# Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure

Marcin Nawrocki
marcin.nawrocki@fu-berlin.de
Freie Universität Berlin
Germany

Maynard Koch
maynard.k@fu-berlin.de
Freie Universität Berlin
Germany

Thomas C. Schmidt
t.schmidt@haw-hamburg.de
HAW Hamburg
Germany

Matthias Wählisch
m.waehlisch@fu-berlin.de
Freie Universität Berlin
Germany

## ABSTRACT

In this paper, we revisit the open DNS (ODNS) infrastructure and, for the first time, systematically measure and analyze transparent forwarders, DNS components that transparently relay between stub resolvers and recursive resolvers. Our key findings include four takeaways. First, transparent forwarders contribute 26% (563k) to the current ODNS infrastructure. Unfortunately, common periodic scanning campaigns such as Shadowserver do not capture transparent forwarders and thus underestimate the current threat potential of the ODNS. Second, we find an increased deployment of transparent forwarders in Asia and South America. In India alone, the ODNS consists of 80% transparent forwarders. Third, many transparent forwarders relay to a few selected public resolvers such as Google and Cloudflare, which confirms a consolidation trend of DNS stakeholders. Finally, we introduce DNSRoute++, a new traceroute approach to understand the network infrastructure connecting transparent forwarders and resolvers.

## CCS CONCEPTS

• **Networks** → **Public Internet**; **Security protocols**; **Network measurement**; • **Security and privacy** → *Security protocols*.

**ACM Reference Format:**
Marcin Nawrocki, Maynard Koch, Thomas C. Schmidt, and Matthias Wählisch. 2021. Transparent Forwarders: An Unnoticed Component of the Open DNS Infrastructure. In *The 17th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '21), December 7–10, 2021, Virtual Event, Germany*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3485983.3494872

## 1 INTRODUCTION

The open DNS infrastructure (ODNS) [37] comprises all components that publicly resolve DNS queries on behalf of DNS clients located in a remote network. This "openness" makes the ODNS system a popular target for attackers, who are in search for amplifiers of DNS requests, for periodic DNS scan campaigns, which try to expose the attack surface, and for researchers, who want to learn more about DNS behavior.

**Table 1: Comparison of known open DNS components.**

| | 2014 | 2020 | 2021 | | | |
|---|---|---|---|---|---|---|
| | [26] | [1] | [8] | [39] | [38] | This Work |
| # Rec. Resolvers | n/a | 20K | 50K | n/a | n/a | 32K (2%) |
| Forwarders | | | | | | |
| # Recursive | n/a | 1.4M | 1.7M | n/a | n/a | 1.5M (72%) |
| # Transparent | 0.6M (2%) | n/a | n/a | n/a | n/a | 0.6M (26%) |
| All ODNSes | 25.6M | 1.42M | 1.75M | 1.8M | 1.6M | 2.125M |

Originally observed in 2013 [31], *transparent* DNS forwarders have not been analyzed in detail since then, but fell off the radar in favor of *recursive* forwarders and resolvers. This raises concerns for two reasons. First, the relative amount of transparent forwarders increased from 2.2% in 2014 to 26% in 2021 (see Table 1). Second, as part of the ODNS, they interact with unsolicited, potentially malicious requests.

In this paper, we systematically analyze transparent forwarders. Our main contributions read as follows:

(1) We characterize transparent forwarders and review DNS measurement methods. (§ 2)
(2) We experimentally show that popular DNS scanning campaigns do not expose transparent forwarders and thus provide an incomplete view on the ODNS threat landscape. (§ 3)
(3) We measure the impact of transparent forwarders and find diverse deployments, heavily dependent on the hosting country. For example, configurations of forwarders in South America and Asia greatly contribute to DNS consolidation. (§ 4)
(4) We introduce `DNSRoute++`, a new traceroute approach that leverages the behaviour of transparent forwarders and explores interconnectivity in the ODNS. (§ 5)
(5) We discuss transparent forwarders in a broader context. Most of the transparent forwarders are CPE devices, either serving single end customers or larger networks. (§ 6)


*CoNEXT '21, December 7–10, 2021, Virtual Event, Germany*

ACM ISBN 978-1-4503-9098-9/21/12…$15.00
https://doi.org/10.1145/3485983.3494872

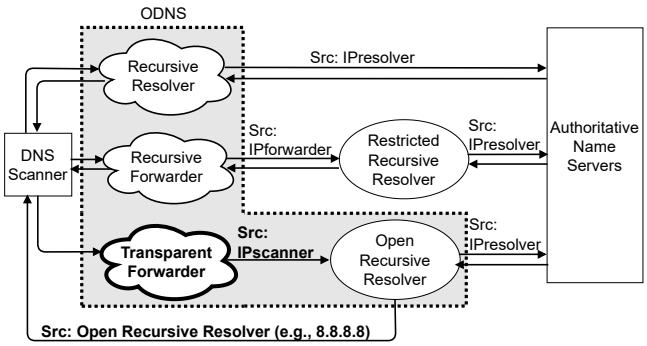


**Figure 1: Overview of various ODNS components and their relation to common Internet-wide scan setup.**

## 2 BACKGROUND AND RELATED WORK

**Open DNS (ODNS).** Various DNS stakeholders [3] such as domain owners and network operators operate autonomously and pursue different goals. A common view on the DNS is the ODNS infrastructure [37], client-side DNS speakers that openly accept requests from any host (not related to oblivious DNS, *i.e.,* ODoH). ODNS components have been previously classified into *recursive resolvers* and *forwarders* [4, 37]. Recent Internet-wide scans show that the majority (95%) of ODNSes are forwarders [1] but prior work does not further distinguish between recursive and transparent forwarders and mainly assume only the presence of recursive forwarders [25].

Figure 1 shows the expected behavior of all three ODNS components, which are commonly used by stub clients. Recursive resolvers send queries recursively to authoritative name servers and respond with the final answer to the original client (*e.g.,* scanner). In contrast, forwarders do not use DNS primitives to resolve names but forward queries to a recursive resolver [16]. Recursive forwarders can relay to restricted resolvers, however, transparent forwarders must forward to open resolvers to act as an ODNS (otherwise, the resolver rejects the scanner IP address). Upon receiving a final answer, a forwarder may cache it and relays it to the client. Forwarders are often susceptible to fragmentation [43] or side-channel [30] attacks.

**Introducing Transparent Forwarders.** In this paper, we argue that there are DNS forwarders that do not receive DNS answers because they operate completely transparent. Such deployment makes the distinction between recursive forwarders and transparent forwarders necessary.

A *recursive forwarder* replaces the original source IP address of the client by its own IP address. A *transparent forwarder* keeps the IP address of the original requester (*e.g.,* $IP_{Scanner}$). The relaying behavior of transparent forwarders has two implications. First, answers are sent back directly from resolvers to the original requester, *i.e.,* they are neither observed nor cached by the forwarder. Second, transparent forwarders spoof the IP address of the requester.

Surprisingly, Internet-wide, single packet scans lead to multiple answers from the same host, *e.g.,* 314k responses from `8.8.8.8`. Our study verifiably links these to transparent forwarders. Prior work removes these in a sanitizing step [1] or describes them as *unexpected* [28] but falls short to identify the root cause. So far, transparent forwarders have been treated as an artifact which can be utilized to measure missing outbound source-address-validations [23, 26].

**Table 2: Comparison of forwarder detection methods.**

| | Custom | |
|---|---|---|
| | Queries [2, 23, 26, 37] | Responses [1, 8], this work |
| Utilization of caches | None | High |
| Load auth. name server | High | Low |
| Forwarder detection | At server | At client |
| Forwarder classification | At client | At client |

**Comparison of ODNS Classification Methods.** Two methods are common to distinguish recursive resolvers and forwarders: (*i*) Destination-specific DNS queries from a scanner, which encode the destination IP addresses as a subdomain into the query name (*e.g.,* `203-0-113-1.mydomain.com`). (*ii*) Source-specific responses from an authoritative name server, which inserts the IP address of the immediate client (*e.g.,* `203.0.113.1`) into a dynamic `A` resource record of the query name (*e.g.,* `mydomain.com A 203.0.113.1`). This method can utilize two `A` resource records, a client-specific record and a static control record to check for DNS manipulations.

The first method enables an analysis at the name server. If the IP source address of an immediate client matches the encoded IP address within the query name, then the scanned destination is a recursive resolver, and a forwarder otherwise. The second method requires an analysis at the node that originally sent the query (*e.g.,* a DNS scanner). If the IP source address of the response matches the IP address within the `A` record, then the scanned destination is a recursive resolver, otherwise the scanned node is a DNS forwarder. This condition does not hold true anymore for transparent forwarders as recursive resolvers reply directly to the scanning node.

Table 2 summarizes the (dis-)advantages of both methods. The query-based method is particularly useful when the measurement objective needs to prevent caching, because the query name is unique for each target. This increases the load at the authoritative name server, though. The response-based method keeps the load at the authoritative name server low since it allows to utilize caches. Although the first method allows to detect forwarders already at the name server, classifying forwarders into recursive and transparent is only possible at the scanning node because the source IP address of the response has to be evaluated. Such a classification requires a correlation of DNS requests and responses to reflect the full DNS transaction. Hence, we will deploy the latter method in § 4.

## 3 POPULAR SCANNING CAMPAIGNS AND TRANSPARENT FORWARDERS

Censys [8], Shadowserver [39], and Shodan are popular scanning campaigns to reveal ODNSes. To verify our assumption that these campaigns underestimate the current number of ODNSes because they only record responses without correlating with the original target IP addresses of requests, we conduct a controlled experiment.

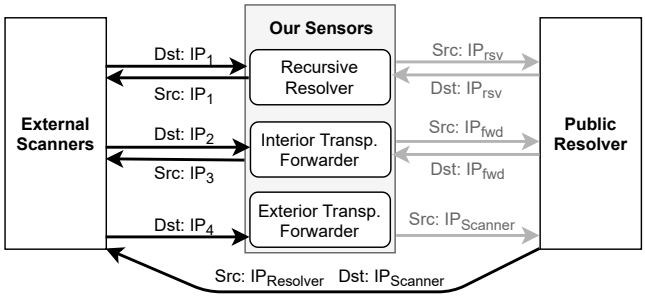


**Figure 2: DNS sensors. Black arrows indicate DNS messages visible to external scanning campaigns.**

## 3.1 Controlled Experiment

We develop and deploy three ODNS honeypot sensors, see Figure 2.

**Sensor 1: Recursive Resolver.** The first sensor behaves exactly like a public recursive resolver. The sensor answers using the same IP address at which it also has received a DNS request, $IP_1$. This configuration is a baseline measurement. We expect every viable Internet-wide DNS scanning campaign to find this sensor.

**Sensor 2: Interior Transparent Forwarder.** We utilize two IP addresses, $IP_2$ to receive DNS requests from a scanner and $IP_3$ to send responses. Both IP addresses are part of the same /24 prefix. This configuration allows for the following inferences: (*i*) Scanners that report $IP_2$ ignore the different IP address $IP_3$ in the response. They are RFC-compliant [33], and implement DNS transactional scans. (*ii*) Scanners that report $IP_3$ only evaluate the responses independently of the sent requests, which is a strong indicator for stateless, response-based analysis. This sensor mimics the key behavior of a transparent forwarder, but, as both addresses belong to the same IP prefix, the setup is easy to deploy. It does not require special network configuration such as disabled source address validation. Moreover, we can ensure that a reply is sent to the scanner.

**Sensor 3: Exterior Transparent Forwarder.** The third sensor implements a transparent forwarder which relays spoofed packets to an external, public resolver. This sensor is reachable at $IP_4$ and forwards a request using the source IP address of the scanner. To allow for spoofing, this sensor should be connected to a network that does not deploy source address validation [13] and peers directly with the network of the public resolver. In contrast to the previous setups, we do not receive the answer from the public resolver since the answer is sent directly to the scanner. Similarly to sensor (2), we can infer the following: (*i*) Scanners that report $IP_4$ ignore the different IP address in the response, indicating transactional scans. (*ii*) Scanners that report the public resolver will miss our forwarding sensor. This is because multiple responses from the same source will be aggregated into a single DNS speaker.

**Deployment Details.** Our sensors resolve incoming requests using Google's public resolver. We verify that source address validation is not deployed in our network. Moreover, our network peers directly with Google at an Internet eXchange point (IXP), so we are not exposed to filters from upstream providers, as required for sensor 3. We confirm the correct operation of all sensors by sending DNS queries and analyzing replies at the scanner. To prevent amplification attacks [35], we configure a strict rate limiting such that each sensor is allowed to answer one request every 5 minutes per source /24 prefix. We use a rate limiting based on the client prefix since it also prevents DoS carpet bombs [14]. We deploy our sensors for multiple weeks and then inspect the scan project results.

**Table 3: Detection of our DNS sensors by popular scans.**

| | Detected | | | |
|---|---|---|---|---|
| | Sensor 1 | Sensor 2 | | Sensor 3 |
| Scanner | $IP_1$ | $IP_2$ | $IP_3$ | $IP_4$ |
| Shadowserver | ✓ | ✘ | ✓ | ✘ |
| Censys | ✓ | ✘ | ✘ | ✘ |
| Shodan | ✓ | ✘ | ✘ | ✘ |

## 3.2 Results

All three sensors received scans from Censys, Shadowserver, and Shodan, but those scanners did not identify all of our sensors as an ODNS component. We use Censys' and Shodans public search API to check which IP addresses of our sensors have been discovered. As owner of the IP prefix that we used for our sensors, we have been informed by Shadowserver about our sensors.

All measurement campaigns discovered our public resolver (Sensor 1). None of them found one of our DNS forwarders, see Table 3. Shadowserver reported the replying IP address $IP_3$ of Sensor 2, which, in real deployment, would represent the address of a recursive resolver. However, Censys and Shodan did not report $IP_3$, which indicates that the responses did not pass a sanitizing step, respectively. We conclude that transparent forwarders are currently missed by these scanning campaigns. Given that the measurement results of these campaigns are used by third parties, the impact of ignoring transparent fowarders is large. National CERTS, for example, rely on data from Shadowserver to identify local ODNS systems.

# 4 MEASURING AND ANALYSING TRANSPARENT DNS FORWARDERS

## 4.1 Measurement Method and Setup

**Method.** To identify transparent forwarders, we need to correlate requests and responses at the scanning node. Our method aims for easy deployment, low measurement overhead, and robustness against manipulations. It requires two steps. First, mapping replies to requests of our scans. Second, classifying ODNS components.

To implement the first step, our scanner records the complete DNS transaction, *i.e.,* source and destination IP addresses, client port, and the ID used in the DNS header [33]. Assigning replies to requests based only on IP addresses would introduce ambiguity since replies triggered via transparent forwarders will include the source IP address of the resolver. Furthermore, to enable Internet-wide parallel scans, we ensure unique tuples of transport port and ID similar to other asynchronous scanners [11]. Then, even if we receive replies from the same resolver used by different transparent forwarders, we can clearly map responses to requests (for a detailed example, compare appendix Figure 7).

Our scanner requests a static name that belongs to a DNS zone which we control. The corresponding authoritative name server replies with two A records similar to other approaches using client-specific responses (details see § 2). Performing full DNS transactions and using a control resource record also helps us to identify distortions introduced by middleboxes [20]. After receiving replies, we correlate the client port number and DNS transaction ID of responses and previously recorded request data. We use a conservative DNS timeout of 20 seconds. Note that this and the subsequent analysis of forwarders is part of post-processing the data. It does not affect the speed of scanning.

We then classify ODNS components. Utilizing the destination address of the request ($IP_{target}$), the response source address ($IP_{response}$) and dynamic A resource record ($A_{resolver}$), we apply:

**Transparent Forwarder** if
$$IP_{target} \neq IP_{response}$$
**Recursive Forwarder** if
$$IP_{target} = IP_{response} \wedge IP_{response} \neq A_{resolver}$$
**Recursive Resolver** if
$$IP_{target} = IP_{response} \wedge IP_{response} = A_{resolver}$$

**Setup.** We deploy our scanner in a network, which allows for high packet rates without triggering a DoS attack mitigation such as packet drops or rate limiting. We probe any public IPv4 address and use moderate scanning rates, *i.e.,* we need 18 hours for a full pass. Our authoritative name server is implemented based on a common high-performance DNS library [32], which supports up to 20k pps.

## 4.2 Results

The subsequent results are based on an Internet-wide scan from April 20, 2021. Ongoing, more recent scans find the same results.

**Detailed Comparison with Shadowserver.** We find ≈536k transparent forwarders, identified by distinct IP addresses. Compared to Shadowserver [39], which does not detect transparent forwarders, this reveals ≈ 18% more ODNS components (compare Table 1).

It is worth noting that we identified, in sum, fewer recursive resolvers and recursive forwarders compared to Shadowserver, because we require responses to include both A-records, with the static control record being unaltered. Shadowserver requires only one correct A record. Omitting this step in our method leads to similar numbers than Shadowserver. To be robust against manipulation, we keep our more strict requirement and still detect more ODNS components in total due to consideration of transparent forwarders.

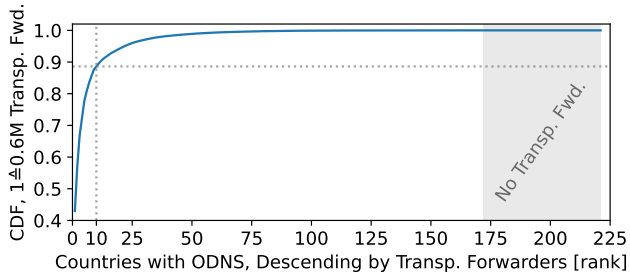


**Figure 3: CDF of transparent forwarders per country. Top-10 countries exhibit ~90% of all transparent forwarders.**

**Geo-Location of Transparent Forwarders.** We now try to understand whether the deployment of transparent forwarders is more popular in specific countries. To this end, we successfully map 99.9% IP addresses to ASes based on Routeviews dumps. Then, we map ASes to country codes with *whois* data und MaxMind. Figure 3 depicts the cumulative number of forwarders per country. Roughly 25% of countries with at least one ODNS component do not exhibit any transparent forwarder (highlighted in gray). We find, though, that ten countries host 90% of all transparent forwarders .

Countries that only expose transparent forwarders to the ODNS may be missed completely by scanning campaigns. Considering our complete data set, we do not find those cases. We find 5 countries hosting over 90% transparent forwarders, 4 of them are among the top-50 countries (see Figure 4). 8 out of 9 countries with over 10k transparent forwarders are classified as an emerging market [7], such as Brazil and India. With respect to all ODNSes in these two countries, transparent forwarders account for more than 80%.

**Common Public Resolvers used by Transparent Forwarders.** DNS consolidation directly correlates with how difficult it is to detect transparent forwarders. This is because the higher the consolidation, the more forwarders are *hidden* behind individual resolver IP addresses. Hence, we analyze the used resolvers and assess the relative popularity of four large public resolver projects (Google, Cloudflare, Quad9, and OpenDNS) per country. Figure 5 unveils that Google and Cloudflare are most common. Almost all transparent forwarders in India relay requests to Google, for example.

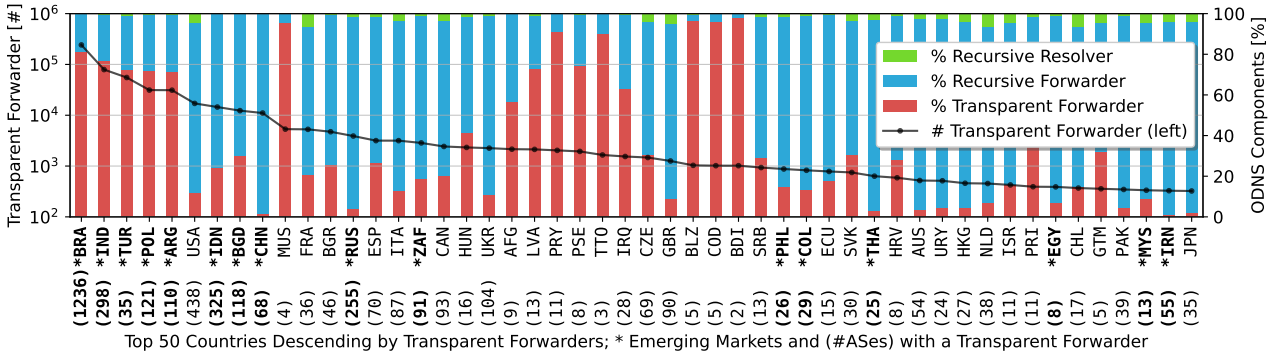


**Figure 4: Top-50 countries with transparent forwarders. Countries with emerging markets exhibit more transparent forwarders.**

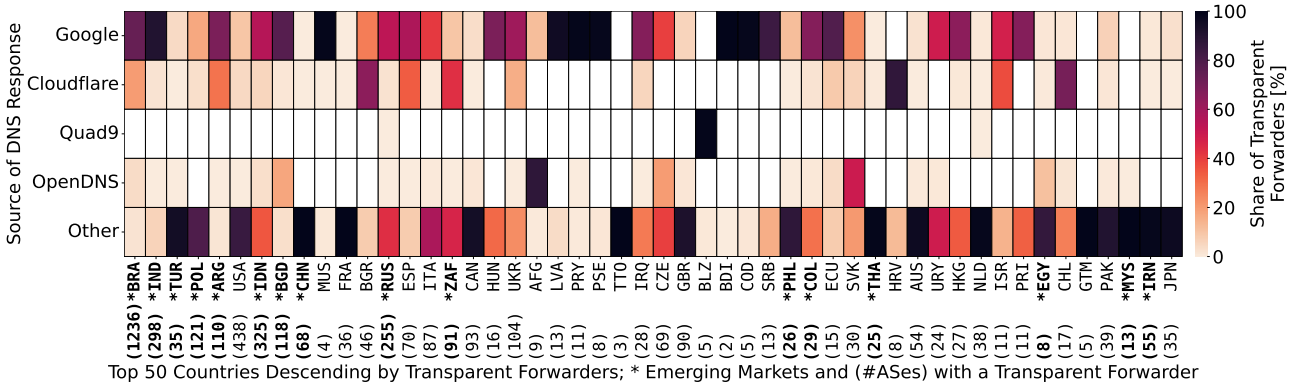


**Figure 5: Popularity of public resolver projects. Google & Cloudflare are commonly used by transparent forwarders.**

This aligns with recent complementary studies, which show that 19% of Google DNS users are located in India [18]. Following these results we can conclude that current scanning campaigns, which only consider DNS replies, underestimate the amount of ODNS components per country since they observe responses only from `8.8.8.8` or other public DNS projects. Comparing Shadowserver and our data, the ODNS rank of the top-20 countries varies up to 12 positions (details see Appendix C).

**Alternative Resolvers used by Transparent Forwarders.** We find countries in which transparent forwarders do not use one of the four common resolver projects (see "other" in Figure 5). In order to understand the usage of alternative resolvers, we analyze the top-10 countries with most transparent forwarders in the "other" share. Our results are summarized in Table 4. We detect two trends. First, countries such as India and Italy that already use popular resolver projects frequently (direct DNS consolidation) also deploy complex forwarding chains. In those cases, at our scanner, we receive DNS responses from IP addresses belonging to the AS of the transparent forwarder. Analysing the IP address in the $A_{resolver}$ record reveals, however, that our authoritative name server received the request from an IP address outside this AS. This unveils a dependency chain in which transparent forwarders relay to local recursive forwarders, which then forward to a popular resolver project (indirect consolidation). Second, we identify countries (Poland, France, China, and Turkey) that tend to not use public resolvers at all. Here, we find larger forwarder pools but those forwarders use only 1 to 10 local resolvers. For example, a single DNS resolver (195.175.39.69, Turkish Telecom) is serving almost all transparent forwarders from Turkey, which again masks their existence (for stateless scans).

**Table 4: Top-10 countries with highest "other" share (absolute) in Fig. 5. We show (*i*) the ASNs from which our scanner received most of the "other" responses, (*ii*) the number of transparent forwarders, (*iii*) the share of responses in "other" for which the ASN of $A_{resolver}$ belongs to one of the four common resolver projects.**

| Country | Top ASN | # Transparent Forwarders | Indirect Consolidation |
|---|---|---|---|
| Turkey | 9121 | 52,663 | 0.3% |
| Poland | 5617 | 24,879 | 1.4% |
| United States | 209 | 14,546 | 18% |
| China | 4812 | 11,030 | 0.9% |
| France | 5410 | 5,268 | 0.8% |
| Indonesia | 4622 | 5,154 | 27% |
| India | 3356 | 5,037 | 48% |
| Brazil | 262462 | 4,920 | 48% |
| Canada | 21724 | 2,303 | 21% |
| Italy | 3269 | 1,824 | 35% |

## 5 DNSROUTE++

In this section, we introduce *DNSRoute++*, a tool to explore network properties around transparent forwarders, and present two results.

**Measurement Approach.** *DNSRoute++* is a traceroute application that exploits the behavior of transparent forwarders. In contrast to common `traceroute`, *DNSRoute++* sends DNS requests and continues incrementing the TTL when the target is reached. If the target IP address is a transparent forwarder, we expect to receive `TTL Exceeded` messages from hosts beyond the forwarder. In detail, *DNSRoute++* (*i*) reveals all hops between a scanner and the (target) transparent forwarder, then (*ii*) all hosts between the transparent forwarder and the recursive resolver used by the forwarder. This works because the IP stack of the transparent forwarder replies when the TTL is exceeded (which stops forwarding) and forwards a DNS request internally to the upper layers otherwise (which reveals hosts beyond a forwarder). We scan all transparent forwarders.

**Path Lengths to Public Resolvers.** We compare path lengths from transparent forwarders to their recursive resolvers, see Figure 6. We obtain over 70k paths to 1.1k ASNs after sanitization. Our sanitization removes incomplete paths due to host churn or traceroute anomalies. Short path lengths indicate sound anycast deployments.

We find that Cloudflare exhibits the shortest paths compared to Google and OpenDNS. On average, Cloudflare resolvers are reachable in 6.3 hops. In case of Google and OpenDNS, we observe 7.9 and 9.3 hops, respectively.

Doan *et al.* [10] performed similar path measurements using 2.5k RIPE Atlas probes in 729 distinct ASes. They also observe shorter paths to the Cloudflare resolver but a reverse ranking in case of Google and OpenDNS. This difference might be due to the

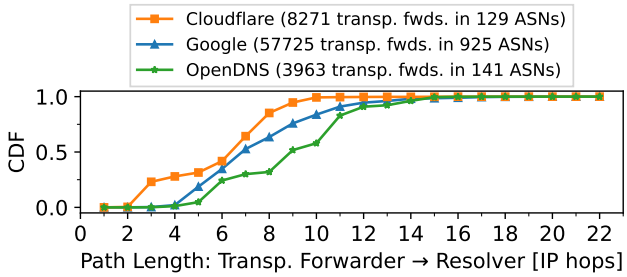


**Figure 6: Distribution of path lengths between transparent forwarders and their recursive resolvers, separated by common recursive resolver projects.**

location of measurement probes. RIPE Atlas probes are more likely located in North America and Europe, transparent forwarders are more common in South America and Asia. It is worth noting that our measurement approach only requires transparent forwarders and no deployment of dedicated probes in external networks. Hence, our methodology is complementary.

**AS Relationship Inference.** Paths acquired with *DNSRoute++* may help to infer AS relationships. The autonomous system (AS) before the AS of a forwarder indicates an inbound network ($AS_{in}$) and the AS after a forwarder the outbound network ($AS_{out}$). If $AS_{in} = AS_{out}$, we can assume a provider-customer relationship, since our scanner is outside the customer cone of $AS_{in}$. After sanitizing AS mappings, we can utilize 27k paths and observe $AS_{in} = AS_{out}$ for 62% of the paths. We detect 41 provider-customer relationships that are currently unclassified by CAIDAs relationship inference [6].

## 6 DISCUSSION

**What is the purpose of transparent forwarders?** Transparent forwarders differ from intentional DNS manipulations. First, they are not part of transparent interception [22, 27, 34, 42], which forwards queries to alternate resolvers and spoofs responses. Also, they differ from DNS redirection, which changes response records for the sake of advertisement [24, 41] or censorship [5, 25]. Lastly, they are not part of DNS tunneling, which carries ancillary information [19] not related to name resolution.

We conjecture that transparent forwarders are misbehaving CPE devices, either serving a single end customer or larger networks. To support this hypothesis, we perform an (*i*) AS-based, (*ii*) device-based, and (*iii*) prefix-based classification. For details about the classification, we refer to Appendix E.

Considering the top-100 ASes by the number of transparent forwarders, we find 79% ASes are eyeball ISPs, 7% of other types, and 14% remain unclassified. 65 ASNs are 32-bit numbers [40], *i.e.,* belong to more recent AS deployments.

MikroTik produces low-cost routers and CPE devices which are often affected by vulnerabilities [9, 21] and have been previously identified as DNS forwarders [26]. MikroTik's price policy seems to attract countries with emerging markets [7]. Overall, we attribute about 18k hosts (23%) to MikroTik.

Finally, we find that 26% of transparent forwarders are located in a /24 IP prefix that hosts less than 25 transparent forwarders. Such sparse population indicates that those forwarders belong to CPE devices (*e.g.,* home gateways) of individual end customers. On the other hand, we also find that 36% of the transparent forwarders cover a /24 network completely. 50% of the MikroTik routers we identified can be assigned to such a scenario.

All these observations strongly indicate that most of the transparent forwarders are misconfigured CPE devices. Whether these devices serve as a middlebox for a single customer or as router for a larger network does not affect our results regarding consolidation and attack potential.

**Should scanning campaigns deploy transactional scans?** Yes. Based on our measurements, current implementations of stateless DNS scans miss transparent forwarders, which account for 26% of all ODNS components. Interestingly, some countries host a disproportional amount of transparent forwarders which makes them far more exposed to misuse than previously assumed. For those 15 countries, we find that they host at least 50% of transparent forwarders and twice as much ODNSes as comparable studies detect.

Our transactional scans show that revealing transparent forwarders does not conflict with fast, stateless scans. Transactional scans require little-to-none changes to existing scanning infrastructures. Required changes include (*i*) the recording of outgoing scan traffic and (*ii*) a lightweight post-analysis, which matches queries and responses based on the client port and DNS transaction ID. These changes do not impair the scanning rate itself. We focus on DNS over UDP [33] as we do not expect transparent forwarding for DoT [17] and DoH [15] since their connection-based requests conflict with IP spoofing. Also, for benevolent scanning campaigns, we recommend utilizing custom responses and *not* custom queries for a forwarder classification to limit adverse effects. Encoding the IP addresses of targets leads to cache pollution due to negative caching [16] and cache evictions of popular, legitimately used names, which resembles random-subdomain [12] and water-torture [29] attacks. We find resolvers serving >40k forwarders, which would introduce >40k cache entries to a single resolver.

**What is the misuse potential?** Transparent forwarders can be misused as invisible diffusers for reflective amplification attacks as they relay the source IP address of the DNS request as-is. Hence, spoofed packets (allegedly from the victim) are forwarded with the source address spoofed by the attacker. Booters offering DDoS services utilize centralized attack infrastructures to reduce costs and maintenance [36]. Misusing transparent forwarders (*i*) allows to reach multiple PoPs of anycast DNS providers despite their centralized infrastructure (*e.g.,* Google allows `ANY` requests) and (*ii*) impedes attribution by further obfuscating the origin of spoofed traffic.

Overall, transparent forwarders likely belong to domestic setups but interact with unsolicited, external requests, which might lead to impaired performance, security risks and liability implications.

## 7 CONCLUSION

We showed that the open DNS infrastructure comprises transparent forwarders—in addition to its recursive components. These forwarders intensify the perceived threat potential of the ODNS. We argue to include them in on-going and future measurements as they account for a relevant impact and share. Our results bolster current concerns regarding consolidation of the DNS, at least for countries that massively host transparent forwarders.

## ACKNOWLEDGMENTS

We would like to thank our shepherd Marinho Barcellos and the anonymous reviewers for their helpful feedback. We are grateful to Markus de Brün, Anders Kölligan, and operators of the LACNIC region for fruitful discussions. We thank Moritz Müller and Martino Trevisan for their feedback on the artifacts. This work was partly supported by the German Federal Ministry of Education and Research (BMBF) within the project PRIMEnet.

## REFERENCES

[1] Arian Akhavan Niaki, William Marczak, Sahand Farhoodi, Andrew McGregor, Phillipa Gill, and Nicholas Weaver. 2021. Cache Me Outside: A New Look at DNS Cache Probing. In *Proc. of PAM* (Virtual conference). Springer International Publishing, Cham, Switzerland, 427–443. https://doi.org/10.1007/978-3-030-72582-2_25

[2] Mark Allman. 2020. Putting DNS in Context. In *Proc. of ACM IMC* (Virtual conference). ACM, New York, NY, USA, 309–316. https://doi.org/10.1145/3419394.3423659

[3] Mario Almeida, Alessandro Finamore, Diego Perino, Narseo Vallina-Rodriguez, and Matteo Varvello. 2017. Dissecting DNS Stakeholders in Mobile Networks. In *Proc. of ACM CoNEXT* (Incheon, Republic of Korea). ACM, New York, NY, USA, 28–34. https://doi.org/10.1145/3143361.3143375

[4] Marios Anagnostopoulos, Georgios Kambourakis, Panagiotis Kopanos, Georgios Louloudakis, and Stefanos Gritzalis. 2013. DNS amplification attack revisited. *Computers & Security* 39, B (2013), 475–485. https://doi.org/10.1016/j.cose.2013.10.001

[5] Anonymous. 2014. Towards a Comprehensive Picture of the Great Firewall's DNS Censorship. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)* (San Diego, CA, USA). USENIX Association, Berkeley, CA, USA. https://www.usenix.org/conference/foci14/workshop-program/presentation/anonymous

[6] CAIDA. 2005. AS Rank API. Website. http://as-rank.caida.org/ Retrieved: May, 2021.

[7] Lourdes Casanova and Anne Miroux. 2020. *Emerging Market Multinationals Report: 10 Years that Changed Emerging Markets.* Technical Report. Cornell University. https://doi.org/10.7298/cvhn-dc87

[8] Censys. 2017. Censys IO Search 2.0. Website. https://search.censys.io/ Retrieved: May, 2021.

[9] João M. Ceron, Christian Scholten, Aiko Pras, and Jair Santanna. 2020. MikroTik Devices Landscape, Realistic Honeypots, and Automated Attack Classification. In *Proc. of IEEE/IFIP Network Operations and Management Symposium (NOMS)* (Budapest, Hungary. Virtual Conference). IEEE, Piscataway, NJ, USA, 1–9. https://doi.org/10.1109/NOMS47738.2020.9110336

[10] Trinh Viet Doan, Justus Fries, and Vaibhav Bajpai. 2021. Evaluating Public DNS Services in the Wake of Increasing Centralization of DNS. In *Proc. of 20th IFIP Networking Conference* (Virtual conference). IEEE, Piscataway, NJ, USA, 00–00. http://dl.ifip.org/db/conf/networking/networking2021/1570700957.pdf

[11] Zakir Durumeric, Eric Wustrow, and J. Halderman. 2013. ZMap: Fast Internet-Wide Scanning and its Security Applications. In *Proc. of the 22nd USENIX Security Symposium* (Washington, D.C., USA). USENIX Association, Berkeley, CA, USA, 605–620. https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric

[12] Shir Landau Feibish, Yehuda Afek, Anat Bremler-Barr, Edith Cohen, and Michal Shagam. 2017. Mitigating DNS Random Subdomain DDoS Attacks by Distinct Heavy Hitters Sketches. In *Proc. of the Fifth ACM/IEEE Workshop on HotWeb* (San Jose, California). ACM, New York, NY, USA, 1–6. https://doi.org/10.1145/3132465.3132474

[13] P. Ferguson and D. Senie. 2000. *Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.* RFC 2827. IETF. https://doi.org/10.17487/RFC2827

[14] Tiago Heinrich, Rafael R. Obelheiro, and Carlos Alberto Maziero. 2021. New Kids on the DRDoS Block: Characterizing Multiprotocol and Carpet Bombing Attacks. In *Proc. of PAM* (Virtual conference). Springer International Publishing, Cham, Switzerland, 427–443. https://doi.org/10.1007/978-3-030-72582-2_16

[15] P. Hoffman and P. McManus. 2018. *DNS Queries over HTTPS (DoH).* RFC 8484. IETF.

[16] P. Hoffman, A. Sullivan, and K. Fujiwara. 2019. *DNS Terminology.* RFC 8499. IETF. https://doi.org/10.17487/RFC8499

[17] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman. 2016. *Specification for DNS over Transport Layer Security (TLS).* RFC 7858. IETF.

[18] Geoff Huston and Joao Damas. 2021. Measuring Recursive Resolver Centrality. RIPE Meeting 82. https://ripe82.ripe.net/wp-content/uploads/presentations/41-2021-05-19-resolver-centrality.pdf Retrieved: June, 2021.

[19] Naotake Ishikura, Daishi Kondo, Vassilis Vassiliades, Iordan Iordanov, and Hideki Tode. 2021. DNS Tunneling Detection by Cache-Property-Aware Features. *IEEE Trans. Netw. Serv. Manag.* 18, 2 (2021), 1203–1217. https://doi.org/10.1109/TNSM.2021.3078428

[20] Liz Izhikevich, Renata Teixeira, and Zakir Durumeric. 2021. LZR: Identifying Unexpected Internet Services. In *Proc. of 30th USENIX Security Symposium* (Virtual conference). USENIX Association, Berkeley, CA, USA, 18 pages. https://www.usenix.org/conference/usenixsecurity21/presentation/izhikevich Prepublication.

[21] Jacob Baines. 2019. RouterOS Post Exploitation. Blog. https://medium.com/tenable-techblog/routeros-post-exploitation-784c08044790 Retrieved: September, 2021.

[22] Ben Jones, Nick Feamster, Vern Paxson, Nicholas Weaver, and Mark Allman. 2016. Detecting DNS Root Manipulation. In *Proc. of PAM* (Heraklion, Greece). Springer, Cham, Switzerland, 276–288. https://doi.org/10.1007/978-3-319-30505-9_21

[23] Maciej Korczynski, Yevheniya Nosyk, Qasim Lone, Marcin Skwarek, Baptiste Jonglez, and Andrzej Duda. 2020. Don't Forget to Lock the Front Door! Inferring the Deployment of Source Address Validation of Inbound Traffic. In *Proc. of PAM* (Eugene, Oregon, USA). Springer International Publishing, Cham, Switzerland, 107–121. https://doi.org/10.1007/978-3-030-44081-7_7

[24] Christian Kreibich, Nicholas Weaver, Boris Nechaev, and Vern Paxson. 2010. Netalyzr: illuminating the edge network. In *Proc. of ACM IMC* (Melbourne, Australia). ACM, New York, NY, USA, 246–259. https://doi.org/10.1145/1879141.1879173

[25] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. 2015. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proc. of ACM IMC* (Tokyo, Japan). ACM, New York, NY, USA, 355–368. https://doi.org/10.1145/2815675.2815683

[26] Marc Kührer, Thomas Hupperich, Christian Rossow, and Thorsten Holz. 2014. Exit from Hell? Reducing the Impact of Amplification DDoS Attacks. In *Proc. of 23rd USENIX Security Symposium* (San Diego, CA). USENIX Association, Berkeley, CA, USA, 111–125. https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/kuhrer

[27] Baojun Liu, Chaoyi Lu, Hai-Xin Duan, Ying Liu, Zhou Li, Shuang Hao, and Min Yang. 2018. Who Is Answering My Queries: Understanding and Characterizing Interception of the DNS Resolution Path. In *Proc. of the 27nd USENIX Security Symposium* (Baltimore, MD, USA.). USENIX Association, Berkeley, CA, USA, 1113–1128. https://www.usenix.org/conference/usenixsecurity18/presentation/liu-baojun

[28] Keyu Lu, Tingting Chai, Haiyan Xu, Shitala Prasad, Jianen Yan, and Zhaoxin Zhang. 2021. Research on Unexpected DNS Response from Open DNS Resolvers. *The Computer Journal* 00, 00 (05 2021), 1–23. https://doi.org/10.1093/comjnl/bxab063

[29] Xi Luo, Liming Wang, Zhen Xu, Kai Chen, Jing Yang, and Tian Tian. 2018. A Large Scale Analysis of DNS Water Torture Attack. In *Proc. of CSAI* (Shenzhen, China). ACM, New York, NY, USA, 168–173. https://doi.org/10.1145/3297156.3297272

[30] Keyu Man, Zhiyun Qian, Zhongjie Wang, Xiaofeng Zheng, Youjun Huang, and Haixin Duan. 2020. DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels. In *Proc. of ACM SIGSAC CCS* (Virtual conference). ACM, New York, NY, USA, 1337–1350. https://doi.org/10.1145/3372297.3417280

[31] Jared Mauch. 2013. Spoofing ASNs. NANOG mailing list. http://seclists.org/nanog/2013/Aug/132 Retrieved: May, 2021.

[32] Miek Gieben. 2010. DNS library in Go. Code repository. https://github.com/miekg/dns Retrieved: April, 2021.

[33] P.V. Mockapetris. 1987. *Domain names - implementation and specification.* RFC 1035. IETF. https://doi.org/10.17487/RFC1035

[34] Audrey Randall, Enze Liu, Ramakrishna Padmanabhan, Gautam Akiwate, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. 2021. Home is Where the Hijacking is: Understanding DNS Interception by Residential Routers. In *Proc. of ACM IMC* (Virtual Event). ACM, New York, NY, USA. https://www.sysnet.ucsd.edu/~voelker/pubs/homejack-imc21.pdf

[35] Fabrice J. Ryba, Matthew Orlinski, Matthias Wählisch, Christian Rossow, and Thomas C. Schmidt. 2015. *Amplification and DRDoS Attack Defense – A Survey and New Perspectives.* Technical Report arXiv:1505.07892. Open Archive: arXiv.org. http://arxiv.org/abs/1505.07892

[36] José Jair Santanna, Roland van Rijswijk-Deij, Rick Hofstede, Anna Sperotto, Mark Wierbosch, Lisandro Zambenedetti Granville, and Aiko Pras. 2015. Booters—An analysis of DDoS-as-a-service attacks. In *Proc. of IFIP/IEEE International Symposium on Integrated Network Management (IM)* (Ottawa, ON, Canada). IEEE, Piscataway, NJ, USA, 243–251. https://doi.org/10.1109/INM.2015.7140298

[37] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. 2013. On Measuring the Client-Side DNS Infrastructure. In *Proc. of ACM IMC* (Barcelona, Spain). ACM, New York, NY, USA, 77–90. https://doi.org/10.1145/2504730.2504734

[38] Shodan. 2014. Shodan - the world's first search engine for Internet-connected devices. Website. https://www.shodan.io/ Retrieved: September, 2021.

[39] The Shadowserver Foundation. 2021. Open Resolver Scanning Project. Website. https://scan.shadowserver.org/dns/ Retrieved: May, 2021.

[40] Q. Vohra and E. Chen. 2007. *BGP Support for Four-octet AS Number Space.* RFC 4893. IETF. https://doi.org/10.17487/RFC4893

[41] Nicholas Weaver, Christian Kreibich, and Vern Paxson. 2011. Redirecting DNS for Ads and Profit. In *USENIX Workshop on Free and Open Communications on the Internet (FOCI)* (San Francisco, CA, USA). USENIX Association, Berkeley, CA, USA, 2–3. https://www.usenix.org/conference/foci11/redirecting-dns-ads-and-profit

[42] Lan Wei and John S. Heidemann. 2020. *Whac-A-Mole: Six Years of DNS Spoofing.* Technical Report arXiv:2109.08783. Open Archive: arXiv.org. https://arxiv.org/abs/2011.12978

[43] Xiaofeng Zheng, Chaoyi Lu, Jian Peng, Qiushi Yang, Dongjie Zhou, Baojun Liu, Keyu Man, Shuang Hao, Haixin Duan, and Zhiyun Qian. 2020. Poison Over Troubled Forwarders: A Cache Poisoning Attack Targeting DNS Forwarding Devices. In *Proc. of 29th USENIX Security Symposium* (Virtual conference). USENIX Association, Berkeley, CA, USA, 577–593. https://www.usenix.org/conference/usenixsecurity20/presentation/zheng

# A ARTIFACTS

This section gives a brief overview of the artifacts of this paper. We contribute tools to conduct follow-up measurements as well as raw data and analysis scripts to reproduce the results and figures presented in this paper.

## A.1 Hosting

All artifacts are available in the following repository:

https://github.com/ilabrg/artifacts-conext21-dns-fwd

This public repository provides up-to-date instructions for installing, configuring, and running our artifacts. We also archive the camera-ready version of our software on Zenodo:

https://doi.org/10.5281/zenodo.5636314

## A.2 Description

This bundle of artifacts includes three tools, each located in a separate directory. Additionally, we provide wrapper scripts for Internet-wide scans. Finally, we provide notebooks to replicate the results of this paper.

**`dnsRoute++/`** Traceroute implementation that maps paths behind transparent forwarders (see § 6). We also add an IP hitlist that includes transparent forwarders at the time of our measurements. Please note that IP churn might have change the current state.

**`dns-honeypot-sensors/`** Honeypot sensors emulating various Open DNS speakers (ODNS), including transparent forwarders. These scripts emulate three different types of open DNS speakers (see § 3.1).

**`recursive-mirror-auth-server/`** A DNS nameserver implementation that replies with an `A` record referring to the IP address of the client that sent a DNS query. This reveals a recursive resolver (see § 4.1).

**`dns-scan-server/`** DNS scanning with *zmap* and *dumpcap* to capture the complete traffic during the scan. This artifact produces raw PCAP files required by the analysis scripts (need to be processed first).

**`dns-measurement-analysis/`** Implementation of our postprocessing and sanitizing method which creates a dataframe that is used for further analyses. This artifact replicates the results of this paper.

## A.3 Dependencies

**Software.** For a full list of dependencies, please, see the `readme.md` file included in each directory. The minimal requirement is a Linux system, GoLang, Python, and Bash.

**Network Infrastructure.** Our software requires specific network setups, such as network access without NAT and DNS names under your control. Again, please compare the *readme* files.

## A.4 Usage and Testing

Each directory includes a `run.sh` and a `test.sh`.

**`run.sh`** wraps multiple sub-scripts of each tool and allows for a quick start.

**`test.sh`** executes tests and provides expected output, which then can be evaluated for correctness. For more complex setups, the tests are run against our servers.

The artifacts can be tested independently and in any order.

## A.5 Future Measurements

We plan to continue our experiments. Future measurement results will be available on https://odns.secnow.net.

# B ETHICAL CONCERNS

We presented a method to discover a new type of public DNS forwarders, which may be misused by attackers. We are in contact with federal security offices to include transparent forwarders in their on-going measurements that inform network operators about vulnerable devices.

# C RANKING COUNTRIES BY ODNS COMPONENTS

In this work, we showed that transparent forwarders amount to more than 25% of all ODNS components. Common ODNS scan campaigns such as Shadowserver rank countries based on the number of ODNS components but miss transparent forwarders (see § 3). Table 5 shows the change of ranks for the top-20 countries when considering the complete ODNS infrastructure by including transparent forwarders.

# D MEASURING TRANSPARENT FORWARDERS

Figure 7 illustrates our response-based measurement method, which we explain in detail in § 4.

# E DETAILS ON THE DEPLOYMENT OF TRANSPARENT FORWARDERS

**AS Classification.** We classify the top-100 ASes by transparent forwarder count. These ASes cover 50% of all transparent forwarders. For each ASN, we map the network type using PeeringDB. 37 ASes are `Cable/DSL/ISP` networks. Since the majority of ASes is not classified in PeeringDB, we also perform a manual classification. We also perform a manual check whether ASes that are classified as `NSP` provide eyeball Internet services. Based on our manual inspection, we identify 42 additional ISPs. In total, out of the top-100 ASes, 79 can be considered `Cable/DSL/ISP` networks.

**Device Fingerprinting.** For device fingerprinting, we use Shodan [38] and Censys [8]. To this end, we analyze all open ports and banner grabbing information. Shodan provides information for 80k

**Table 5: Top-20 countries ranked by number of ODNS components, comparing this work and Shadowserver.**

| | This Work | | Shadowserver | | Difference Δ | |
|---|---|---|---|---|---|---|
| Country | Rank | #ODNS | Rank | #ODNS | Rank | #ODNS |
| China | 1 | 632428 | 1 | 717706 | 0 - | 85278 ↓ |
| Brazil | 2 | 297828 | 6 | 49616 | 4 ↑ | 248212 ↑ |
| United States | 3 | 144568 | 2 | 137619 | 1 ↓ | 6949 ↑ |
| India | 4 | 102910 | 8 | 33510 | 4 ↑ | 69400 ↑ |
| Russia | 5 | 93498 | 3 | 102368 | 2 ↓ | 8870 ↓ |
| Turkey | 6 | 76168 | 18 | 19298 | 12 ↑ | 56870 ↑ |
| Indonesia | 7 | 59972 | 5 | 56319 | 2 ↓ | 3653 ↑ |
| South Korea | 8 | 49143 | 4 | 73790 | 4 ↓ | 24647 ↓ |
| Argentina | 9 | 43648 | 20 | 16974 | 11 ↑ | 26674 ↑ |
| Poland | 10 | 43431 | 10 | 29175 | 0 - | 14256 ↑ |
| Bangladesh | 11 | 40917 | 16 | 22940 | 5 ↑ | 17977 ↑ |
| Taiwan | 12 | 37550 | 7 | 38525 | 5 ↓ | 975 ↓ |
| Iran | 13 | 36659 | 9 | 33444 | 4 ↓ | 3215 ↑ |
| France | 14 | 25320 | 12 | 25763 | 2 ↓ | 443 ↓ |
| Italy | 15 | 24766 | 14 | 24483 | 1 ↓ | 283 ↑ |
| Vietnam | 16 | 21407 | 15 | 24266 | 1 ↓ | 2859 ↓ |
| Ukraine | 17 | 20780 | 13 | 25307 | 4 ↓ | 4527 ↓ |
| Thailand | 18 | 19694 | 17 | 20474 | 1 ↓ | 780 ↓ |
| Bulgaria | 19 | 18443 | n/a | 16239 | >1 ↑ | 2204 ↑ |
| Germany | 20 | 16243 | 19 | 17788 | 1 ↓ | 1545 ↓ |

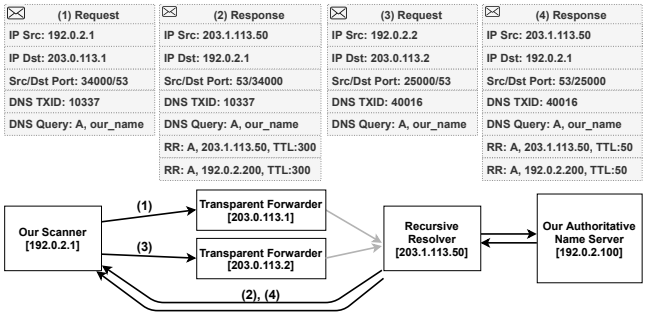


**Figure 7: Two transparent forwarders trigger DNS responses from the same recursive resolver, identified by the same source IP address. Black arrows indicate DNS messages observed by our infrastructure.**

of 600k queried hosts. Inspecting the open port distribution, we find a strong correlation for 10 MikroTik ports [9]. OS and product information collected by Shodan confirm our observations becasuse the most common tags specify MikroTik. Censys data confirms our results and also identifies the hosts as MikroTik devices.

**Distribution in /24 Prefixes.** We map each transparent forwarder to a (non-overlapping) covering /24 IP prefix and count the number of forwarders per prefix. If all IP addresses of a prefix reply to our transparent forwarder scans, we may assume that these replies are initiated by a single device (*e.g.,* some kind of middlebox that serves the whole prefix). In contrast, for sparsely populated prefixes, we may assume multiple deployed devices (*e.g.,* several CPE devices that serve differnet customers).

41k distinct IP prefixes cover 0.6M transparent forwarders. Figure 8 shows the distribution of the number of transparent forwarders in each /24 prefix. Overall, we observe a mixed picture. 26% of all transparent forwarders are located in sparsely populated prefixes ($\leq$ 25 transparent forwarders in a /24 prefix), and 36% in completely populated prefixes ($\geq$ 254 transparent forwarders in a /24 prefix). Only 806 prefixes are completely populated. In those cases, we argue that a CPE device serves as a router for larger networks instead of a single end customer. Using CPE devices outside of individual DSL/cable customers is not uncommon because CPE devices are cheap and some implement routing protococols (*e.g.,* MikroTik even BGP). In any case, whether the transparent forwarder function runs on a device that serves a single end customer or a larger network, our results hold, the transparent forwarder interacts as an ODNS component and uses the resolvers we observed.

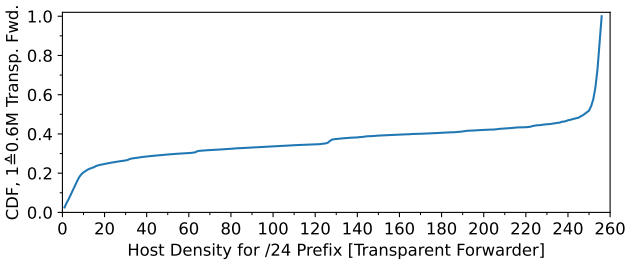


**Figure 8: We map all transparent forwarders to a covering /24 prefix. Some transparent forwarders belong to individual end customers, others may serve a larger network.**