. Our methodology complements [2] for non-SNMPv3 responsive IPv6 routers by developing a classification method based on ICMPv6 error message rate limiting behavior of routers.

ICMPv6 Rate Limits. Ravaioli et al. classified ICMP rate limiting in different categories such as on-off behavior or generically rate-limited routers. They also explored the effects of increasing probing rates and found that higher probing rates lead to more irregular on-off behavior of routers [29]. While we keep the probing rates low, we also noticed irregular behavior, but assume it is triggered by other entities impacting the routers global rate limit. Previous work exploited routers' rate-limiting behavior for purposes other than router vendor classification. Vermeulen et al. [37] conducted alias resolution, i.e., identifying IP addresses belonging to the same router. As aliased addresses are subject to the same rate limit, probing them simultaneously triggers rate limiting and distinct loss patterns. Security-wise, Pan et al. [28] showed how to exploit remote routers' global error message rate limits to use them as vantage points for network scans. The same concept is used by Albrecht et al. [4] to perform UDP idle scans through remote routers. To the best of our knowledge, we are first to explore ICMPv6 error message rate limiting in such detail, as well as its exploitability for router classification.

7 Discussion

Error Message Classification. Our classification of error message types does not provide a guarantee that every router behaves accordingly. While we used BValue steps to quantify the response behavior of active and inactive IPv6 networks, the classification could be impacted by either the correctness of inferred active and inactive networks or routers that misuse the respective error message type. However, we cannot distinguish between those two cases. Most of the classified IPv6 networks behave accordingly and for 95% of networks identified as active we receive $AU_{RTT>1s}$.

Prefix Boundary Precision. We used BValue steps to separate active from inactive networks, but the precision of network border detection could be further improved. First, the randomization of the address bits might results in an overlap with the original hitlist address. The probability for the first bit to overlap is 50%, the probability for the eight bits to overlap 0.4%. A pseudo-random address flipping the first bit of the BValue would increase precision as the generated address would not overlap. However, we generate five addresses per BValue step indicating that on average 2.5 addresses deviate already in the first bit. Second, the step width impacts prefix boundary precision. We opted for eight bits as a trade-off to cover major prefix boundaries. A change at a non-eight-bit prefix boundary, e.g., /60, is misclassified as a /56. However, the occurrences appear to be limited in practice, Table 11 in Appendix C shows that we received only one response type in 97% of BValues, suggesting that changes within a BValue step are a minor phenomena.

Unique Vendor Identification. Our approach allows us to determine a router's vendor and/or operating systems based on the ICMPv6 rate limiting behavior, turning a protection mechanism into a privacy leak. There is room for improvement: First, we were limited to the fingerprints collected in our laboratory setup and the SNMPv3 data set. These fingerprints do not cover the whole Internet population as our Internet measurement revealed unknown patterns. Second, certain fingerprints overlap. While this does not allow unambiguous classification, it still allows to narrow the field down to a few vendors. This information could then be combined with other approaches. For example, one could investigate the rate limiting behavior of different error message types and compare with Table 9.

Old Kernels used in Periphery Routers. In our measurements, we identified 1.2M routers operating a Linux version of 2018 or older that have already reached end of life. This does not mean that they are exploitable, but in case of a vulnerability no updates will be made available for this significant share of periphery routers.

Countermeasures. Strict adherence to RFC4443 [10] only facilitates network activity classification by making router behavior more consistent. For router classification, the consequences are the opposite. More congruent ICMPv6 rate limiting (e.g. more specific values could be proposed by the RFC) would hinder classification of vendor and operating system. Disabling ICMPv6 error messaging mitigates both, Network Activity Classification and Router Classification, and is also compliant with the specification. In fact, this is the case for 38% of the investigated networks. For these networks,

our approaches were not successful, but also previous work collecting addresses from ICMPv6 error messages [22] would fail for such networks. In addition, disabling hinders network diagnosis by administrators. Removal of the rate limits would, according to the specification, put the routers at risk of denial-of-service attacks; still, some of investigated routers appear to operate properly without such limits.

8 Conclusion

In this paper, we developed two new measurement methods, exploiting ICMPv6 error messages beyond the mere extraction of source addresses, to to gain more insight into remote networks. For each method, we first established our hypothesis in a virtual laboratory setup, then validated the results via measurements involving ground truth, and, finally, conducted exemplary Internet measurements. Our work is summarized as follows: (I) Routers return ICMPv6 messages in non-specified ways with RFC4443 which negatively impacts the messages' diagnostic value. Nevertheless, we were able to classify them regarding the activity status of the remote network by combining ICMPv6 message type, subcode, and timing behavior. Our method is able to guide scanning efforts towards active networks where responsive IPv6 addresses reside. (II) ICMPv6 rate limits protect routers against denial-of-service, but the implementations vary significantly among different router vendors and operating systems. We use them for router classification and, by measuring 1.4 million routers on the Internet, we discovered different populations for core and periphery routers; the latter are primarily Linux-based (96.0%). Most prevalent (83.3%) is Linux kernel version 4.9 and older which have reached end of life in January 2023.

Acknowledgements

This material is based upon work partially supported by (1) the Christian-Doppler-Laboratory for Security and Quality Improvement in the Production System Lifecycle; the financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development and the Christian Doppler Research Association are gratefully acknowledged; (2) SBA Research (SBA-K1), a COMET Centre within the framework of COMET – Competence Centers for Excellent Technologies Programme and funded by BMK, BMDW, and the province of Vienna.(3) Project DynAISEC FO999887504 funded by the Program "ICT of the Future" – an initiative of the Austrian Ministry of Climate Action, Environment, Energy, Mobility, Innovation and Technology, (4) the Austrian Science Fund (FWF) (SFB SPyCoDe F85).

References

[1] David Adrian, Zakir Durumeric, Gulshan Singh, and J Alex Halderman. 2014. Zippier zmap: internet-wide scanning at 10 gbps. In *8th USENIX Workshop on Offensive Technologies (WOOT 14)*.

[2] Taha Albakour, Oliver Gasser, Robert Beverly, and Georgios Smaragdakis. 2021. Third time's not a charm: Exploiting SNMPv3 for router fingerprinting. In *Proceedings of the 21st ACM Internet Measurement Conference*. 150–164.

[3] Taha Albakour, Oliver Gasser, Robert Beverly, and Georgios Smaragdakis. 2023. Illuminating Router Vendor Diversity Within Providers and Along Network Paths. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. 89–103.

[4] Martin Albrecht. 2019. UDP idle scanning. Retrieved Nov 20, 2023 from https://martinralbrecht.wordpress.com/2019/10/25/udp-idle-scanning.

[5] Shehar Bano, Philipp Richter, Mobin Javed, Srikanth Sundaresan, Zakir Durumeric, Steven J Murdoch, Richard Mortier, and Vern Paxson. 2018. Scanning the internet for liveness. *ACM SIGCOMM Computer Communication Review* 48, 2, 2–9.

[6] Robert Beverly. 2016. Yarrp'ing the Internet: Randomized High-Speed Active Topology Discovery. In *Proceedings of the 2016 Internet Measurement Conference*. ACM, 413–420.

[7] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P Rohrer. 2018. In the IP of the Beholder: Strategies for Active IPv6 Topology Discovery. In *Proceedings of the Internet Measurement Conference 2018*. ACM, 308–321.

[8] Wesley G Bofman and Fernando Maniego. 2019. *Fingerprinting IPv4 and IPv6 routers using ICMP*. Ph. D. Dissertation. Monterey, CA; Naval Postgraduate School.

[9] Caida. 2007. Archipelago (Ark) Measurement Infrastructure. http://www.caida.org/projects/ark/

[10] A. Conta, S. Deering, and M. Gupta (Ed.). 2006. Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification. RFC 4443 (Internet Standard). https://doi.org/10.17487/RFC4443 Updated by RFC 4884.

[11] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide scanning and its security applications. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*. 605–620.

[12] Tobias Fiebig, Kevin Borgolte, Shuang Hao, Christopher Kruegel, and Giovanni Vigna. 2017. Something from Nothing (There): Collecting Global IPv6 Datasets from DNS. In *Passive and Active Measurement*, Mohamed Ali Kaafar, Steve Uhlig, and Johanna Amann (Eds.). Springer International Publishing, Cham, 30–43.

[13] Pawel Foremski, David Plonka, and Arthur Berger. 2016. Entropy/IP: Uncovering Structure in IPv6 Addresses. In *Proceedings of the 2016 Internet Measurement Conference* (Santa Monica, California, USA) *(IMC '16)*. ACM, New York, NY, USA, 167–181. https://doi.org/10.1145/2987443.2987445

[14] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczynski, Stephen D. Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the Expanse: Understanding and Unbiasing IPv6 Hitlists. In *Proceedings of the 2018 Internet Measurement Conference* (Boston, MA, USA). ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3278532.3278564

[15] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. 2016. Scanning the IPv6 Internet: Towards a Comprehensive Hitlist. In *Proc. of 8th Int. Workshop on Traffic Monitoring and Analysis*. Louvain-la-Neuve, Belgium.

[16] Eric W Gaston. 2017. *High-frequency mapping of the IPv6 Internet using Yarrp*. Technical Report. Naval Postgraduate School Monterey United States.

[17] Fernando Gont. 2013. scan6 - An IPv6 host scanner. http://manpages.ubuntu.com/manpages/cosmic/man1/scan6.1.html

[18] F. Gont and T. Chown. 2016. Network Reconnaissance in IPv6 Networks. RFC 7707 (Informational). https://doi.org/10.17487/RFC7707

[19] Allan Grønlund, Kasper Green Larsen, Alexander Mathiasen, Jesper Sindahl Nielsen, Stefan Schneider, and Mingzhou Song. 2017. Fast exact k-means, k-medians and Bregman divergence clustering in 1D. *arXiv preprint arXiv:1701.07204* (2017).

[20] Jordan Holland, Ross Teixeria, Paul Schmitt, Kevin Borgolte, Jennifer Rexford, Nick Feamster, and Jonathan Mayer. 2020. Classifying Network Vendors at Internet scale. *arXiv preprint arXiv:2006.13086* (2020).

[21] Juniper. 2022. ICMP Features. Retrieved Mai 25, 2023 from https://www.juniper.net/documentation/us/en/software/junos/transport-ip/topics/topic-map/icmp.html.

[22] Xiang Li, Baojun Liu, Xiaofeng Zheng, Haixin Duan, Qi Li, and Youjun Huang. 2021. Fast IPv6 Network Periphery Discovery and Security Implications. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 88–100.

[23] Markus Maier and Johanna Ullrich. 2023. In the loop: A measurement study of persistent routing loops on the IPv4/IPv6 Internet. *Computer Networks* 221 (2023), 109500.

[24] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. Target generation for Internet-wide IPv6 scanning. In *Proceedings of the 2017 Internet Measurement Conference*. ACM, 242–253.

[25] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. 2007. Neighbor Discovery for IP version 6 (IPv6). RFC 4861 (Draft Standard). https://doi.org/10.17487/RFC4861 Updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028, 8319, 8425.

[26] "RIPE NCC". 2018. Routing Information Service (RIS). https://www.ripe.net/analyse/internet-measurements/routing-information-service-ris

[27] Ramakrishna Padmanabhan, John P Rula, Philipp Richter, Stephen D Strowes, and Alberto Dainotti. 2020. DynamIPs: Analyzing address assignment practices in IPv4 and IPv6. In *Proceedings of the 16th international conference on emerging networking experiments and technologies*. 55–70.

[28] Long Pan, Jiahai Yang, Lin He, Zhiliang Wang, Leyao Nie, Guanglei Song, and Yaozhong Liu. 2023. Your Router is My Prober: Measuring IPv6 Networks via ICMP Rate Limiting Side Channels. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society. https://www.ndss-symposium.org/ndss-paper/your-router-is-my-prober-measuring-ipv6-networks-via-icmp-rate-limiting-side-channels/

[29] R. Ravaioli, G. Urvoy-Keller, and C. Barakat. 2015. Characterizing ICMP rate limitation on routers. In *2015 IEEE International Conference on Communications (ICC)*. 6043–6049. https://doi.org/10.1109/ICC.2015.7249285
[30] RIPE. 2020. *IPv6 Address Allocation and Assignment Policy*. Technical Report. https://www.ripe.net/publications/docs/ripe-738/
[31] Jan Rüth, Torsten Zimmermann, and Oliver Hohlfeld. 2019. Hidden Treasures – Recycling Large-Scale Internet Measurements to Study the Internet's Control Plane. In *Passive and Active Measurement*, David Choffnes and Marinho Barcellos (Eds.). Springer International Publishing, Cham, 51–67.
[32] Erik Rye and Dave Levin. 2023. Ipv6 hitlists at scale: Be careful what you wish for. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 904–916.
[33] Erik C. Rye and Robert Beverly. 2020. Discovering the IPv6 Network Periphery. In *Passive and Active Measurement*, Anna Sperotto, Alberto Dainotti, and Burkhard Stiller (Eds.). Springer International Publishing, Cham, 3–18.
[34] Lion Steger, Liming Kuang, Johannes Zirngibl, Georg Carle, and Oliver Gasser. 2023. Target Acquired? Evaluating Target Generation Algorithms for IPv6. In *Proceedings of the Network Traffic Measurement and Analysis Conference (TMA)* (Naples, Italy).
[35] J. Ullrich, P. Kieseberg, K. Krombholz, and E. Weippl. 2015. On Reconnaissance with IPv6: A Pattern-Based Scanning Approach. In *2015 10th International Conference on Availability, Reliability and Security*. 186–192. https://doi.org/10.1109/ARES.2015.48
[36] Yves Vanaubel, Jean-Jacques Pansiot, Pascal Mérindol, and Benoit Donnet. 2013. Network fingerprinting: TTL-based router signatures. In *Proceedings of the 2013 conference on Internet measurement conference*. 369–376.
[37] Kevin Vermeulen, Burim Ljuma, Vamsi Addanki, Matthieu Gouel, Olivier Fourmaux, Timur Friedman, and Reza Rejaie. 2020. Alias resolution based on ICMP rate limiting. In *Passive and Active Measurement: 21st International Conference, PAM 2020, Eugene, Oregon, USA, March 30–31, 2020, Proceedings 21*. Springer, 231–248.
[38] Johannes Zirngibl, Lion Steger, Patrick Sattler, Oliver Gasser, and Georg Carle. 2022. Rusty clusters? dusting an ipv6 research foundation. In *Proceedings of the 22nd ACM Internet Measurement Conference*. 395–409.

A Ethics

In our measurements, we followed the rules of good Internet citizenship [11]. We received a single request for opt-out and acted accordingly. For both measurements, we only sent requests which are typical for Internet traffic. Overall, for network activity classification, the number of requests has been moderate. For validation with BValue Steps, we send 62 requests to a /32 prefix; in our Internet measurements, we send a single request per /48 and /64 prefix respectively. The targets were randomized to prevent the overloading of individual routers. For router classification, we intentionally triggered ICMPv6 rate limiting behavior of routers. Measurements exploiting ICMP rate limiting have been conducted before and related work found no significant performance degrades on routers [28, 37]. Our measurements do not impact the forwarding abilities of routers, it could only deny the origination of ICMPv6 error messages for routers with global rate limits. For example, a potential consequence might be that the affected routers would not reply to tracerouting efforts of other Internet hosts during our measurement period. We found such global rate limited routers to be present in the Internet-core. Periphery routers apply peer-based rate limits and only refrain from returning further ICMPv6 error messages to our measurement host. To minimize our impact on the routers, we limited our measurements to 200 pps and a maximum measurement period of 10 s, resulting in a total of 2,000 requests. Thereby, we did our best to minimize the impact on the router: Packet sizes were kept small to minimize bandwidth consumption; as payload, they only included a request ID and the sent timestamp. Beyond that, we decided to elicit *TX* like in tracerouting campaigns (instead of *AU*) to prevent stateful and, thus, more resource-intensive address resolution. It has to be highlighted that IPv6 rate limits are a protection measure. They prevent routers from sending too many ICMPv6 error messages in order to maintain their main functionality of packet forwarding.

B Vendor Coverage

We provide details on vendor coverage for tested routers and state how the router images relate to each vendor. The individual vendors are listed in Table 9. A lot of router operating systems nowadays rely on the Linux kernel. However, operating systems are still customized based on the vendor's needs. We will take a closer look at each RUT. We test three different images for ***Cisco***. Cisco IOS (Internet Operating System) version 15.9 (2019) is the original monolithic operating system developed by Cisco. The Cisco CSR1000v represents a virtual router series that has been designed for cloud services. It runs a subset of Cisco IOS XE, which supports the same commands as IOS as it runs IOS as a separate process, but is built on Linux. In contrast, IOS XR has a completely different codebase. The XRv 9000 Version 7.2.1 is a virtualized router implementing the feature set of IOS XR. IOS XR is originally based on a microkernel provided by QNX, but has changed to Wind River Linux since version 6. With these images, our lab setup covers Cisco's main networking software for routers. The remaining Cisco NX-OS is the operating system for a series of Cisco switches. We found different ICMPv6 error message implementations for all three router OSes with IOS and IOS-XE being more similar. IOS-XR shows a more diverse behavior with unique Neighbor Discovery timings and response type usage in filtering scenarios. ***Juniper*** runs Junos as a single operating system across its router and switches. The image in our lab is Junos VMx Version 17.1. In contrast to the Cisco operating systems it is based on FreeBSD. It is the second appliance

	Protocols	S1 Active Network	S2 Inactive Network	S3 Active Network with ACL	S4 Inactive Network with ACL	S5 Null Route	S6 Routing Loop
Cisco IOS XR (XRv 9000 7.2.1)	All	AU [18s]	NR	∅	AP	∅	TX
Cisco IOS (15.9 M3)	All	AU [3s]	NR	AP/FP•	AP/FP•	RR	TX
Cisco IOS-XE (CSR1000v17)	All	AU [3s]	NR	AP	AP	RR	TX
Juniper Junos (VMx 17.1)	All	AU [2s]	NR	AP	AP	AU/∅•	TX
HPE (VSR1000)*	All	AU [3s]	NR	AP	AP	∅	TX
Huawei (NE40)	All	∅	NR	-	-	∅	TX
Arista (vEOS 4.28)	All	AU [3s]	NR	-	-	∅	TX
VyOs (1.3)	All	AU [3s]	NR	PU	NR★	∅	TX
Mikrotik (6.48/7.7)	All	AU [3s]	NR	NR	NR★	NR/AP/	TX
OpenWRT (19.07/21.02)	ICMP/U	AU [3s]	FP	PU	FP★	NR/AP/	TX
	TCP	AU [3s]	FP	RST	FP★	NR/AP/	TX
ArubaOS (OS-CX)	All	AU [3s]	NR	∅	∅	AP	TX
Fortigate (7.2.0)*	All	AU [3s]	NR	∅	∅	∅	TX
PfSense (2.6.0)*	ICMP	AU [3s]	NR	∅	∅	-	TX
	TCP	AU [3s]	NR	∅/RST•	∅/RST•	-	TX
	UDP	AU [3s]	NR	∅/PU•	∅/PU•	-	TX

• Multiple ACL/route options. [] Minimum delay. - Not supported. ★ ACL on forward chain. Error messages indicating active, ambiguous and inactive networks.

Table 9: ICMPv6 error message behavior of routers as observed in GNS3 laboratory setup. For router configurations see https://github.com/sbaresearch/router-lab.

BValues	active	ambiguous				inactive			ER	Responsive	Targets
	$AU_{RTT>1s}$	NR	AP	FP	PU	$AU_{RTT<1s}$	RR	TX			
B127	49.3%	3.1%	1.4%	0.1%	0.0%	2.2%	0.9%	2.8%	40.2%	20,319	47,922
B120	71.3%	3.7%	1.1%	0.1%	0.0%	3.9%	1.8%	7.4%	11.1%	28,693	47,922
B112	78.3%	4.2%	1.3%	0.1%	0.0%	4.5%	2.2%	8.7%	0.7%	25,447	47,922
B64	77.4%	4.4%	1.3%	0.1%	0.0%	4.7%	2.4%	9.6%	0.1%	24,981	47,879
B56	29.2%	11.8%	1.7%	0.1%	0.0%	16.1%	9.1%	31.8%	0.1%	19,102	46,847
B48	24.5%	12.8%	1.6%	0.1%	0.0%	17.6%	9.9%	33.4%	0.2%	18,176	46,743
B40	13.3%	16.3%	1.7%	0.0%	0.0%	14.8%	15.6%	38.1%	0.1%	6,246	19,585
B32	12.2%	15.9%	0.9%	0.1%	0.0%	16.4%	18.5%	35.9%	0.1%	3,670	10,669

Table 10: Selected BValue steps for 47,922 IPv6 prefixes showing the transition from active to inactive error message types. The two rightmost columns show the number of responsive targets vs. the total ones.

No. of message types		Protocol	No. of responses 1	2	3	4	5
	1	ICMPV6	4.0%	3.0%	4.0%	6.0%	80.0%
		TCP	4.0%	4.0%	5.0%	7.0%	79.0%
		UDP	4.0%	4.0%	4.0%	6.0%	79.0%
	2	ICMPV6	1.0%	0.0%	0.0%	1.0%	0.0%
		TCP	1.0%	0.0%	0.0%	0.0%	0.0%
		UDP	1.0%	0.0%	0.0%	1.0%	0.0%
	3	ICMPV6	0.0%	0.0%	0.0%	0.0%	0.0%
		TCP	0.0%	0.0%	0.0%	0.0%	0.0%
		UDP	0.0%	0.0%	0.0%	0.0%	0.0%

Table 11: Mean number responses in relation to number of message types for BValue Steps.

that shows unique response timings by returning *AU* after a timeout of 2 seconds. It is also the only appliance that returns *AU* in the presence of null routes.

The ***HP*** image in our lab belongs to the virtual router series (VSR1000). The virtual router runs the same Comware version 7 operating system as HPE routers and switches. Since version 7 the operating system has switched to the Linux kernel. HP was the only vendor where ICMPv6 error messages were deactivated by default. After enabling them it shows simlar behavior to previous vendors. ***Huawei*** runs its own operating system Versatile Routing Platform (VRP) on its routers. The Huawei image is a virtualized version of the NetEngine 40E series. However, the image has limited capabilities, i.e. configuring ACLs is not possible. It is also the only image that does not return *AU* for unassigned IP addresses. Networks with huawei routers behaving in a similar way could therefore be missing in our list of active networks. Howerver, an analysis of EUI-64 addresses for M2 in §5.3 yielded Huawei routers to be the most prominent for active periphery. Thus other versions of VRP might behave differently in regards of active networks. The same limiation for ACL configuration accounts for ***Arista*** vEOS 4.28. Arista's Extensible Operating System (EOS) which is again a Linux-based network operating system.

VyOS is an open virtual Linux-based network operating system. It arose as a community fork from Vyatta which was based on the Debian Linux-distribution. ***Mikrotik*** products are more targeted towards small office, home office (SOHO) customers. Mikrotik features its own router operating system RouterOS. We include both version 6.48 and 7.7.1 in our lab, as they run different versions of the Linux kernel. While we see no difference in error message type, we found the rate limiting behavior to change between these versions. With Mikrotik we also begin to see a clear change in error message behavior for scenarios including ACLs and null routes. ***OpenWrt*** is a vendor-independent network operating system based on Linux for embedded devices. We include both version 19.07 which is based on kernel version 4.14 and 21.02 which is based on kernel version 5.4. OpenWRT is the only appliance to return *FP* in (S2). ***Aruba*** ArubaOS-CX is a virtual switch simulator implementing the features of the Linux-based AruboaOS-CX operating system. However, its layer 3 functionality allows it to be tested in the lab setup. To show that also firewall appliances return ICMPv6 error messages we included ***PfSense*** and ***Fortigate***. While by default any inbound traffic is rejected, we configured rules to explicitly forward traffic to the target network.

C BValue Steps Responsiveness & Borders

We provide details about the share of different message types for our BValue Steps approach. Table 10 gives an overview of the received ICMPv6 responses for the different generated addresses, sorted by message types associated with active respective inactive networks and ambiguous ones. For B127 we find that from the original 47,922 prefixes only 20,319 are responsive. This highlights that traffic in these networks is only forwarded to the hitlist address. For the 42% of prefixes we receive a response in 40% of cases we target an assigned IP, while in the other 60% we target an unassigned IPs. For B120 we also see a higher share of responses from assigned IPs. This shows the presence of other assigned IPs close to the hitlist address. Tools like scan6 [17] can exploit these address patterns to perform host discovery in these networks. We also notice a clear shift from networks returning $AU_{RTT>1s}$ from B127 to B64 to networks returning *NR*, $AU_{RTT<1s}$, *RR* and *TX* for B56 to B32. *BValue Step Width* In the beginning we experimented with step widths of 4, 8 and 16 bits. We decided for a step width of 8-bit as a trade-off in number of probes and covering the major prefix boundaries. A change at a prefix boundary such as /60 currently results in a change in B56 in Figure 4. However, the overall occurrence of such non 8-bit boundaries appear to be limited as highlighted in Table 11. In 97% of the BValue steps, we receive a single error message type suggesting that these cases are . Future work could repeat the measurements with lower step sizes for more fine-grained prefix boundary detection. For most BValue Steps (80%) one message type and five responses are received.

D Linux & BSD Kernel Default Behavior

Table 12 shows the change in response behavior between Linux kernel versions 4.9 and 4.19. We automated the network configuration of Debian-live CDs (https://cdimage.debian.org/mirror/cdimage/archive/) in qemu by redirecting the serial console. This way, we can trigger error messages and measure the behavior of the underlying Linux kernels beginning in 2014. With each major Debian version,

	Kernel Version	Release	IPv4	IPv6
Linux	2.6.26-1-2	2008	15	15
	3.16.0-4-6	2014	15	15
	4.9.0-3-13	2016	15	15
	4.19.0-5-21	2018	15	45
	5.10.0-8-22	2020	15	45
	6.1.0-9	2022	15	45
Freebsd	11.0	2016	2000	1000
Netbsd	8.2	2020	1000	1000

Table 12: Error messages (NR(10)) for TX for IPv4 and IPv6 of different Linux kernels yielding a change between version 4.9 and 4.19.

the underlying Linux kernel version also changed. We tested Debian version 5 with Linux kernel 3.16.0 up to 12 with Linux kernel version 6.1.9. We failed to automate different versions for FreeBSD and manually verified the rate-limiting behavior of version 11. The error rate matches that of PfSense, which we tested in our GNS3 lab. Similarly, we also checked the rate limit of NetBSD version 8.2. For NetBSD we find overlapping behavior with FreeBSD. We classify this fingerprint similar to a multi vendor fingerprint as *FreeBSD/NetBSD*.