Towards Encrypted DNS with DDR: Standards, Adoption and Security Implications

Verschlüsseltes DNS mit DDR: Eine Betrachtung von Standards, Adoption und Sicherheitsrisiken

Steffen Sassalla

Universitätsmasterarbeit
zur Erlangung des akademischen Grades

Master of Science
(M. Sc.)

im Studiengang
IT Systems Engineering

eingereicht am 18. Dezember 2024 am
Fachgebiet Data-Intensive Internet Computing der
Digital-Engineering-Fakultät
der Universität Potsdam

Gutachter	Prof. Dr. Vaibhav Bajpai
	Prof. Dr. Anja Lehmann
Betreuer	Dr. Vasilis Ververis

Abstract

The Domain Name System (DNS), a cornerstone of the Internet, traditionally transmits queries in plaintext, thus leaving it vulnerable to security and privacy threats such as monitoring, user tracking, and censorship. Despite efforts to standardize DNS encryption, adoption remains limited. Users often lack awareness of privacy risks and the knowledge required to enable encryption. To address this, the IETF standardized a new protocol; Discovery of Designated Resolvers (DDR), enabling automatic discovery and seamless upgrade from unencrypted to encrypted DNS traffic. As such, we present the first large-scale, in-depth analysis of DDR by measuring its deployment on a semi-weekly basis for four months. From 4M IPv4 (287K IPv6) DNS servers, we have found that 7.59 % (2.65 % IPv6) of resolvers return a DDR configuration, with DNS over HTTPS/2 (DoH/2) being the most popular protocol advertised (>99 %). Despite the performance and privacy benefits of DNS over QUIC (DoQ), fewer than 7 % of DDR-enabled resolvers advertise support for this protocol. Alarmingly, more than 99 % of DDR-compliant clients fail the verified discovery process when using a resolver's IP address, primarily due to misconfigurations in real-world deployments, which undermines DDR's promise of seamless encryption upgrades. Finally, we have observed that over 97 % of DDR-enabled resolvers delegate clients to major DNS cloud providers such as Google and Cloudflare, raising concerns about DNS resolver market centralization and its adverse implications for user privacy and governance.

Zusammenfassung

Das Domain Name System (DNS), ein Grundpfeiler des Internets, überträgt Anfragen traditionell im Klartext, wodurch es anfällig für Sicherheits- und Datenschutzrisiken wie Überwachung, Nutzerverfolgung und Zensur wird. Trotz Bemühungen zur Standardisierung von DNS-Verschlüsselung bleibt die Akzeptanz begrenzt. Häufig mangelt es Nutzern an Bewusstsein für Datenschutzrisiken und dem erforderlichen Wissen zur Aktivierung von Verschlüsselung. Um dieses Problem zu adressieren, hat die IETF ein neues Protokoll standardisiert: Discovery of Designated Resolvers (DDR). DDR ermöglicht die automatische Erkennung und das nahtlose Wechseln von unverschlüsselten auf verschlüsselten DNS-Übertragungen. In dieser Arbeit präsentieren wir die erste umfassende und zugleich detaillierte Kartierung von DDR, basierend auf Messungen, die über einen Zeitraum von vier Monaten im halbwöchentlichen Rhythmus durchgeführt wurden. Ausgehend von 4M IPv4 und 287K IPv6 DNS Servern stellen wir fest, dass 7,59% (2,65% IPv6) der Resolver eine DDR-Konfiguration anbieten, wobei DNS over HTTPS/2 (DoH/2) das am häufigsten angebotene Protokoll ist (>99 %). Trotz der Leistungs- und Sicherheitsvorteile von DNS over QUIC (DoQ) wird es von weniger als 7 % der DDR-fähigen Resolver angeboten. Alarmierenderweise scheitern in mehr als 99 % der Fälle DDR-kompatible Clients am verifizierten Upgrade-Prozess auf ein verschlüsseltes DNS-Protokoll, hauptsächlich aufgrund von Fehlkonfigurationen in realen Implementierungen. Dies untergräbt das Versprechen von DDR, eine nahtlose Verschlüsselung zu ermöglichen. Letztendlich haben wir beobachtet, dass über 97 % der DDR-fähigen Resolver ihre Clients an große DNS-Cloud-Anbieter wie Google oder Cloudflare weiterleiten, was erhebliche Bedenken hinsichtlich der Marktkonzentration von DNS-Resolvern und deren nachteiligen Auswirkungen auf Datenschutz und Governance aufwirft.

Contents

1 Introduction

When a user or client attempts to connect to a network resource, the process typically involves resolving a domain name (or hostname) into an IP address to establish a connection at the IP layer. For example, when accessing the website `www.example.com`, the domain name must first be translated into an IP address, which the computer then uses to initiate the connection. This resolution process is facilitated by the Domain Name System (DNS) [112, 113], a cornerstone protocol of the Internet [50] and one of the largest distributed databases globally [76]. Despite its immense scale, DNS operates with remarkable efficiency, remaining largely invisible to users as it functions seamlessly in the background during activities such as browsing web pages, sending emails, or performing online transactions.

However, the traditional DNS resolution process is inherently insecure because it transmits queries and responses in plaintext. This exposes users to potential risks, such as eavesdropping and data manipulation [164]. For example, previous research has demonstrated that DNS queries can be leveraged to track users across multiple websites [60, 73, 94, 114, 164]. Moreover, DNS traffic can reveal the presence of Internet-of-Things (IoT) devices within home networks and even expose behavioral patterns in how these devices are used [92].

In response to the growing concerns about user privacy, efforts have intensified to enhance the security of DNS communication. The Internet Engineering Task Force (IETF) introduced several encrypted DNS protocols, including DNS over HTTPS (DoH) [64], DNS over TLS (DoT) [70], and DNS over QUIC (DoQ) [72]. These protocols aim to protect users' DNS queries and responses by encrypting communication between clients (e.g., stub resolvers) and recursive resolvers, i.e., *stub-to-recursive* communication. By ensuring confidentiality and integrity, these encrypted DNS protocols — collectively referred to as DNS over Encryption (DoE) protocols — mitigate risks associated with eavesdropping and tampering.

Despite these advancements, the majority of DNS traffic remains unencrypted, as the transition from plaintext DNS to encrypted DNS poses challenges for many users [18]. To address this issue, the IETF standardized the Discovery of Designated Resolvers (DDR) protocol [124] in November 2023. DDR streamlines the adoption of encrypted DNS by enabling clients to automatically discover

and transition to designated encrypted resolvers. As a result, DDR reduces the need for manual intervention, thereby inherently providing mechanisms enhancing security and privacy for end users.

1.1 Motivation

While the DDR protocol holds promise for leveraging DNS security and privacy, its deployment and configuration in real-world scenarios remain underexplored. To the best of our knowledge, no study has yet investigated the actual deployment of DDR. However, major DNS cloud providers like *Google* and *Cloudflare* already support DDR, signaling its growing adoption. As its use expands, understanding the adoption rates, configuration patterns, and challenges associated with DDR is crucial for evaluating its effectiveness in facilitating secure transitions to encrypted DNS protocols. This study addresses the need for an empirical investigation into DDR's deployment and its role within the DNS ecosystem.

1.2 Research Questions

Throughout this study, we aim to answer the following research questions (RQs):

RQ1: What are the adoption rates and trends of DDR-enabled resolvers in IPv4 and IPv6, and how do they vary across geographical regions and network types over time?

The DDR protocol [124], standardized in November 2023, establishes a unified mechanism for clients to automatically discover and transition to encrypted DNS (DoE) resolvers. As a newly introduced protocol, its deployment and adoption on a global scale remain largely uncharted. This research question aims to provide a comprehensive analysis of the adoption rates of DDR-enabled resolvers in both IPv4 and IPv6 environments. Recognizing that numerical metrics alone may not fully encapsulate the complexity of DDR adoption. Thus, this research question also aims at investigating the adoption diversity across geographical regions, networks (e.g., Autonomous Systems (ASes)), and network categories (e.g., Internet service provider, enterprise, or academic networks). Through frequent, large-scale measurements of DDR-enabled resolver adoption

rates, this research seeks to reveal the dynamics and trends of the protocol's deployment over time.

RQ2: What configuration patterns are observed in DDR-enabled resolvers, and how do these patterns differ across networks and over time?

Each DDR-enabled resolver provides a response containing a set of encrypted resolvers to which clients can be delegated. This response includes all the necessary information for a client to seamlessly upgrade to a DoE protocol through the designated resolver. This research question examines the configuration patterns of DDR-enabled resolvers, focusing on the advertised DoE protocols and their respective configurations. By analyzing these patterns, it seeks to uncover the diversity of DDR deployments and explore how configuration practices vary across networks. Incorporating a temporal dimension, this question also aims to identify trends in configuration patterns observed in-the-wild over time, revealing how these practices evolve as the protocol matures.

RQ3: What observable challenges hinder clients from successfully transitioning from plain DNS to DoE protocols in real-world DDR deployments?

DDR's effectiveness in enabling secure and reliable transitions to DoE resolvers depends on the correctness and completeness of the configurations provided by DDR-enabled resolvers. For example, configurations that delegate clients to DoE resolvers incapable of supporting the specified protocols may hinder automatic upgrades. Additionally, DDR's specification requires clients to validate DDR responses, which may fail in cases of invalid or incomplete configurations. In the worst case, these challenges may force clients to fall back to unencrypted DNS, exposing end-users to privacy risks. This research question explores the observable issues that impact the functionality and reliability of transitions from plain DNS to DoE, aiming to uncover potential barriers and their broader implications.

1.3 Contributions

To the best of our knowledge, this study is the first to empirically examine the deployment, adoption, and configuration of the DDR protocol in real-world environments, with a comprehensive analysis for both IPv4 and IPv6. The key contributions of this study include:

- We developed an open-source, reactive, adaptable, and highly scalable three-stage measurement architecture called `DoE-Hunter`, written in Go [143]. The platform is fully containerized, monitored, unit-tested, and designed to operate across multiple Vantage Points (VPs), enabling efficient and large-scale (DDR) measurements.

- We conducted frequent large-scale DDR and DoE measurements on a semi-weekly basis over a four-month period, targeting both IPv4 and IPv6 resolvers. These measurements reveal adoption trends and deployment dynamics across geographical regions and networks. Our study highlights severe resolver consolidation induced by current DDR deployments, as >97% of DDR-enabled resolver delegate to major DNS cloud providers, raising concerns about privacy and governance. Additionally, we identified critical challenges preventing clients from successfully transitioning from plain DNS to DoE, including incomplete or incorrect DDR configurations. Alarmingly, in over 99% of cases, DDR-compliant clients may fail to transition to DoE resolvers due to these misconfigurations, underscoring the current limitations of DDR in the wild.

- As a byproduct of our measurements, we collected the largest known dataset of unique DoQ resolvers by Authentication Domain Names (ADNs) (>3K). This dataset could provide other researchers with valuable resources to study the DoQ protocol in greater depth.

- We contributed to Go's open-source DNS library [53] by implementing support for Service Binding and Parameter Record (SVCB)-related standards, including RFC 9540 [123] (Oblivious DNS over HTTPS (ODoH)). This contribution ensures full compliance with the latest standards and has been merged into the library's latest release [145].

- To the best of our knowledge, we are the first to measure the adoption of *recursive-to-authoritative* encrypted communication as specified in RFC 9539 [54]. By setting up an authoritative name server supporting DDR and DoE protocols, we probed discovered DoE resolvers in real-world deployments and found no evidence of resolvers employing encrypted communication with authoritative servers.

- We observed *traffic shadowing* in the context of DoE protocols, a phenomenon previously noted in research [165]. However, this study is the first to document *traffic shadowing* behavior in DoE, where recursive

resolvers replay DoE queries to authoritative name servers at intervals ranging from seconds to days. Some uniquely crafted DoE queries were replayed by 115 servers across 15 ASes, with instances exceeding 5K replays, predominantly originating from China.

- We present a first view on the Transport Layer Security (TLS) in the context of DoE protocols, providing an overview of the TLS versions and cipher suites negotiated by DoE resolvers. Our findings show that TLS 1.3 is the most negotiated version, followed by TLS 1.2, with no resolvers having negotiated older TLS versions. Additionally, we observe that all DoE resolvers comply with an early IETF draft [83], which specifies that resolvers discovered via DDR must not require client certificates, i.e., Mutual Transport Layer Security (mTLS).

1.4 Outline

This study is organized as follows: Chapter 2 provides an overview of the DNS, its architecture, and the key security and privacy challenges. It introduces the DoE protocols (DoT, DoH, DoQ, and ODoH) examined in this study, along with the DDR protocol, which plays a pivotal role in enabling seamless transitions to encrypted DNS. Chapter 3 surveys prior research on DNS datasets, measurement tools, encrypted DNS adoption, and security and privacy issues related to DoE protocols, while highlighting existing gaps addressed in this work. Chapter 4 justifies and describes the design of the `DoE-Hunter` measurement architecture, including its multi-stage approach to data collection and analysis, and discusses the ethical considerations of this study.

Chapter 5 examines the DNS resolvers observed in this study, providing context for the findings presented in Chapter 6 and Chapter 7. These chapters analyze DDR adoption trends, configuration patterns, and their implications for security, DNS centralization, and resulting privacy. They also assess the reliability and security of DoE protocols uncovered through DDR. Chapter 8 discusses the study's constraints and suggests potential directions for future research. Finally, Chapter 9 answers the research questions and underscores the broader implications of the findings.

2 Background

This chapter provides the foundational background necessary to understand the topics discussed throughout this study. We begin by outlining the DNS architecture and operations in Section 2.1, including its hierarchical structure, resolution processes, and the role of resource records. Next, the chapter explores protocols designed to enhance DNS security and privacy, with a focus on DoE protocols described in Section 2.3. We follow by an introduction to mechanisms for discovering encrypted DNS resolvers, including the DDR protocol, discussed in Section 2.4. Together, these sections provide a comprehensive overview of advancements in secure DNS communication.

2.1 Domain Name System (DNS)

Whenever a user or client attempts to connect to a resource on a network, the initial step oftentimes involves resolving the domain name (or hostname) of the resource to establish a connection on the IP layer. For instance, when accessing the website `www.google.de`, the domain name must be resolved into an IP address, which the computer then uses to initiate the actual connection. This resolution process is managed by the domain name system DNS [112, 113], which is a fundamental protocol of the Internet [50] and one of the largest distributed databases globally [76]. Despite its vast scale, the DNS operates with such efficiency that it remains largely invisible to users, functioning seamlessly in the background whenever a web page is visited, an email is sent, or any online transaction is carried out.

2.1.1 Database Structure and Delegation

The DNS is organized as a hierarchical tree structure [112, 128]. A resource within this structure is uniquely identified by a Fully-Qualified Domain Name (FQDN). A FQDN consists of a series of labels separated by periods (see red boxes in Figure 2.1). For example, the FQDN `de.wikipedia.org.` can be interpreted from right to left: starting from the root node, denoted by · (period), which represents a fixed number of root servers, the hierarchy progresses through

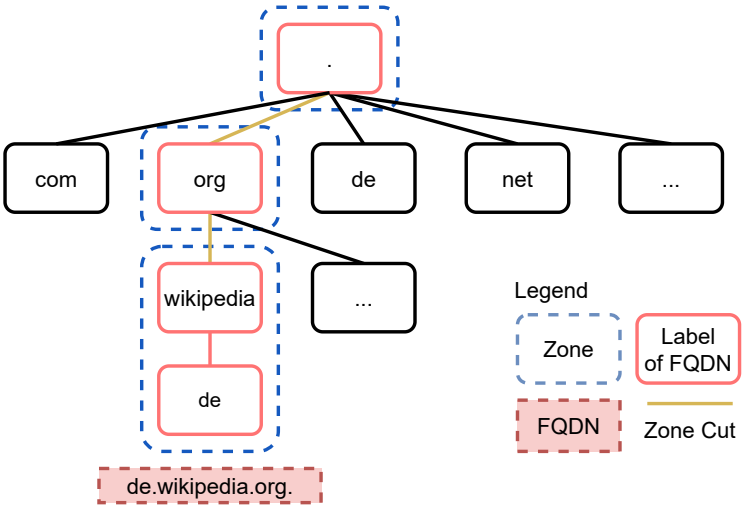


Figure 2.1: Exemplary organization of the hierarchical DNS structure.

the Top-Level Domain (TLD) `org`, followed by the Sub-Level Domain (SLD) `wikipedia`. In this instance, the FQDN encodes language preferences; accessing the subdomain `de` directs the user to the German-language version of the Wikipedia website. Thus, FQDNs not only serve as pointers to resources but also encode human-readable information about the resources being accessed.

Resources within the DNS reside in zones, as illustrated by the blue dotted boxes in Figure 2.1. Zones may include multiple labels from a FQDN within their scope. For instance, the labels `de` and `wikipedia` may be managed within the same zone. From an infrastructure perspective, zones are managed by authoritative DNS servers, which oversee the domain names and records within the boundaries defined by zone cuts. Additionally, each authoritative server maintains information about which entity or server is responsible for its child zones. For example, the server managing the `org` zone holds information about its child zones, including `wikipedia`. Consequently, it can delegate a DNS query to the appropriate child zone within the DNS hierarchy. This delegation information is stored in `NS records`.

2.1.2 Domain Name Resolution

The Recursive Resolver (RR) is preconfigured on the end-user device either automatically (e.g., via Dynamic Host Configuration Protocol (DHCP)) or manually (e.g., through configuration files). The RR attempts to resolve the desired DNS resource by traversing the DNS hierarchy in a top-down manner,

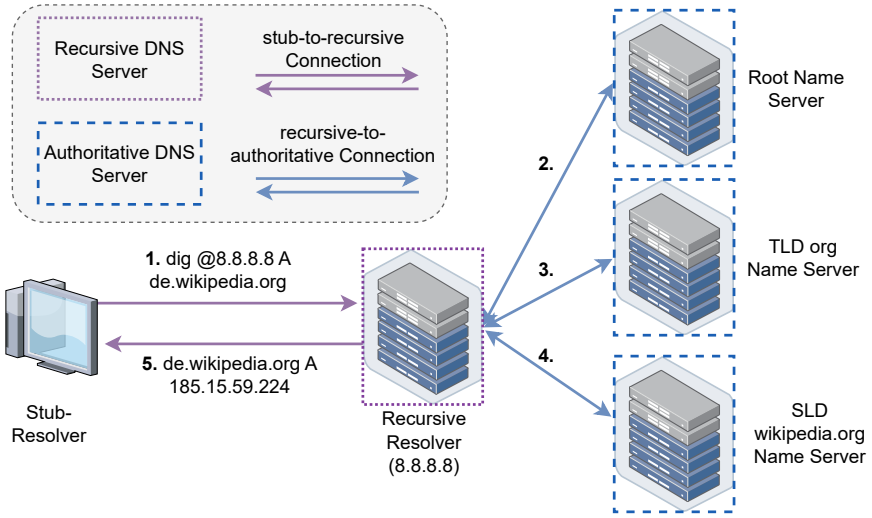


Figure 2.2: Exemplary process of recursively resolving a DNS record.

beginning at the DNS root name servers. Since each authoritative DNS server is aware of its child zones, it can provide information about the DNS server that might be responsible for the desired DNS resource. The RR iteratively repeats this process until the resource is found or the responsible DNS server responds with `NXDOMAIN`, indicating that the zone does not contain the desired DNS resource [9].

Recursive resolving offers several performance advantages for clients (see Figure 2.2). By delegating the resolution task to a third party, clients conserve resources for their primary functions instead of using them for DNS resolution. This is particularly advantageous for IoT devices, which typically have limited resources. Additionally, recursive resolving reduces the client's bandwidth usage, as it requires only a single DNS query to the RR instead of multiple queries to servers distributed across the Internet (iterative resolving). Most importantly, the RR can employ caching techniques to respond to DNS queries more quickly, thereby reducing the overall query load on the DNS infrastructure.

At the transport level, DNS queries and replies have historically been transmitted unencrypted and without integrity protection via DNS over UDP (Do53) [113]. Data in User Datagram Protocol (UDP) is encapsulated in datagrams, which are limited to a size of 512 bytes. If larger data needs to be transmitted, the DNS server can instruct the client to fall back to DNS over TCP (DoTCP53) by setting the truncation (TC) bit in the header of the DNS response (see also Section 2.1.3). However, DNS is a time-critical protocol that

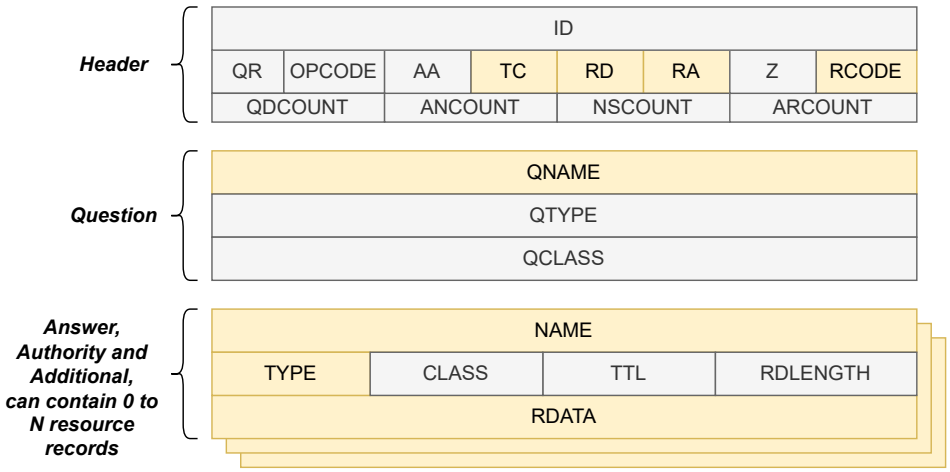


Figure 2.3: Format of a DNS message. Yellow highlighted fields are of particular interest in this study.

serves as the foundation for nearly every other protocol relying on domain names to connect to network services. Multiple round-trips and protocol changes can significantly slow down the DNS resolution process, thereby affecting the performance of dependent protocols. To address these challenges, Extension Mechanisms for DNS (EDNS0) was standardized to allow both communicating parties in the DNS resolution process to negotiate larger response sizes based on their respective capabilities [23].

2.1.3 Resource Records, Queries, and Responses

Structurally, both DNS queries and responses adhere to a standardized format comprising a header and four sections: `Question`, `Answer`, `Authority`, and `Additional` [113], as illustrated in Figure 2.3. In the following, we focus on fields of particular relevance.

The DNS message `Header` contains meta-information critical for managing queries and responses. The truncation bit (`TC`) indicates when a response exceeds the size limit for UDP, prompting the client to retry the request using Transmission Control Protocol (TCP) for reliable transmission. The recursion desired (`RD`) bit signals to the server that the client requests recursive resolution, whereas the recursion available (`RA`) bit specifies whether the server supports recursive resolving. This distinction enables differentiation between DNS servers functioning as RRs (with the `RA` bit set), indicating support for recursive resolution, and those that do not provide this functionality, for example

authoritative name servers (Non-Recursive Resolver (NRR)). Additionally, the `DNS Return Code (RCODE)` field encodes status and error codes. For instance, an RCODE value of zero signifies successful resolution with no errors, whereas a value of five indicates that the server has refused to answer the query.

The `Question` section contains the core DNS query, which is mirrored in the DNS response. The `Query Name (QNAME)` field specifies the FQDN of the resource being queried (unless QNAME Minimization (QNAME Min.) is utilized [8], see Section 2.2). The QNAME also determines the traversal path through the DNS hierarchy to locate the desired resource. However, exceptions exist. The Internet Assigned Numbers Authority (IANA) has defined a category of domains known as Special Use Domain Name (SUDN) [78], which are not resolvable within the public DNS hierarchy. Each SUDN serves a special purpose and typically points to resources that reside exclusively on the DNS server itself.

The `Answer`, `Authority`, and `Additional` sections provide actual DNS data in the form of Resource Records (ResRs) [112]. The `Answer` section contains the response to the actual DNS query, while the `Authority` section provides details about related authoritative DNS servers. The `Additional` section includes supplementary information that enhances the data from the other sections. A common example of such supplementary information is glue records, which provide IP addresses for FQDNs to reduce additional DNS lookups in certain cases.

Each section can include multiple ResRs. A ResR consists of a name (analogous to QNAME), a class, and a Time-To-Live (TTL) value specifying how long the information may be cached. Common types include `A` for resolving IPv4 addresses (see example in Section 2.1.2) and `AAAA` for IPv6. Other types include `NS` for encoding name server delegation information (see Section 2.1.1) and `MX` for encoding mail server exchange information. The actual payload data, such as the resolved IP address, is contained in the `RDATA` field.

SVCB Records

The resource record type SVCB, standardized in November 2023, allows clients to retrieve detailed information about accessing a service (e.g., an HTTPS service) via DNS [150, 151]. This feature enables clients to gather additional details about an endpoint through DNS before establishing a connection, thereby preserving privacy and enhancing performance.

For instance, when a client connects to a service via HTTP, it may be redirected to an HTTPS endpoint through an HTTP redirection. However,

Figure 2.4: Format of an `SVCB` ResR.

the initial connection via HTTP is unencrypted, potentially exposing sensitive information. By leveraging SVCB records, the client can retrieve information about the service and any delegations beforehand, avoiding such vulnerabilities.

While SVCB records are useful for HTTP and HTTPS scenarios, they are not limited to this use case. They support various scenarios by allowing arbitrary key-value pairs as payload data. For example, SVCB records are utilized in Encrypted ClientHello (ECH) [135] to enable encrypted handshakes, further enhancing user privacy. Additionally, they are employed in DDR to signal support for encrypted DNS protocols, as discussed further in Section 2.4.

The format of an `SVCB` record, shown in Figure 2.4, consists of three main fields. The `SvcPriority` field defines the record's priority relative to others, with lower values indicating higher priority. Priorities start at 1, while a value of 0 signifies `AliasMode`, which delegates to another resource [150], similar to the ResR `CNAME`. The `TargetName` field specifies the domain name of either the alias target or an alternative endpoint for the service. Finally, the `SvcParams` field contains optional key-value pairs providing additional configuration information for client connectivity. For instance, the `alpn` key lets the server indicate supported protocols, removing the need for protocol negotiation.

2.1.4 DNS Authenticity — DNSSEC

DNS, in its original standardization [112, 113], lacks built-in security mechanisms [109], making it an attractive target for malicious actors. One prominent vulnerability is cache-poisoning [109], which targets DNS resolvers (e.g., RRs) by exploiting their trust in upstream authoritative name servers. An off-path malicious actor tricks the resolver into sending a query to an authoritative name server and then attempts to inject a fake response with a spoofed IP address of the authoritative server. If the fake response arrives before the legitimate one and correctly matches the query's parameters, the resolver accepts and caches the malicious response. Subsequent client requests

for the resource record will return the maliciously altered cached entry until the TTL expires.

These vulnerabilities stem from the absence of integrity and authenticity mechanisms for DNS responses. DNS Security Extensions (DNSSEC) addresses this issue by introducing security extensions and additional ResRs, enabling public-key authentication and integrity verification for DNS responses. It also provides mechanisms for authenticated denial of the existence of ResRs. DNSSEC introduces four new ResRs: Resource Record Signature (RRSIG), DNS Key (DNSKEY), Delegation Signer (DS), and Next Secure (NSEC).

Authentication is achieved through cryptographic signatures generated over a set of ResR types within the respective zone file. These signatures are stored in the same zone as RRSIG records (see Figure 2.5). Public keys are maintained in DNSKEY records within the respective zone, while the parent zone retains hashed information about the child zone's DNSKEY in DS records, thereby forming an authentication chain. This chain allows a resolver to verify the authenticity of DNS responses by validating each signature, starting from the root name server and proceeding to the authoritative server that holds the requested information.

However, DNSSEC does not inherently prevent a malicious actor from returning a DNS response with a status code of `NXDOMAIN`, which indicates the absence of a resource record. This limitation arises because only existing ResRs are signed within the zone file. To address this, DNSSEC introduces the NSEC ResR. Domain names within a zone are arranged alphabetically, and the NSEC record provides the next domain name in the sequence. As the NSEC record is itself a ResR, it can be verified using cryptographic signatures, offering a secure method to confirm the non-existence of a ResR within a zone.

While DNSSEC ensures the authenticity and integrity of DNS data by preventing tampering and forgery, it does not provide confidentiality, meaning an eavesdropper can still observe the queries being made by users. Consequently, sensitive information such as the websites a user is attempting to visit remains visible to third parties, leaving privacy unprotected.

2.2 DNS Security and Privacy

DNS is a fundamental component of the Internet, supporting services such as email communication, financial transactions, and basic website access [104, 147]. Traditionally, DNS queries and responses are transmitted over UDP, and the majority of DNS traffic remains unencrypted [70, 164], posing privacy

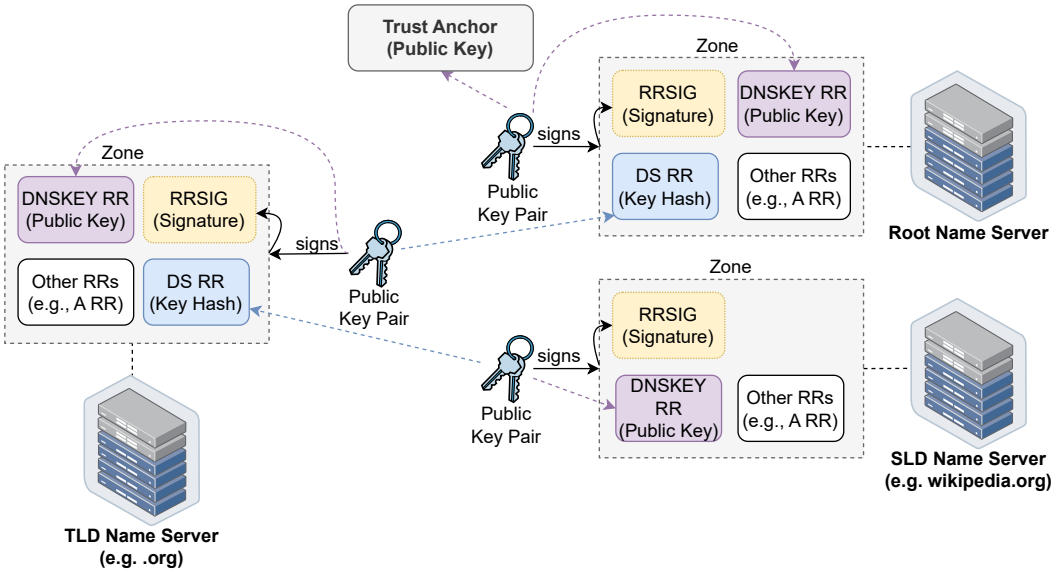


Figure 2.5: Simplified representation of DNSSEC's extensions in DNS.

risks. Unencrypted DNS traffic enables eavesdroppers to observe users' online activities and infer detailed user profiles [60, 73, 94]. Beyond simply revealing the websites a user has visited, basic web traffic can expose additional information. For instance, visiting a website often generates multiple DNS queries [164]. The primary DNS query resolves the website's FQDN into its corresponding IP address, while secondary DNS requests may be issued to load embedded content, such as media files, JavaScript libraries, or images [164]. These additional requests can reveal further user details, as embedded content may depend on the user's browser configuration, such as language settings, which may be reflected in the FQDN. In addition to privacy concerns, unencrypted DNS traffic can be exploited by malicious actors to inject or manipulate DNS packets, especially when DNSSEC is not deployed. This vulnerability exposes systems to attacks such as DNS cache poisoning [2]. Such exploits have been used in the past by state actors, including the NSA, which monitored and manipulated DNS traffic through projects like *MoreCowBell* [57] and *QuantumDNS* [104].

In response to these risks, the IETF has declared DNS monitoring to be "an attack on the privacy of Internet users and organizations" [48]. Pervasive Monitoring (PM) refers to widespread and often covert surveillance involving the intrusive collection of protocol artifacts, including application content and metadata such as headers. This surveillance includes both active and passive

wiretapping, as well as traffic analysis. The IETF classifies PM as an attack because it subverts the intended communication between parties without their consent, and the intent of the monitoring actor cannot be reliably distinguished from that of a malicious attacker [48].

As a result, the IETF declared DNS monitoring as "an attack on the privacy of Internet users and organizations" [48]. PM refers to widespread and often covert surveillance involving the intrusive collection of protocol artifacts, including application content and metadata, such as headers. This form of monitoring includes both active and passive wiretapping as well as traffic analysis. PM is considered an attack by the IETF because it subverts the intended communication between parties without their consent, and the intent of the monitoring actor cannot be reliably distinguished from that of a malicious attacker [48].

In 2009, *DNSCurve* [25] was introduced to provide authentication and confidentiality for *recursive-to-authoritative* DNS communication, thereby mitigating both active and passive attacks. However, *DNSCurve* does not secure *stub-to-recursive* communication [4]. To address this gap, *DNSCrypt* was developed in 2011 [26]. Unlike TLS (see Section 2.3.1), *DNSCrypt* does not rely on the X.509 Public Key Infrastructure (PKI). Instead, it uses an out-of-band method to retrieve the resolver's public key, which is then used to verify the key material exchanged between the client and the resolver, ensuring encryption and authentication of DNS packets [26].

To address PM and security challenges in DNS, several protocols have been developed to improve privacy and integrity of the DNS system [147]. In 1997, DNSSEC has been standardized by the IETF to preserve origin authentication of DNS data [63]. However, it mitigates only threats that target to manipulate DNS zone data [97] but does not provide any privacy features. Additionally to DNSSEC, *DNSCurve* [25] was designed in 2009 to mitigate active and passive attacks by providing authentication and confidentiality to recursive-to-authoritative DNS communication. However, it does not provide stub-to-recursive security [4]. *DNSCrypt* was developed in 2011 to tackle this gap [26]. Different to TLS (see Section 2.3.1), it does not rely on the X.509 PKI but uses an out-of-band method to retrieve the resolver's public key which is used to verify the key material that is exchanged between client and resolver to encrypt and authenticate the DNS packets [26]. *DNSCrypt* is actively maintained [36] and also supported by some RRs [149]. However, neither *DNSCurve* nor *DNSCrypt* have ever been standardized by the IETF [25, 36]. *DNSCrypt* remains actively maintained [36] and is supported by some

RRs [149]. However, neither *DNSCurve* nor *DNSCrypt* has been standardized by the IETF [25, 36].

Another protocol standardized to enhance DNS privacy is QNAME Min. [8]. Unlike cryptographic protocols such as *DNSCurve* or *DNSCrypt*, which encrypt DNS traffic, QNAME Min. focuses on minimizing the information included in the DNS query itself, specifically within the QNAME field (see Section 2.1.2). Since the QNAME reveals the specific resource the client is attempting to resolve, it exposes sensitive details about user behavior [164]. Within the DNS hierarchy (see Figure 2.1), authoritative servers — particularly root and TLD authoritative servers — can gather considerable user information, as the full QNAME is typically forwarded from the RR without modification [164].

QNAME Min. mitigates this issue by instructing RRs to send only the minimal necessary portion of the query to the upstream name server required for continuing the resolution process. For instance, when resolving `de.wikipedia.org` (see Figure 2.2), a QNAME Min.-enabled resolver would first send `.org` to the root node, followed by `wikipedia.org` to the TLD authoritative server, and finally the complete QNAME, `de.wikipedia.org`, to Wikipedia's authoritative server. This ensures that only information relevant to the resolution process is passed to each authoritative server, thereby reducing the exposure of user-related data.

However, this approach requires the RR to be aware of zone cuts, which do not necessarily align with every label boundary (see Figure 2.1). Additionally, QNAME Min. does not address the exposure of *stub-to-recursive* DNS queries and responses, leaving them vulnerable to eavesdropping.

Stub-to-recursive communication remains the most advantageous interception point for eavesdroppers, as this traffic is unaffected by DNS caching. To address this vulnerability, the IETF has standardized protocols designed to secure the integrity and confidentiality of *stub-to-recursive* DNS traffic. These protocols, commonly referred to in the literature as DoE protocols, aim to mitigate security and privacy concerns [104].

2.3 DNS over Encryption (DoE) Protocols

DoE protocols leverage existing connection protocols to securely transmit DNS queries and responses over encrypted channels, ensuring both the authenticity and integrity of the transmitted data. Although initially introduced to secure *stub-to-recursive* communication, technically they can also be used to secure *recursive-to-authoritative* communication [54]. Three such protocols have been

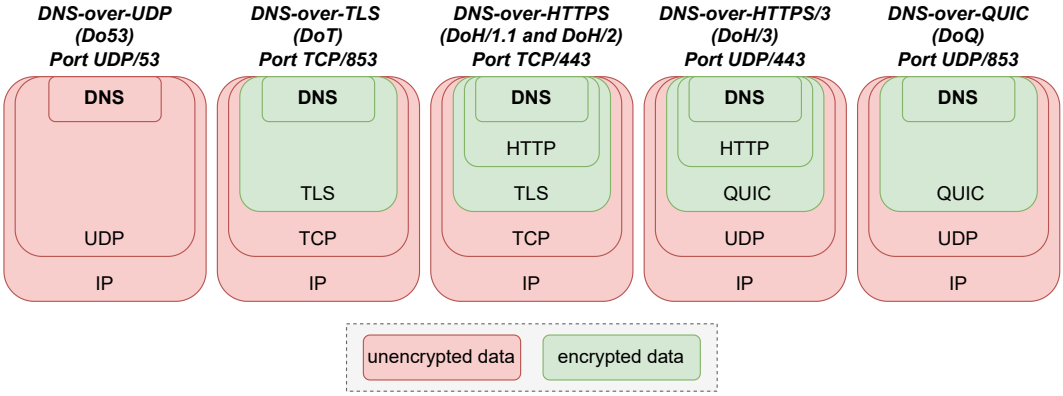


Figure 2.6: The structure of the packet layers starting from the IP layer [107].

standardized: DoT in May 2016 [70], DoH in October 2018 [64], and DoQ in April 2024 [72]. To provide their security characteristics, they rely on TLS and its PKI based on X.509 certificates [21]. In the case of the Quick UDP Internet Connections (QUIC) protocol [81], the TLS 1.3 handshake is integrated directly into the protocol itself [159].

2.3.1 Transport Layer Security (TLS)

TLS is designed to create a secure communication channel between two endpoints over an insecure network, ensuring confidentiality, integrity, and authenticity. Currently, TLS 1.2 [32] and TLS 1.3 [134] are the most widely deployed versions [129], with one of its primary applications being HTTPS [133] on the web. However, TLS is a versatile protocol used in various Internet applications.

TLS consists of two main components: the record protocol and the handshake protocol. The record protocol is responsible for the integrity-protected and encrypted transmission of data using symmetric session keys. Meanwhile, the handshake protocol handles server (and client) authentication and establishes the session key for the record protocol.

To ensure authenticity of the server and optionally the client in the handshake phase, TLS relies on the X.509 PKI [21]. This PKI establishes authentication chains through a hierarchical structure (similar to DNSSEC), where trust is anchored in a root Certificate Authority (CA) and cascades down through intermediate CAs, enabling secure verification of digital certificates across a network. This hierarchical model ensures that each entity's identity is

authenticated by a trusted authority, creating a chain of trust from the root CA to the end user. Thus, a client can validate the identity and authenticity of a server's response, for example.

2.3.2 DNS over TLS (DoT)

DoT is a DoE protocol designed to securely transmit DNS queries and responses over a TLS-encrypted channel, ensuring privacy and integrity of the exchanged data [70]. While initially defined for *stub-to-recursive* communication (see Figure 2.2), DoT can also be extended to secure *recursive-to-authoritative* server connection.

By default, DNS servers that run DoT should use port TCP/853 unless an alternative arrangement is negotiated between both parties [70]. Clients are responsible for maintaining knowledge of which DNS servers support TLS and must be capable of handling connections with both encrypted and unencrypted resolvers. When establishing a connection, clients are expected to follow the TLS protocol, including the proper verification of the server's certificate chain in accordance to best practices for secure communication [154].

Once the TLS handshake is completed, DNS queries and responses are exchanged over the TLS channel via the DNS wire format [113], a byte-level representation of DNS messages. This ensures that while the underlying data remains consistent with existing DNS standards, the communication is protected against eavesdropping and tampering.

2.3.3 DNS over HTTPS (DoH)

DoH is a protocol designed to secure the transmission of DNS queries and responses by encapsulating them within HTTPS traffic, leveraging TLS to provide both authenticity and encryption [64]. In contrast to DoT, which operates DNS directly over TLS, DoH utilizes the intermediate layer HTTP between DNS and TLS (see Figure 2.6). This architectural difference allows DoH to take advantage of various HTTP features, such as redirection, proxying, client authentication, compression, and response format negotiation. Furthermore, DoH allows HTTP clients such as web applications to interact with the DNS ecosystem seamlessly by standard APIs. However, a challenge with DoH lies in the potential ambiguity surrounding caching mechanisms and error codes, since both HTTP and DNS natively implement these features. DoH can operate on TCP and TLS with HTTP/1.1 (legacy) and HTTP/2. In addition, it also

supports HTTP/3, which operates over UDP and uses QUIC as its transport layer, similar to DoQ. By default, HTTPS traffic runs on port 443 [133].

Unlike DoT and DoQ, where clients only need to know the resolver and the port to exchange DNS information, DoH requires additional configuration. Specifically, DoH clients must be aware of the specific URI path where the DNS queries are processed. For instance, Google's DoH resolver accepts queries at `https://dns.google/dns-query`. Additionally, DoH supports both HTTP GET and HTTP POST methods for transmitting DNS queries. When using HTTP POST, the DNS query is encoded in its wire format and embedded in the body of the HTTP request. For HTTP GET, the query is base64url [87] encoded and appended to the URI as a query parameter. For example, a GET request to Google's resolver would appear as `https://dns.google/dns-query?dns=<b64-dns-query>`. Still, the DoH standardization specifies a URI path `dns-query` and parameter `dns`, but leaves a different configuration to the resolver.

2.3.4 DNS over QUIC (DoQ)

QUIC is a secure, general-purpose, connection-oriented transport protocol built on top of UDP [81]. QUIC is specifically designed to minimize protocol-induced delays [72]. This is achieved through features such as 0-RTT data transmission during session resumption, advanced packet-loss recovery mechanisms, the mitigation of head-of-line blocking, and parallel delivery of data across multiple streams. Unlike TCP, which provides reliability and ordered packet delivery natively, UDP offers only minimal protocol mechanisms without any packet order or delivery guarantees. QUIC addresses these limitations by implementing reliability, flow control, and packet ordering at the transport layer, ensuring the features necessary for reliable communication. This is particularly important since QUIC utilizes the TLS 1.3 handshake to negotiate session keys to provide authentication, encryption and integrity protection of the application data exchanged. Without a reliable transport layer, the handshake process cannot be guaranteed to complete successfully. However, unlike traditional TLS, which provides packet protection through its record layer, QUIC introduces its own transport layer with so-called `CRYPTO` frames.

DoQ is a protocol that maps the DNS protocol onto the QUIC transport layer [81, 159], offering a modern and secure alternative to traditional DNS transport mechanisms like Do53. It provides the same security and privacy guarantees as DoT, including server authentication and its defined usage profiles (see Section 2.3.2) but with the benefits of the QUIC protocol. By default, a

DNS server supporting DoQ listens for QUIC connections on UDP port 853, unless a different port is mutually agreed upon.

In contrast to DoT, where multiple DNS queries are multiplexed over a single connection, DoQ allocates a separate QUIC stream for each individual DNS query. The server replies on the same stream, ensuring efficient, stream-specific communication. The structure enhances the protocol's flexibility by allowing concurrent queries without the head-of-line blocking that occurs in TCP-based protocols which limits throughput.

DNS queries and responses in DoQ are encoded in the DNS wire format [112]. Additional to the `RCODEs` a DNS response provides (see Section 2.1.3), DoQ introduces new error codes specific to its transport layer. For example, `DOQ_EXCESSIVE_LOAD` signals the client that the server is under heavy load and must close the connection to manage its resources. These additional error codes provide more granular control over error handling and resource managed.

2.3.5 Oblivious DNS over HTTPS (ODoH)

In general, DoE protocols (see Section 2.3) ensure encryption and integrity protection for DNS queries and responses exchanged between stub-resolver and RR. However, despite this protection, the RR can still identify the client through its IP address, potentially exposing sensitive information such as the client's identity and location. As a result, the RR can track user behavior in a manner similar to eavesdropper intercepting unencrypted DNS traffic (Do53). To enhance client privacy, the IETF introduced ODoH as a standard in June 2022 [88]. ODoH builds upon DoH by enabling proxied resolution, wherein DNS messages are encrypted in such a way that no single server has access to both the client's IP address and the contents of the DNS queries.

Figure 2.7 demonstrates the protocol in a simplified version. The communication between the stub-resolver and the proxy, as well as between the proxy and the RR, occurs over DoH (see Section 2.3.3). The stub resolver uses the public key to encrypt its DNS query and encapsulate an ephemeral symmetric key. The public-key cryptography scheme used provides confidentiality and integrity protection. The stub-resolver begins by retrieving the public key of the RR through an unspecified method to encrypt its DNS query and an ephemeral symmetric key (blue boxes). This encryption provides integrity and confidentiality of the encrypted data through Hybrid Public Key Encryption (HPKE). Along with the encrypted query, the stub-resolver includes routing information (i.e., `targethost` and `targetpath`), which indicates where the proxy should forward the encrypted DNS query. Once the encrypted

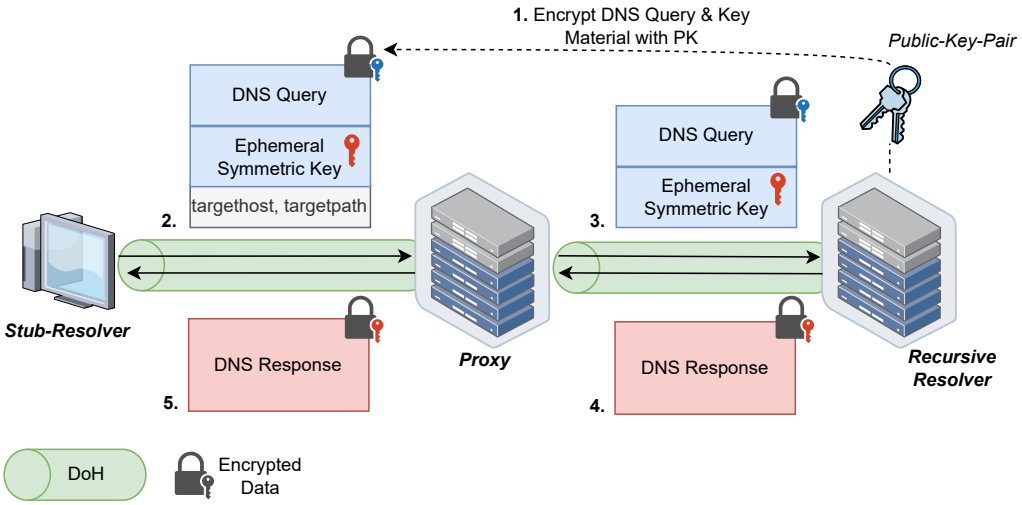


Figure 2.7: Simplified illustration of the ODoH protocol. Green tunnel symbolizes the DoH protocol. The blue boxes are encrypted using the recursive resolver's public key, the red boxes are encrypted using the stub-resolver's ephemeral symmetric key.

DNS query reaches the proxy, it forwards the query to the RR. Since the DNS query and ephemeral symmetric key are encrypted with the RR's public key, only the RR can decrypt the query and extract the symmetric key using its private key. The RR resolves the desired DNS information and forms an encrypted DNS response with the ephemeral symmetric key using Authenticated Encryption with Associated Data (AEAD) scheme which ensures confidentiality and integrity protection. This prevents the proxy from inferring the contents of the DNS response or modifying them. The RR sends the encrypted DNS response back to the proxy, which, in turn, forwards it to the stub-resolver. Since the stub-resolver generated the symmetric key earlier, it can validate and decrypt the DNS response.

Although an active malicious actor could potentially manipulate the routing information (`targethost` and `targetpath`), the RR's private key ensures that only it can decrypt the query and respond with valid, encrypted data. Thus, the integrity of the DNS response remains intact, as only the resolver itself has access to the ephemeral symmetric key material.

2.3.6 Security and Privacy Considerations

Since every DoE protocol builds on top of TLS, they inherit the same security characteristics of the respective version, which have been elaborated in [153].

However, security and privacy concerns emerge when a client attempts to discover a DNS resolver that supports a DoE protocol. Typically, the clients probe on the DoE protocol's default port (e.g. 853 in the case of DoT) to establish a connection [29]. This opens an attack surface on redirect and downgrade attacks. For example, an attacker on the wire could simply drop the probing packets to leave the client on unencrypted Do53 traffic. Thus, RFC 8310 [29] specifies two usage profiles that attempt to increase user privacy: the *Opportunistic Privacy profile* and the *Strict Privacy profile*. Although these usage profiles are specifically defined for DoT, they conceptually apply to all DoE protocols.

In the *Opportunistic Privacy profile*, the client prefers to maintain privacy but does not enforce strict requirements for resolver authentication [29]. For example, the client may use a resolver discovered through an untrusted source, such as one provided via DHCP, and attempt to establish a TLS connection on the DoE protocol's default port. While this configuration offers protection against passive eavesdropping (simply because a DoE protocol is used), it leaves the client vulnerable to active downgrade or redirection attacks since the resolver's identity may not be verified. Thus, the privacy benefits are limited, particularly in environments with active adversaries.

Conversely, the *Strict Privacy profile* assumes that a pre-existing trust relationship has been established between the client and the DoE resolver [29]. For example, administrators may supply the client with the resolver's public key fingerprint, similar to SSH's method of server key verification [47], which is also known as key pinning. The client stores the fingerprint and verifies the resolver during future connections, ensuring the resolver's authenticity and effectively preventing downgrade attacks. However, if a downgrade attack occurs, the client may be not able to access the DNS, leaving Denial of Service (DoS) attacks open. This profile offers stronger protection as the opportunistic profile, as it requires authentication of the resolver by relying on manual or preconfigured trust.

2.4 Encrypted DNS Server Discovery — DDR

While DoE protocols provide confidentiality and integrity for the *stub-to-resolver* communication, none of the DoE protocols offer provisions for resolver selection by client applications [164]. As such, stub-resolvers have limited means to discover encrypted DNS resolvers and end up relying on the (often unencrypted) resolver assigned by the Internet Service Provider (ISP) via DHCP.

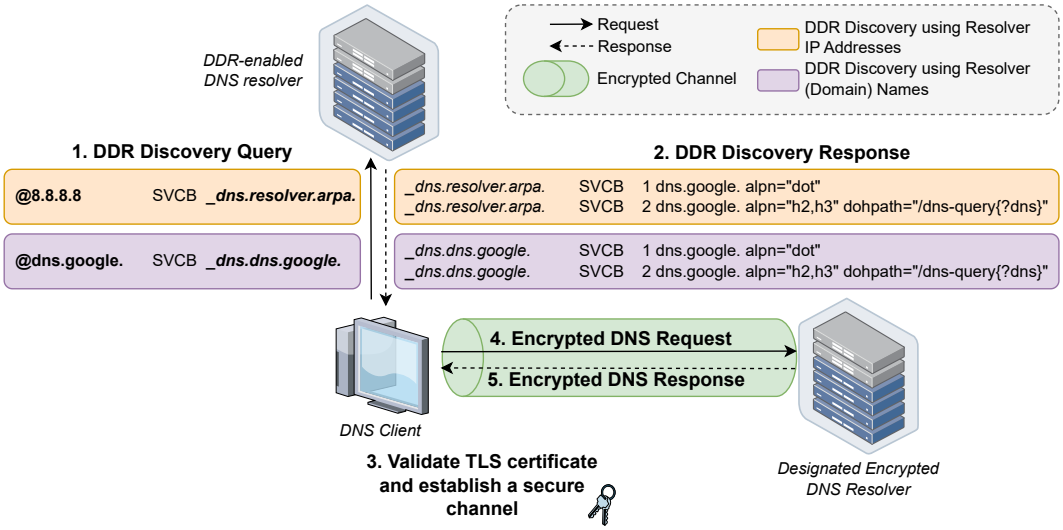


Figure 2.8: Illustration of the DDR protocol. Orange boxes indicate the DDR discovery query and response using the resolver IP address, while purple boxes indicate the same discovery using the resolvers' domain name. We use Google's DDR-enabled resolver (`8.8.8.8`/`dns.google`) to illustrate a sample response. Bold text highlights differences between the DDR discovery queries and responses.

In addition, users frequently remain unaware of the privacy risks associated with defaulting to unencrypted DNS, and there is insufficient understanding on how to effectively transition to encrypted DNS configurations [64]. Consequently, there is a pressing need for mechanisms that enable clients to both identify available encrypted DNS services and automatically switch to them without requiring user intervention. To address this gap, the IETF has standardized DDR, a mechanism for clients to use SVCB ResRs (see Section 2.1.3), to leverage DNS queries to discover a resolver's encrypted DNS configuration [124]. The encrypted DNS resolvers discovered by this mechanism are called "Designated Resolver(s)" operated by the same entity or cooperating entities [124]. With DDR, stub-resolvers are enabled to automatically upgrade from unencrypted DNS to encrypted DNS. Although the DoE protocols target secure *stub-to-resolver* communication, DDR can also be applied to discover DoE endpoints of authoritative endpoints, i.e., *stub-to-resolver* DoE traffic [54]. The DDR discovery process is illustrated in Figure 2.8. DDR consist of two steps: the encrypted resolver discovery, and the encrypted resolver verification.

2.4.1 Discovery Queries and Types

DDR supports two types of discoveries [124] (see Figure 2.8): **discovery using resolver IP addresses** (orange boxes) and **discovery using resolver (domain) names** (purple boxes).

DDR's *discovery using resolver IP addresses* applies to scenarios where the client knows only the IP address of a DNS resolver (e.g., a RR) and seeks to discover and upgrade to DoE endpoints. In this case, the client can issue a DNS query of type SVCB with the query name `_dns.resolver.arpa.` directed to the resolver's IP address [124]. The `resolver.arpa.` domain is a SUDN, serving as a locally defined zone specifically designated for DDR discovery [14, 124].

As a SUDN, `resolver.arpa.` is not part of the public DNS hierarchy and cannot be resolved recursively (see Section 2.1.1). Instead, the ResRs within the `resolver.arpa.` zone are locally served, enabling efficient discovery of DoE endpoints without dependence on the traditional DNS hierarchy.

Conversely, the *discovery using resolver (domain) names* applies to scenarios where the client already knows an encrypted resolver by its FQDN and seeks to determine the resolver's supported DoE protocols or its current configuration [124]. In this process, the client issues an SVCB query with the query name `_dns.<FQDN>`. This query can either be sent directly to the resolver or resolved recursively, as — different to the first discovery method — the SVCB resource resides within the public DNS hierarchy (see Section 2.1.1).

For instance, if the client already knows the encrypted resolver `dns.google.`, it can issue an SVCB query with the QNAME `_dns.google.dns.` to retrieve its current DoE configuration (see purple boxes in Figure 2.8).

2.4.2 Discovery Response

In response to the DDR discovery, the resolver returns a set of SVCB ResRs [150]. Each SVCB record contains a priority field (`SvcPriority`), the encrypted resolver's domain name (`TargetName`), and the supported protocol(s), specified in the `SvcParam` *alpn* (application layer protocol negotiation) fieldFigure 2.4. For example, the DDR response from Google's DNS resolver advertises the DoT protocol on its default port 853, at the domain `dns.google.`, with the highest priority set to 1 (see Figure 2.8). A secondary advertised resolver supports both DoH/2 and DoH/3 on port 443 with a lower priority of 2, and the URI path `/dns-query{?dns}` (see Section 2.3.3). Ultimately, the decision regarding which DoE resolver to use remains with the client.

Further, DDR also allows for the advertisement for deviating default ports. For example, if the encrypted resolver runs DoH/2 on 8884 instead of the default port 443, the DDR-enabled resolver can advertise a deviating port with `port=8884`, for example.

An encrypted resolver is advertised by DDR with its FQDN. Thus, if a client chooses to upgrade to the advertised encrypted resolver, it first needs to look up its IPv4 or IPv6 address. To minimize these additional Round-Trip Times (RTTs), the DDR-enabled resolver can advertise the encrypted resolver's IP addresses with `ipv4hint` and `ipv6hint`.

The DDR-enabled resolver can indicate ODoH (see Section 2.3.5) by setting the `SvcParam ohttp` without any value, i.e., just `ohttp` [123]. Additional information about public key material and proxy information can be retrieved using a `.well-known/` URI at the advertised encrypted resolvers domain name.

2.4.3 Discovery Verification

In all cases, when a client performs DDR to discover encrypted resolvers, it must verify the response [124]. The verification method depends on the discovery approach employed, whether it involves *discovery using resolver IP addresses* or *discovery using resolver (domain) names*.

If a client performs DDR's **discovery using resolver IP addresses**, two verification methods are available: *verified discovery* and *opportunistic discovery*. Alternative approaches, such as policy- or heuristic-based methods, are left to the discretion of the client.

The *verified discovery* method requires two verification steps that a client must complete before accepting an automatic upgrade to an encrypted resolver. First, the certificate chain presented during the TLS handshake must be valid (see Section 2.3.1). Second, the IP address of the DDR-enabled resolver advertising the encrypted resolver must be included in the subjectAltName TLS extension (SAN) (Subject Alternative Name) field of the encrypted resolver's certificate. The SAN field allows additional FQDNs or IP addresses to be specified, under which the certificate is also valid [69, 142]. For instance, if a server is accessible through two FQDNs, such as `example.com` and `example.de`, the latter can be listed in the SAN field to validate the certificate for both domains.

However, the verified discovery process may not always be applicable [124]. For example, when a client attempts to verify an advertised encrypted resolver accessible via a local IP address, the verification process may fail. This limitation arises because local IP addresses are not globally unique [132] and therefore cannot definitively establish ownership or control of the advertised service. In

such cases, the decision to use the encrypted resolver defaults to the client, following the `opportunistic discovery` method. Under this approach, clients may rely on the information in the SVCB records only if the IP address of the advertised encrypted resolver matches that of the DDR-enabled resolver.

If a client performs DDR's **discovery using resolver (domain) names**, it must execute the verified discovery procedure, similar to the one specified for *discovery using a resolver's IP address* [124]. In this process, the domain name of the DDR-enabled resolver, on which the DDR discovery query was executed, must appear in the encrypted resolver's certificate SAN field. For instance, in the example shown in Figure 2.8, the encrypted resolver name `dns.google.` must correspond to the advertised encrypted resolver name, which, in this case, is also `dns.google.`.

Encrypted resolvers may, however, delegate to other encrypted resolvers. For example, if the DoH service is provided by `doh.dns.google.`, the certificate for `doh.dns.google.` must include `dns.google.` in the SAN field to enable proper resolver verification. Additionally, since the SVCB record resides within the public DNS hierarchy, DNSSEC can be applied to DDR responses to provide additional response authentication (see Section 2.1.4). However, DNSSEC is not explicitly addressed in the DDR specification.

If any of the verification methods outlined above fail, the client should disregard the information in the DDR response [124]. In such cases, the client would fall back to traditional, unencrypted Do53 traffic.

2.4.4 Encrypted DNS Server Redirection — EDSR

Complementary to DDR, the IETF is, at the time of writing this study, developing an additional protocol that leverages encrypted DNS resolver redirections: Encrypted DNS Server Redirection (EDSR) [160]. This protocol allows an encrypted DNS resolver already known to clients to redirect them to alternative resolvers that may better suit their needs. This redirection process eliminates the need for anycast support or pre-configuration of the client with alternative resolvers. For example, to reduce network latency, the resolver might redirect a client to an unencrypted resolver that is geographically closer, thereby improving performance. Unlike DDR, this protocol applies only when the initial DNS traffic is already encrypted, meaning it operates after DDR has been completed.

The execution of the EDSR protocol proceeds as follows (see Figure 2.9): The client initiates a DDR discovery query to the encrypted resolver (`_dot.example.com`), and if redirection is needed, the resolver includes a glue

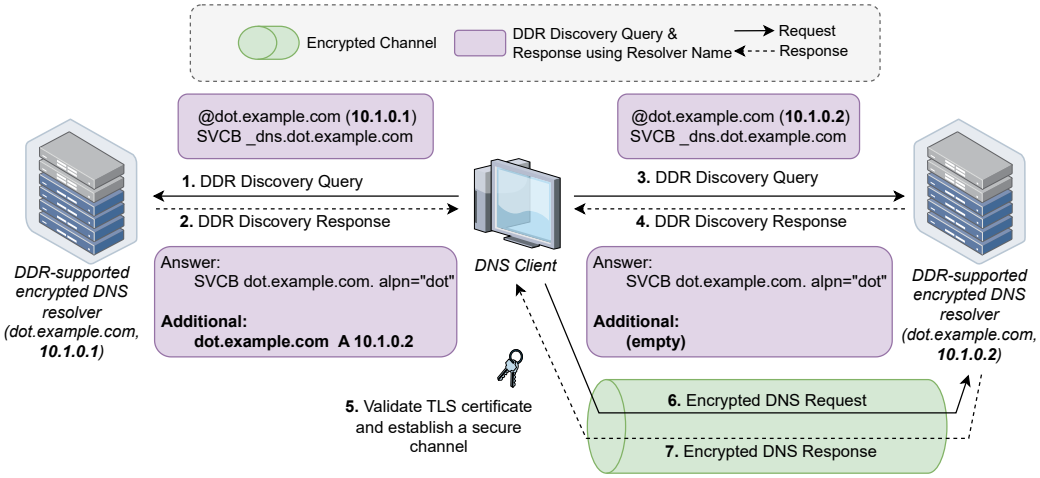


Figure 2.9: Illustration of the EDSR protocol. The purple boxes represent DDR's discovery of encrypted resolvers using resolver names. Bold text highlights the differences in queries and responses. This example depicts a single redirection; however, multiple redirections are possible.

record in the response's additional section (see Section 2.1.3), pointing to another encrypted resolver with a different IP address. The client repeats this process by querying the newly discovered IP address, continuing until the IP address in the additional section matches the resolver's or the glue record is absent. At this point, the client connects to the encrypted resolver using the advertised DoE protocol, verifies its certificate as defined by the DDR specification, and starts an encrypted DNS communication. To avoid excessive or infinite redirection chains, clients may detect loops, limit redirection hops to a predefined maximum, or cease following redirections at any stage [160].

3 Related Work

This chapter reviews prior work that forms the foundation of this study. We examine recent research on existing DNS datasets (see Section 3.1), tools and methodologies for DNS measurements (see Section 3.2), and studies exploring encrypted DNS communication (see Section 3.3). Additionally, we discuss security and privacy aspects of encrypted DNS communication (see Section 3.4) and address the emerging topic of DNS centralization (see Section 3.5).

3.1 Actively and Passively collected DNS Datasets

Numerous recent studies have conducted Internet-wide scans for various purposes, leading to the creation of a wide range of datasets. These datasets provide researchers with a more comprehensive and representative understanding of the diverse nature of Internet traffic dynamics [3], while also minimizing the error-prone, tedious and cost-intensive development and execution of measurements [43]. This section offers an overview of datasets related to DNS, which can be broadly classified into two categories: those collected through active probing and querying of publicly available hosts, and those gathered through passive monitoring of DNS traffic.

Censys is a measurement platform designed to enhance the accessibility of Internet-wide scanning by offering a near real-time and comprehensive "bird's view eye" on the Internet [41, 43]. *Censys* actively collects a wide range of data and metadata about IPv4 and IPv6 hosts such as availability, geographic location, open services by port, and protocol specific details (e.g., TLS version or cipher suites offered). In the context of DNS, *Censys* provides information about DNS hosts, whether they function as public resolver or authoritative servers, when they were last seen, and whether they responded correctly to DNS probes. The data collected is made publicly available through a search engine where the dataset can be filtered on specific labels. For instance, applying the label `services.service_name: DNS` returns in around 5M hosts running a DNS service. Additionally, the data can be queried through a REST API or via *Google BigQuery*, where SQL-based queries are supported. While access to the platform is generally limited to the public, researchers are eligible to gain free

access. In its early developments stages, *Censys* relied on *ZMap* [82] and *ZGrab* for its scanning architecture [43]. *ZMap* was used to quickly identify responsive IP addresses on specific ports, while *ZGrab* in a second stage conducted specific protocol handshakes, such as for TLS. Over time, *Censys* has developed its proprietary scanning tools.

Shodan [155] is one of the most widely used public platforms for actively scanning and cataloging IoT devices [163]. Like *Censys*, *Shodan* collects a variety of metadata about hosts, such as host names, geolocation, supported IP versions, or available services. It was primarily designed to assist in vulnerability assessment and penetration testing of online resources [163]. While it is generally a paid service for the public, researchers are granted a limited number of monthly queries at no cost. Similar to *Censys*, they allow data filtering based on labels. However, its filtering capabilities for identifying DNS-related hosts are limited to searching for services running on port 53, which is typically associated with DNS but does not accurately distinguish DNS servers from other services that may use the same port. As of the writing of this thesis, *Shodan* has identified approximately 8M hosts with a service running on port 53.

Rapid7 Open Data [130] provides access to internet telemetry data. It utilizes data from the Project Sonar [131], which was launched in September 2013 with the goal of enhancing security via active analysis of the IPv4 address space. The project focuses on DNS, HTTPS, UDP and TLS services. The IPv4 address space is scanned on a weekly basis, and DNS data is made available for download. However, the collected data does not provide insights into SVCB ResRs, nor does it share the IP addresses of name servers included in the scans.

Another search engine specifically for IoT devices is ***Zoomeye*** [168]. It can be considered complementary to *Shodan* and *Censys*, but is often used specifically by researchers to quickly retrieve information about devices when a specific search query is known [162]. For example, the query `device="webcam"` returns all webcams with an Internet connection [162]. Similar to *Shodan*, they do not provide any specific filter for DNS services. Filtering for open port 53 results in around 8.5M hosts. In general, *Zoomeye* is a paid service.

RIPE Atlas [139] is a measurement platform specifically designed to assess region-based connectivity and the reachability of hosts and services [6]. Measurements are conducted from distributed VPs, known as probes, which are contributed by participants providing network access and computing resources to the *RIPE Atlas* network. Currently, the platform consists of approximately 12K probes. While the majority of these probes operate within core or access networks, a notable number are also hosted by volunteers within their home networks [6], providing a very user-centric view on the Internet. All

measurement results are publicly available, and registered users can configure custom measurements, allowing them to define specific parameters for the probes. However, the platform does not support measurements aimed at probing DNS servers for SVCB resource records (i.e., DDR discovery queries), nor does it allow for comprehensive discovery of all hosts running DNS services across the IP address space.

When focusing specifically on DNS-related datasets within the broader landscape of active Internet measurements, ***OpenINTEL*** [137] stands out as one of the largest continuously collected and publicly available databases. They conduct daily active DNS measurements by querying authoritative DNS servers to collect zone file data resulting in over 1.85 billion queries and 240 GB data a day [137]. At time writing this study, their dataset comprises around 306M domains [119]. Their data comprises zone files from TLDs such as `.com`, `.net` or `.org`. To handle this enormous amount of data, their architecture is based on an Apache Hadoop Cluster in combination with Apache Impala engine to execute SQL-like queries [137]. Their dataset is open-access. However, they do not collect SVCB ResRs but more common ResRs such as `A`, `AAAA` or `NS` [166].

Similar to *OpenINTEL*, ***DNS Coffee*** [51, 102] actively collects data on TLD root zone growth and domain distribution across DNS zones by probing DNS name servers and attempting zone file transfers to obtain full zone information (covering approximately 1.2K zones daily). Again, their dataset does not include SVCB ResRs.

DNSDB is a passively collected, historical DNS database [39, 138]. It gathers DNS data from a global network of sensors, including RRs that voluntarily share their data. The collected DNS transactions are filtered and verified before being interested into the database [100]. The dataset is considered representative of broader Internet trends, making it valuable resource for gaining insights into DNS activity in the Internet [5]. However, `DNSDB` is a paid service. While passive data collection is useful for observing protocol usage patterns under user interest, it cannot provide a comprehensive measure of the adoption of specific protocols [161], such as DDR, since it only reflects a subset of global DNS traffic.

Similar, ***DNS Observatory*** is a novel stream analytics platform that passively collects and aggregates DNS traffic observed by hundreds of globally distributed RRs [50]. To manage traffic rates of up to 200K DNS queries per second, the platform aggregates and tracks top-k DNS objects, such as the most frequent authoritative name server IP addresses and top domains. Over a four-month period, the platform analyzed 1.6 trillion DNS transactions. They showcased that approximately half of the observed traffic was handled by only

1K authoritative name servers and just 10 AS operators, indicating that a large portion of DNS traffic relies on a concentrated infrastructure, rather than being evenly distributed across the IP address space.

While most studies focus on scanning the IPv4 address space, there is a growing demand for scanning and analyzing services the IPv6 address space. However, due to the vastness of the IPv6 address space (2^{128} addresses), performing a full scan is practically infeasible [141]. To address this challenge, "hitlists" of responsive IPv6 addresses have been developed to reduce the search space for scanners [141]. Among these efforts, Gasser et al. [52, 146] collected one of the most comprehensive datasets to date: the ***IPv6 Hitlist***. Their approach combines both passive and active measurement techniques: They first passively collect possible IPv6 addresses from other data sources like the *RIPE Atlas* project, `AAAA` ResRs from DNS datasets like *Rapid7* or Certificate Transparency (CT) logs. To measure responsiveness, they leveraged *ZMap* to be capable of IPv6 scans to probe the collected dataset on various ports, including responsiveness on various ports like 443 (HTTPS) or 53 (DNS), resulting in around 1.9M active addresses spread across 10K different ASes [146]. As of this writing, their dataset includes around 314K responsive addresses on port 53. However, we have noticed that the frequency of their data publication does not appear to follow a regular schedule.

3.2 Tools and Measurements in DNS

Although existing DNS datasets provide valuable insights for research (see Section 3.1), the dynamic nature of DNS and the specific needs of various studies often require customized data collection. In this section, we review recent advancements in DNS measurement tools and methodologies, highlighting how researchers have developed or adapted tools to address gaps in available datasets.

Probing the public IPv4 address space is challenging, time-consuming, error-prone and tedious, as highlighted by Durumeric et al. [43]. To address this gap, they introduced ***ZMap***, a modular and open-source network scanner designed specifically for Internet-wide scans [43]. *ZMap* is capable of surveying the entire IPv4 address space in under 45 minutes using a single machine, outperforming other tools like *nmap* [106] by a factor of 1.3K. *ZMap* achieves this efficiency be sending probes as quickly as the source's network interface card and CPU can support, reaching speeds of up to gigabit levels. To avoid overloading networks by scanning IP addresses in numerical order, it employs a random permutation

of the address space, distributing the scanning load in a more network-friendly manner. This permutation is based on a seed, which can be used to execute *ZMap* on multiple nodes in parallel scanning while dividing the address space in chunks for each instance (*sharding*). It also supports probing for various protocols including DNS. Over time, *ZMap* has become a de facto standard for active network measurements due to its speed and scalability [42].

Since DNS is a foundation of the Internet and one of the biggest, highly distributed databases, it became an attractive research area. ***ZDNS*** is a pendant to *ZMap* but specifically tailored for active measurements in the field of DNS, providing researchers with a ready-made and efficient solution, avoiding the time-consuming and error-prone process of developing custom tools [82]. *ZDNS* is faster than the popular GNU dig command, performing around 90K lookups per second when using an external RR, but also supports internal recursion, allowing it to act as a RR. It has been widely utilized in various Internet measurement studies and serves as the foundation for several open datasets [82]. However, *ZDNS* is tailored for specific DNS queries, such as performing `MX` lookups on the Alexa top one million domains, and does not support discovering DNS resolvers that implement the DDR protocol. Unlike *ZMap*, *ZDNS* does not offer full IP address space scanning based on random permutation but requires a predefined list of servers to be scanned.

MassDNS [7] and ***dnsrecon*** [126] share similarities with *ZDNS* but differ in functionality and scope. *MassDNS* is a high-performant DNS stub resolver developed in C, designed to handle large-scale subdomain enumeration by resolving millions or even billions of domain names at speeds exceeding 350K queries per second using publicly available RRs. Similar to *ZDNS*, it is primarily designed for querying a vast number of domains [34], rather than focusing on specific RRs across a wide set of resolvers. Likewise, *dnsrecon* can query multiple DNS record types for a given domain and perform specific tasks such as checking records zone transfers. It is a Python-based tool built for security assessments and network troubleshooting in the area of DNS. Notably, it can scan CIDR block for `PTR` ResRs but does not offer any similar functionality to scan for other ResRs. While both tools are optimized for scanning large sets of domains rather than focusing on specific ResRs across a broad set of resolvers, neither tool offers the full IP address space scanning capabilities.

In addition to established scanning tools like *ZMap*, researchers have adapted existing tools or developed new ones to meet specific research requirements. Below, we highlight studies that have conducted large-scale DNS measurements.

Kührer et al. [90] conducted a long-term, large-scale empirical study of open DNS resolvers on the Internet, focusing on changes over time and classifying resolvers based on characteristics such as device type and software version. Over a 13-month period from January 2014 to February 2015, they scanned the entire IPv4 address space to identify hosts that responded on special-crafted DNS QNAMEs targeting their own DNS zone to evaluate and track DNS servers. Furthermore, they found DNS resolvers with a selected set of 155 domain names across 13 different categories to investigate DNS based censorship. They built their own scanning architecture, similar to *ZMap*, and analyzed the resolvers based on geolocation, network distribution, DNS software, and underlying hardware. Notably, they found around 26M `NOERROR`-responding resolvers at the start of the study, though this number dropped to approximately 18M by the end. 40% of the scanned resolvers changed their IP address within the first day (IP address churn), likely due to short IP lease times on consumer routers. Their fingerprinting analysis showed that *BIND* was the most widely used DNS software on the hosts scanned. Additionally, they found discrepancies between the responses of validated and open resolvers, with thousands of resolvers manipulating DNS resolutions to censor communication channels, inject advertisements, serve malicious files, or engage in phishing.

In like manner, Takano et al. [156] conducted an active measurement study, scanning the entire IPv4 address space once to gather DNS-related information in 2013, including the distribution of DNS server types, software versions, and FQDNs. Their scan architecture was built around a *MongoDB* database and a DNS prober, implemented in C++, which sent `A` ResR requests with the `RD` flag set (see Section 2.1). If a response was received, they performed fingerprinting of the server. Their study identified approximately 30M DNS servers, with around 25M being public open resolvers. Among the around 7M servers that disclosed their software, *BIND* emerged as the most popular DNS software. Additionally, they found a presence of open resolvers on domains frequently associated with spamming activities.

Schomp et al. [148] developed a set of methodologies for efficiently discovering client-side DNS infrastructure. They classified DNS servers into ingress servers, which directly receive DNS queries; egress servers, which communicate with authoritative DNS servers; and hidden servers, which act as intermediaries between ingress and egress servers. Additionally, they further categorized these three server types. For example, ODNSes are open DNS servers, which accept requests from any host, while FDNSes are forwarders and a subset of ODNSes that do not resolve queries by themselves but merely forward them to other servers within the server's resolution chain. To gain these insights,

they conducted around 267M domain queries in 2012, targeting resources on their own authoritative servers from 100 nodes of the *PlanetLab* network [16], a global overlay network designed for broad-coverage network services. Their study identified around 32M open resolvers on the Internet and observed that DNS queries often travel long distances within the resolving infrastructure.

Luo et al. [105] further explored the dependencies between ingress and egress DNS resolvers in 2023. Their study revealed that around 8% of the egress resolvers had the ability to influence 90% of the ingress resolvers by their responses. Furthermore, these egress resolvers were highly concentrated, with around 2% of them spanning across all ASes. The authors highlight that this concentration poses security risks, including increased susceptibility to single points of failure and potential threats to user privacy.

In 2019, Park et al. [122] conducted a study similar to the previous works of Takano et al. [156], Schomp et al. [148] and Kührer et al. [90], investigating the state of open resolvers and their potential negative impact on the security and stability of the Internet. Utilizing *ZMap* for their scans, they queried their own domain to analyze how open resolvers behaved in *recursive-to-authoritative* communication, though they did not provide details on timeouts or other insights into their scanning architecture. They identified over 3M open resolvers, a significant reduction compared to the approximately 25M and 18M open resolvers found by Takano et al. [156] and Kührer et al. [90], respectively. Despite this decrease in the number of open resolvers, Park et al. noted an increase in resolvers providing malicious responses. They also observed that many resolvers deviated from DNS standards, improperly marking fields such as the `RA` and `AA` bits in the DNS message header.

Hohlfeld [65] criticized the narrow focus of many studies that concentrate on single protocols over short time frames, noting that Internet operations often involve multiple protocols such as QUIC, HTTP, and TLS. In response, they developed a scan architecture called *Internet Observatory* that regularly scans the IPv4 address space using *ZMap* and approximately 50% of the domain name space, drawing data from sources such as *OpenINTEL* [119] and CT logs. They implemented a message broker architecture based on *RabbitMQ* to distribute the scan workload across several worker nodes, which enables easy scalability and the addition of further scans, while also facilitating multi-VPs seamlessly.

Analogous to Hohlfeld [65], Braakhuis [11] developed a scalable and reactive DNS measurement system compatible with *OpenINTEL* [119], designed to capture large portions of the DNS through daily snapshots. According to Braakhuis, this system addresses a limitation of *OpenINTEL*, which lacks

the flexibility to perform diverse reactive measurements, such as responding to triggers or executing on-demand scans. Their architecture is a message broker system based on *Apache Kafka*, enabling the scheduling of tasks across consumers and supporting a distributed scanning framework across multiple virtual machines.

Mao et al. [110] investigated the support for DoTCP53 form the perspective of RRs and authoritative resolvers in 2021. They found that up to 4.8% of the resolvers did not properly fall back to TCP when instructed by the authoritative DNS via the truncation bit (TC) (see Section 2.1). To conduct this research, they probed every server in the entire IPv4 public address space with DNS queries targeting their own experimental DNS zone to identify open resolvers. If the resolver responded with `NOERROR` RCODEs, their scanner sent follow-up queries to assess the resolvers support for DoTCP53 to avoid timeouts due to IP address churn during the study. To further investigate resolvers not exposed to the public, they utilized *RIPE Atlas* which resulted in a distinct dataset from their public IP address scans. Overall, they discovered around 3M open resolvers. However, they do not provide any detailed information about their measurement architecture.

Since previous studies primarily focused on the IPv4 address space, Hendriks et al. [59] proposed an active measurement approach to identify open resolvers in the IPv6 environment that could potentially be exploited in Distributed Denial of Service (DDoS) attacks. Their approach is based on the hypothesis that a portion of open IPv4 resolvers also have IPv6 connectivity and are openly resolving over IPv6. Given that a full IPv6 scan is impractical, they created a DNS zone managed by an authoritative server which only responded with a valid IPv6 address on a delegation record (`NS`, see Section 2.1). As a result, the open resolver is effectively compelled to use IPv6 to resolve resource records within the zone managed by the delegated name server. Their measurements identified approximately 1K IPv6-enabled open resolvers and revealed that many of these were misconfigured, creating a potential attack surface for DDoS exploits.

Zirngibl et al. [167] were the first to our knowledge to investigate SVCB and HTTPS ResRs in the wild (see Section 2.1.3) in 2023. They collected domain names from various sources and queried them with the necessary `_dns` label, as specified in the standard [150]. Their findings revealed that only around 4K servers provided SVCB records, while approximately 10M domains had HTTPS records. Additionally, they demonstrated that HTTPS records were generally well-configured, allowing clients to utilize them effectively to reduce handshake costs and minimize the leakage of meta-data.

3.3 Adoption and Performance Measurements of DoE

While Do53 imposes a risk on security and privacy of the transmitted DNS messages, DoE protocols have been deployed in the wild (see Section 2.3), thus becoming an interesting research topic for many studies.

Lu et al. [104] conducted the first large-scale, end-to-end analysis of DoE protocols in 2018. Using *ZMap*, they probed the port 853 to discover DoT resolvers. However, discovering DoH resolvers proved to be more challenging due to their diverse configurations. As a result, they relied on a dataset provided by an industry partner to identify DoH resolvers. In total, they discovered 150 DoT and 17 DoH resolvers, with approximately 25% of the DoT resolvers offering invalid TLS certificates. Their study also revealed the presence of censorship (e.g., blocking) within DoE protocols. Despite the additional cryptography overhead introduced by DoE (see Section 2.3), the authors found it to be minimal, adding only a few milliseconds of latency. Furthermore, they observed that while the adoption of DoE protocols is still small compared to traditional DNS, it is steadily growing. These findings were based on a measurement platform involving around 123K VPs across 166 countries, allowing for an evaluation of client-side performance when using DoE endpoints for DNS resolution.

Lyu et al. [107] provided a comprehensive literature review on DoE encryption techniques, covering studies from 2016 to 2021. They focused on opportunities, risks, adoption, performance, and security vulnerabilities towards DoE protocols. Among the studies investigated, DoH is the most widely adopted encryption method in industry, with a few major cloud service provider dominating the DoE ecosystem, raising concerns about data monopolization. Their analysis revealed that DoE protocols tend to under-perform in non-ideal network conditions and that clients often use opportunistic encryption by default (see Section 2.3.6), leaving them vulnerable to downgrade attacks.

Several studies have investigated the performance of DoH. Sharma et al. [152] found that major cloud service providers oftentimes offered better median response times, largely due to their use of Content Delivery Networks (CDNs). In contrast, Chhabra et al. [15] discovered that clients in countries with higher-quality Internet infrastructure — characterized by faster speeds and a greater number of ASes — as well as those in higher-income countries, were less likely to experience performance slowdowns when switching to DoH. In fact, these clients may even experience a speedup when using DoH. These findings align

with those of Hounsel et al. [66]. However, they observed that the standard deviation of DoH latency was three times smaller than that of Do53, likely due to the caching capabilities inherent in the HTTP protocol. Deccio and Davis [24] investigated the deployment of TCP Fast Open (TFO) in DoT and DoH resolvers, concluding that only a few resolvers supported TFO, presenting an open opportunity to reduce communication latency. Whereas all recent studies in DoE performance focused on the IPv4 address space, Jia et al. [84] examined the performance differences of major IPv4 and IPv6 DoE resolvers, such from *Google*, *Cloudflare*, or *AdGuard*, in late 2022. They found that DoT was particularly faster over IPv6 compared to IPv4.

While DoH/2 and DoT require multiple round trips in their handshake to establish a secure connection, DoQ offers 0-RTTs to reduce handshake latency (see Section 2.3.4). Kosek et al. [89] studied the 0-RTT performance of DoQ in comparison to DoH/2 and DoT in the wild, by measuring 313 encrypted resolvers from 6 different VPs. They found those resolvers by probing scanning the IPv4 address space with `ZMap` on port 784, 853 and 8853, resulting in around 1.2K DoQ resolvers, while 313 had the intersection of advertising every DoE protocol. They showed that DoQ was around 33% faster than DoT and DoH/2, while the web performance of browsers improves by 10% overDoH/2 for simple webpages. Interestingly, DoQ is only around 2% slower than Do53. While Kosek et al. investigated specifically the 0-RTT performance of DoQ, Nawrocki et al. [116] showed that the overall performance of the HTTP/3 handshake suffered form certificates exceeding QUIC's amplification limits in 87% of all their measured connection attempts. This led to an overall reduced performance.

3.4 Security, Privacy and Censorship in DoE

DoE protocols are primarily designed to secure DNS communication and to enhance user privacy. However, recent studies have evaluated the security implications of DoE and its impact on user privacy [33]. While DNS has long been a common target for censorship (e.g., DNS hijacking), studies have further investigated whether DoE protocols can circumvent censorship.

Hoang et al. [62] investigated the DoH and DoT protocols in the context of censorship given the fact that DNS blocking in the unencrypted Do53 is a common censorship technique. They developed ***DNSEye***, a measurement system built on a distributed network of VPs, which they used to assess the efficacy and accessibility of DoH and DoT in circumventing censorship. Over a

period of six months, their study examined the accessibility of 1.6K domains from around 20K VPs, targeting 71 DoH and DoT resolvers. Their findings demonstrated that the use of DoE protocols enabled them to unblock over 55% of blocked domains in China and more than 95% of blocked domains in other countries employing DNS-based filtering. Differently, Jin et al. [85] found evidence of DNS manipulation and censorship in DoE protocols. They performed around 7M DNS lookup measurements on approximately 3.8M DoT and 75 DoH resolvers. They found that more than two-thirds of the DoT and DoH resolvers manipulated DNS responses.

At first glance, one might assume that encrypted communication, such as DoT, does not reveal information about user activities. However, Houser et al. [68] developed a DoT fingerprinting method aimed at analyzing DoT traffic to determine whether a user has visited a specific website of interest, e.g, health insurance, gambling or dating websites. Their approach infers visited websites by modeling the temporal patterns of packet sizes and sequences. Interestingly, their method showed a false negative rate of less than 17%, which drops to less than 0.5% if DNS messages are not padded. As a mitigation, they proposed the use of padding and uniform time intervals for message exchanges. While message delays could also help to obfuscate packet sequences, this stands in conflict with the demand for low-latency DNS communications.

As already depicted in Section 2.3.6, DoE protocols in general have two usage profiles. Huang et al. [71] investigated downgrade attacks on DoH due to the opportunistic privacy profile [29], which allows a fallback to unencrypted DNS if the DoH channel cannot be established. They found that every major browser like *Firefox*, *Safari* or *Google Chrome* used the opportunistic privacy profile, while all of them are vulnerable to every downgrade attack vector. Furthermore, none of them notified users when the connection falls back to unencrypted DNS. As a result, users are often unaware of attacks on their privacy. Conversely, DoE in web browsers can also pose a privacy risk, as private data may be collected by major DNS cloud providers without users' awareness. Nisenoff et al. [117] confirmed that most users are unaware of the development of DoE. For instance, *Firefox* forwards all DNS requests to *Cloudflare* in the U.S. by default using DoE, with these settings being changed without explicit user consent. This occurs because users do not fully understand the implications of DoE settings in their browsers, owing to their technical complexity and the lack of sufficient information provided by web browsers.

Li et al. [95] investigated the deployment of DoH and DoT with a focus on the strict privacy profile (see Section 2.3.6). They conducted monthly scans with `ZMap` between November 2021 and September 2022 to discover DoT and

DoH resolvers, along with daily scans of TLS/HTTPS-related security features in the resolvers found. To discover valid DoT resolvers, they simply probed DNS queries via DoT on port 853. In the case of DoH, they queried on the URI paths `/dns-query`, `/query`, `/resolve`, and `/` on port 443. Their scans identified around 26K DoH and 21K DoT resolvers, while only approximately 65 DoH and 290 DoT resolvers were authoritative DNS servers. They found that around 60% of the DoT and 44% of the DoH RRs lacked valid certificates. In line with previous studies, they confirmed that DoT and DoH resolvers were becoming centralized. Additionally, they noted that 25% of the DoH resolvers supporting strict privacy failed to meet the minimum privacy requirements.

In a follow-up study, Li et al. [96] further examined DoE resolvers and were the first to look at DDR. Over a 15-month scan, they identified approximately 1.3K operational DoE resolvers, of which 448 supported IPv6. They conducted 10M IPv4 and 570K IPv6 DoE queries from around 5K VPs over two months in 2023, discovering that approximately 6% of IPv4 and 5% of IPv6 queries were blocked. Their study also revealed that IPv6 DoE resolvers, particularly for DoQ and DoH, exhibited better reachability than their IPv4 counterparts. As an underlying measurement technique, they used *ZMap* probing the IPv4 address space on ports TCP/853 (DoT), TCP/443 (DoH/2), UDP/443 (DoH/3), and UDP/853 (DoQ). While it is possible to directly check for valid DNS resolver service in the case of DoQ and DoT, they mentioned it is challenging in the case of DoH due to the more complex configuration needed, leading to the same probing configuration as in their previous study [95]. They also classified various blocking types and claimed their dataset to be the most comprehensive on DoE resolvers to date. Finally, they pointed out the lack of a standardized method for clients to discover DoE configuration details for open resolvers. In a related DDR scan, they identified around 317K DDR-enabled resolvers, with 77% redirecting to *Google* and 12% to *Cloudflare*. However, neither did they analyze the DDR results in detail (e.g., priorities of advertised DoE protocols, DoE configurations, discrepancies between DDR configuration and real-world, validating advertised DoE resolvers, security considerations, etc.) nor did they conduct a long-term DDR scan.

3.5 DNS Centralization

Though DoE enhances the security and privacy of DNS queries avoiding man-in-the-middle attacks such as eavesdropping, it also contributes to DNS centralization. Unlike traditional DNS, where clients typically send queries

to a local resolver pre-configured via DHCP, clients using DoE protocols rely on centralized architectures, sending all queries to a single RR [67]. Further, limited discovery methods for DoE resolvers and their configuration [96] and the small number of DoE resolvers concentrate user data among a few third-party providers [74, 95, 101], posing risks to the Internet's decentralized structure and user privacy [67]. As a result, recent studies have focused on the growing centralization trend in DNS.

For example, Doan et al. [37] investigated the latency between centralized public DoE and ISP resolvers by conducting probes via *RIPE Atlas*. Their findings revealed that approximately one in three users relies on at least one public DNS service, with *Google* being the most widely used, accounting for around 78% of the probes utilizing a cloud provider. While other studies suggest that one motivation for using public cloud providers is improved performance, they demonstrated that some public DNS services actually achieve lower lookup latency compared to local ISP resolvers.

The centralization trend has tremendous impact on the DNS structure. The resilience of the Internet partly stems from its diversity, whereas centralization clusters pose multiple risks, ranging from technical to economic, and amplifies the potential consequences of any single point of failure [91]. Additionally, a centralized architecture can even degrade DNS performance, as DoE resolvers depend on CDNs and replica selection [91]. This assumes that the location of a client's resolver provides an accurate approximation, but in specific regions, certain third-party DNS services may lack nearby servers, leading to negative performance effects [120].

To address the centralization problem and support distribution of queries across multiple resolvers, several studies [61, 67, 91] proposed architectures designed to enhance privacy while supporting decentralization. Essentially, their solutions allow for pre-configuration of multiple DoE resolvers, enabling queries to be distributed among them, thereby preventing any single DoE resolver from reconstructing the full user query history.

While these approaches offer potential to improve the current situation, the fundamental challenge lies in the large-scale unified discovery of DoE endpoints including their configuration, even at the ISP level. In this context, DDR could play a crucial role in fostering decentralization if appropriately implemented.

4 Methodology

In this chapter, we present the rationale for developing a custom measurement platform, including our tailored measurement approach, in response to the limitations of existing datasets and tools in addressing our specific research questions (see Section 1.2). Our platform is capable of discovering IPv4 addresses running potential DNS services via *ZMap* [43], downloading the latest responsive IPv6 addresses from *IPv6 Hitlist Service* [52, 146] on a daily frequency, and executing DDR probes in-time to prevent bias through IP address churn [90]. Based on the DDR responses, it can schedule follow-up scans, including DoE probes and TLS certificate scans, to analyze the broader DDR ecosystem.

While datasets like *OpenINTEL* [119], *DNS Coffee* [102], and *DNSDB* [39] offer valuable insights into DNS traffic and records, they lack information about IP addresses or hosts running DNS services, nor do they collect SVCB ResRs, which are essential for studying the DDR protocol. *Censys* appeared to be the closest match as a starting point, which provides daily snapshots of the entire Internet including DNS hosts. However, the prohibitive costs associated with querying their data via *Google BigQuery*, coupled with unsuccessful negotiations for smaller, more specific datasets, made this option unfeasible. As a result, we were compelled to discover IPv4 DNS servers ourselves, while relying on *IPv6 Hitlist Service* for UDP/53 responsive IPv6 addresses.

Existing DNS measurement tools, such as *ZDNS* [82], *dnsrecon* [126], and *MassDNS* [7], though effective for scanning for common DNS records such as `A` or `MX` ResRs, do not support large-scale scanning of IP addresses for SVCB ResRs, which are vital for our research. Integrating the required functionality into their complex codebases would be inefficient compared to the simplicity offered by lightweight DNS libraries like *miekg/dns* [53] in Go. Furthermore, *ZMap* already provides modules to discover potential DNS servers, so our platform only needed to extend this by adding DDR probes and DoE scans, making it both efficient and adaptable the context of our research.

In Section 4.1, we describe our measurement approach and procedure in detail. We compare our measurement architecture and findings in terms of found DNS servers, response times and retry strategy on a meta-level with existing research and databases in Section 4.1, providing a validation of our

methodology. Acknowledging the responsibility that comes with large-scale measurements, particularly regarding potential impacts on networks and privacy, we incorporate ethical considerations drawn from established best practices and guidelines, as discussed in Section 4.3.

4.1 Measurement Architecture and Stages

To address the research questions outlined in Section 1.2, we developed an open-source, reactive, adaptable and highly-scalable three-stage, fully containerized and monitored measurement architecture called `DoE-Hunter` written in Go [143]. Go is a fast programming language, provides an active community, well maintained libraries for DNS [53], and supports lightweight but highly efficient threads (*Goroutines*) for parallel execution.

Our measurement architecture covers multiple standards (see Section 4.1.1), includes our own DNS authoritative name servers (see Section 4.1.2) and follows a three-stage measurement approach. In the first stage (see Section 4.1.3), we collect responsive IPv4 and IPv6 addresses on port UDP/53. In the second stage (see Section 4.1.4), these addresses are used to discover DDR-enabled resolvers and their delegated encrypted resolvers. Finally, in the third stage (see Section 4.1.5), we query these encrypted resolvers using the respective DoE protocols and query for other protocols like DNSSEC as well. To facilitate a more fine-grained analysis of the results, we enhance the collected dataset by incorporating AS-related data (see Section 4.1.6).

4.1.1 RFCs, Scans and their Relation

Our architecture implements several standards (RFCs), as illustrated in Figure 4.1. Initially, we use *ZMap* [43] and the *IPv6 Hitlist Service* [52, 146] to gather IP addresses that respond on UDP port 53 (for more details, refer to Section 4.1.3). These addresses are then processed by our DDR scanner. If the server replies with any DNS response, we schedule Pointer Record (PTR) and fingerprinting scans to track the discovered servers over time. In cases where the DNS server replies with a DDR response, the system parses the information and schedules DoE scans based on the advertised encrypted resolvers and their associated protocols. Notably, we establish a tree-like data structure to interconnect scans, with the DDR scan as the root node. This structure allows us to trace relationships between scans, such as tracking which DoE

scan followed a particular DDR scan or which certificate belongs to a specific DoE scan.

To avoid redundant scans of already-known encrypted resolvers, we implement a caching mechanism. This ensures that DoE scans are only performed for newly discovered encrypted resolvers, reducing both the amount of disk space required for measurement results and the overall network traffic. Further, this reduces the likelihood of being blocked by recursive resolvers. To maintain the tree-like structure, we link the root (DDR) scan with the corresponding DoE, certificate and DNSSEC scans from the cache. Since DoE protocols rely on TLS, every DDR scanner schedules separate scans to collect certificate information about the encrypted resolver. These are executed in a dedicated scan using Go's *crypto/tls* library, as not all connection handlers (e.g., Go's QUIC implementation) support certificate extraction from an established connection. This ensures that every stored certificate follows a consistent scheme. Additionally, we set the Server Name Indication (SNI) to the hostname of the advertised DoE resolver (`targetName`) to signalize the server which certificate to return for the requested domain.

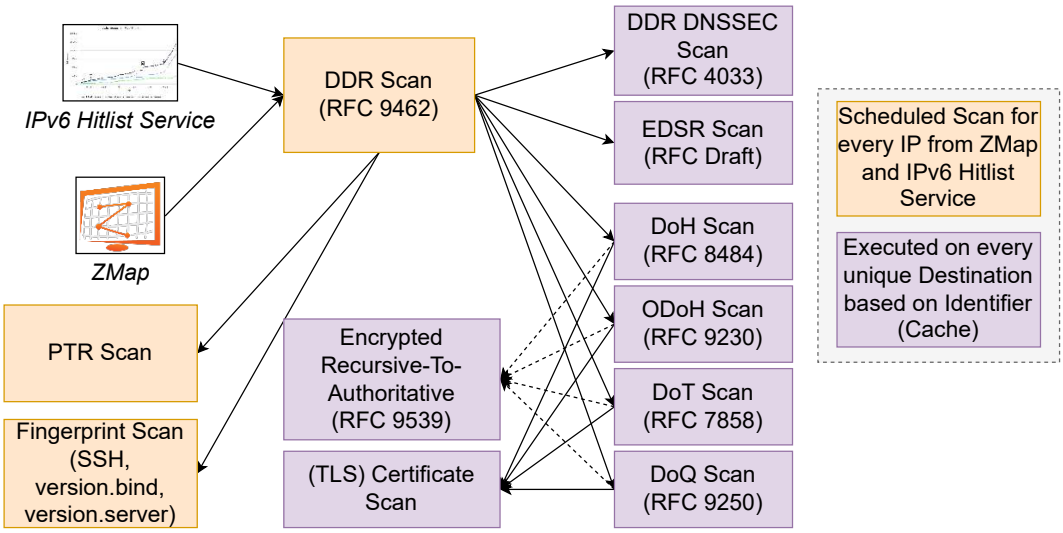


Figure 4.1: Relation of standards (RFCs) and their scheduled scans in our measurement architecture.

4.1.2 Authoritative DNS Server Setup

We also analyze the behavior of potential encrypted *recursive-to-authoritative* communication, which is standardized in RFC 9539 [54]. To analyze the adoption rate, we query every discovered DoE resolver for a uniquely crafted QNAME targeting an `A` ResR on our domain `raiun.de`. These uniquely crafted QNAMEs are of the pattern `<id>.measurement.raiun.de`. In our DNS zone, we set up the wildcard `*.measurement.raiun.de` pointing to an IPv4 address that is used by our measurement hint's website (see Section 4.3). This has three essential advantages. First, every uniquely crafted QNAME is resolvable and identifiable, enabling us to link the DoE probes and resolving results to the queries being made on our name servers as we enable logging for every query being made, additionally providing us with insights in the DoE recursive resolving behaviors and infrastructure. Second, we can track whether the probed DoE resolvers use encrypted communication to our authoritative servers. This includes possible DDR queries to discover our current configuration of encrypted endpoints. Third, we can track whether DDR is being studied in other research projects, including our own measurements.

For our authoritative name servers, which are located in Germany and Austria, we use *bind9* [79], which supports DoT and DoH/2 natively. However, *bind9* does not differentiate query logs based on the transport protocol used

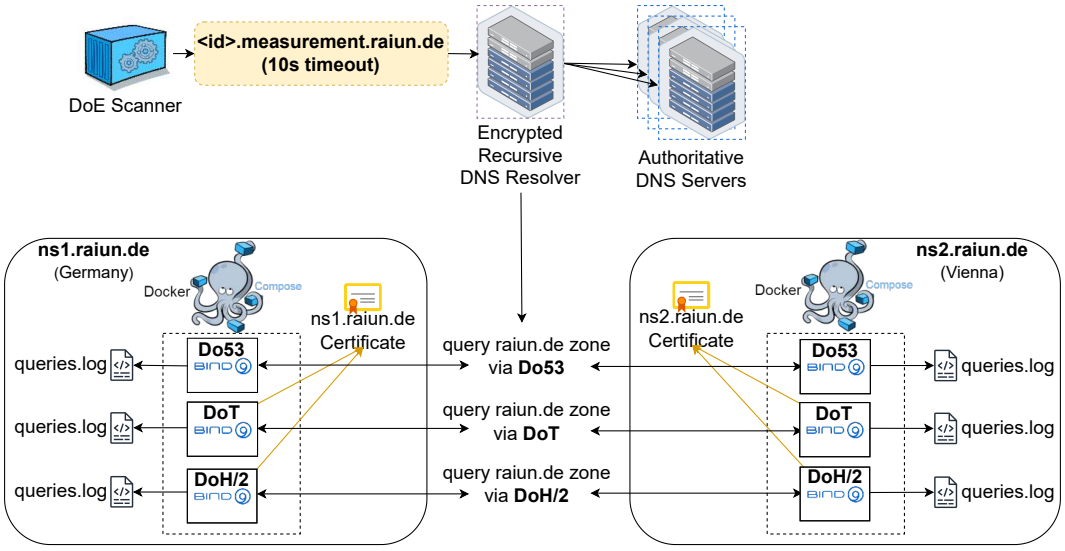


Figure 4.2: Setup of our authoritative DNS servers serving the *raiun.de* zone.

(i.e., Do53, DoT, or DoH), as all queries are recorded in a single log without specifying which protocol was used for the query. To overcome this limitation, we run a separate Docker container for each protocol, with each container sharing the same zone file (see Figure 4.2). These containers are based on the latest Docker images maintained by Linux distributor *Canonical* for *bind9* on *Ubuntu* [12]. Additionally, we use *certbot* [44] with the *Let's Encrypt* CA [46] to provide valid TLS certificates for DoT and DoH/2 communication.

4.1.3 Stage 1: Discovery of IPv4 and IPv6 DNS Resolvers

To gather IPv4 DNS resolvers, we perform consecutive large-scale DNS scans from a single VP within a network of an educational institution (AS *8881*) (refer to Section 8.1 for a discussion of limitations). We employ the *ZMap* network scanner [43] to frequently identify publicly available IPv4 DNS servers that respond to DNS queries on UDP/53[1] by resolving the `A` record for *www.google.com* ① . Given that *www.google.com* is one of the most frequently accessed websites on the internet [103], this increases the likelihood of hitting cached entries in ResR, enabling quicker responses rather than waiting for full DNS resolution.

However, it should be noted that *ZMap*'s DNS probing module does not distinguish between proper DNS responses and any other packets replied. This means that any received byte is considered a hit in the result set, including non-DNS servers.

Given that the IPv4 address space includes address blocks reserved for private use [132] and special purposes [22], and that we occasionally receive requests from system administrators to exclude specific addresses from scanning (see also Section 4.3), we employ a blocklist[2]. This blocklist is further enhanced with address spaces sourced from the *MassDNS* project [7], which includes addresses and abuse reports requesting exclusion from DNS large-scale measurement scans.

As shown in previous work (see Section 3.2), scanning the full IPv6 address space with tools like *ZMap* is infeasible due to the sheer scale of its space (2^{128} addresses). To address this, we utilize the *IPv6 Hitlist Service* provided by Gasser et al. [52, 146], which offers a curated set of responsive IPv6 addresses on UDP/53. This service provides a weekly updated list of responsive IPv6

1 Note that we send only one query per IPv4 address, as sending multiple probes does not considerably increase the success rate. For further details, refer to the discussion on this topic at https://github.com/zmap/zmap/wiki/Sending-Multiple-Probes.

2 The full blocklist is available in the architecture's public repository [143].

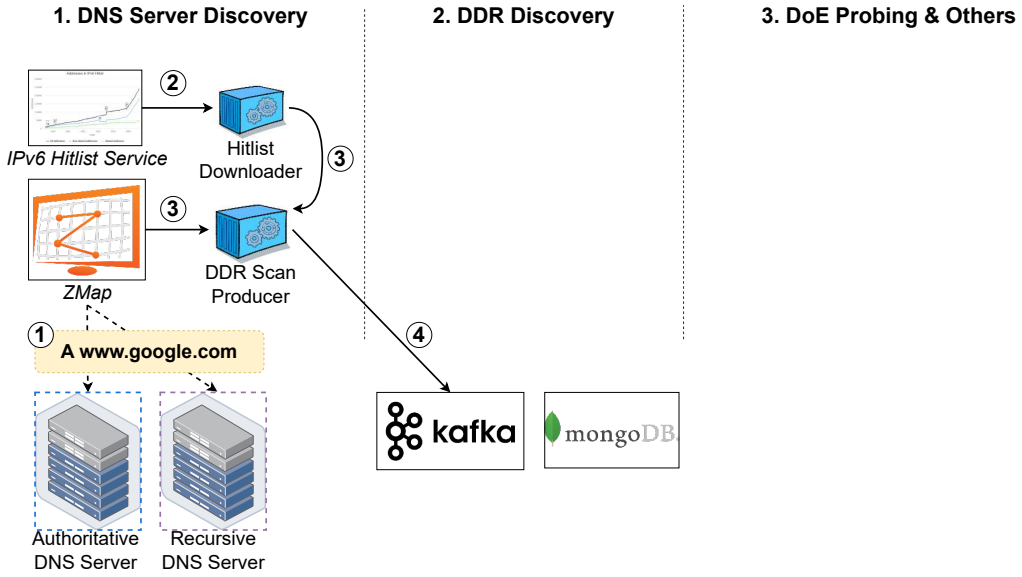


Figure 4.3: Methodology Stage 1: Potential IPv4 and IPv6 DNS servers are discovered and DDR scans are scheduled.

addresses. However, we observe that there is typically a one-week delay between the start date of their scan and the publication of their result set. This delay introduces potential bias in our findings, as recent studies have demonstrated that despite the vast IPv6 address space, IP address churn remains prevalent [127]. Since the service first collects all data before publishing the processed result set, we are unable to mitigate this bias. Additionally, similar to the *ZMap* result set, the responsive IPv6 addresses may include services that do not necessarily run DNS.

To automate the retrieval of these addresses, we developed an open-source and dockerized Python application, *Hitlist-Downloader* [144], which checks for any updates on a daily basis and downloads the latest data from the *IPv6 Hitlist Service* ② .

To enable inter-process communication between *ZMap* and *Hitlist-Downloader*, and our measurement architecture, we developed a generic scan producer ③ . Since named pipes offer a suitable mechanism for inter-process communication [121], our architecture is capable of tailing named pipes or regular files. However, named pipes have a limited buffer size of 64 KB in Linux [121] (see Figure 4.4). If *ZMap* outputs IP addresses faster than the listener on the named pipe can process and the buffer space is filled, back pressure is created, potentially causing *ZMap* to block or to terminate

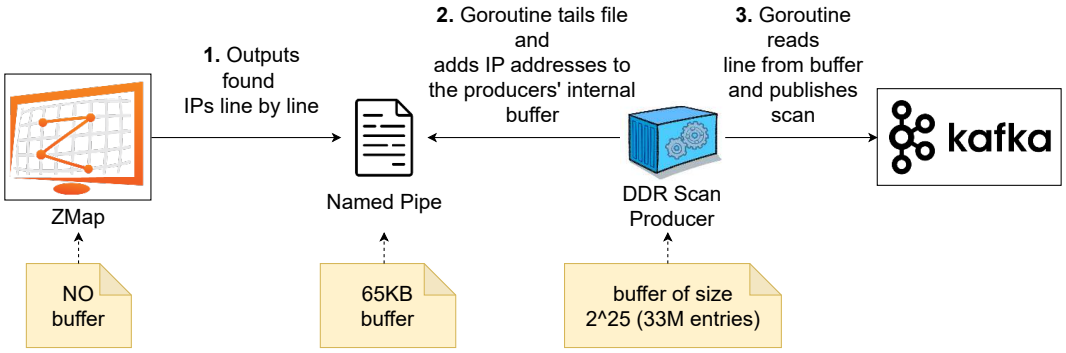


Figure 4.4: To resolve back pressure issues on the named pipe, our architecture implements a buffer to prevent occasional exits and incomplete scans from *ZMap*. This buffer ensures smoother data flow and stability during large-scale scans, preventing the scanners like *ZMap* from terminating prematurely.

unexpectedly. To address this issue, our architecture employs an internal buffer of 33M entries[3], along with a dedicated *Goroutine* (thread) responsible for continuously reading from the named pipe, ensuring the pipe's buffer remains clear and preventing back pressure. Another *Goroutine* simultaneously reads from the internal buffer and schedules the scans to *Apache Kafka*. This design ensures that the architecture is capable of handling large-scale outputs from scanning tools without causing disruptions.

Furthermore, the scan producer only requires knowledge of the output format of the result set entries (e.g., from *ZMap*). This offers several advantages: First, scanning tools like *ZMap* or *ZDNS* typically output the found IP addresses line by line. Therefore, our scan producer merely needs to parse this output, avoiding any modifications to the scanning tools themselves for integration with our architecture. Second, our architecture allows for seamless integration of new versions of scanning tools like *ZMap*, as updates to these tools generally do not require changes to the architecture, unless the output format changes. Third, by scheduling follow-up scans immediately after a hit is detected by *ZMap*, we mitigate the effects of IP address churn [90, 110, 136], ensuring timely and accurate scans of discovered resolvers.

To distribute scans among scanners in our architecture (e.g, scanning for DDR-enabled resolvers), we utilize the dockerized version of *Confluent Kafka* [20], an open-source message broker system based on *Apache Kafka*. It allows the

3 We use the upper bound of ~30M IPv4 addresses found by *ZMap* during our scans, including some margin space (see also Section 4.2).

use of a producer-consumer pattern, where messages are produced to specific topics, and consumers register to pull these messages at their own pace [86]. In our architecture, the DDR scan producer ④ acts as a producer that generates scan tasks and sends them to a designated topic in *Apache Kafka*. Each topic represents a set of scans to be performed from a particular VP. For instance, if scanning from ten different VPs is required, the scan producer schedules the scans to ten separate topics in *Apache Kafka*, allowing the respective scanners to pull and execute the scan tasks from their assigned topics.

4.1.4 Stage 2: DDR Discovery

In the second stage, the DDR scanner, subscribed to the `Kafka` topic representing the VP within the educational network, consumes the scheduled scans from the DNS server discovery ⑤ . Since only the IP address of the potential DNS server is known, our scanner performs the DDR discovery according to the *Discovery Using Resolver IP Addresses* method (see Section 2.4.1). Consequently, the DDR scanner sends a query for the SVCB ResR with the query name `_.dns.resolver.arpa`, applying a five-second timeout ⑥ . This timeout orientates on the default timeout used by the popular DNS tool *dig* [80] and Linux' stub resolver [99]. For a detailed

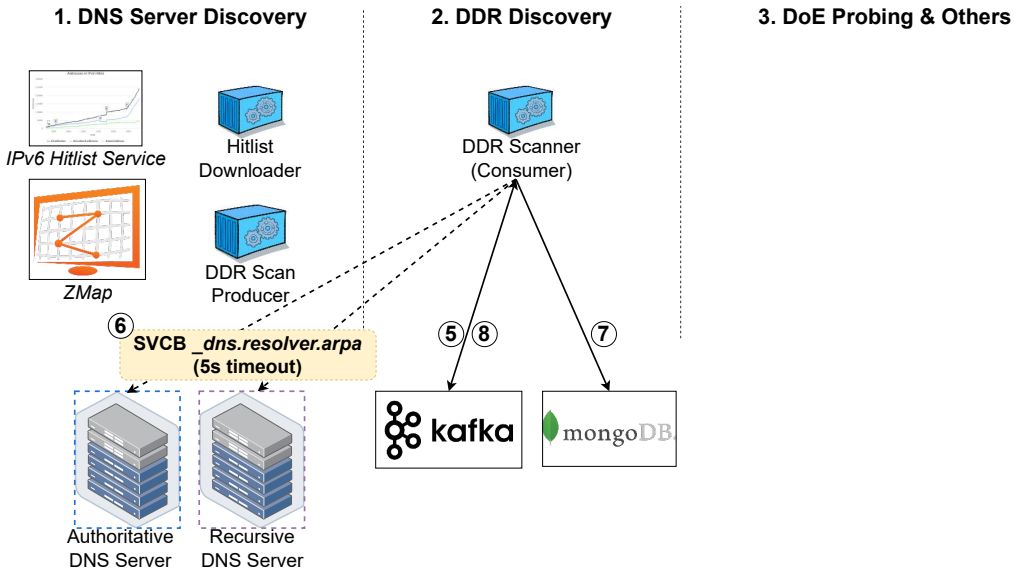


Figure 4.5: Methodology Stage 2: DDR probes are executed and DoE scans are scheduled.

analysis of this approach, please refer to Section 4.2. To keep up with the pace of *ZMap*'s hit rate of around 310 IP addresses found per second, we run 10K Go routines in parallel consuming and executing the DDR discovery. To minimize the overhead associated with socket binding and allocation, each thread pre-allocates a socket before executing a scan. Since each socket requires binding to a unique port, this imposes a practical limit on the number of threads, constrained by the 65,536 (2^{16}) available ports in Linux systems.

Every scan is subject to a retry and back-off strategy ⑥ . The retry strategy similar to Linux's stub resolver [99]. Before considering a potential DNS server as timed out, we retry two times on UDP/53, followed by a final attempt on TCP/53. The number of retries is recorded in the metadata of each scan and can be individually adjusted for each scheduled scan (e.g., a DDR scan). To avoid overwhelming requests with frequent requests due to timeouts at the same time, we incorporated a randomized back-off mechanism. This strategy dynamically adjusts the delay between the successive retries, progressively increasing the wait time after each timeout. Specifically, the back-off strategy begins with selecting an initial retry value within an interval of one second. The delay doubles after each failure, with a maximum interval cap of five seconds. An overview of the effectiveness of the retry and back-off strategy is provided in Section 4.2.

Although we support EDNS0, a server may signal data truncation due to UDP's datagram size limit [28]. In such cases, we immediately fall back to TCP/53 if the server indicates truncation via the truncation bit (`TC`) [45]. The recursive desired (`RD`) bit in the DDR discovery is set to false, as the `resolver.arpa` ResR is a SUDN and thus resolved locally, outside the standard DNS hierarchy. Our test runs also revealed that setting the `RD` bit to true results in more DNS servers responding with an `RCODE` other than zero, i.e., failed resolving attempts.

Every scan result is stored to a `MongoDB` instance for later analysis ⑦ . If the DNS sever supports DDR with a valid configuration, the scanner parses the configuration, extract relevant parameters such as ports and protocols (see Section 2.4.2), and schedules DoE scans according to the advertised designated encrypted resolvers (`targetName`) ⑧ . As SVCB ResRs can also provide IP hints for optimized routing, and some DNS servers may include glue records, separate DoE scans are scheduled to check for discrepancies between the `targetName`, glue records, and IP hints. In addition, we schedule EDSR scans to evaluate the recent RFC draft [160]. To investigate whether the designated encrypted resolver also supports DDR and whether it further employs DNSSEC

to ensure the authenticity and integrity of DDR discoveries, we further schedule DNSSEC scans according to the pattern `_dns.<targetName>`.

4.1.5 Stage 3: DoE Probes and Other Scans

For each DoE protocol, we use a dedicated consumer that pulls the scheduled scans from `Apache Kafka` ⑧ and executes them from the educational network (VP). Each scan queries a unique `A` ResR, with the pattern `<id>.measurement.raiun.de` ⑨, encoding an identifier for later matching of our DNS server's logs and the query being made. This allows us to correlate recursive-to-authoritative communications ⑩, including potential encrypted interactions. Our authoritative name servers support DoT, DoH/2, and DDR (see Section 4.1.2).

The RFC 9539 [54] recommends a four-second timeout for *encrypted recursive-to-authoritative* communications, while `Quad9` sets a two-second timeout for its DoH recursive resolver [104], for example. We decided to set a relaxed ten-second timeout for DoE probes (see also Section 8.2). Still, studies show that the overhead introduced by DoE protocols is on the order of milliseconds [104], suggesting that doubling the timeout from plain DNS to DoE provides sufficient margin for potential cryptography-related overhead.

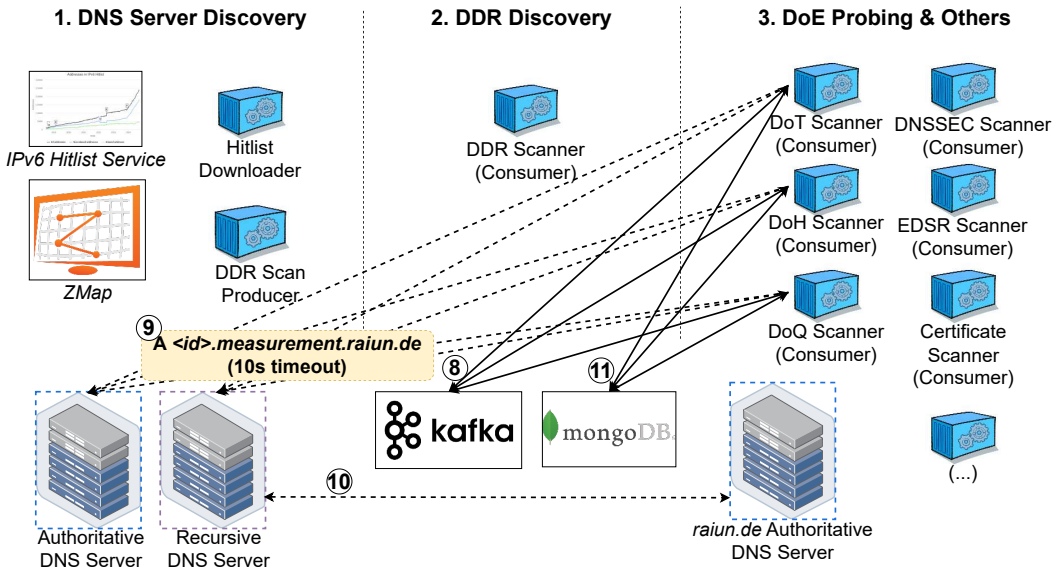


Figure 4.6: Methodology Stage 3: DoE probes are executed, and additional scans are scheduled. Note that note every scan is depicted in this illustration.

In case a DoE resolver does not time out but fails to provide a valid certificate, we schedule a new scan which bypasses the certificate verification to assess the resolvers' plain DNS functionality. All scan results including every error occurred during the scan execution are ultimately stored in `MongoDB` ⑪ .

To ensure full compliance with the latest development of SVCB-related standards, we contributed to Go's open-source DNS library [53]. We implemented the standard RFC 9540 [123], which introduces the `ohttp` SVCB parameter to indicate support for ODoH in a DDR discovery (see Section 2.3.5). Our implementation has been reviewed and merged into the latest release of the library [145].

4.1.6 Data Enrichment

To analyze the adoption of DDR by various network types, we map the IP addresses to ASes by using *GeoLite2* [111]. To further understand the characteristics of these ASes and the adoption of (encrypted) resolvers across network types, we make use of *PeeringDB* [125], a publicly accessible, user-maintained database that provides essential information for interconnecting networks, data centers, and other facilities. Specifically, we use *PeeringDB*'s `info_type` classification, which categorizes an AS in eight network types (see Table 4.1). For simplicity, we merge *Route Server* and *Route Collector* (purple) as the network *Route Server*; and *Cable/DSL/ISP*, *Network Services*, and *NSP* (blue) as the network category *Network Services*.

Table 4.1: Network categories from *PeeringDB* [125]. For simplicity, we merge the blue colored categories to *ISP* and the purple colored to *Route Server*.

Network Classification	Example
Cable/DSL/ISP	Vodafone (*AS 3209*)
Content	Netflix (*AS 2906*)
Enterprise	Tesla (*AS 394161*)
Educational/Research	Alabama Research and Education Network (*AS 3464*)
NSP	Deutsche Telekom AG (*AS 3320*)
Network Services	Online Telecom Brazil (*AS 271529*)
Non-Profit	DENIC (*AS 31529*)
Government	Irish Government Network (*AS 15806*)
Route Server	Eurasia Peering IX (*AS 63602*)
Route Collector	Riyadh IX Route Collectors (*AS 216406*)

While *PeeringDB* offers a useful classification of ASes, its results may introduce bias, as some classifications are ambiguous, and certain ASes remain unclassified, potentially affecting our analysis (see Section 8.1 for limitations). For every unclassified AS, we introduce a new category *unknown*.

4.1.7 Monitoring and Logging

As the complexity of our architecture increases with the number of scanners and VPs in use, a monitoring solution becomes indispensable for tracking key hardware metrics, such as CPU, memory and disk consumption, bandwidth, packet drops and I/O usage. Such monitoring should encompass not only our DDR and DoE scanners but also external scan software such as *ZMap* or *ZDNS*. To achieve this, we deployed *node_exporter* [157] to collect hardware metrics of the nodes running scanners. Additionally, we enabled the *dockerd* [38] metrics API and used *cAdvisor* [55] to export fine-grained metrics about the performance of dockerized applications on the nodes. The metric data is centrally scraped and stored by *Prometheus* [158], and visualized in a unified dashboard using *Grafana* [56], which retrieves the collected data from *Prometheus*. This setup

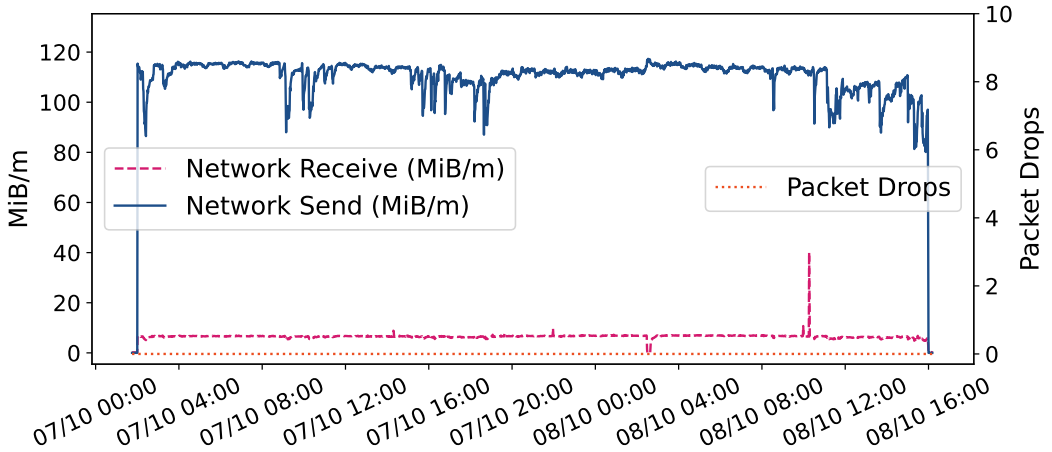


Figure 4.7: An exemplary monitoring output from *Prometheus* which shows the network traffic (send/receive in MiB/minute on the left-hand Y-axis) and the packet drops (right-hand Y-axis) from a full *ZMap* IPv4 address space scan.

helps us to monitor both the performance of our application and detect outages or interruptions during the measurements[4].

For example, in Figure 4.7, the bandwidth usage (receive/send MiB) of the worker node running *ZMap* during a full IPv4 scan is displayed, alongside the packet drops at the driver or hardware level. Throughout the scan, no packets were dropped, indicating that the hardware system was able to sustain *ZMap*'s scanning speed. The figure also highlights fluctuations in bandwidth usage, particularly during typical working hours. Despite our architecture achieving a test coverage of approximately 89% [143], errors or uncovered edge cases may still occur, particularly as the scanners rely on internet responses to determine new scans (e.g., DDR discovery responses). This is especially critical during the initial stages of running scans. To enhance visibility, we log every encountered error, including the exact code line where it occurred. Errors are logged to `STDOUT` and stored in *MongoDB*, with each error linked to the metadata of the corresponding scan.

We have also implemented unified error codes for all known errors encountered during scans, with 53 codes established to date. These codes reduce ambiguity during analysis[5]. For example, we can determine how many servers respond with an invalid or non-parsable DDR configuration by reviewing the error codes in the scan metadata.

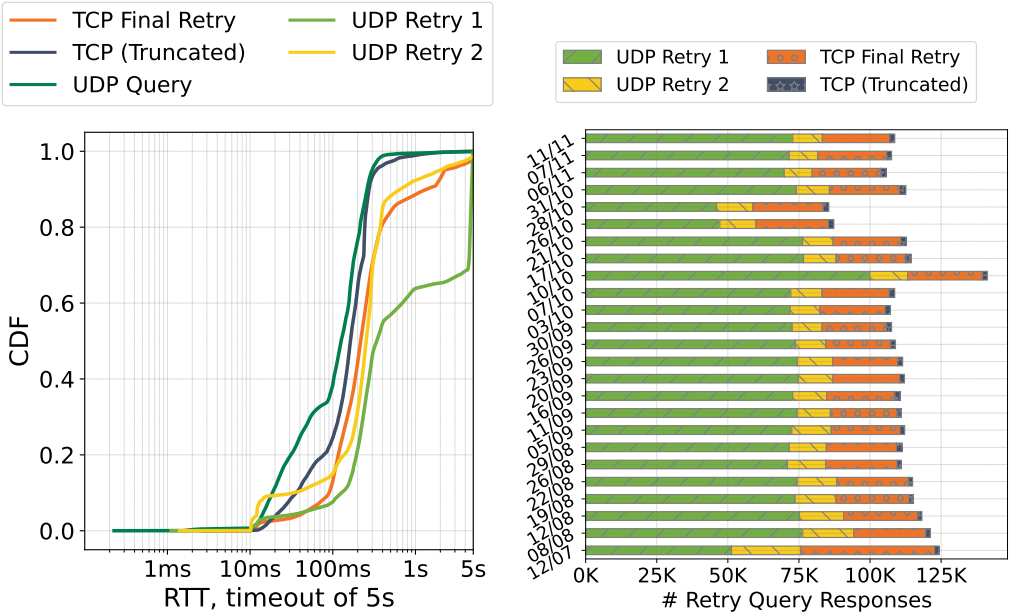
4.2 Measurement Meta-Analysis

To further verify and contextualize our discovered DNS servers, we compare our findings with contemporary databases as *Censys* [41] and research in the field of DNS (cf Section 3.2). Our *ZMap* scans detect an average of approximately 29.9M services responding on UDP/53. According to the second stage of our methodology (see Section 4.1.4), we verify whether an IP address runs a DNS service on UDP/53. Ultimately, we discover an average of 4M IPv4 (287K IPv6) DNS resolvers that reply with any DNS response.

The literature reflects varying numbers of discovered IPv4 DNS servers. Schomp et al. (2012) [148] identified 32M open (recursive) resolvers, while Takano et al. (2013) [156] reported 25M, and Kührer et al. (2014) [90] found only 18M. More recent studies by Park et al. (2019) [122] and Mao et al.

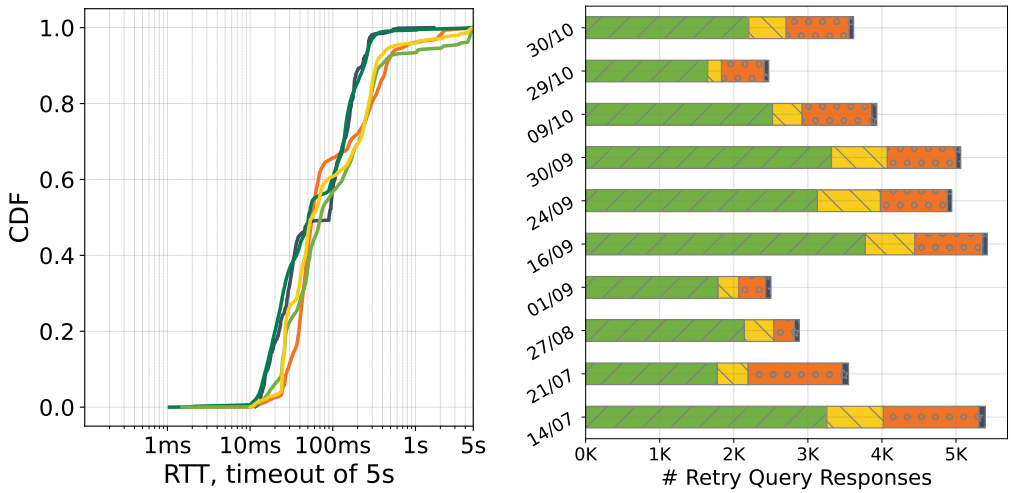
4 The full Prometheus and Grafana setup can be seen in the public repository of *DoE-Hunter* [143].

5 The full error list is available at https://github.com/steffsas/doe-hunter/blob/main/lib/custom_errors/types.go.



(a) RTT CDF for all IPv4 DDR DNS Requests

(b) IPv4 DDR DNS Retries per Run



(c) RTT CDF for all IPv6 DDR DNS Requests

(d) IPv6 DDR DNS Retries per Run

Figure 4.8: CDFs (left-hand side) of all 105M IPv4 and IPv6 RTTs from DDR discovery responses and the effectiveness of our retry strategy (right-hand side).

(2021) [110] report notable fewer, around 3M open resolvers. Our results are relatively close to these findings, though we discovered a slightly higher number of resolvers. We want to highlight that most studies in literature depict the number of open RRs but not DNS servers in general. In contrast, we classify a DNS server as such if we get any DNS response on our DDR discovery.

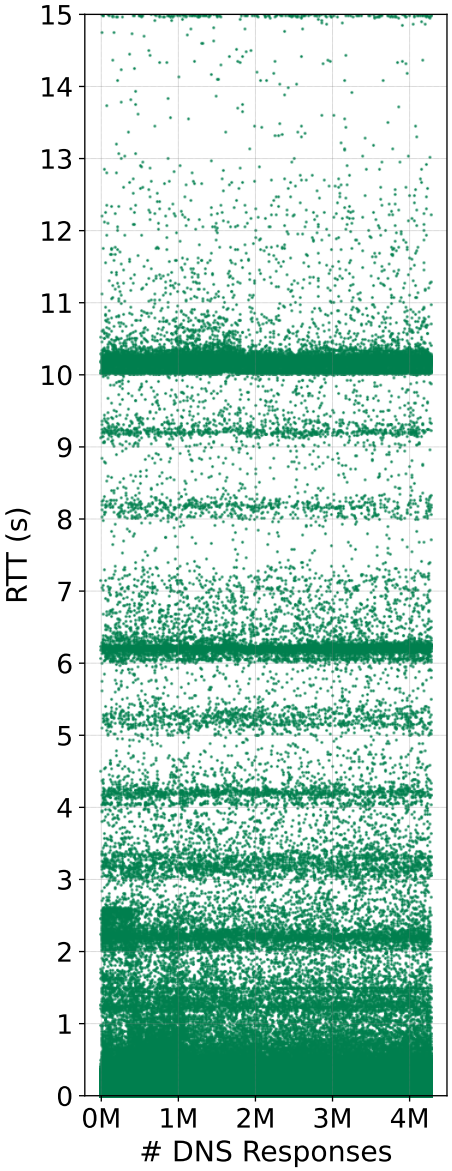


Figure 4.9: RTTs for initial IPv4 UDP DDR discovery queries which answered with any DNS response. It depicts a full IPv4 scan conducted on October 26, 2024, with a maximum timeout of 15 seconds. The queries are arranged in order of their execution time.

We additionally validated our results with databases of *Shodan* [155], *Zoomeye* [168], and *Censys* [41]. *Shodan* and *Zoomeye* identified around 8M to 8.5M hosts with port UDP/53 open, while *Censys* identifies approximately 5.1M DNS servers. We believe that these variations likely stem from differences in the methodologies like retry strategies, VPs, timeouts used for DNS server identification, or VPs.

Our approach differs to other studies due to our retry and backoff strategy (see Figure 4.5). This involves sending an initial DNS request to potential IPv4 or IPv6 DNS servers. If a response is not received within a five-second timeout, we retry twice on UDP/53, followed by a final attempt on TCP/53. The number of DNS servers responding on DDR discovery retry queries is illustrated in Figure 4.8 (b) for IPv4 and Figure 4.8 (d) for IPv6. Overall, the retry strategy allowed us to identify in addition 111K IPv4 (4K IPv6) DNS servers on average,

representing a 2.73% increase for IPv4 (1.39% IPv6) compared to initial requests without retries. By comparing all DDR discovery RTTs in our result set, we observe faster response times when querying servers via IPv6 compared to IPv4 (see Figure 4.8 (a) and Figure 4.8 (c)). Notably, the initial UDP DDR discovery query's in IPv6 is very close in round-trip time to the truncated TCP query, whereas TCP is slower in the IPv4 case. Interestingly, in the case of IPv4, the second UDP query during retries exhibits the slowest response times across all retries.

To investigate the impact of the five-second timeout on missing DNS responses, we conducted a test scan with a 15-second timeout on October 26, 2024 (see Figure 4.9). The results show that 7,840 resolvers responded with a DNS response above a RTT of five seconds, accounting for 1.1% of all responding DNS resolvers. When focusing on responses without errors (*NOERR* RCODE), this figure rises to 1.42% of all non-error responses. Due to resource constraints, subsequent scans will continue to use a five-second timeout (see Section 8.1 for limitations).

Observation: Retry Strategy.

The additional gain in responses from our implemented retry strategy is approximately 2.73% for IPv4 and 1.39% for IPv6, which, given the associated effort and resource consumption, is limited in its justification. This marginal increase aligns with the observations of the *ZMap* authors regarding multiple probes sent [40], who advise against using multiple probes due to the marginal benefit it provides.

4.3 Ethical Considerations

In conducting our measurements, we adhere to established best practices and guidelines [3, 13, 58, 93, 115] to ensure responsible and ethical research. To mitigate any privacy concerns, we do not collect any user-related or personally identifiable information. Our focus is solely on gathering publicly resolvable data regarding DNS servers' DDR and DoE configurations, without attempting to elicit user-specific data. Additionally, we do not seek to exploit or circumvent systems with inadequate security.

To avoid overloading servers with excessive measurement traffic, we limit our DNS server discovery frequency to twice per week. Furthermore, we implement

a caching mechanism to prevent multiple queries for the same DoE resolver, thereby reducing unnecessary traffic. By using well-established scanning tools such as *ZMap* [43] and data from *IPv6 Hitlist Service* [52, 146], we ensure that our measurements do not overwhelm networks. For instance, *ZMap* employs a permutation approach, which randomly selects IP addresses to scan rather than following a sequential numerical order, thus preventing concentrated network load [43].

We also have proactively configured our scanner nodes and authoritative name servers to ensure that our measurements are transparent and traceable. This includes adding `TXT` ResRs for the wildcard query name *measurement.raiun.de*, used in DoE probes, which resolves to an email distribution address allowing system administrators to contact our research team. Additionally, we have set up reverse DNS (rDNS) such that the IP addresses of our measurement worker nodes can be resolved to hostnames. These hostnames link to a website (measurement hint)[6] that explains our measurement approach and provides contact information to our research team.

To date, one system administrator has contacted us via the provided email address, requesting that their network not be scanned. In response, we added their network to our blocklist. This blocklist, which prevents scans on specific networks (in the form of a CIDR IP address), is further enriched by incorporating information from abuse reports collected by other research projects, such as *MassDNS* [7]. This helps to mitigate potential abuse concerns in advance.

In order to support transparency and promote further research in the areas of DDR and DoE, we plan to publish all collected data online, while ensuring that privacy concerns are fully addressed.

6 The measurement hint can be accessed at www.measurement.raiun.de.

5 DNS Resolvers

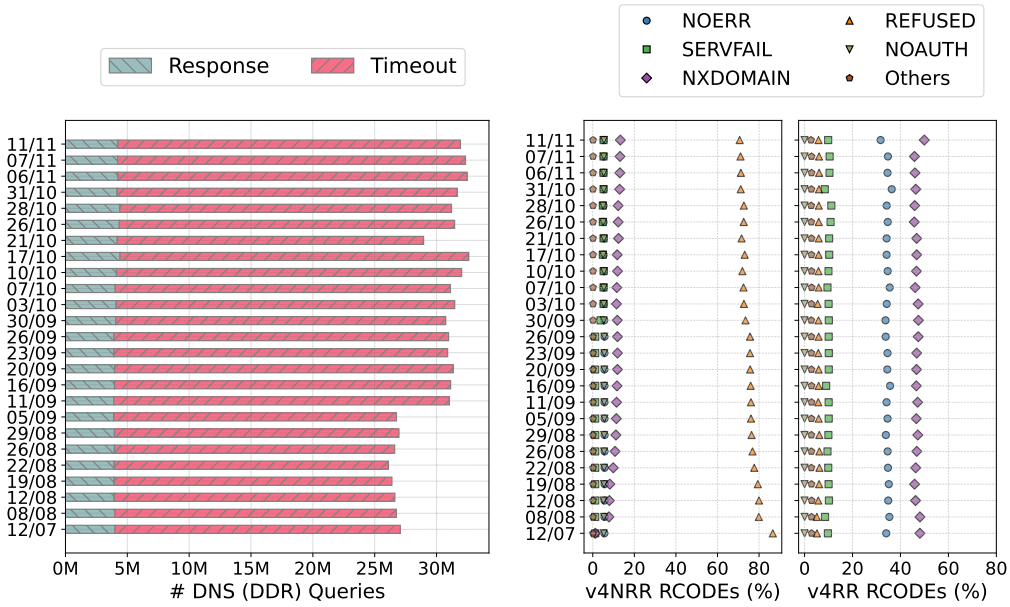
This chapter investigates the first two methodological stages of our study (see Section 4.1.3 and Section 4.1.4), focusing on the measurement and analysis of DNS resolvers and their behavior across IPv4 and IPv6 address spaces over a four-month period. In Section 5.1, we examine the change rates and response patterns of DNS servers, with distinctions between recursive and non-recursive resolvers and notable differences in RCODE distributions. Section 5.2 explores the geographical distribution of DNS servers, uncovering regional disparities in deployment and response behaviors. Finally, Section 5.3 investigates server classification by AS and network categories. These analyses provide a baseline to understand DDR's operational characteristics and broader implications for the DNS ecosystem.

5.1 Change Rates, Response Patterns and Trends

Over a four-month period, from 12 July 2024 to 11 November 2024, we scanned 746.6M IPv4 addresses that potentially offer DNS services. For the IPv4 address space, *ZMap* covered 3.7B addresses for each scan run, yielding an average hit rate of about 0.8%. Consequently, each run resulted in an average of around 29.9M IPv4 servers potentially running a DNS service (see Figure 5.1 (a)). Note that these results do not necessarily represent actual DNS servers, as *ZMap* counts any response as a "hit", including those from services other than DNS that reply to DNS probes.

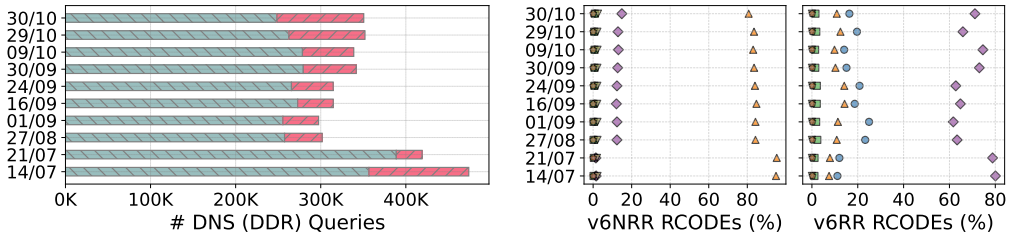
Using *ZMap*'s result set, we conducted DDR discovery queries, which yielded responses from approximately 4M (13.9%) DNS servers on average. While the number of IPv4 addresses detected by *ZMap* increased from the initial scan in July to the final scan in November by around 4.9M (see Figure 5.1 (a)), the number of actual DNS servers responding remained stable at approximately 4.1M throughout all scans.

For IPv6, we conducted DDR discovery queries directly from the set of *IPv6 Hitlist Service*'s responsive IPv6 addresses. In total, we scanned 3.5M IPv6 addresses while the number of addresses scanned in each run varied due to the fluctuation of *IPv6 Hitlist Service*'s result set (see Figure 5.1 (c)). In contrast



(a) IPv4 addresses that responded or timed out to the DDR discovery queries.

(b) DDR discovery replies and their RCODE for IPv4 non-recursive resolvers (v4NRR, left) and IPv4 recursive resolvers (v4RR, right).



(c) IPv6 addresses that responded or timed out to the DDR discovery queries.

(d) DDR discovery replies and their RCODE for IPv6 non-recursive resolvers (v6NRR, left) and IPv6 recursive resolvers (v6RR, right).

Figure 5.1: Overview of the responses to every DDR discovery scan over a time frame of four months in 2024. The left-hand side shows the DNS replies and timeouts for every IPv4 and IPv6 we have scanned. The right-hand side zooms into the DDR discovery (DNS) replies and shows the number of RCODEs for each run for both, RR and NRRs resolvers. For detailed figures refer to Section B.1.

to the relatively low response rate for IPv4 addresses (13.7%), an average of 287K (82%) returned a DNS response. This high response rate is almost the inverse of the IPv4 timeout rate, depicting a reversal in responsiveness between IPv4 and IPv6.

Focusing on the IPv4 DNS servers that replied with any DNS response to the DDR discovery (see Figure 5.1 (b)), we observe a shift in RCODEs between IPv4 Recursive Resolver (v4RR) and IPv4 Non-Recursive Resolver (v4NRR) over time. Specifically, there is an increase in v4NRRs returning `NXDOMAIN` (non-existing domain), rising from around 1.2% (35K) in the first scan to 13.3% (406K) in the latest scan. This trend likely reflects the fact that DDR is primarily standardized for RRs [124], meaning that authoritative name servers are not the intended targets of the protocol. Similarly, we see a rise in `SERVFAIL` (server failure) responses from v4NRRs, increasing from around 33K (1.2%) on the 26th of September to approximately 159K (5.2%) by mid-November. Conversely, the portion of `REFUSED` responses from v4NRRs decreased, from 2.5M (86.6%) to 2.2M (70.7%). The number of `NOERR` (no error) replies remained stable throughout the scanning period. Interestingly, the RCODE distribution for v4RRs (the target group for the DDR protocol) was relatively stable across scans. Notably, while v4RRs returned `NOERR` in about 379K (34.5%) responses on average, v4NRRs returned this RCODE in only about 163K responses (5.5%). This indicates that v4RRs are more likely to successfully respond to DDR discovery queries than v4NRRs, highlighting again DDR's focus on RRs. Additionally, v4NRRs refused the DDR discovery in 2.2M responses (75%) on average, compared to only 64K (5.8%) refusals from v4RRs per scan.

We observe a different RCODE distribution for IPv6 DNS servers compared to IPv4. For instance, IPv6 Recursive Resolver (v6RR) returned `NOERR` in only about 16% of the responses, compared to around 35% for IPv4, indicating that v4RRs are more familiar with DDR discovery queries than v6RRs (see Figure 5.1 (d)). Similar to IPv4 resolvers, we see an increasing trend of IPv6 Non-Recursive Resolver (v6NRR) returning `NXDOMAIN` and a declining trend in `REFUSED` responses. Initially, around 4K (1.6%) v6NRRs returned `NXDOMAIN`, which rose to 28K (14.9%) by the final scan. For `REFUSED` responses, about 240K (94.9%) of v6NRRs returned this code in the first scan, which dropped to around 152K (80.7%) by the last scan. In general, it appears that v6RRs exhibit greater instability in RCODEs distribution over time compared to IPv4 resolvers (see right side of Figure 5.1 (d)).

Please note that detailed figures of the plots can be found in Section B.1.

Observation: DNS RCODES.

Our scans revealed a notable difference in RCODE distributions between IPv4 and IPv6 DNS servers. IPv4 recursive resolvers (v4RR) consistently returned more `NOERR` responses compared to non-recursive resolvers (v4NRR), highlighting their better compatibility with DDR discovery queries. Conversely, IPv6 resolvers showed greater instability in RCODE distribution over time, with an increase in `NXDOMAIN` responses from non-recursive resolvers (v6NRRs).

5.2 Geographical Insights and Global Distribution

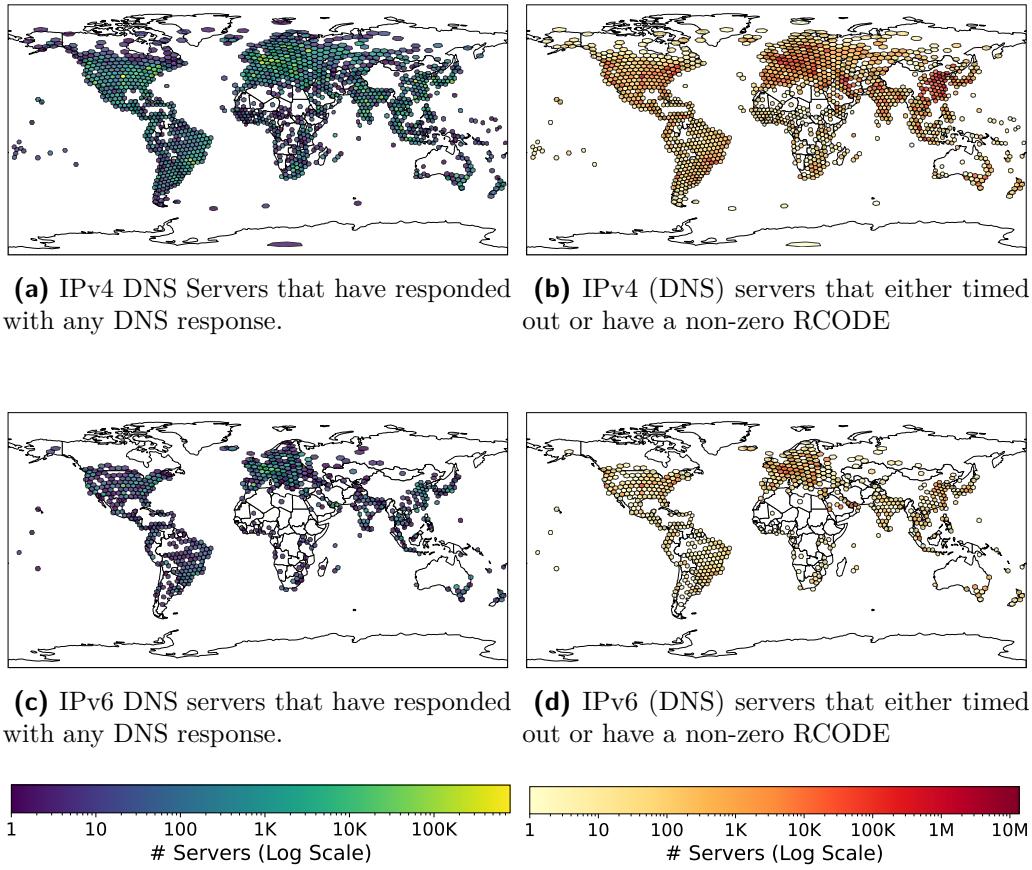
By mapping the IP addresses to their geolocations using *GeoLite2* [111], we can more accurately visualize the global distribution of IP addresses and DNS servers.

In analyzing the IP addresses from our latest scan on November 11, 2024, we observe that over 88.7% (28.2M) of the IPv4 addresses identified by *ZMap* are located in Asia, followed by 1.5M (4.76%) in Europe and 1.5M (4.72%) in North America (for detailed figures, refer to Section B.1). Notably, the number of IPv4 addresses in Asia increased by 4.8M from mid-July to mid-November 2024, while the counts for other continents remained relatively stable.

More than 65% (20M) of all IPv4 addresses are located in China, followed by Iran (20.3%, 6.5M), the United States (4.2%, 4.2M), and Germany with only 0.78% (250K). We want to highlight that the number of IPv4 addresses located in China raised from 16M (59%) to over 20M (65%) from the first scan in mid-July to the last in mid-November 2024.

Interestingly, the distribution of IPv6 addresses from the *IPv6 Hitlist Service* presents a different pattern. Most IPv6 addresses are located in Europe (151K, 45.8%), followed by Asia (94K, 28.2%) and North America (66K, 19.8%). On a country level, most IPv6 addresses are located in the United States (17%, 58K), followed by France (9.8%, 34K), United Arab Emirates (9.77%, 33K), and Germany 26K (7.53%). We observe a high fluctuation in IPv6 addresses within the hitlist data. For example, the number of IPv6 addresses in China nearly halved by 21K from approximately 46K in mid-July to mid-November 2024. Similarly, IPv6 addresses in the United Arab Emirates dropped from around 83K to 33K within our measurement period.

Turning focus on IP addresses that timed out or returned errors (RCODE $\neq 0$), we observe again differences between IPv4 and IPv6 addresses. Around



(a) IPv4 DNS Servers that have responded with any DNS response.

(b) IPv4 (DNS) servers that either timed out or have a non-zero RCODE

(c) IPv6 DNS servers that have responded with any DNS response.

(d) IPv6 (DNS) servers that either timed out or have a non-zero RCODE

Figure 5.2: (DNS) servers that responded to our DNS queries, along with their distribution across the globe. The figures on the left display servers that returned any DNS response. The figures on the right show servers that either timed out or returned a response with a non-zero RCODE (error). The data reflects the latest scans from November 11, 2024, for IPv4 and October 30, 2024, for IPv6.

89.5% of the IPv4 addresses found by *ZMap* timed out or returned an error in the Asia region, while in contrast only around 5% of the IPv4 addresses in Europe and North America did so (see Figure 5.2 (b)). Zooming in on the level of countries, we see that China (20.6M, 65.78%) and Iran (6.4M, 23.6%) are the most prevalent ones, followed by the United States (1.2M, 3.8%) and Germany (239K, 0.76%). Further, the number of timed out or error-returning IPv4 addresses in China increased by around 4M from the first scan in July to the last scan in November, while the other regions are more or less stable. In general, we believe these figures result from censorship or filtering mechanisms

in place in China and Iran. In comparison, most IPv6 addresses that timed out or returned an error are located in Europe (153K, 46.56%), followed by Asia (around 97K, 29.51%) and North America (around 62K, 18.86%). Notably, Belarus, Saudi Arabia and the United Arab Emirates had the highest percentage of timed out or faulty DNS responses in the last IPv6 scan with 11K (99.97%), 12K (99.95%), and 33K (99.89%), respectively. These top three remained with the highest percentage of timed out or faulty DNS responses throughout all IPv6 scans.

Since we cannot draw conclusions about IP addresses that time out, we focus exclusively on the IP addresses that provided a DNS response, as these must necessarily be DNS servers. Our analysis shows that the majority of IPv4 DNS servers are located in the United States (29.4%), followed by China (257K, 6.3%), Germany (225K, 5.5%), and Brazil (218K, 5.3%) (see Figure 5.2 (a)). The distribution of DNS servers across countries remained largely stable throughout the scan period.

In contrast, the distribution of IPv6 DNS servers showed more variation over time (see Figure 5.2 (c)). In the first scan, the United Arab Emirates had the highest number of responsive DNS servers (57.2K, 16.79%), but this number more than halved to 21.5K (8%) in the final IPv6 scan conducted in mid-November 2024. By the last scan, the United States had the most responsive IPv6 DNS servers (45K, 16.7%), followed by Germany (23.6K, 8.8%), the United Arab Emirates (21.5K, 8%), France (20.3K, 7.5%), and China (15.9K, 5.9%). It is important to note that these figures also include servers returning a non-zero RCODE, as these still constitute DNS responses.

Observation: Geographical Distribution of (DNS) Servers.

Our analysis of the distribution of IPv4 addresses identified by *ZMap* shows that the majority are located in Asia (88.7%), with a particularly high concentration in China (65%). In contrast, most IPv6 addresses from the *IPv6 Hitlist Service* are located in the United States (17%). The proportion of timed-out or error-returning IPv4 addresses is notably higher in Asia, especially in China (20.6M, 65.78%) and Iran (6.4M, 23.6%). For IPv6, the majority of timed-out or error-returning IP addresses are located in Europe (46.56%). Focussing on DNS servers only, the United States hosts the largest share of IPv4 and IPv6 servers with 29.4% and 16.7%, respectively.

5.3 Distribution across ASes and Network Categories

To provide a more detailed picture of the distribution of DNS servers, we classify them based on their AS and network type. This classification utilizes geolocation data from *GeoLite2* [111] and network categorization data from *PeeringDB* [125] (see Section 4.1.6). We further separate the servers into RR and non-RR, based on the presence of the `RA` bit in the DNS response header.

The approximately 746.6M scanned IPv4 addresses are distributed across 48,921 unique ASes, while the 3.5M IPv6 addresses scanned span across 9,506 unique ASes. The distribution of DNS servers across different network categories and ASes for both RRs and non-RRs in the IPv4 space is shown in Table 5.1. Analogous figures for IPv6 can be found in Table 5.2.

There are approximately three times more non-RRs than RRs in both IPv4 and IPv6 families. However, while the number of IPv4 DNS servers increased from the first to the last scan, we observe the opposite trend for IPv6 servers.

Table 5.1: Classification of v4RR and v4NRR DNS servers by network category for the first and last scan, including the percentage change in the number of DNS servers and ASes.

a Classification of v4RR servers by network category.

Category	First Scan 12.07.24		Last Scan 11.11.24		% Change First to Last	
	# DNS Servers	# ASes	# DNS Servers	# ASes	# DNS Serv.	# ASes
Network Services	656,324 (61.18%)	8,421 (35.59%)	693,919 (59.01%)	8,375 (35.48%)	37,595 (5.73%)	-46 (-0.55%)
unknown	318,288 (29.67%)	14,001 (59.18%)	318,024 (27.04%)	14,011 (59.36%)	-264 (-0.08%)	10 (0.07%)
Content	66,233 (6.17%)	643 (2.72%)	66,879 (5.69%)	645 (2.73%)	646 (0.98%)	2 (0.31%)
Educational/Research	24,749 (2.31%)	211 (0.89%)	91,132 (7.75%)	193 (0.82%)	66,383 (268.22%)	-18 (-8.53%)
Enterprise	6,279 (0.59%)	274 (1.16%)	5,545 (0.47%)	276 (1.17%)	-734 (-11.69%)	2 (0.73%)
Non-Profit	496 (0.05%)	72 (0.30%)	289 (0.02%)	66 (0.28%)	-207 (-41.73%)	-6 (-8.33%)
Route Server	307 (0.03%)	9 (0.04%)	71 (0.01%)	9 (0.04%)	-236 (-76.87%)	0 (0.00%)
Government	90 (0.01%)	27 (0.11%)	125 (0.01%)	30 (0.13%)	35 (38.89%)	3 (11.11%)
Total	1,072,766	23,658	1,175,984	23,605	103,218(9.62%)	-53(-0.22%)

b Classification of v4NRR servers by network category.

Category	First Scan 12.07.24		Last Scan 11.11.24		% Change First to Last	
	# DNS Servers	# ASes	# DNS Servers	# ASes	# DNS Serv.	# ASes
Content	1,347,575 (45.84%)	1,364 (3.55%)	1,342,896 (43.89%)	1,369 (3.58%)	-4,679 (-0.35%)	5 (0.37%)
unknown	864,065 (29.39%)	25,493 (66.28%)	866,592 (28.32%)	25,170 (65.83%)	2,527 (0.29%)	-323 (-1.27%)
Network Services	634,256 (21.57%)	10,060 (26.16%)	641,475 (20.97%)	10,149 (26.54%)	7,219 (1.14%)	89 (0.88%)
Enterprise	65,783 (2.24%)	729 (1.90%)	54,270 (1.77%)	727 (1.90%)	-11,513 (-17.50%)	-2 (-0.27%)
Educational/Research	24,765 (0.84%)	460 (1.20%)	150,892 (4.93%)	463 (1.21%)	126,127 (509.30%)	3 (0.65%)
Non-Profit	2,915 (0.10%)	274 (0.71%)	3,009 (0.10%)	280 (0.73%)	94 (3.22%)	6 (2.19%)
Government	416 (0.01%)	54 (0.14%)	388 (0.01%)	49 (0.13%)	-28 (-6.73%)	-5 (-9.26%)
Route Server	117 (0.00%)	26 (0.07%)	154 (0.01%)	28 (0.07%)	37 (31.62%)	2 (7.69%)
Total	2,939,892	38,460	3,059,676	38,235	119,784(4.07%)	-225(-0.59%)

Table 5.2: Classification of v6RR and v6NRR DNS servers by network category for the first and last scan, including the percentage change in the number of DNS servers and ASes.

a Classification of v6RR servers by network category.

Category	First Scan 12.07.24		Last Scan 30.10.24		% Change First to Last	
	# DNS Servers	# ASes	# DNS Servers	# ASes	# DNS Serv.	# ASes
Network Services	84,145 (80.70%)	1,384 (53.71%)	45,144 (74.78%)	1,247 (53.11%)	-39,001 (-46.35%)	-137 (-9.90%)
unknown	16,805 (16.12%)	851 (33.02%)	12,555 (20.80%)	795 (33.86%)	-4,250 (-25.29%)	-56 (-6.58%)
Content	2,353 (2.26%)	167 (6.48%)	1,939 (3.21%)	143 (6.09%)	-414 (-17.59%)	-24 (-14.37%)
Educational/Research	674 (0.65%)	86 (3.34%)	495 (0.82%)	83 (3.53%)	-179 (-26.56%)	-3 (-3.49%)
Enterprise	176 (0.17%)	48 (1.86%)	147 (0.24%)	43 (1.83%)	-29 (-16.48%)	-5 (-10.42%)
Non-Profit	109 (0.10%)	38 (1.47%)	86 (0.14%)	35 (1.49%)	-23 (-21.10%)	-3 (-7.89%)
Government	3 (0.00%)	3 (0.12%)	2 (0.00%)	2 (0.09%)	-1 (-33.33%)	-1 (-33.33%)
Total	104,265	2,577	60,368	2,348	-43,897(-42.10%)	-229(-8.89%)

b Classification of v6NRR servers by network category.

Category	First Scan 12.07.24		Last Scan 30.10.24		% Change First to Last	
	# DNS Servers	# ASes	# DNS Servers	# ASes	# DNS Serv.	# ASes
Content	122,583 (48.58%)	676 (9.27%)	94,077 (50.01%)	638 (9.61%)	-28,506 (-23.25%)	-38 (-5.62%)
unknown	76,844 (30.45%)	2,858 (39.20%)	52,293 (27.80%)	2,531 (38.11%)	-24,551 (-31.95%)	-327 (-11.44%)
Network Services	43,337 (17.17%)	3,005 (41.22%)	33,032 (17.56%)	2,779 (41.85%)	-10,305 (-23.78%)	-226 (-7.52%)
Educational/Research	5,323 (2.11%)	335 (4.59%)	5,173 (2.75%)	309 (4.65%)	-150 (-2.82%)	-26 (-7.76%)
Enterprise	2,453 (0.97%)	200 (2.74%)	2,103 (1.12%)	184 (2.77%)	-350 (-14.27%)	-16 (-8.00%)
Non-Profit	1,696 (0.67%)	190 (2.61%)	1,383 (0.74%)	177 (2.67%)	-313 (-18.46%)	-13 (-6.84%)
Government	60 (0.02%)	15 (0.21%)	43 (0.02%)	12 (0.18%)	-17 (-28.33%)	-3 (-20.00%)
Route Server	37 (0.01%)	12 (0.16%)	28 (0.01%)	11 (0.17%)	-9 (-24.32%)	-1 (-8.33%)
Total	252,333	7,291	188,132	6,641	-64,201(-25.44%)	-650(-8.92%)

Simultaneously, the AS diversity decreased for DNS servers in both IP families, pointing in the direction of DNS centralization, which is also discussed in recent literature [74, 91].

If we analyze the categories presented in Table 5.1a and Table 5.1b, most v4RRs are classified as Network Services (694K, 59.01%), whereas the majority of v4NRRs are categorized as Content Providers (1.3M, 43.89%). The high proportion of Network Services across v4RRs stems likely from the fact that recursive resolvers are part of the ISP's infrastructure to serve their client's DNS requests.

Over time, we see the largest positive percentage changes in the Educational/Research category, with v4RRs increasing by 66K (268.22%) and v4NRRs by 126K (509.3%). As a result, the number of DNS servers in the Educational/Research category surpassed those in the Enterprise category in the final scan. However, at the same time, the Educational/Research category has the most negative percentage change in ASes across v4RRs with a loss of 18 ASes (-8.53%). For v4NRR it is the Government category with 4 fewer

ASes (-9.26%). In general, the percentage changes in the IPv4 AS distribution reveals a stable trend with slight decreases.

When comparing IPv4 DNS servers to those in the IPv6 space, distinct patterns emerge (see Table 5.2a and Table 5.2b). Unlike IPv4, where the total number of DNS servers increased, both v6RRs and v6NRR showed considerable decreases. In total, v6RRs decreased by 44K (-42.10%) and v6NRRs by 64K (-25.44%). This trend is evident across all categories, with the largest decrease observed in Network Services for v6RRs (-39K, -46.35%) and Content for v6NRRs (-25.5K, -23.25%). The decline in servers also results in a reduction of AS diversity, decreasing by approximately 9% for both v6RRs and v6NRRs. We attribute these variations primarily to fluctuations in the *IPv6 Hitlist Service* data.

Turning perspective on the ASes, Table 5.3a and Table 5.3b show the top ten ASes of IPv4 and IPv6 DNS servers ordered by the number of servers within the respective AS. For IPv4, AS *46606* has the largest share of DNS servers (172K,

Table 5.3: Top 10 ASes by number of DNS servers, including their country and network type.

a IPv4 DNS server figures as of November 11, 2024.

	AS No.	AS Organization	Country	Network Type	# DNS Servers
1	46606	UNIFIEDLAYER-AS-1	United States	Content	171,793 (4.06%)
2	19551	INCAPSULA	United States	Content	160,040 (3.78%)
3	53166	UNIVERSIDADE ESTADUAL PAULISTA	Brazil	Educational/Research	126,777 (2.99%)
4	16276	OVH SAS	France	Content	99,995 (2.36%)
5	4538	China Education and Research Network	China	Educational/Research	90,183 (2.13%)
6	19871	NETWORK-SOLUTIONS-HOSTING	United States	unknown	73,694 (1.74%)
7	24940	Hetzner Online GmbH	Germany	Content	70,948 (1.68%)
8	4837	CHINA UNICOM Backbone	China	Network Services	67,206 (1.59%)
9	4134	Chinanet	China	Network Services	65,729 (1.55%)
10	17488	Hathway IP Over Cable Internet	India	Network Services	50,685 (1.20%)

b IPv6 DNS server figures as of September 30, 2024.

	AS No.	AS Organization	Country	Network Type	# DNS Servers
1	8966	Emirates Telecommunications Group	United Arab Emirates	Network Services	23,691 (9.53%)
2	198066	Grupo Loading Systems, S.L.	Spain	unknown	14,653 (5.90%)
3	16276	OVH SAS	France	Content	13,362 (5.38%)
4	205016	HERN Labs AB	unknown	unknown	9,931 (4.00%)
5	20940	Akamai International B.V.	United States	Content	7,110 (2.86%)
6	19551	INCAPSULA	United States	Content	4,717 (1.90%)
7	12876	Scaleway S.a.s.	France	Content	4,408 (1.77%)
8	20857	Signet B.V.	The Netherlands	Content	4,162 (1.67%)
9	63949	Akamai Connected Cloud	United States	Content	4,036 (1.62%)
10	4837	CHINA UNICOM Backbone	China	Network Services	3,595 (1.45%)

4.06%), followed by AS *19551* (160K, 3.78%). Both are content providers, with AS *46606* belonging to *Newfold Digital Inc.*, a web domain and hosting provider, and AS *19551* to *Incapsula*, a cloud-based security services provider. In contrast, for IPv6, AS *8966* (*Emirates Telecommunications Group*), an ISP, hosts the largest share of servers (21K, 7.94%)

Focusing only on RRs, notable regional trends between IPv4 and IPv6 emerge. Most v4RRs are based in Asia, with the most prominent AS being *4538* (*China Education and Research Network*), hosting 88K (7.5%) of all v4RRs, followed by *Chinanet* with 59K (4.99%) and *CHINA UNICOM Backbone* with 56K (4.78%). For v6RRs, the most prevalent AS is *8966* (Emirates Telecommunications Group), with an impressive 24K (39.24%) of all v6RRs, followed by AS *56041* (CHINA UNICOM Backbone) with comparably fewer servers (3.3K, 5.60%).

Observation: ASes and Network Categories.

We observe that recursive resolvers are more prevalent than non-recursive ones, with a ratio of approximately three to one. While the total number of IPv4 DNS servers shows a slight increase, the number of IPv6 DNS servers is declining more noticeably. Geographically, the majority of RRs are located in Asia, whereas non-RRs are more commonly found in the United States and Europe. Generally, the most prominent ASes hosting RR servers are associated with network services, while content providers are more prevalent among non-RRs. The observed reduction in ASes diversity may suggest trends toward DNS centralization, a topic explored in recent literature [74, 91].

6 Discovery of Designated Resolvers (DDR) Ecosystem

This chapter investigates the responses and adoption patterns of DNS servers implementing the *Discovery of Designated Resolvers* (DDR) protocol across both IPv4 and IPv6 networks, based on data collected during the second stage of our measurements (see Section 4.1.4). Over a four-month period (July 12 to November 11, 2024), we executed over 750M DNS queries, obtaining 105M responses related to DDR discoveries, targeting both authoritative and recursive resolvers. Using this dataset, we classify DNS servers, assess adoption patterns and trends in Section 6.1, and evaluate how DDR contributes to DNS centralization in Section 6.2.3. Additionally, we examine configuration patterns and advertised DoE protocols in Section 6.2 and analyze resulting security risks and privacy implications in Section 6.4.

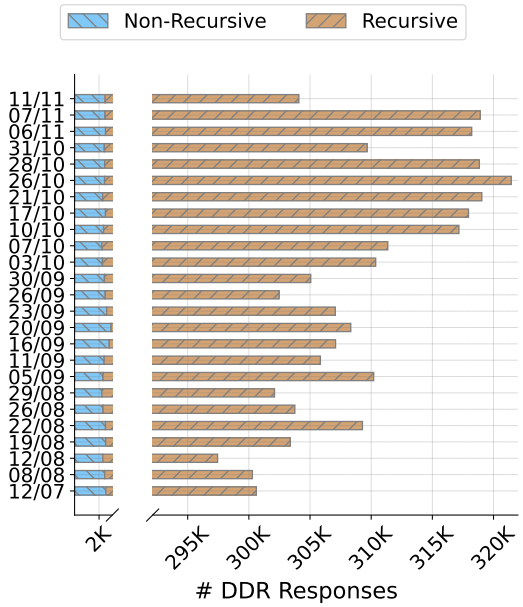
Throughout this chapter, we classify DNS servers as DDR-enabled or DDR-disabled based on their responses to DDR discovery queries: DDR-enabled servers return non-empty responses with a zero RCODE (*NOERR*), while DDR-disabled servers provide empty responses with an RCODE of zero. Since both DDR-enabled and DDR-disabled servers reply with a zero RCODE, we define a DNS server as supporting DDR if it falls into either category. In addition to better analyze DDR's adoption rate over time, we introduce the concept of *density* (ρ). We define the DDR density as the percentage of DDR-enabled servers relative to all DNS servers: $\rho_{DDR} = \frac{\#\text{ DDR-enabled Servers}}{\#\text{ DNS Servers}} \cdot 100$. This metric accounts for fluctuations in the total number of DNS servers when evaluating DDR adoption trends.

6.1 Adoption Rates and Density Trends

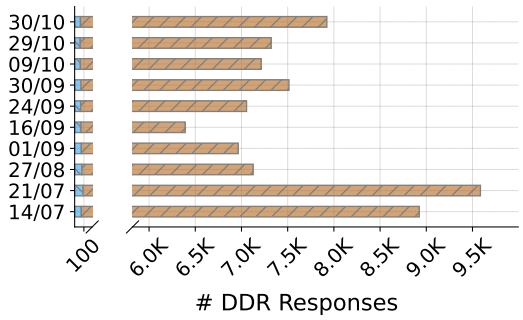
The development of DDR-enabled DNS servers is shown in Figure 6.1 (a) for IPv4 and Figure 6.1 (b) for IPv6. The figures show the number of DDR-enabled DNS servers over time, distinguishing between recursive and non-recursive resolvers based on the `RA` bit in the DNS response header.

From the on average 4M IPv4 DNS servers discovered (see Section 5.1), 541K (13.29%) DNS servers support DDR. Of these 541K, 309K (57.10%) are DDR-enabled, i.e., they return a DDR configuration. As a result, 7.59% of all

IPv4 DNS resolvers are DDR-enabled on average, which is also the average density of DDR-enabled servers.



(a) Number of IPv4 DDR-enabled DNS recursive and non-recursive resolvers over time.



(b) Number of IPv6 DDR-enabled DNS recursive and non-recursive resolvers over time.

Figure 6.1: Evolution of the DDR-enabled DNS servers discovered over a time frame of around four months in 2024. The upper figure shows DDR-enabled resolvers in the IPv4 address space, the lower figure IPv6 DDR-enabled resolvers.

For IPv6, the proportion of DDR-supporting DNS servers is lower. From the 287K IPv6 DNS servers discovered on average, 12K (4.08%) support DDR. Among these 12K, around 8K (65.06%) are DDR-enabled, indicating a higher proportion within this subset than IPv4. Consequently, there is an average DDR density of 2.65% in the IPv6 space, meaning 2.65% of all IPv6 DNS resolvers are DDR-enabled.

In the first IPv4 scan conducted in mid-July 2024, we observed 300K DDR-enabled servers. This number reached its peak at 321K on October 26, before declining to 304K in the final scan on November 11, 2024. Overall, we observe only a slight upward trend in the number of DDR-enabled servers during our measurement period (+3,479).

Conversely, in the IPv6 space, the number of DDR-enabled servers fluctuates more. The number of DDR-enabled servers increased from 9K in the first scan to 9.6K in the second scan, before dropping to 7.9K in the final scan on October 30, 2024. As a result, we see a decreasing number of DDR-enabled servers in the IPv6 space over time (-1,000).

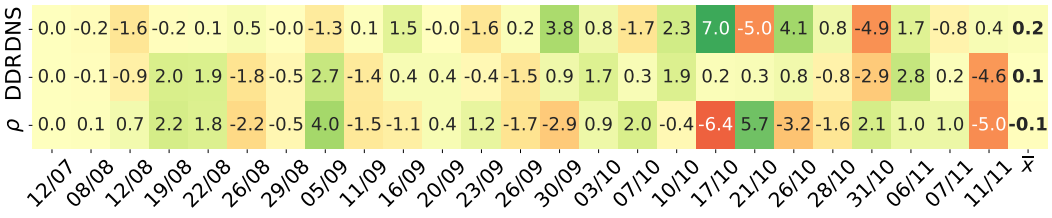
The vast majority of IPv4 DDR-enabled servers belong to DDR's primary target group, recursive

resolvers (RRs), with an average of 99.21% (307K) of the DDR-enabled servers. If we compare the DDR-enabled servers among v4RRs and v4NRRs separately, we see that only 1.51% (2.5K) of the v4NRR advertise any DDR configuration on average, compared to 80.97% (307K) within the set of v4RR DDR-enabled servers.

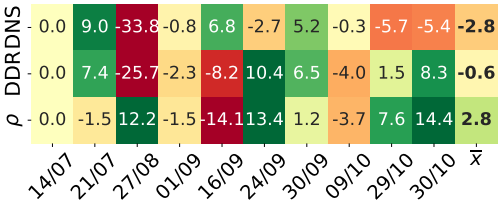
The differences between v6RR and v6NRR DDR-enabled servers are similar to those in IPv4. On average, 99.09% (7.5K) of all DDR-enabled servers are v6RR, while only 3.6% (69) of all v6NRR servers are DDR-enabled. This is slightly higher than the 1.54% (305K) adoption rate of v4NRR servers. If we only consider the set of v6RRs, 77.24% (7.5K) of them are DDR-enabled on average, which is slightly lower than the 80.97% (307K) of v4RR servers.

The density of DDR-enabled servers over time is shown in Figure 6.2 for both IPv4 and IPv6. The data shows that, on average, the density of DDR-enabled servers in the IPv4 space slightly decreases (-0.1%) over the measurement period, despite increases in the absolute number of DDR servers (0.1%) and DNS servers (0.2%). This indicates that the number of DNS servers grows faster than the number of DDR-enabled resolvers. This becomes even more apparent if we only consider recursive resolvers (RRs). While the number of RRs increases by 0.4%, the adoption rate only did by 0.1%, resulting in a percentage change of -0.3% for the density of DDR-enabled RRs. Especially the scan on the November 11, 2024, shows a large decrease in the density of DDR-enabled servers of -4.6% and -9.7% if we only consider RRs.

In the case of IPv6, it is the opposite. While the number of DNS servers



(a) IPv4 Percentage Changes



(b) IPv6 Percentage Changes

Figure 6.2: The figures show the percentage change for the number of DNS and DDR servers over time. ρ depicts DDR's density percentage change. That is, the number of DDR servers divided by the number of DNS servers. $\overline{x}$ shows the average percentage change.

(-2.8%) and DDR-enabled servers (-0.6%) decreases on average, the density of DDR-enabled servers increases by 2.8%. This becomes even more prevalent when focusing only on RRs: The density increases then by 9.1% on average.

Observation: DDR Adoption and Density.

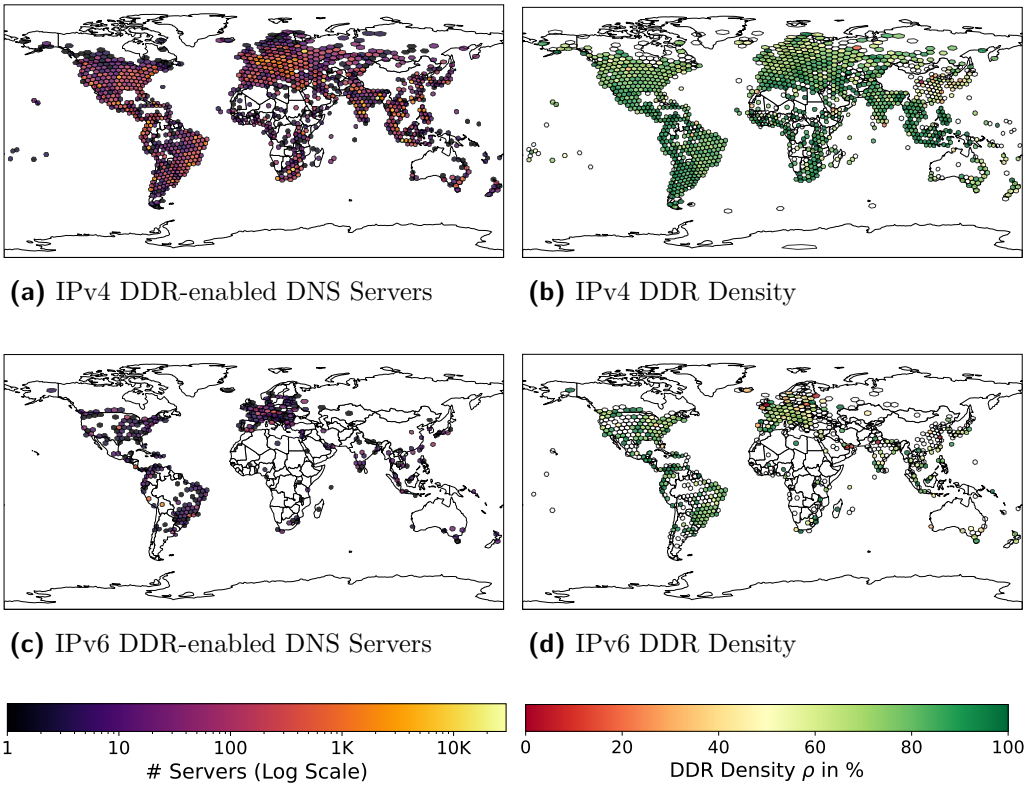
Form 4M IPv4 (287K IPv6) DNS servers discovered, 7.59% (2.65% IPv6) are DDR-enabled on average. The number of IPv4 DDR-enabled resolvers increased slightly by 3.5K during the measurement period, but their DDR density declined (-0.1%). Conversely, IPv6 DDR-enabled resolvers saw a 1K decrease in absolute numbers, while DDR density rose by 2.8%. As the four-month measurement period provides limited insights into the adoption of a newly standardized protocol, longer-term observations are required for more reliable conclusions.

6.1.1 Geographical Insights and Global Distribution

The distribution and density of DDR-enabled IPv4 and IPv6 DNS servers across the globe is shown in Figure 6.3. Figure 6.3 (a) and Figure 6.3 (c) show the number of DDR-enabled servers in different regions for IPv4 and IPv6. Dark-purple hexagons (cold regions) indicate a low number of DDR-enabled resolvers, while yellowish hexagons indicate regions with a high number of DDR-enabled resolvers (log-scale). To address the relationship between the number of DDR-enabled servers and the total number of DNS servers in a region, the density of DDR-enabled servers is shown in Figure 6.3 (b) and Figure 6.3 (d). Greenish hexagons indicate a high density of DDR-enabled servers, red hexagons indicate a low density, and white hexagons indicate regions with no DDR-enabled servers but DNS servers.

Observing the global distribution of IPv4 DDR-enabled servers, we see only a few yellowish hexagons, indicating regions with a high concentration of DDR-enabled servers. These regions include Russia (Moscow), Iran, Bangladesh, Indonesia, and South Africa, whereas in America and Europe, the distribution appears more uniform. In the IPv6 space, three orange hexagons stand out in the areas corresponding to Costa Rica, Bolivia, and Peru.

In terms of raw figures, Asia has the highest number of IPv4 DDR-enabled servers, accounting for 50.05% (152K) of the total DDR-enabled servers in the IPv4 space, followed by Europe (53K, 17.34%), South America (42K, 13.77%), North America (33K, 10.94%), Africa (21K, 7.04%), and Oceania 3K (0.87%). However, when considering the DDR density instead, Africa has the highest



(a) IPv4 DDR-enabled DNS Servers **(b)** IPv4 DDR Density

(c) IPv6 DDR-enabled DNS Servers **(d)** IPv6 DDR Density

Figure 6.3: The left-hand figures show DDR-enabled DNS servers across the globe (log-scale), while the right-hand figures depict the density of DDR-enabled resolvers. White hexagons indicate regions with no DDR-enabled servers but DNS servers. The data corresponds to the latest scans conducted on November 11, 2024, for IPv4 and October 30, 2024, for IPv6.

ratio at 34.46% as 21K out of 62K IPv4 DNS servers are DDR-enabled. South America follows with a density of 14.24% (24K out of 294K). Although Europe has the second-highest number of DDR-enabled servers (53K), its density is only 4.24% (from 1.4M total IPv4 DNS servers), while notably North America lags behind with 2.46% (33K out of 1.4M).

During our scan period, Asia gained the largest number of DDR-enabled resolvers, with additional 7.3K resolvers — a percentage change of 5.10%. Africa showed the highest percentage change at 7.07%, with an increase of 1.4K DDR-enabled servers. Europe also experienced a moderate increase of 1.7K servers, representing a 3.43% change. Conversely, both South America

and North America saw declines, losing 4.2K servers (-9.05%) and 1.6K servers (-4.56%), respectively.

Zooming in at country level, Bangladesh emerges as the leader in IPv4 DDR-enabled servers, hosting around 50K resolvers. As a result, Bangladesh alone accounts for 16.4% of all IPv4 DDR-enabled servers. Indonesia ranks second with 27K servers (8.77%), followed by the United States with 22K (7.06%) and India with 19K (6.2%). Within Europe, Ukraine stands out as the country with the highest number of DDR-enabled servers, accounting for 5.7K (1.87%), with an increasing trend. Ukraine is followed by Germany with 4,2K (1.6%) IPv4 DDR-enabled resolvers. Notably, Bangladesh has one of the highest densities of IPv4 DDR-enabled servers, with 81.89% of all IPv4 DNS servers being DDR-enabled.

In contrast, the IPv6 space exhibits different patterns. South America leads with the highest number of DDR-enabled servers, accounting for a remarkable 47.5% (3.8K) of all IPv6 DDR-enabled servers. North America follows with 1.7K servers (21.71%), and Europe ranks third with 1.5k servers (18.77%). In contrast to the observations in IPv4, Asia contributes only 798 servers (10.07%) to the total number of IPv6 DDR-enabled servers. Oceania contributes with 83 servers (1.05%).

During our measurements, the number of IPv6 DDR-enabled servers stayed more or less the same, with only minor fluctuations. However, the total number of DDR-enabled servers in South America decreased to 3.8K in the final scan on October 30, 2024, from 4.9K in the first scan (-1.1K). Yet, South America has by far the best ratio of IPv6 DDR densities among all continents with 29.32% of all 12.8K IPv6 DNS servers being DDR-enabled. Within South America, Bolivia stands out, hosting 1,8K IPv6 DDR-enabled resolvers — the highest number among all countries contributing with 22.31% to the total number of IPv6 DDR-enabled resolvers. Brazil, Peru and Costa Rica follow with 853 (12.74%), 822 (10.37%), and 693 (8.75%) IPv6 DDR-enabled servers, respectively. Among these countries, Bolivia has an astonishing DDR density of 98.11%, with almost all of its 1.8K IPv6 DNS servers being DDR-enabled, followed by Costa Rica (93.15%) and Peru (90.92%).

In contrast, North America and Europe have low IPv6 DDR densities of 3.58% and 1.28%, respectively. Africa has a DDR density of 7.63%, Asia of 1.42%, and Oceania of 2%. The U.S. leads within North America with 837 servers and a density of 1.97%, while the Czech Republic leads Europe with 283 servers and a density of 5.48%, and India leads Asia with 129 servers and a density of 1.63%. Detailed figures are in Section B.2.

Observation: DDR Adoption across Continents and Countries.

The global distribution of DDR-enabled DNS servers shows notable regional differences. Asia hosts the largest number of IPv4 DDR-enabled servers (152K, 50.05%), while Africa leads in DDR density (34.46%). Bangladesh has the highest DDR density for IPv4 with 81.89%. South America dominates the number of IPv6 DDR-enabled servers, with Bolivia hosting the majority of servers achieving the highest DDR density (98.11%). Europe, despite having the most IPv6 DNS servers overall, shows low DDR-enabled proportions for IPv4 (3.47%) and IPv6 (0.95%).

6.1.2 Adoption Rates across ASes and Network Categories

Similar to our analysis of DNS resolvers in Section 5.3, we classify DDR-enabled servers by network categories in Table 6.1. These tables include the number of DNS and DDR-enabled resolvers, along with the number of unique ASes they span. Since the raw number of DDR-enabled servers provides only limited insights into DDR adoption, especially in providing any trends, we also calculate the density of DDR-enabled servers (ρ_{DDR}) and the average percentage change in this density over all measurements conducted ($\overline{\rho}_{DDR}$).

To offer additional context about DDR adoption across ASes, we calculate the density of ASes hosting DDR-enabled servers (ρ_{AS}, also in percent), defined as the ratio of unique ASes with DDR-enabled servers to the total number of unique ASes with DNS servers. Additionally, we calculate the average percentage change in AS density ($\overline{\rho}_{AS}$) and the standard deviations of both densities (σ_{DDR} for server density and σ_{AS} for AS density). These standard deviations serve as a measure of variability, indicating how much the density of DDR-eanbled servers (σ_{DDR}) and the density of ASes hosting DDR-enabled servers (σ_{AS}) fluctuate over time. A higher standard deviation suggests greater inconsistency or variability in the density values across the measurement period, while a lower standard deviation indicates more stability. By analyzing these fluctuations, we gain insights into the temporal dynamics of DDR adoption, helping us to understand whether certain network categories or ASes experience sporadic adoption patterns or maintain consistent growth over time. We do not differentiate between RRs and non-RR in this analysis because a notable proportion of DDR-enabled servers are RRs (see Section 6.1). For visibility and clarity, we therefore refrain from splitting the analysis.

The approximately 309K IPv4 DDR-enabled DNS servers discovered span

Table 6.1: Classification of DDR-enabled servers by network category. The figures represent average values across all measurements, including the number of DDR and DNS servers, DDR density (ρ_{DDR}), average percentage change in DDR density ($\overline{\rho}_{DDR}$) and its standard deviation (σ_{DDR}), AS density (ρ_{AS}), and the average percentage change in AS density ($\overline{\rho}_{AS}$) with its standard deviation (σ_{AS}).

a IPv4 DDR-enabled server categories.

Category	# DDR Servers	# DDR ASes	# DNS Servers	# DNS ASes	ρ_{DDR} (%)	$\overline{\rho}_{DDR}$ (%)	σ_{DDR}	ρ_{AS} (%)	$\overline{\rho}_{AS}$ (%)	σ_{AS}
Netw. Services	181,186 (58.60%)	6,334 (41.25%)	1,315,828 (32.30%)	11,478 (25.82%)	13.77	0.04	2.10	55.19	-0.16	0.69
unknown	113,361 (36.66%)	8,340 (54.31%)	1,177,530 (28.91%)	29,955 (67.39%)	9.63	-0.11	2.70	27.84	-0.02	1.11
Content	10,144 (3.28%)	371 (2.41%)	1,399,216 (34.35%)	1,396 (3.14%)	0.73	0.72	4.33	26.56	-0.12	2.74
Enterprise	2,608 (0.84%)	166 (1.08%)	73,653 (1.81%)	766 (1.72%)	3.63	1.57	17.35	21.70	-0.30	3.28
Ed./Research	1,646 (0.53%)	106 (0.69%)	103,307 (2.54%)	479 (1.08%)	2.49	-4.89	17.91	22.07	-0.30	2.15
Non-Profit	189 (0.06%)	25 (0.16%)	3,309 (0.08%)	283 (0.64%)	5.68	0.58	29.69	8.74	-0.77	8.25
Route Server	48 (0.02%)	5 (0.04%)	224 (0.01%)	30 (0.07%)	19.74	-2.84	16.99	18.10	-0.99	11.26
Government	18 (0.01%)	8 (0.05%)	499 (0.01%)	62 (0.14%)	3.63	2.81	20.71	13.43	2.63	17.70
Total	309,200	15,355	4,073,567	44,448	7.59	-0.14	13.97	34.55	-0.00	5.90

b IPv6 DDR-enabled server categories.

Category	# DDR Servers	# DDR ASes	# DNS Servers	# DNS ASes	ρ_{DDR} (%)	$\overline{\rho}_{DDR}$ (%)	σ_{DDR}	ρ_{AS} (%)	$\overline{\rho}_{AS}$ (%)	σ_{AS}
Netw. Services	5,881 (77.37%)	656 (56.77%)	91,514 (31.94%)	3,355 (42.99%)	6.66	5.59	21.42	19.54	-0.80	4.41
unknown	1,030 (13.55%)	368 (31.89%)	79,529 (27.76%)	3,042 (38.97%)	1.31	3.53	8.61	12.12	0.39	5.63
Content	531 (6.99%)	72 (6.19%)	106,137 (37.04%)	667 (8.55%)	0.50	0.22	3.46	10.70	-2.60	8.65
Ed./Research	75 (0.98%)	29 (2.50%)	5,293 (1.85%)	346 (4.43%)	1.41	-2.16	13.32	8.34	-2.09	7.29
Enterprise	56 (0.74%)	19 (1.62%)	2,500 (0.87%)	205 (2.62%)	2.25	0.64	9.41	9.10	-3.82	5.50
Non-Profit	28 (0.37%)	12 (1.04%)	1,545 (0.54%)	190 (2.44%)	1.83	1.03	11.95	6.30	-0.08	12.73
Total	7,601	1,156	286,517	7,805	2.65	2.82	11.36	14.81	-1.50	7.37

across 15K ASes on average (see Table 6.1a). While the DDR density ρ_{DDR} is relatively low at 7.59%, the AS density ρ_{AS} is much higher at 34.55%. This indicates that although the overall density of DDR servers is not very high, these servers are distributed across a relatively large number of ASes. We also observe that the DDR density fluctuates notably over the measurement period, with a standard deviation σ_{DDR} of 13.97. In contrast, the fluctuation in AS density is much smaller, with a standard deviation σ_{AS} of 5.9. This suggests that the density of DDR servers tends to change more over time compared to the distribution of DDR servers across ASes. Interestingly, the average change in AS density $\overline{\rho}_{AS}$ is effectively zero (0.00%), indicating that the number of ASes hosting DDR servers remains stable over time. Despite an overall increase in the number of DDR servers during the measurement period, the DDR density shows a negative trend, with an average change $\overline{\rho}_{DDR}$ of -0.14%. This highlights that while the number of DDR servers grows, it does not keep pace with the growth in the total number of DNS servers, leading to a decline in density.

In the IPv6 space, 7.6K DDR-enabled resolvers span 1.2K ASes on average, resulting in a DDR density ρ_{DDR} of 2.65% and an AS density ρ_{AS} of 14.81%. Both densities are more than half of those observed in the IPv4 space. Thus, in general, there are less IPv6 DDR-enabled servers on the total number of DNS servers and the DDR-enabled servers are more concentrated in fewer ASes compared to IPv4. While the standard deviation of the percentage change in DDR density is smaller for IPv6 at 11.36 compared to IPv4, the standard deviation of AS density is higher, at 7.37, indicating greater fluctuation in the distribution of ASes hosting DDR-enabled servers. We highlight that the average percentage change in DDR density $\overline{\rho}_{DDR}$ shows an increasing trend at 2.82%, whereas the average percentage change in AS density $\overline{\rho}_{AS}$ is negative at -1.5%. This suggests that DDR-enabled servers increasingly concentrate on fewer ASes over time. We believe this as a potential sign of DNS centralization in the IPv6 space.

Focusing on network categories, we observe that most IPv4 DDR-enabled servers (181K, 58.60%) belong to the Network Services category, which aligns with our expectations, as DDR is the target of these servers. In contrast, most DNS servers overall are categorized as Content (1.4M, 34.35%). The highest DDR densities ρ_{DDR} appear in the Network Services category (13.77%) and the Route Server category (19.74%), while the lowest densities are found in the Content (0.73%) and Educational/Research (2.49%) categories. The Educational/Research category shows the largest negative trend in density, with an average percentage change $\overline{\rho}_{DDR}$ of -4.89%. However, these values require cautious interpretation due to a relatively high standard deviation σ_{DDR} of 17.91, indicating considerable variability. In contrast, the target group of DDR (Network Services) exhibits a slight positive trend in DDR density growth ($\overline{\rho}_{DDR} = 0.04\%$) and a comparatively low standard deviation of σ_{DDR} with 2.1, suggesting the highest stability. Turning focus on the development of AS density in the IPv4 space, all categories except Government show a negative trend, meaning that DDR-enabled servers are increasingly concentrated in fewer ASes. The Government category, however, experiences a positive trend, accompanied by one of the highest standard deviation across all categories (20.71), indicating substantial fluctuation.

The distribution of DDR-enabled servers across different categories in IPv6 is similar to that in IPv4. However, Network Services accounts for a much larger proportion of DDR-enabled servers, with 5.9K servers (77.37%). Categories such as Government and Route Server contain no DDR-enabled servers at all. The highest DDR density ρ_{DDR} is also found in the Network Services category, at 6.66%. While this is about half the density observed in IPv4, the Network

Table 6.2: Top 10 ASes by number of IPv4 DDR-enabled servers, including their country and network type. The figures depict the results from the latest scan from 11.11.2024.

	AS No.	AS Organization	Country	Network Type	# DDR Servers	ρ_{DDR} (%)
1	17488	Hathway IP Over Cable Internet	India	Network Services	9,495 (3.12%)	18.73
2	7713	PT Telekomunikasi Indonesia	Indonesia	Network Services	8,728 (2.87%)	73.55
3	36994	Vodacom-VB	South Africa	Network Services	7,465 (2.45%)	98.00
4	58224	Iran Telecommunication Company PJS	Moldova	unknown	4,631 (1.52%)	71.02
5	22773	ASN-CXA-ALL-CCI-22773-RDC	United States	Network Services	2,586 (0.85%)	67.84
6	16637	MTN Business Solutions	Botswana	Network Services	2,375 (0.78%)	87.77
7	134540	Tata Teleservices Maharashtra Ltd	India	Network Services	2,324 (0.76%)	61.38
8	9299	Philippine Long Dist. Telephone Comp.	Philippines	Network Services	2,231 (0.73%)	53.26
9	8151	UNINET	Mexico	Network Services	2,125 (0.70%)	21.35
10	4134	Chinanet	China	Network Services	2,102 (0.69%)	3.20

Services category shows a relatively strong positive trend in density growth, with an average percentage change $\overline{\rho}_{DDR}$ of 5.59%. However, this trend should be interpreted cautiously due to the high standard deviation σ_{DDR} of 21.42, the highest among all categories. Similar to IPv4, the Research/Education category experiences a negative trend in DDR density, with $\overline{\rho}_{DDR}$ equals -2.16%. Notably, this is the only category in IPv6 to exhibit a negative trend in DDR density. The highest AS density ρ_{AS} occurs in the Network Services category (19.54%), although it is much lower than in IPv4. This indicates that, comparatively, more DDR servers are concentrated in fewer ASes in the Network Services category. Overall, the average percentage change in AS density $\overline{\rho}_{AS}$ shows a negative trend (-1.5%), suggesting that IPv6 DDR-enabled servers continue to consolidate within fewer ASes. Again, the trends we depict here should be interpreted with caution, as the standard deviation across all categories of AS density σ_{AS} is 7.37, which is higher than that of the IPv4 family. We believe this variability is partly due to fluctuations in the dataset from the *IPv6 Hitlist Service*.

In Table 6.2 and Table 6.3, we display the top 10 ASes hosting the most DDR-enabled servers. Notably, most of the ASes in these lists are categorized as Network Services.

For IPv4, AS 17488 (*Hathway IP Over Cable*) from India tops the list with 9,495 DDR-enabled servers. This AS was already ranked 10th in the list of ASes with the most DNS servers (see Table 5.3a). As a result, its DDR density ρ_{DDR} is relatively low at 18.73%. At the bottom of the list, AS *4134* (*Chinanet*) is also among the top 10 ASes with the most DNS servers but has the lowest DDR density of just 3.20%. In contrast, AS *36994* (*Vodacom-VB*) exhibits the

Table 6.3: Top 10 ASes by number of IPv6 DDR-enabled servers, including their country and network type. The figures depict the results from the latest scan from the 30.09.2024.

	AS No.	AS Organization	Country	Network Type	# DDR Servers	ρ_{DDR} (%)
1	27839	Comteco Ltda	Bolivia	Network Services	1,768 (22.31%)	99.33
2	267831	TELECABLE	Peru	Network Services	791 (9.98%)	100.00
3	263762	Coopeguanacaste	Costa Rica	Network Services	688 (8.68%)	98.43
4	266423	Connectvy	Brazil	Network Services	358 (4.52%)	84.63
5	6939	Hurricane Electric	Portugal	Network Services	296 (3.74%)	11.53
6	203936	Iberwix Telecom S.l.	Spain	Network Services	177 (2.23%)	100.00
7	1929	UMASSNET-NET	United States	unknown	144 (1.82%)	100.00
8	201565	Etruria Wi-fi S.r.l.	Italy	unknown	135 (1.70%)	100.00
9	16276	OVH SAS	Hong Kong	Content	95 (1.20%)	0.53
10	44489	STARNET, s.r.o.	Czechia	Network Services	95 (1.20%)	6.41

highest DDR density at 98%, meaning almost all DNS servers in this AS offer DDR configurations.

The IPv6 top 10 ASes show even higher DDR densities. Four ASes in this list achieve a DDR density of 100%. Among these, AS *16276* (*OVH SAS*), which is also listed among the top 10 ASes with the most DNS servers, hosts only 95 DDR-enabled servers and has the lowest DDR density in this list at just 0.53%. It is worth highlighting that the top three ASes in the IPv6 list are located in South America, which we identified them already on the world map in Section 6.1.1. Among these, AS *27839* (*Comteco Ltda*) stands out as it hosts 22.31% of all DDR-enabled servers in the IPv6 family and boasts a DDR density of 99.33%. All three South American ASes show relatively high DDR densities, ranging from 98.43% to 100%.

Observation: DDR Adoption and DDR Density.

DDR-enabled servers are mainly hosted in the Network Services category, comprising 58.60% of IPv4 and 77.37% of IPv6 servers. IPv4 has a low DDR density of 7.59% with a slight decline (-0.14%), while IPv6 shows a smaller density of 2.65% but a positive trend (2.82%). IPv4 AS density remains stable, while IPv6 servers increasingly centralize within fewer ASes. The top IPv6 ASes, particularly in South America, reach nearly 100% DDR density, highlighting their critical role in DDR's ecosystem.

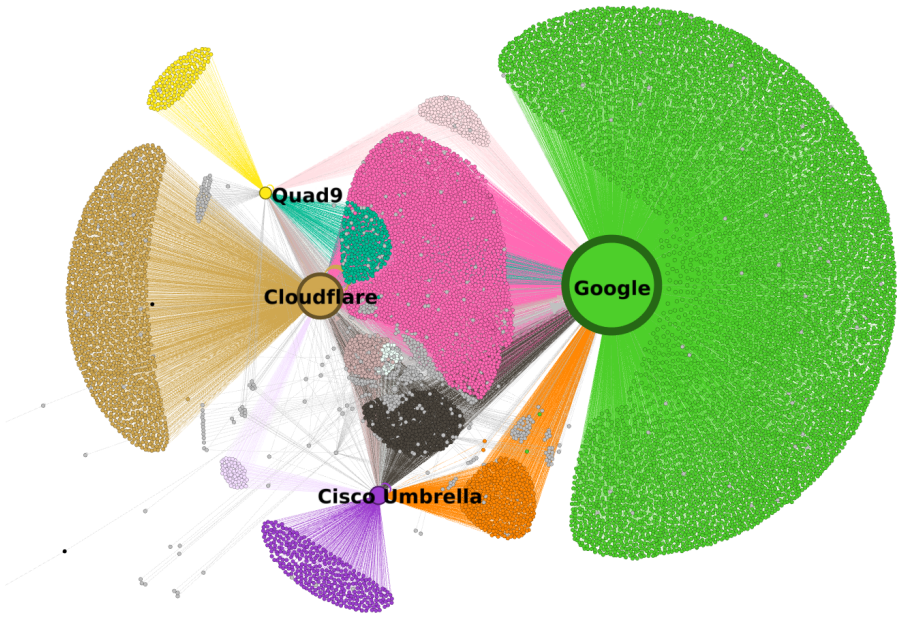
6.2 Configuration Dynamics and their Influence on DNS Centralization

In this section, we examine the configuration dynamics of DDR-enabled resolvers. We begin by analyzing the diversity and prevalence of real-world DDR configurations, followed by a detailed exploration of delegation trends across network categories. Through visualizations and data-driven insights, we highlight how configurations align with broader trends in DNS infrastructure. Additionally, we discuss compliance issues regarding the DDR standard observed during our measurements, identifying patterns and potential implications for the deployment of DoE protocols. This analysis provides a foundation for understanding how DDR contributes to the centralization of DNS infrastructure.

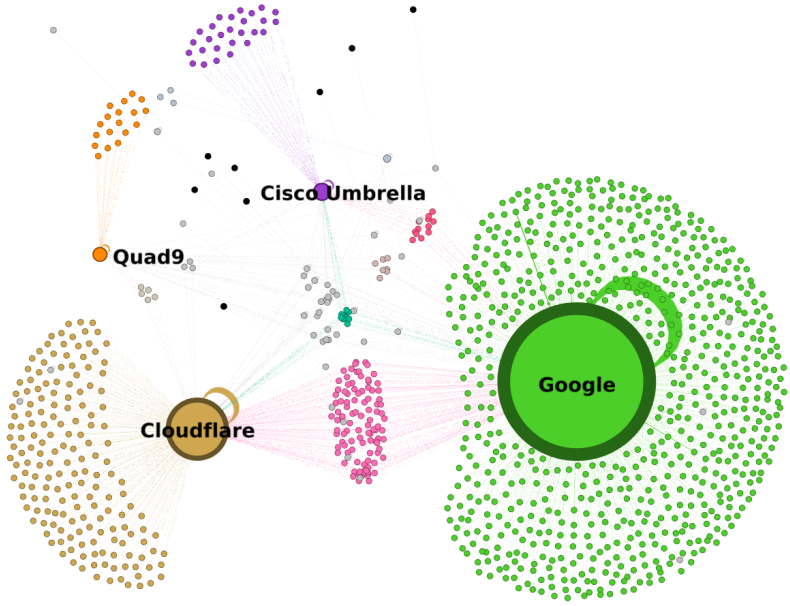
6.2.1 Diversity and Prevalence of DDR Configurations

We analyze the payloads (i.e., DDR configurations) of the 309K IPv4 DDR-enabled resolvers (7.6K IPv6) to investigate their real-world configurations. To identify the most common configurations, we hash and group them by their hash values. In total, we identify 3,378 unique configurations among IPv4 DDR-enabled resolvers and 263 among IPv6 ones, indicating low configuration diversity compared to the overall number of DDR-enabled resolvers. The most common configuration is those of Google's cloud DNS service (`dns.google.`), used by an average of 79.3% of IPv4 DDR-enabled resolvers and 82.54% of IPv6 ones, redirecting to the DoE resolvers of Google. The configuration of *Cloudflare* follows with 12.17% (9.73% IPv6). The third and fourth most deployed configurations are from *Cisco Umbrella (OpenDNS)*. While the third most deployed configuration occurs in 4.46% (3.33% IPv6) of cases and delegates to `dns.opendns.com` and `dns.umbrella.com`, the fourth most used configuration delegates to `familyshield.opendns.com` in 0.76% of the cases (none in IPv6). However, both DoE resolver belong to *Cisco Umbrella*, as the latter one just provides additional features like content filtering. The fifth most used configuration is from Quad9, delegating to their servers in 0.72% (0.99% IPv6). These five configurations collectively represent over 97% of all DDR-enabled resolvers' configurations, demonstrating a high degree of consolidation of configurations within the DDR ecosystem. The details of these five most advertised configurations are provided in Section B.3.

Assuming clients use DDR to automatically upgrade to DoE protocols, the current in-the-wild configurations suggest that DDR-enabled resolvers delegate



(a) IPv4 DDR Resolver Delegation Graph



(b) IPv6 DDR Resolver Delegation Graph

Figure 6.4: Nodes represent ASes, while edges illustrate redirections to either the same or a different AS. Colors group DDR-enabled resolvers that share the same configuration in terms of AS target combination. The size of each node reflects the number of incoming edges, representing how many other ASes delegate their clients to that specific AS.

their resolving activities away from their own AS to other ASes, primarily those of major cloud DNS providers like *Google*, *Cloudflare*, *Cisco*, or *Quad9*. To better understand this delegation, we visualize redirections as a graph network for IPv4 and IPv6 in Figure 6.4. Nodes represent ASes, while edges illustrate redirections to either the same or different ASes. Colors group ASes with identical AS-target combinations. For example, if all DDR-enabled resolvers within the AS X and AS Y redirect only to *Cloudflare* and *Google*, they share the same color. Only the top eight AS-target combinations are color-coded. The node size reflects the number of incoming edges, indicating how many other ASes redirect their clients to a given AS.

The graph reveals the dominance of *Google*, *Cloudflare*, *Cisco*, and *Quad9*, but also highlights variation in DDR configurations within the same AS. For instance, pink-colored nodes in both figures demonstrate that DDR-enabled resolvers in the same AS may not share identical configurations. This is due to the fact that the top five most common DDR configurations do not include any configurations jointly offered by *Cloudflare* and *Google* but at the same time, the pink-colored nodes make up to 17%. This suggests that there must be DDR-enabled resolvers within the same AS that are configured with either one or the other configuration.

On a closer look, the graph also shows only a few nodes with self-loops, representing DDR-enabled resolvers that redirect traffic within the same AS without delegating externally. These include the major cloud DNS providers like *Google* and *Cloudflare*, as they offer a DDR configuration targeting themselves. In the IPv4 space, 0.23% (48) of ASes hosting DDR-enabled resolvers have at least one resolver that does not delegate clients to another AS. For IPv6, this figure is slightly higher at 1.42% (27). However, considering the absolute number of DDR-enabled resolvers, only 0.69% (8K) in IPv4 and 1.60% (327) in IPv6 refrain from redirecting their requests to other ASes.

We emphasize that the distributions of configurations observed during our measurement period have shown only negligible changes. For instance, we cannot confirm a shift away from configurations relying on DNS cloud providers towards independently crafted setups delegating to DoE servers within the same organization, as intended by the DDR standard [124].

6.2.2 Delegation Trends across Network Categories

To better understand how DDR-enabled resolvers are configured across different network categories, we analyze the distribution of configurations concerning the most offered DoE target delegations (alternative domains). The results,

depicted in Table 6.4, present the alternative domains of IPv4 and IPv6 DDR-enabled resolvers, sorted by the total number of occurrences. To gain further insights, we also provide the relative proportions of each configuration within its respective network category. We only consider the latest available measurement results for this analysis, as DDR configurations exhibited minimal changes over our measurement period. Note, that different to Section 6.2.1, where we focus on the most common configurations, we now analyze the distribution of alternative domains within each DDR configuration, as each configuration can have multiple alternative domains (SVCB ResRs) advertised (see Section 2.4.2).

We observe 1,668 unique alternative domains across 902,179 advertisements in configurations of IPv4 DDR-enabled resolvers. The most common alternative domain is `dns.google.`, advertised approximately 700K times (77.63%), making *Google* the most advertised alternative domain across all network categories. The second most common domain is `one.one.one.one.` from *Cloudflare*, observed 120,420 times (13.35%), followed by *Cisco*'s `.opendns.com` and `.umbrella.com`. In general, these findings align with the AS delegation observations in Section 6.2.

Examining the delegations to *Google* across network categories reveals that resolvers classified as Route Servers delegate the most of their clients to *Google* (82%), while in the case of Educational/Research resolvers it is only 39.90%. Resolvers classified as Network Services account for the largest number of delegations to *Google*'s DNS, with 402,278 delegations, as they host the most DDR-enabled resolvers (see Section 6.1.2). This observation is particularly relevant because Network Services resolvers include ISP resolvers, meaning that their clients, i.e., oftentimes residential broadband consumers, are frequently redirected to *Google* when automatically upgrading to DoE using DDR. Among all unique IPv4 DDR-enabled resolvers classified as Network Services, only 0.71% offer at least one entry in their DDR configuration pointing to a DoE resolver within the same AS. For IPv6, this figure is slightly higher at 1.21%.

In IPv6 DDR-enabled resolvers, we observe 152 unique alternative domains across 23,822 advertisements. Two network categories — Government and Route Server — are absent, as no IPv6 DDR-enabled resolvers are identified in these categories. IPv6 resolvers delegate to *Google*'s DNS more frequently than IPv4 resolvers, with 83.12% (19.8K) entries referring clients to *Google*. *Cloudflare* remains the second most common, used by 2.1K (9.17%) IPv6 resolvers, followed by *Cisco*'s `dns.opendns.com` and `dns.umbrella.com`.

Similar to the alternative domains in IPv4, resolvers in the Network Services category show a higher tendency to delegate to *Google* in IPv6 (87.76%) but redirect to *Cloudflare* less frequently (7.58% versus 14.29% in IPv4).

Table 6.4: Top 10 advertised alternative domains of IPv4 and IPv6 DDR-enabled resolvers categorized by their respective network types. Note that IPv6 DDR-enabled resolvers are distributed across fewer network categories, resulting in fewer columns.

a IPv4 DDR-enabled resolvers' most advertised alternative domains (November 11, 2024).

Alternative Domain	*Total*	Content	Ed. Research	Enterprise	Govern-ment	Network Services	Non-Profit	Route Server	Unknown
dns.google.	700,376 77.63%	19,912 64.94%	1,549 39.90%	5,257 75.32%	51 62.96%	402,278 75.89%	213 78.31%	96 82.05%	271,020 82.11%
one.one.one.one.	120,420 13.35%	6,198 20.21%	396 10.20%	1,044 14.96%	15 18.52%	75,741 14.29%	18 6.62%	21 17.95%	36,987 11.21%
dns.opendns.com.	16,221 1.80%	752 2.45%	124 3.19%	122 1.75%		10,656 2.01%	6 2.21%		4,561 1.38%
dns.umbrella.com.	16,220 1.80%	752 2.45%	124 3.19%	122 1.75%		10,655 2.01%	6 2.21%		4,561 1.38%
doh.umbrella.com.	8,006 0.89%	374 1.22%	62 1.60%	61 0.87%		5,260 0.99%	3 1.10%		2,246 0.68%
doh.opendns.com.	8,005 0.89%	374 1.22%	62 1.60%	61 0.87%		5,260 0.99%	3 1.10%		2,245 0.68%
dns.quad9.net.	6,924 0.77%	560 1.83%	24 0.62%	68 0.97%		4,310 0.81%	2 0.74%		1,960 0.59%
familyshield.opendns.com.	5,068 0.56%	28 0.09%	958 24.68%	108 1.55%	2 2.47%	3,110 0.59%			862 0.26%
family.cloudflare-dns.com.	3,699 0.41%	117 0.38%	84 2.16%	9 0.13%		2,733 0.52%	15 5.51%		741 0.22%
security.cloudflare-dns.com.	2,706 0.30%	93 0.30%	6 0.15%	15 0.21%	12 14.81%	2,151 0.41%			429 0.13%

b IPv6 DDR-enabled resolvers' most advertised alternative domains (October 30, 2024).

Alternative Domain	*Total*	Content	Ed. Research	Enterprise	Network Services	Non-Profit	Unknown
dns.google.	19,800 83.12%	558 37.20%	150 66.96%	102 58.29%	16,221 87.76%	45 56.96%	2,724 81.05%
one.one.one.one.	2,184 9.17%	336 22.40%	24 10.71%	33 18.86%	1,401 7.58%	15 18.99%	375 11.16%
dns.opendns.com.	298 1.25%	130 8.67%	2 0.89%	4 2.29%	112 0.61%		50 1.49%
dns.umbrella.com.	298 1.25%	130 8.67%	2 0.89%	4 2.29%	112 0.61%		50 1.49%
dns.quad9.net.	212 0.89%	44 2.93%	2 0.89%	6 3.43%	138 0.75%	4 5.06%	18 0.54%
doh.opendns.com.	146 0.61%	65 4.33%		2 1.14%	54 0.29%		25 0.74%
doh.umbrella.com.	146 0.61%	65 4.33%		2 1.14%	54 0.29%		25 0.74%
family.cloudflare-dns.com.	63 0.26%		27 12.05%		21 0.11%	12 15.19%	3 0.09%
dns.adguard-dns.com.	55 0.23%				45 0.24%		10 0.30%
DOH.COX.NET.	53 0.22%				53 0.29%		

Additionally, the domain `doh.cox.net`, associated with Cox, a major U.S. broadband ISP, is notable, although its overall share remains low (0.22%).

As a result, these findings highlight the dominance of major DNS cloud providers, particularly *Google*. We can conclude that the current deployment of DDR in-the-wild does not contribute to creating a diversified DoE landscape but rather supports the centralization of DNS infrastructure.

6.2.3 Reinforcing DNS Centralization through DDR

As previous studies have shown (see Section 3.5), the recursive resolver market in DNS, and in particular the use of DoE protocols, exhibits signs of DNS centralization [37, 74, 101]. The primary issue with DoE protocols has been the limited options for automatically discovering DoE resolvers and their configurations [96]. This challenge is one reason why the IETF standardized DDR, enabling clients to automatically discover encrypted resolvers and their configurations such that an automatic upgrade to encryption protocols is possible. However, the DDR standard states: "[DDR] mechanisms are designed to be limited to cases where Unencrypted DNS Resolvers and their Designated Resolvers are operated by the same entity or cooperating entities" [124].

Yet, our measurements show that 97% of all DDR-enabled resolvers fully delegate to the four major cloud DNS providers. This widespread configuration may stem from the major providers' ability to ensure high availability and performance through globally distributed infrastructure [37], their support for DoE protocols enabling encrypted communication without additional deployment efforts, or simply their reputation as trusted entities in the DNS ecosystem. Nevertheless, it may simply be convenience, as the configurations of major providers appear to be replicated in most DDR-enabled resolvers, suggesting they were directly copied.

However, this heavy reliance on cloud providers raises concerns about DNS centralization, where a few entities control a large portion of the recursive resolution process. Such concentration poses risks to user privacy, security, and the broader decentralization of internet governance, as the resolution process is critical to the internet's integrity and honest functionality.

6.2.4 Non-Compliant Configurations

During our measurement period, an average of 1.05% (9.7K) of all IPv4 DDR-enabled resolvers offered non-compliant SVCB RRs according to the standards [124, 150, 151]. In comparison, IPv6 resolvers showed a slightly

Observation: DDR and DNS Centralization.

Our analysis reveals significant DDR configuration consolidation, with over 97% delegating to only four major providers: *Google*, *Cloudflare*, *Cisco*, and *Quad9*. *Google*'s DDR configuration alone is deployed to 79.3% of IPv4 and 82.54% of IPv6 DDR-enabled resolvers, highlighting the ecosystem's reliance on a few dominant players. In contrast, only 0.69% of IPv4 and 1.60% of IPv6 DDR-enabled resolvers delegate clients within their own AS. These findings raise concerns about the privacy and governance implications of increasing DNS centralization from the perspective of DDR.

lower error rate at 0.29% (67). Among these, we observe that in 98.23% of the non-compliant IPv4 DDR-enabled resolvers, mandatory keys such as `priority` are missing (84.11% in IPv6). Interestingly, 1.65% of the non-compliant IPv4 DDR-enabled resolvers advertise the root domain ("·") as an alternative domain. In IPv6, this figure is considerably higher at 15.89%. While the use of the root domain ("·") is originally defined in the SVCB RR standard [150] to refer to the queried host, the DDR standard explicitly disallows this usage in DDR configurations [124]. Additionally, 0.053% (128) of the non-compliant IPv4 DDR-enabled resolvers include DoE targets and priorities but fail to specify the DoE protocols to be used. In contrast, every configuration offered by IPv6 DDR-enabled resolvers specify the DoE protocols.

We also detected a drastic spike in non-compliant DDR configurations on September 5, 2024. The number of non-compliant configurations surged from an average of 9.7K to 24K, representing an increase of 147%. This phenomenon was not observed among IPv6 DDR-enabled resolvers in this time frame. In 99% of these faulty configurations, mandatory parameters such as `priority` were missing. Analyzing the origin of these non-compliant SVCB RRs, we found that resolvers operated by *Korea Telecom* (*AS 4766*) were primarily responsible for this surge, with their non-compliant resolvers increasing from 52 (as of August 29, 2024) to 7.7K — an increase of 14,884%, accounting for almost 85% of all DDR-enabled resolvers operated by *Korea Telecom*. Other organizations, such as *China Unicom Backbone* (*AS 4837*) and *Chinanet* (*AS 4134*), also contributed to this rise, with increases of 351% (+1,887) and 479% (+402), respectively. Although we observed non-compliant configurations globally across various organizations, the growth was not as pronounced as

with the aforementioned operators. By the next measurement on September 11, 2024, the number of non-compliant resolvers had returned to approximately 9K. The reasons behind this temporary spike remain unclear.

In addition to the SVCB keys specified by the *IANA* [77], we observe that 1.1K IPv4 DDR-enabled resolvers and 19 IPv6 resolvers include at least one SVCB RR with the key `mandatory=port`. Although this key is defined in the SVCB standard [150], it has no application in DDR and is ignored by clients [124]. In one case, we identify the SVCB key `key42`, which is used to specify the *dohpath*. However, since this key is not defined in the SVCB standard and is ignored by clients [124], the entire configuration is deemed faulty because the *dohpath*, incorrectly linked to the `key42`, is effectively missing. In addition, two IPv4 and one IPv6 DDR-enabled resolvers fail to advertise any *dohpath*. However, according to the standard [151], specifying this parameter is mandatory, as there is only a suggestion for its value (`/dns-query{?dns}`), but it is not a default value.

Furthermore, two resolvers advertise a `priority` lower than 1, which is not permitted [124].

While the proportion of non-compliant configurations is relatively small compared to compliant DDR configurations, such errors can prevent clients from automatically upgrading to DoE protocols. This inability may force clients to continue using unencrypted DNS (Do53), potentially compromising user privacy.

Observation: Non-Compliant DDR Configurations.

1.05% of IPv4 and 0.29% of IPv6 DDR-enabled resolvers provide non-compliant SVCB RRs, mostly due to missing mandatory keys like `priority`. On September 5, 2024, we detect a 14,884% spike in non-compliance within *Korea Telecom*'s AS, accounting for 85% of the surge, with minor contributions from other operators. Additional issues include undefined SVCB keys like `key42` and improperly specified *dohpath* parameters. These errors, though rare, hinder client upgrades to DoE protocols, risking user privacy.

6.3 Distribution of DoE Protocols in DDR Configurations

This section explores the distribution of DoE protocols in DDR configurations across various network types and highlights adoption trends for DoE protocols. We analyze protocol preferences, prioritization, and deviations from standard configurations, including non-standard ports and custom *dohpath* settings. By comparing IPv4 and IPv6 configurations and excluding large providers in certain analyses, we focus on the low adoption of DoQ and the persistence of legacy protocols like DoH/1.1. Finally, we investigate DDR adoption across DoE resolvers.

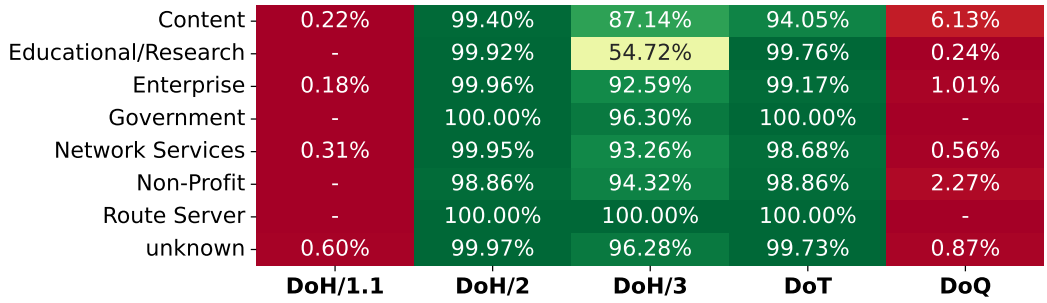
6.3.1 Distribution across Network Types

Shifting the perspective solely to the DoE protocols observed in DDR configurations, we can analyze which protocols are preferred by DDR-enabled resolvers in each network category. The heatmaps depicted in Figure 6.5 illustrate the distribution of protocols for IPv4 and IPv6 DDR-enabled resolvers. The color scale indicates the likelihood of a specific protocol being advertised within a particular category, with greener cells representing higher likelihoods and redder cells indicating lower likelihoods.

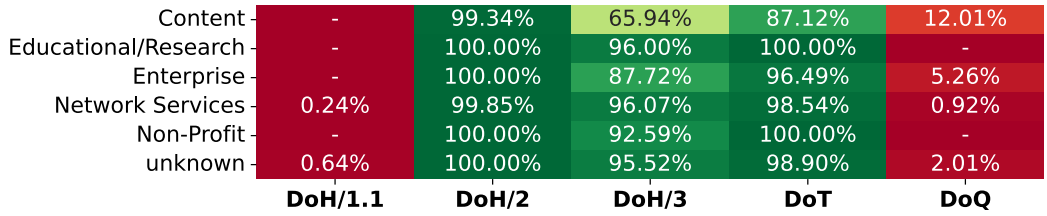
It is evident that IPv4 and IPv6 DDR-enabled resolvers most frequently advertise DoT, DoH/2, and DoH/3 across all network categories. However, IPv4 DDR-enabled resolvers in the Educational/Research category advertise DoH/3 less frequently compared to other categories (54.72%). In the case of IPv6 DDR-enabled resolvers, the Content category exhibits a relatively low adoption rate of DoH/3 with 65.94%.

Overall, DoQ is offered in comparatively few cases, despite the relatively frequent offering of DoH/3, which is also based on the QUIC protocol but incorporates an additional HTTP layer (see Section 2.3). The low likelihood of DoQ adoption is primarily attributable to its lack of support from major cloud DNS providers such as *Google* and *Cisco*. This is notable given that QUIC was initially invented at *Google* [81], and contributed to the DoQ standard by *Cisco* employees [72]. Throughout our measurement period, DoQ was offered relatively consistently, with configurations ranging between 2,406 and 3,291, showing neither an increasing nor decreasing trend.

The legacy protocol HTTP/1.1 is surprisingly still advertised. When advertised by a DDR-enabled resolver, it is associated with a >88% likelihood



(a) IPv4 DDR protocol distribution across network types (November 11, 2024)

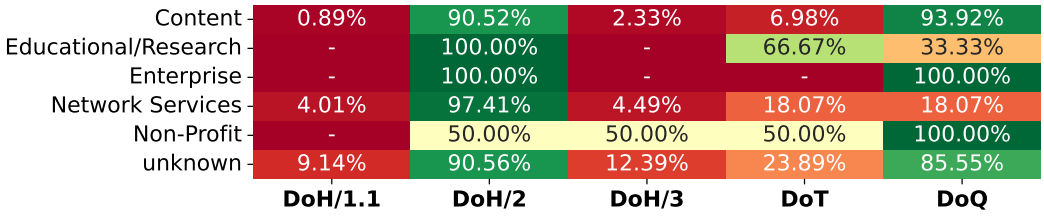


(b) IPv6 DDR protocol distribution across network types (October 30, 2024)

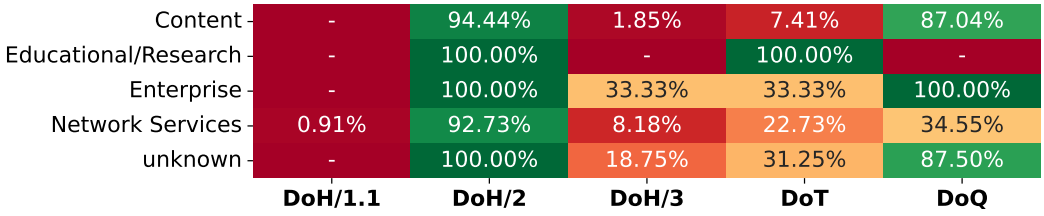
Figure 6.5: Heatmaps depicting the distribution of advertised encryption protocols across network types for IPv4 and IPv6 DDR-enabled resolvers. Green cells indicate a higher probability of protocol advertisement, while red cells indicate a lower probability.

of delegation to *AdGuard*'s DoE resolvers. We observed no notable decline or increase in these configurations throughout our measurement period, which consistently ranged between 1,218 and 1,758 configurations offering the DoH/1.1 protocol.

Considering the high delegations to major cloud DNS providers, it is unsurprising that protocols like DoQ exhibit limited adoption among DDR-enabled resolvers. To investigate further, we attach similar heatmaps to those depicted in Figure 6.5, but without considering the five most advertised cloud DNS providers — *Google*, *Cloudflare*, *Cisco*, *Quad9*, and *AdGuard*. These revised heatmaps, presented in Figure 6.6, reveal a noteworthy shift: DoQ is now advertised more frequently than DoH/3 across all network categories. For instance, Non-Profit, Content, and Enterprise DDR-enabled resolvers advertise DoQ in >87% of cases, while DoH/3 is offered in only $\leq 50\%$ of cases. Nevertheless, DoH/2 remains the most commonly advertised protocol. Note that some categories are missing due to the fact that DDR-enabled resolvers within these categories only delegate to the excluded cloud DNS providers.



(a) IPv4 DDR protocol distribution across network types (November 11, 2024)



(b) IPv6 DDR protocol distribution across network types (October 30, 2024)

Figure 6.6: Heatmaps illustrating the distribution of advertised encryption protocols across different network types for IPv4 and IPv6 DDR-enabled resolvers, without considering the cloud DNS providers *Google*, *Cloudflare*, *Cisco*, *Quad9*, and *AdGuard*. Greener cells indicate a higher likelihood of the protocol being advertised, whereas red cells represent a lower likelihood.

Observation: Distribution of DoE Protocols.

The distribution of DoE protocols in DDR configurations shows that most commonly DoT, DoH/2, and DoH/3 are advertised, while DoQ adoption remains low due to limited support from major cloud DNS providers. Excluding these providers, DoQ becomes more prevalent in Non-Profit, Content, and Enterprise networks, surpassing DoH/3 and DoT in some categories.

6.3.2 Protocol Prioritization

In general, there are no notable differences in the priorities of protocols offered by IPv4 and IPv6 DDR-enabled resolvers. A complete overview of the protocols and their associated priorities for both address families can be found in Section B.4. The following analysis focuses solely on the priorities associated with protocols advertised by IPv4 DDR-enabled resolvers.

Every SVCB RR within DDR must specify a priority, indicating the preferred

order of advertised DoE resolvers and their protocols (see Section 2.4.2). However, ultimately clients determine which DoE resolver and protocol to use. Among the configurations, DoT is assigned priority 1 most frequently (26.29%), followed closely by DoH/2 and DoH/3 with priority 2 in 26.27% of cases. This pattern is influenced by Google's configurations (see Section 6.2), where DoT is given a higher priority than DoH. In contrast, DoQ is assigned priority 1 in only 0.163% of cases, but when DoQ is offered, it holds the highest priority in 54.51% of such configurations. Interestingly, the legacy protocol DoH/1.1, when included in a DDR configuration, is consistently assigned the highest priority. Notably, DoH/1.1 appears exclusively in configurations targeting *AdGuard*'s and *Control D*'s resolvers.

Examining the configurations in terms of priority gaps, we observe that priorities are not always offered sequentially. Instead, gaps often exist between assigned priorities. For example, the highest priority recorded is 30 for DoQ in six cases, with the next highest being 20 for DoH/2. These gaps are particularly noticeable in *Cisco*'s DDR configurations, where DoT is assigned the highest priority (5), followed by DoH/2 with priorities 10 and 20, each targeting different DoE resolver endpoints. The rationale behind such priority assignments remains unclear to us but complies with the relevant standards [124, 150].

A figure with the total number of advertised protocol-priority combinations for IPv4 and IPv6 DDR-enabled resolvers can be found in Table B.16.

6.3.3 Default Configuration Deviation

Unlike other studies that probe DoE resolvers using default ports (see Section 3.3), DDR enables the discovery of DoE resolvers with arbitrary configurations, including non-standard ports and custom *dohpath* settings in the case of DoH. In the following, we consider the most recent scans, dated November 11, 2024, for IPv4 DDR-enabled resolvers and October 30, 2024, for IPv6. Among IPv4 DDR-enabled resolvers, 0.081% advertise ports for DoE protocols that deviate from the respective standards. In IPv6, this proportion increases to 0.41%. Interestingly, port deviations for both IPv4 and IPv6 DDR-enabled resolvers are only observed in DoH/2 and DoQ protocols.

For DoQ, the default port specified by the standard is 853 [72]. Among IPv4 DDR-enabled resolvers, only 784 as an alternative port is advertised in 190 SVCB records. In contrast, deviations are more diverse for DoH/2. Here, 506 SVCB records advertise ports different from the standard port 443, listing a total of 115 distinct ports. The most common alternative ports include 8443 (22.73%), 444 (12.25%), and 4443 (6.92%).

IPv6 DDR-enabled resolvers exhibit two non-standard ports for DoQ: port 20784 in one case and port 784 in 44 cases, which is similar to IPv4. For DoH/2, 50 different ports are advertised. The most prevalent among them are 444 and 8443, each accounting for 20% of the cases, followed by 11443 at 12%.

Deviations in DDR configurations are not confined to ports; they also extend to the *dohpath* used in DoH protocols. While the standard suggests `/dns-query{?dns}` as a URI path, resolvers retain the flexibility to define custom paths for issuing DNS requests. The most frequently used *dohpath* deviations are illustrated in Table 6.5. In total, there are 469 *dohpaths* different to the standard's suggested path advertised by IPv4 DDR-enabled resolvers, and 10 in the case of IPv6.

Notably, a total of 36 advertised *dohpaths* fail to comply with the DDR-related standard [151], which mandates that the path must include the parameter `{?dns}` to specify where the encoded DNS request should be placed. All these non-compliant *dohpaths* are associated with SVCB RRs targeting *dns0.eu.* resolvers. We identified the DDR-enabled resolvers as being hosted by *AS 50902*, operated by *Dns0.eu A.d.*, a French non-profit organization. This provider claims to offer a free, sovereign, and GDPR-compliant recursive DNS resolvers to protect the citizens and organizations of the European Union [35]. We contacted the organization to address and correct these improperly specified *dohpaths*. If the issue remains unresolved, clients discovering DoH endpoints via DDR on this provider will likely fail and revert to Do53.

Several DDR-enabled resolvers offer the same DoE resolver (`*.controld.com`)

Table 6.5: Top 10 most advertised *dohpath* SVCB keys in DDR configurations that deviate from the standard's suggested path `/dns-query{?dns}` [64].

a Top 10 *dohpath* deviations offered by IPv4 DDR-enabled resolvers.

DoE Target	dohpath	Count
dns.controld.com.	/comss{?dns}	90 (19.19%)
freedns.controld.com.	/ads{?dns}	42 (8.96%)
dns.controld.com.	/2adx05pxl71{?dns}	27 (5.76%)
freedns.controld.com.	/unfiltered{?dns}	24 (5.12%)
freedns.controld.com.	/uncensored{?dns}	18 (3.84%)
zero.dns0.eu.	/	16 (3.41%)
freedns.controld.com.	/family{?dns}	15 (3.20%)
dns.controld.com.	/290q6sg4y9n{?dns}	15 (3.20%)
dns.controld.com.	/2adace8hybt{?dns}	12 (2.56%)
dns0.eu.	/	12 (2.56%)
Total		469

b All *dohpath* deviations offered by IPv6 DDR-enabled resolvers.

DoE Target	dohpath	Count
dns.controld.com.	/1fyodh4yzbn{?dns}	3 (30%)
dns0.eu.	/	2 (20%)
doh.dns-ga.net.	/query{?dns}	2 (20%)
zero.dns0.eu.	/	2 (20%)
doh.ticklers.org.	/dns{?dns}	1 (10%)
Total		10

but with different *dohpaths*. This behavior likely stems from *Control D* operating as a cloud service offering customizable DNS solutions, where unique DoH URI paths are randomly generated for individual clients.

Although configuration deviations are rare within the context of DDR, these findings highlight the utility of DDR in gaining insights into DoE deployments that deviate from standard configurations. By identifying DoE resolvers operating outside default settings, DDR enables the detection of "hidden" deployments that would otherwise remain unnoticed in studies relying solely on default protocol configurations.

6.3.4 Indication of Oblivious DNS over HTTPS (ODoH)

Since the standardization of ODoH in SVCB RR through RFC 9460 [150] in November 2023, resolvers can indicate their support for ODoH by setting the SVCB key `ohttp` (for a detailed explanation of the ODoH protocol, see Section 2.3.5). As DDR builds on SVCB RRs, this signaling can also be utilized by DDR-enabled servers. However, despite being standardized for over a year, our analysis shows that no DDR-enabled resolver set the `ohttp` flag during our entire scanning period, in either IPv4 or IPv6.

During our scans for DDR-enabled resolvers, we recorded their exact configuration offered. By analyzing SVCB keys not defined by DDR or related standards, the AS *3303* stands out. This AS is part of *Swisscom*, the largest ISP in Switzerland and the largest IT company in the country. This AS contains the only three resolvers that advertise the SVCB key *32769*, in combination with advertising an encrypted resolver that belongs to *Akamai*, a big CDN provider.

Interestingly, the SVCB key *32769* originates from an earlier draft of the RFC 9230 [88], which listed it as the SVCB key for ODoH until August 2021. The draft was withdrawn and replaced by RFC 9230, which no longer includes this key. Moreover, the key was never standardized by the IANA [77], making it continued use by *Swisscom* puzzling.

Unlike the `ohttp` flag, which is the current signal for ODoH support, the *32769* key was designed to hold a value. This value, known as *ObliviousDoHConfigs*, was intended to convey a ODoH configuration as a base64-encoded string. The value typically contains meta-information necessary for encryption algorithms, such as supported cipher suites or public-key material. This implementation is documented in an older, archived repository from *Cloudflare* [19]. However, the value provided by *Swisscom* is not base64-decodable (see Listing 6.1), diverging even from the specifications in the older draft. Furthermore, we can not identify

Listing 6.1: DDR SVCB ResR from one of Swisscom's Resolvers

```
_dns.resolver.arpa.       0 IN SVCB
      1 sc.r15.doh.dns.akasecure.net.
          alpn="h2"
          dohpath="/dns-query{?dns}"
          key32769="\000\001\000(\000
          \000\001\000\001\000 \023
          \010\197\0144\022d\219X(
          Cw\150-!R\240\148\204\180
          <Dk\201\148&\241\193P[\0221"
```

any information about ODoH support in the public available information of *Akamai*'s DNS service. It remains unclear whether this is an implementation error or a deliberate behavior.

6.4 Analysis of Verified Discovery

DDR enables the discovery of delegated DoE endpoints and their configuration through two distinct methods: *discovery using a resolver's IP address* and *discovery using resolver names* (see also Section 2.4.1). The first method is applicable when only the resolver's IP address is known, querying the resolver for the SVCB RR `_dns.resolver.arpa`. Essentially, we used this method in our measurement's second stage to discover DDR configurations (see Section 4.1.4). However, the DDR standard requires clients to verify a DDR response. In the following, we analyze the suggested *Verified Discovery* method for DDR's *discovery using a resolver's IP address*. In Section 7.2.1 we analyze the suggested verification method for the *discovery using resolver names*.

6.4.1 IP-based Verified Discovery

In the case DoE resolvers are uncovered by the *discovery using a resolver's IP address*, the DDR standard suggests the *Verified Discovery* method. This method is "... a mechanism that allows the automatic use of a Designated Resolver that supports DNS encryption that performs a TLS handshake" [124]. Essentially, the client must check for the DoE resolver's certificate validity and additionally validating, whether the IP address of the DDR-enabled resolver occurs in the SAN field of the TLS certificate. The standard further explicitly

states that if these checks fail, "[...] the client MUST NOT automatically use the discovered Designated Resolver if this designation was only discovered via a *_dns.resolver.arpa.* query". For the analysis of this verification method, we focus on certificates from our most recent scans on November 11, 2024, for IPv4 and on October 30, 2024, for IPv6.

Of the 1,535 TLS certificates from IPv4 DoE resolvers, only 31 (2.02%) include any IPv4 address in the SAN field. For IPv6, 21 out of the 229 certificates (9.5%) meet this criterion.

A single DDR configuration can reference multiple DoE resolvers. Among the 304K IPv4 DDR-enabled resolvers, this results in 322K unique DDR-to-DoE resolver combinations. Of these, only 75 (0.0002%) would theoretically pass the *Verified Discovery* method. For IPv6, there are 8,344 such combinations, with only 40 (0.0048%) successfully meeting the *Verified Discovery* criteria.

Large DNS providers such as *Google*, *Cloudflare*, *Cisco* and *Quad9* stand out, as all their DDR configurations comply with the *Verified Discovery* method. This compliance arises because their DDR-enabled resolvers point to themselves in their configurations. Excluding the large cloud DNS providers, only four other organizations host *Verified Discovery*-compliant DDR configurations on at least one of their resolvers: *Swisscom* (AS 3303), *Amazon* (AS 8987), *DNS0.EU* (AS 50902), and *Levonet, s.r.o* (AS 50242). For IPv6, only *Levonet, s.r.o* hosts such configurations.

6.4.2 Resulting Security Implications

The high prevalence of non-compliant DDR configurations with respect to the *Verified Discovery* method raises critical concerns about the practicality of automatic upgrades to encrypted resolvers. Such upgrades are generally infeasible unless clients initially configure their default resolver to one of the major cloud DNS providers' IPs. Clients relying on ISP-assigned nameservers (e.g., via DHCP) encounter barriers to upgrading, as shown in our analysis, most ISP resolvers delegate their clients to large cloud DNS providers (see Section 6.2). Although the DDR standard permits alternative verification methods, investigating these remains outside the scope of this study. Furthermore, the extent to which clients implement and adhere to the *Verified Discovery* method in practice is unclear (see Section 8.2).

In the absence of a verification method, clients are exposed to vulnerabilities such as redirection attacks, commonly known as *DNS Hijacking* [1], which is not specific to DDR but DNS in general. Another potential risk arises if attackers redirect clients to rogue servers under their control, compromising

user privacy by intercepting and potentially manipulating all ongoing DNS requests [164]. However, DDR expands DNS's attack surface by including not only the specification of DoE targets but also the details on how to connect to them. For instance, DoH configurations include the *dohpath*, instructing clients where to send DNS requests when upgrading to DoH. This additional complexity creates opportunities for attackers to manipulate the *dohpath*, potentially redirecting clients to execute harmful HTTP requests to arbitrary destinations. Such attacks could occur automatically, without user consent or awareness, thereby broadening the risks beyond those posed by unprotected DNS.

Observation: Real-World Security Challenges in DDR Configurations.

The *IP-based Verified Discovery* method in DDR requires clients to validate TLS certificates and ensure that the DDR-enabled resolver's IP is listed in the certificates' SAN field. However, this method succeeds in only 75 (0.0002%) IPv4 DDR-to-DoE resolver combinations and 40 (0.0048%) in IPv6. Large DNS providers like *Google* and *Cloudflare* adhere to the requirements of the verification method to full extend, while most other resolvers, including resolvers assigned by ISPs do not. As a result, DDR-compliant clients would fail to upgrade to the advertised DoE protocols in >99% of cases, leaving clients exposed to (privacy) risks in Do53. Even worse, DDR expands the DNS attack surface by including specifications like the *dohpath*, which attackers could exploit to manipulate connections if no response verification is applied.

7 Encrypted Resolvers (DoE) from the Viewpoint of DDR

In this chapter, we evaluate the reliability and security of DoE resolvers discovered via DDR. We analyze >75K encrypted DNS requests and responses to examine the geographical distribution of discovered resolvers, their adoption trends, and their network distribution (see Section 7.1). We assess their ability to establish secure connections and successfully resolve queries, focusing on the TLS versions and cipher suites used. Additionally, we investigate whether these resolvers implement DDR to allow clients to discover their configurations and adhere to the verification methods specified in the DDR standard (see Section 7.2). By probing resources hosted on our authoritative name servers, we uncover insights into resolver infrastructure, such as encrypted *recursive-to-authoritative* communication and unexpected behaviors like *traffic shadowing* (see Section 7.3). Ultimately, we identify critical shortcomings in the reliability, security, and functionality of advertised DoE resolvers, offering insights to inform improvements in their deployment and compliance with the DDR standard.

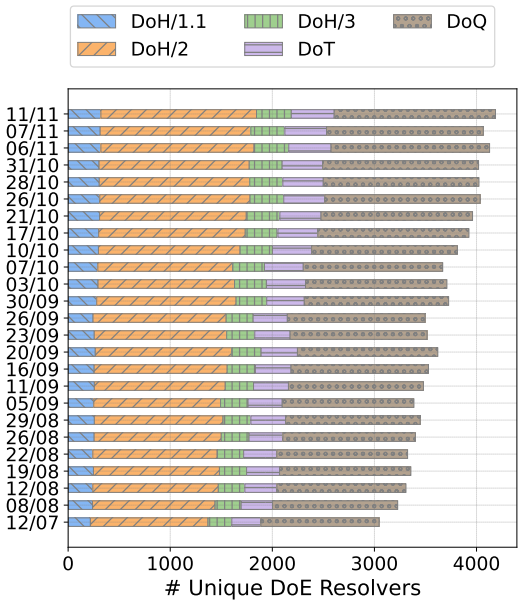
7.1 Analysis of Discovered DoE Resolvers

This section examines the deployment of DoE resolvers, analyzing their global distribution, connection failures across five categories, and operational nuances of DoE protocols. We also assess TLS versions, cipher suites, and mTLS, offering insights into the security of the TLS layer in DoE resolvers.

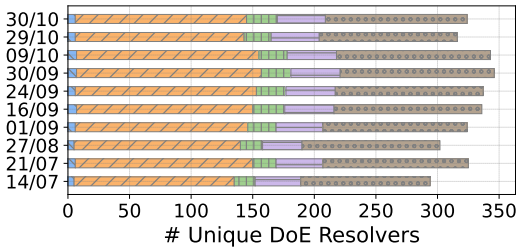
7.1.1 Unique DoE Resolvers

By executing the DDR discovery query over our four-month measurement period, we discovered a total of 3,288 unique DoE resolvers. We define a unique DoE resolver as one that is advertised with a distinct domain name (target) in a DDR configuration. The number of unique DoE resolvers per measurement run is shown in Figure 7.1, which also breaks down the resolvers by protocol and origin, i.e., whether the DoE resolver was discovered through an IPv4 or IPv6 DDR-enabled resolver.

Breaking this down by protocols, we identified 566 DoH/1.1, 2,935 DoH/2, 596 DoH/3, 3,064 DoQ, and 711 DoT resolvers. It is important to note that a single unique DoE resolver may support multiple protocols, so the sum of individual protocols does not equal the total number of discovered unique DoE resolvers.



(a) Number of DoE resolvers discovered by IPv4 DDR-enabled resolvers.



(b) Number of DoE resolvers discovered by IPv6 DDR-enabled resolvers.

Figure 7.1: Evolution of unique DoE resolvers discovered through IPv4 and IPv6 DDR-enabled servers, separated by respective DoE protocol.

The number of DoQ resolvers stands out: while our analysis of DDR configurations revealed that DoQ is relatively infrequently advertised (see Section 6.3), the number of unique DoQ resolvers is the highest among all DoE protocols. It is comparable to the number of unique DoH/2 resolvers, which is the most frequently advertised DoE protocol by DDR-enabled resolvers. This discrepancy arises because, although there are a similar number of distinct DDR configurations offering DoQ and DoH/2, the most common DDR configuration for DoH/2 was observed approximately 6.1M times throughout all of our measurements, whereas the most frequent configuration for DoQ was seen only about 22K times. This suggests that while there is a wide variety of DoQ resolvers, they are advertised much less frequently.

Notably, our dataset of unique DoQ resolvers constitutes the largest collection of DoQ resolvers by ADN identified and analyzed in any comparable study to date (see Section 3.3). This dataset provides a unique opportunity to analyze DoQ adoption and deployment in

the wild, as well as investigate the performance and reachability of DoQ resolvers (see Section 8.2).

Incorporating the temporal dimension into the analysis of unique DoE resolvers reveals an increase in the number of DoE resolvers discovered through IPv4 DDR-enabled resolvers during our measurements. This growth is primarily driven by new DoQ resolvers (rising from 1,157 to 1,580) and DoH/2 resolvers (increasing from 1,152 to 1,524). Interestingly, when considering percentage growth, the legacy protocol DoH/1.1 recorded the highest increase at 47.25% (from 218 to 321). In contrast, we do not observe this growth trend among unique DoE resolvers discovered through IPv6 DDR-enabled resolvers.

7.1.2 Global and AS-Level Distribution

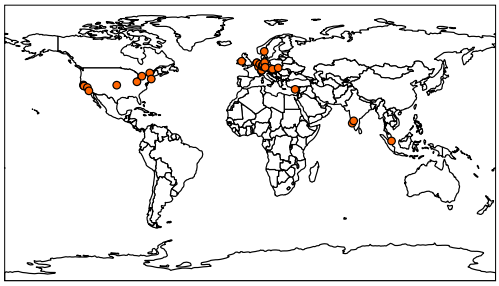
We enriched the DoE resolver data with AS and location data (see Section 4.1.6) and visualized the global distribution of servers for each protocol across the globe in Figure 7.2. Each orange-colored dot represents a unique DoE resolver. It is important to note that a single DoE resolver may support multiple DoE protocols; therefore, dots can appear in the same location across multiple maps but representing the same server. Additionally, dots may overlap if servers are mapped to the same location. Furthermore, the locations of 752 out of 3,208 DoE resolvers (23.44%) could not be determined and are not included in the plots.

The distribution of DoE resolvers indicates that DoQ and DoH/2 resolvers are the most globally dispersed, whereas the remaining protocols show fewer points with many servers concentrated in the same locations. For instance, all 566 DoH/1.1 resolvers map to only three locations (Canada, Cyprus, Ireland), while the 3,064 DoQ resolvers are spread across 72 countries. Similarly, DoH/2 resolvers are distributed across 71 countries, whereas the 711 DoT resolvers span only 14 countries, with none located in China.

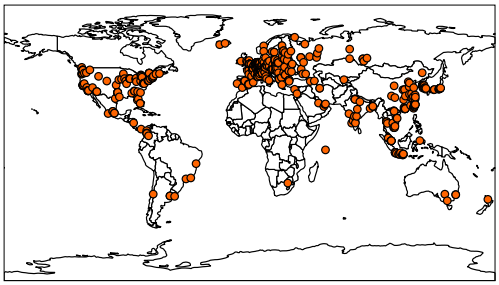
Focusing on the ASes hosting the DoE resolvers, we observe that for each DoE protocol, the majority of resolvers are hosted in *AS 212772* (*AdGuard*). Notably, more than 95% of unique DoH/1.1 resolvers and over 91% of unique DoH/3 resolvers reside in this AS. In contrast, DoQ resolvers demonstrate greater AS diversity, with only 17.68% of resolvers hosted by *AdGuard*.

We observe the highest AS diversity among DoQ and DoH/2 resolvers, which are distributed across more than 479 ASes. In comparison, DoT and DoH/3 resolvers are confined to only 33 and 17 ASes, respectively.

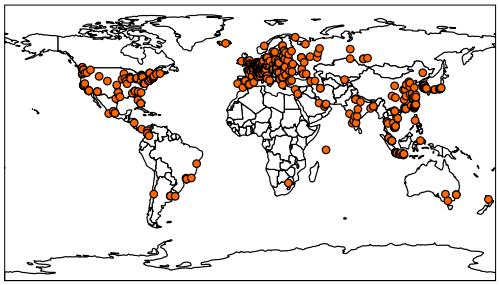
(a) DoH/1.1 Resolvers



(b) DoT Resolvers



(c) DoH/2 Resolvers



(d) DoQ Resolvers

(e) DoH/3 Resolvers

Figure 7.2: The figures illustrate the global distribution of servers for each DoE protocol. Each dot represents a unique DoE resolver. Note that a single DoE resolver may support multiple DoE protocols; consequently, dots may appear at the same location in two or more figures.

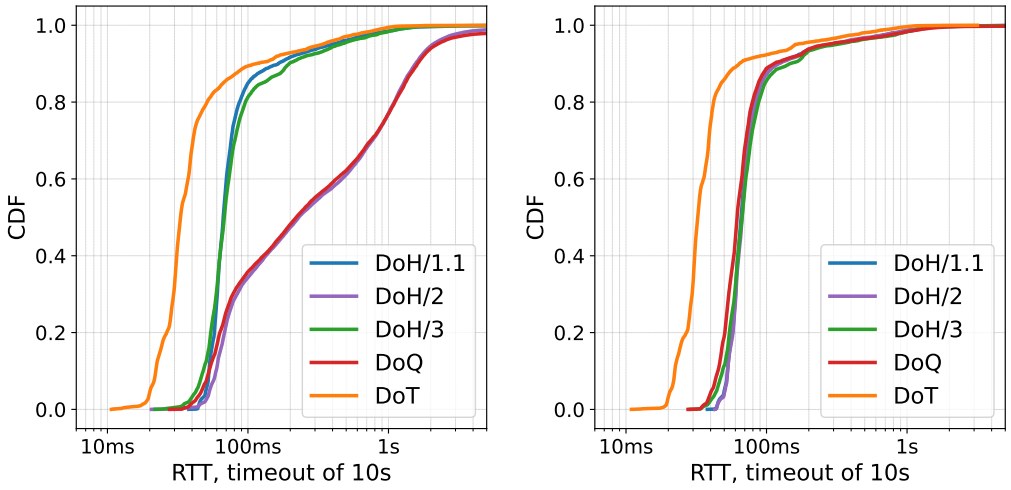
Observation: Distribution and Counts of DoE Resolvers.

We identified 3,288 unique DoE resolvers, with DoQ and DoH/2 being the most prevalent. DoQ resolvers show the widest global spread, appearing in 72 countries and over 479 ASes, whereas protocols like DoH/1.1 and DoT are hosted in fewer locations, often by entities like *AdGuard*, which dominates DoH/1.1 and DoH/3 deployments. Notably, 23% of resolver locations remain unidentified, reflecting gaps in visibility.

7.1.3 Performance

For probing the discovered DoE resolvers, we sent specially crafted DNS requests to the resolvers, which resolved a resource hosted on our authoritative DNS servers (see Figure 4.6). In total, we collected the RTTs from 28,792 DoE requests successfully returning a DNS response. The corresponding CDF for each protocol is shown in Figure 7.3. It is important to note that these values should be interpreted with caution, as our measurements are conducted from a single VP. Depending on the geographic location of the DoE resolvers, RTT values may vary crucially (see Section 7.1.2). The measurements include all DoE probes conducted from September 1, 2024, onward, as reliable RTT tracking became available on our measurement platform only from that date.

Contrary to our expectation that DoQ and DoH/3 would exhibit similar RTT values due to their shared use of QUIC, and that DoH/2 and DoH/1.1 would have comparable performance as only the application layer differs, our measurements revealed a different pattern. DoH/2 and DoQ demonstrate similar performances, with DoH/2 being approximately 100ms faster on average.



(a) CDF of all DoE probe's RTTs.

(b) CDF of DoE probes and their RTTs targeting only *AdGuard* and *Control D* ASes.

Figure 7.3: The CDF of the RTTs for the discovered DoE resolvers during our probing phase is presented. The left-hand figure depicts the CDF for all DoE probes that returned a DNS response, while the right-hand figure focuses specifically on ASes hosting resolvers for all DoE protocols, namely *AdGuard* and *Control D*.

Surprisingly, DoH/3 is faster than both DoQ and DoH/2 but slower than DoH/1.1. The best performing DoE protocol during our measurements is DoT.

We believe that the observed performance results are influenced by the lack of multiple VPs in our study. Our VP is stationed in Berlin, Germany. For example, most DoH/1.1 resolvers are concentrated in Europe and North America, whereas DoH/2 and DoQ resolvers are comparatively more prevalent in Asia, Oceania, and South America (see Section 7.1.2). Consequently, it is reasonable that DoH/1.1 demonstrates better performance than the more geographically distributed hosts running DoE protocols like DoQ.

For example, considering only the ASes operating DoE resolvers that support all four DoE protocols (e.g., *AdGuard* and *Control D*), DoT remains the best-performing protocol, with an average RTT of 60ms. However, the remaining protocols exhibit similar performance, with average RTT values ranging from 115ms to 123ms.

Further large-scale research involving diverse VPs is necessary to gain deeper insights into the performance characteristics of DoE protocols in the wild (see Section 8.2).

7.1.4 Errors and Resulting Reliability

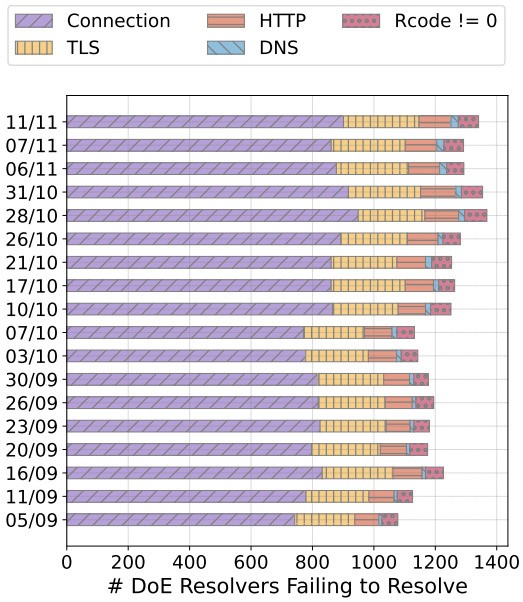
If clients use DDR to automatically upgrade to DoE protocols, it is crucial to assess the reliability of the advertised DoE resolvers they are designated to. This section analyzes the reliability of the DoE resolvers we discovered during our measurements. To this end, our measurement architecture logs any error encountered during a DoE request and response. We categorized the observable errors into five categories: *Connection*, *TLS*, *HTTP*, *DNS*, and *RCODE != 0*.

Connection errors occur when no connection to the DoE resolver can be established. For example, this is the case if the given hostname cannot be resolved in the public DNS. If no IP address is associated with the hostname, establishing a connection becomes impossible. We also observed unreachable hosts, hosts rejecting connections, and redirection loops. Overall, we identified 15 distinct errors in this category.

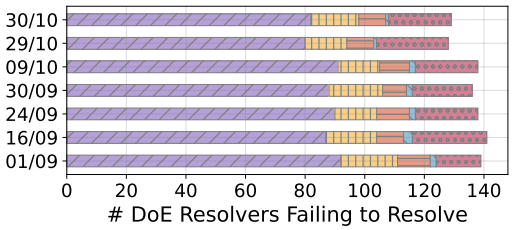
Every DoE protocol is based on TLS (see Section 2.3). Within this context, we identified nine distinct errors, including issues with unknown CAs, self-signed or expired TLS certificates, or handshake failures such as invalid Message Authentication Codes (MACs). In some cases, servers returned plaintext responses even after a successful TLS connection was established.

For DoH protocols, the application layer additionally introduces specific errors through HTTP status codes. We observed 18 distinct errors in this layer,

15 of which corresponded to 400-series or 500-series error codes. For example, the status code 403 (*Forbidden*) is returned when a client is not authorized to access the requested resource (i.e., the *dohpath*).



(a) Error distribution of DoE resolvers discovered by IPv4 DDR-enabled resolvers.



(b) Error distribution of DoE resolvers discovered by IPv6 DDR-enabled resolvers.

Figure 7.4: Evolution of the encountered error categories of all DoE resolvers. The upper figure shows the error distribution across DoE resolvers discovered by IPv4 DDR-enabled resolvers, while the lower one depicts the distribution across DoE resolvers discovered by IPv6 DDR-enabled resolvers.

When the DoE resolver returns a DNS response, it is encoded in the wire format [113]. Consequently, parsing can also fail on the client side. In this context, we observed two distinct errors: either the encoded response was not in the wire format (malformed), or no SVCB RRs were returned. These errors are classified as *DNS* errors.

Finally, if the DNS response can be parsed, the RCODE field provides information about the DoE resolvers' success of looking up the unique resource on our authoritative name servers. A non-zero RCODE indicates that the resolving failed. We identified four distinct RCODE errors: code 1 (*FormErr*), code 2 (*ServFail*), code 3 (*NXDomain*), and code 5 (*Refused*).

The evolution of the different errors is shown in Figure 7.4 for the DoE resolvers we discovered throughout our measurements through IPv4 and IPv6 DDR-enabled resolvers. While the total number of errors is increasing for DoE resolvers discovered by IPv4 DDR-enabled resolvers, it is the opposite for IPv6 DDR-enabled discovered resolvers. Roughly, the increasing number of DoE resolvers returning errors aligns with the increasing trend in the total number of DoE resolvers (see

Figure 7.2). However, it is the opposite for DoE resolvers discovered by IPv6 DDR-enabled resolvers: the number of error returning DoE resolvers is slightly decreasing while the total number of DoE resolvers is slightly increasing. Since we only consider two months for the error analysis, these figures may not be enough to indicate solid trends.

Having a closer look at the errors, we observe a broad variety of 44 distinct errors, highlighting the complexity of the issues encountered. The overall distribution of errors across each DoE protocol can be found in Table 7.1.

We observe that DoH/1.1 (98%), DoH/3 (95%), and DoT (93%) most frequently provided successful responses to our DNS queries. However, the most commonly observed protocol in DDR configurations, DoH/2, performed relatively poorly. More than every third request resulted in errors. The least frequently advertised protocol in DDR, but with the most resolvers, DoQ, performed worst, with almost half of all requests (42%) resulting in non-resolvable resources.

Focusing on the error causes for DoH/2 resolvers, over 50% of errors were attributed to connectivity issues, i.e., no connection could be established with the DoE resolver, most of them due to connection timeouts. We see a similar trend with DoQ, where over one-third of requests failed. As with DoH/2, the majority of errors (around 66%) were caused by connectivity problems, with timeouts being the dominant issue. Analyzing the origin of these DoQ and DoH/2 resolvers reveals that most of them are hosted in the AS *13335* (*Cloudflare*) (10% DoH/2, 18% DoQ). Whether these servers are directly hosted by *Cloudflare* or only use its network infrastructure remains unclear.

It is concerning not only that DDR configurations often point to resolvers that are unreachable but also that many DoQ and DoH/2 resolvers experience issues establishing secure connections via TLS. Approximately 16% of DoQ errors and 20% of DoH/2 errors are attributed to TLS issues. For all TLS errors we observed with DoQ, certificate-related problems are the root cause: in

Table 7.1: Distribution of error categories across each DoE protocol. Percentages within each error category represent their share of the total number of errors observed for the respective DoE protocol.

Protocol	# Req.	# Errors	*Connection*	*TLS*	*HTTP*	*DNS*	*RCODE != 0*
DoH/1.1	6,055	70 (1.16%)	15 (21.43%)	1 (1.43%)	1 (1.43%)	-	53 (75.71%)
DoH/2	27,758	10,713 (38.59%)	6,052 (56.49%)	2,187 (20.41%)	1,638 (15.29%)	300 (2.80%)	536 (5.00%)
DoH/3	6,606	317 (4.80%)	126 (39.75%)	30 (9.46%)	107 (33.75%)	1 (0.32%)	53 (16.72%)
DoQ	27,074	11,433 (42.23%)	9,204 (80.50%)	1,857 (16.24%)	-	-	372 (3.25%)
DoT	7,806	545 (6.98%)	360 (66.06%)	-	-	-	185 (33.94%)

78% of cases, the certificate is expired, 18% are signed by an unknown CA, and the remaining cases involve invalid certificates (e.g., mismatched signatures). For DoH, while expired certificates account for only 66% of TLS issues, we could identify a total of eight different TLS errors.

In comparison, we could not observe any TLS connection issues with DoT, and only one single TLS failure with DoH/1.1.

HTTP status codes in the 400 or 500 range are also rare for requests using DoH/1.1. However, DoH/2 and DoH/3 exhibited error rates of 15% and 34%, respectively. For DoH/3, 90% of errors are due to an incorrect URI, where the hostname of the DoH/3 resolver combined with the *dohpath* from the DDR configuration do not form a valid URL. In comparison, for DoH/2, approximately 60% of HTTP errors are caused by the HTTP status code 404 (resource not found). This indicates that the DDR configuration specified a *dohpath* pointing to a resource that does not exist on the DoH/2 resolver, showing discrepancies between the DDR configuration and the actual server configuration.

Finally, we examine DNS responses with non-zero RCODEs. Over 87% (1,045) of these cases returned a *Refused* RCODE, indicating that the DoE resolver rejected the query. Similar to the 404 HTTP status code, this error may indicate a discrepancy between the DDR configuration and the actual server configuration, as the DDR configuration indicates a DoE resolver that is not willing to resolve the requested resource. DoH/1.1, DoH/3 and DoT resolvers returned this RCODE in more than 98% of the non-zero RCODE cases, while DoH/2 and DoQ resolvers returned it in 86% and 80% of cases, respectively. Notably, in 33 cases (2.75%), the RCODE is *NXDomain*, which is unusual since the requested resource remains available on our authoritative name servers throughout the entire measurement period without any downtime. Yet, the resolver claimed a non-existence of the requested resource. These responses originated from servers in Taiwan and Singapore. The exact cause of this error remains unclear. The remaining RCODEs *FormErr* and *ServFail* occurred in 10% (33) and 0.083% (1) of all requests, respectively.

7.1.5 TLS Analysis

Our measurement architecture also tracks information that is negotiated on the TLS layer during connection establishment, i.e., negotiated TLS version and cryptographic protocols (cipher suites), and whether DoE resolvers require clients to present a certificate (mTLS) [83]. During probing, we universally support TLS versions greater than or equal to 1.0 and all available cipher

Observation: DoE Server Errors and resulting Reliability.

We observe noteworthy variability in error rates across DoE protocols, with 44 distinct error types highlighting their complexity. Protocols DoH/1.1 (98%), DoH/3 (95%), and DoT (93%) had the highest success rates, whereas DoH/2 and DoQ exhibited elevated error rates, with 38.6% and 42.2% of requests failing, respectively. Connectivity issues dominated, with over 50% of DoH/2 errors and 66% of DoQ errors caused by timeouts, and HTTP errors in DoH/2 (15%) and DoH/3 (34%) often resulting from invalid paths. Furthermore, 87% of non-zero RCODEs reflected query refusals, indicating misalignments in DDR configurations, while unexpected *NXDomain* errors arose in servers located in Taiwan and Singapore. In general, these errors may leave clients unable to resolve DNS requests when upgrading to these DoE resolvers or force them to fall back to unprotected Do53, undermining the intended security and privacy benefits.

suites, regardless of the DoE protocol. The DoE resolver then selects the most appropriate TLS version and cipher suite, as described in the relevant TLS standards [30, 31, 32, 134].

None of the DoE resolvers required clients to present a certificate during the TLS handshake, i.e., mTLS. This behavior aligns with an early IETF draft, which defines mTLS in the context of DoE protocols and specifies that DoE resolvers must not offer client authentication for connections established through prior discovery via DDR [83]. Consequently, all the DoE resolvers we discovered adhere to the specifications outlined in this early draft.

All DoE resolvers negotiate either TLS 1.2 or TLS 1.3. Notably, every connection to DoH/1.1 and DoT resolvers uses TLS 1.3. Since QUIC's handshake is based on TLS 1.3 [81, 159], DoQ and DoH/3 connections also consistently use version 1.3. For DoH/2, TLS 1.3 is negotiated in over 99% of cases.

We classified all negotiated cipher suites using the *Ciphersuite.info API* [140]. We only observed recommended or secure cipher suites. The most commonly negotiated cipher suite is `TLS_AES_128_GCM_SHA256`, which is used in over 94% of TLS connections, followed by `TLS_AES_256_GCM_SHA384` (3%) and `TLS_CHACHA20_POLY1305_SHA256` (2%). In all cases, Diffie-Hellman key exchange is executed to determine the session key.

These results do not imply that DoE resolvers are inherently secure in terms of TLS configurations. According to the standards [32, 134], the highest TLS version supported by both parties is selected, while the choice of cipher suite typically depends on the server operator and their security policies. Our measurements do not include probes to determine whether DoE resolvers support insecure TLS versions or cipher suites (see Section 8.2).

7.2 DDR Adoption Among DoE Resolvers

If the hostname of a DoE resolver is known, DDR can be utilized to discover its current configuration through the *discovery using resolver (domain) names* method (see Section 2.4.1). To perform this analysis, we executed DDR using this discovery method on all 3,204 unique DoE resolvers identified during the second stage of our measurements (see Section 4.1.4). Of these resolvers, 626 (19.54%) responded to the DDR discovery query. However, 601 (96%) of the responses contained invalid DDR configurations, missing mandatory keys such as the priority.

Interestingly, among the resolvers providing invalid DDR configurations are DoE resolvers from *AdGuard*, whose hostnames follow the pattern `*.d.adguard-dns.com`. *Quad9*, in contrast, does not provide any DDR configuration for its DoE resolvers.

Analyzing the remaining 25 DoE resolvers with valid DDR configurations, we observe that all delegate to themselves. We expect this behavior, as these DoE resolvers already offer DoE protocols on the same host. Among these 25 resolvers, two belong to our DDR-enabled resolvers and measurement architecture (see Section 4.1.2). Notably, 18 of the 25 resolvers belong to major DNS cloud providers, including *Google*, *Cloudflare* and *Cisco*. A detailed list of the remaining DoE resolvers and their target destinations is provided in Table B.17.

7.2.1 Name-based Verified Discovery and DNSSEC

When clients use the *discovery using resolver (domain) names* method, the DDR standard [124] requires them to verify the hostname's presence in the TLS certificate of the advertised resolvers (see Section 2.4.3). All 25 DDR-enabled DoE resolvers comply with this requirement, as they include their hostname in the TLS certificates. This is due to the fact that they do not delegate outside

their own AS as observed in most of the DDR configurations (see Section 6.2.1), but delegating to themselves.

Another method to validate DDR configurations discovered by the *discovery using resolver (domain) names* method is DNSSEC (see Section 2.1.4). However, DNSSEC applies only to *discovery using resolver (domain) names*, as these records exist within the public DNS hierarchy, enabling resolvers to sign and clients to validate them. Among the 626 resolvers that responded to our DDR discovery queries, only 24 (3.83%) implement DNSSEC. While this adoption rate remains low, it is marginally higher than the rates reported in recent studies [17]. Further, 8 of the 25 DDR-enabled DoE resolver returning a valid configuration have DNSSEC enabled (32%). Of the 17 having no DNSSEC support, 11 belong to *Cisco* and two to *AdGuard*. We want to note that we did not validate the returned signatures. In general, further research is necessary to evaluate the cryptographic robustness of the signature and the reliability of client-side validation for both, the DNSSEC signatures in the context of DoE and DDR's verification method (see Section 8.2).

Despite the fact that many DoE resolvers provide non-compliant configurations, clients have viable options for validating DDR responses retrieved via the discovery using hostnames. All DoE resolvers offering a valid DDR configuration theoretically pass the verification methods defined by the standard, and some resolvers additionally leverage DNSSEC for enhanced authenticity validation.

Observation: DDR Adoption and Response Verification among DoE Resolvers.

We analyzed DDR responses from 3,204 DoE resolvers and found that only 626 (19.54%) responded, with 96% containing invalid configurations, mostly missing mandatory keys like priority. Among the 25 resolvers with valid configurations, all comply with DDR's *discovery using resolver (domain) names*, ensuring their hostnames are included in TLS certificates by delegating to themselves. All DoE resolvers with valid DDR configurations provide clients with reliable means to verify responses, including DNSSEC, which is implemented by 8 out of 25 resolvers.

7.3 Recursive Resolving Behavior

In this section, we investigate the recursive resolving behavior of DoE resolvers by analyzing our authoritative name server logs in conjunction with the uniquely crafted DoE probes we sent. Our analysis explores several critical aspects, including unusual query replays spanning months (*traffic shadowing*), instances of non-compliant or misbehaving resolvers, and the extent to which encrypted *recursive-to-authoritative* communication is utilized. By examining these behaviors, we aim to uncover patterns and anomalies that impact the security and reliability of DoE resolvers.

7.3.1 Traffic Shadowing Behaviors

From September 1, 2024, to November 11, 2024, we sent 75,299 uniquely crafted DNS queries to all available DoE resolvers. In the following, we refer to DoE queries as the DNS queries we initially sent to the DoE resolvers during the DoE probing in our methodology (see Figure 4.6). The resources to be resolved by the DoE queries resided on our authoritative name servers. Our name servers received 52,392 of these requests, none of which were logged as errors, meaning every request was successfully answered by our name servers.

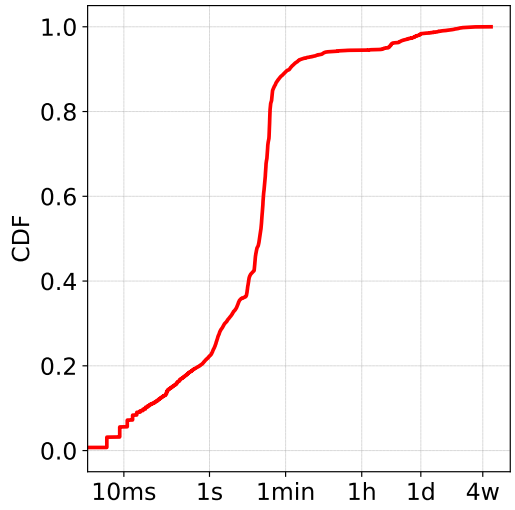


Figure 7.5: The CDF considers all replayed DNS queries and their time difference between the first and the last replayed query.

However, we observe a high number of DoE queries that are repeated. In total, the name servers' logs contain 218,352 query requests with our uniquely crafted DNS query names. Approximately 12K of the 49K (22%) DoE queries are repeated one or more times.

Notably, one query (i.e., having the same QNAME) is repeated 5,250 times. Tracing these 5,250 queries on our name server reveals that they originate from 115 different servers across 15 distinct ASes. Of these servers, 68 (59%) are located in China, with 33 (28%) belonging to AS *4837* (*China Backbone*). Based on

the organization names associated with these ASes, all but three appear to belong to Chinese companies. Interestingly, 34 requests are replayed by 26 different servers in Google's network (AS *15169*), and 16 requests originate from 13 distinct servers in *Cloudflare*'s network (AS *13335*). We believe these requests are replayed through *Cloudflare*'s and *Google*'s cloud DNS services. A notable outlier includes 21 requests from four different servers hosted in AS *49544* (*i3D.net B.V.*), which provides servers for video games and is owned by *Ubisoft*, a French video game company. We probed these IP addresses ourselves and none of these source IP addresses seem to act as open resolvers. The reason why these DNS servers appear to replay our original DNS queries multiple times remains unclear. We have reported this unusual activity to the company.

We shift the perspective from absolute numbers of repeated DoE queries to their temporal behavior. Specifically, we analyze the time difference between the first and the last repeated DoE query. Figure 7.5 shows the cumulative distribution function of these time differences. We observe that 50% of the repeated DoE queries are replayed within 16 seconds. The longest observed time span between the first and the last repeated DoE query is 70 days, with these queries being replayed at irregular intervals.

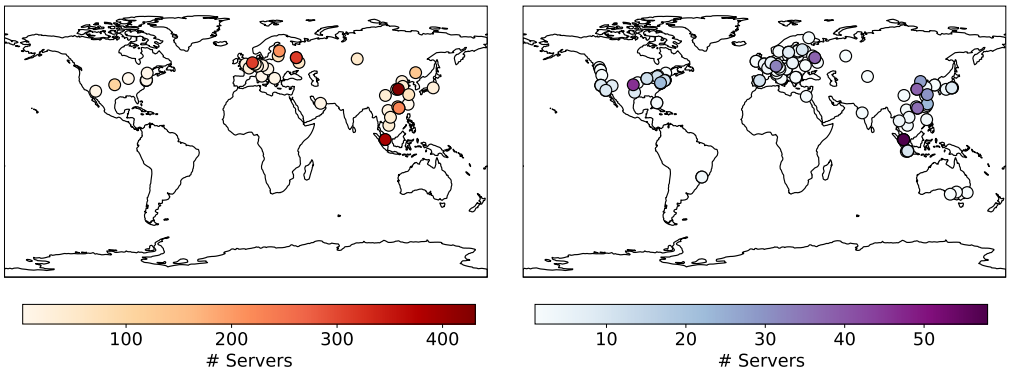
We determine the locations of servers that send more than one DoE query, with a time difference between the first and last query exceeding one day. Since each DoE request is uniquely identifiable due to its one-time QNAME, we can link the original query to the repeated queries on our name servers. Figure 7.6 illustrates the locations of servers initially queried for our DoE probes (discovered DoE resolvers via DDR) and the locations of servers from which repeated queries were received on our name servers.

First, we focus on servers that may have intercepted and replayed our initial DoE queries. These servers are distributed across 20 countries and 59 ASes. Over 22% are located in China, followed by approximately 15% in Russia and over 7% in the United States (see Figure 7.6 (a)). However, these servers do not replay the queries themselves; instead, they utilize recursive resolvers worldwide to perform the repeated requests. In most cases, recursive resolvers from Singapore are used (35%), followed by China (17%) and the United States (13%) (see Figure 7.6 (b)). On our name servers, we observe queries originating from 36 countries and 109 distinct ASes. In 45% of the cases, recursive resolvers from *Google*'s AS *15169* are used, followed by *Cloudflare* (AS *13335*) with over 8% and *Yandex* (AS *13238*) with more than 7%. As a result, most of the queries are intercepted in China, Russia and the United States, and then mostly

replayed through recursive resolvers in Singapore through *Google*'s cloud DNS services.

This behavior is described as *traffic shadowing* by Xing et al. [165] and as *DNS Zombies* by Geoff Huston [75]. However, our study differs from theirs. Xing et al. analyzed behavior using Do53, HTTP, and TLS, including non-DNS protocols. For Do53, they assumed that transmitted Do53 packets were intercepted and subsequently replayed.

Through our methodology (see Figure 4.6), we can narrow down the interception possibilities. Since we sent queries through secure channels to the DoE resolvers, they cannot be intercepted between the client and the recursive resolver. Additionally, the adoption rate of QNAME Min. has risen to over 50% in recent years [108], making the interception and replay of our queries on the recursive-to-authoritative connection even less likely. Consequently, it is more plausible that the recursive resolvers themselves, which received the initial DoE query, record and replay the queries after some time. However, we suggest further analyses and investigations into the *traffic shadowing* in the context of DoE protocols to understand why this behavior occurs (Section 8.2).



(a) Server locations of DoE resolvers initially queried from the perspective of our measurement architecture.

(b) Server locations of DoE request sources from the perspective of our name servers.

Figure 7.6: Both figures consider only server locations where the initial DoE query was observed multiple times in our name server logs, originating from multiple distinct ASes with a time difference exceeding one day. The left-hand figure displays the locations of servers we initially queried for our DoE probes, while the right-hand figure illustrates the locations of servers from which the repeated requests were received.

7.3.2 Misbehavior

Analyzing the resolved ResR of our DoE requests, we observed 142 cases where the DoE resolver returned a non-zero RCODE, but the request never reached our name servers. In these cases, the DoE resolver returned a bogus IPv4 address that does not match the one we set up in our DNS zone. In half of these cases, the IP `0.0.0.0` is returned. Every third of these misbehaving resolvers is hosted in the AS *64089* (DNSFilter, Inc.), while in total, most of them are hosted in the U.S. ($> 70\%$) and South Korea ($> 18\%$) and none from China. It remains unclear to us why these resolvers do not comply with the DNS protocol but misbehave.

7.3.3 Encrypted Recursive-To-Authoritative Communication

Finally, our uniquely crafted DoE queries allow us to determine to what extent DoE resolvers employ encrypted communication to resolve our requested resources over a secure channel. This encrypted *recursive-to-authoritative* communication was standardized by the IETF in March 2024 through the RFC 9539 [54]. To enable encrypted communication with our name servers, we support DoT and DoH/2 in their standard configurations, i.e., DoT on port 853 and DoH/2 on port 443, with DoH/2 receiving DNS queries on the URI path */dns-query{?dns}*. Additionally, to facilitate discovery of these configurations by recursive resolvers, our name servers also support DDR.

None of the DoE resolvers tried to resolve the requested resource via DoH/2 or DoT. Although the DoE resolvers themselves received DNS queries through a DoE protocol, indicating they have implemented this functionality, they did not use DoE for resolving resources in the public DNS. Moreover, none of the recursive resolvers utilized DDR to discover our encrypted endpoints.

However, RFC 9539 [54] leaves the choice of using encryption during the recursive resolving process to the recursive resolver. This decision is due to the computational and network overhead that encryption adds to each resolving process.

Of particular interest is that RFC 9539 states in the context of DoH: "Currently, there are no mechanisms for a DNS recursive resolver to predict the [*dohpath*] on its own, in an opportunistic or unilateral fashion, without incurring an excessive use of resources" [54]. Yet, DDR was specifically designed for this purpose and was standardized prior to RFC 9539, in November 2023.

Observation: Recursive Resolving Behavior of DoE Resolvers.

We analyzed 75,299 DoE queries and found that 52,392 reached our authoritative name servers without errors, while 22% were repeated, originating from 115 servers across 15 ASes, predominantly in China. These repeated queries, replayed over intervals from seconds to 70 days, likely stemmed from recursive resolvers themselves, indicating *traffic shadowing*. Additionally, 142 queries returned invalid responses with bogus IPv4 addresses, mostly from resolvers in the U.S. and South Korea. Despite DoE support, no recursive resolvers used encrypted *recursive-to-authoritative* communication or DDR to discover encrypted endpoints.

8 Limitations & Future Work

In this chapter, we aim to critically evaluate the constraints of this study (see Section 8.1) and address unresolved questions to improve future research in the field of DDR and DoE protocols (see Section 8.2).

8.1 Limitations

The biggest limitation of this study is the reliance on measurements collected from a single VP, which restricts the ability to capture a detailed and representative view of DNS resolvers, DDR's adoption, and the DoE landscape. The decentralized nature of the Internet, while ensuring resilience, inherently prevents a unified perspective of its global state [98]. As a result, measurement studies must approximate this global view by carefully selecting representative datasets or focusing on specific aspects supported by reliable data sources. Employing distributed VPs is therefore essential to reduce biases caused by localized outages, routing anomalies, or region-specific infrastructure constraints. For instance, while numerous IPv4 addresses were retrieved via *ZMap* from China, many timed out in response to subsequent DDR discovery queries (see Section 5.2). This behavior may result from the Great Firewall of China (GFW) blocking such queries. However, due to the absence of a VP within China, we cannot confirm this and the actual reason remains unclear.

Plonka et al. [127] investigated classification and measurement methods for IPv6. They found that although the IPv6 address space is vast, many IPv6 addresses are reused — i.e., assigned to different users over time, often within a week. This behavior may also lead to churn in IPv6 addresses assigned to responsive servers on UDP port 53 in the *IPv6 Hitlist Service* [52]. The *IPv6 Hitlist Service* typically has a one-week delay between the start date of its scan and the publication of its result set. This delay may cause us to miss some responsive IPv6 addresses in our study. Yet, we lack detailed information about IPv6 address churn in the context of DNS resolvers.

During data enrichment, we use *PeeringDB* to classify the ASes of DNS resolvers (see Section 4.1.6). However, *PeeringDB* does not classify every AS, leaving some as "unknown" categories. Additionally, *PeeringDB* classifications

are not always fine-grained. For instance, *Google* is categorized as a content provider but also functions as an enterprise. This behavior may result in misclassifications of ASes in our study.

Furthermore, we use *GeoLite2*[111] to geolocate DNS resolvers. However, the Internet, and particularly ASes and their associated addresses, are not static but evolve over time [27]. In our study, we use a fixed version of the *GeoLite2* database from November 13, 2024. As a result, any changes during our measurement period in ASes or their associated addresses are not reflected and may be inaccurately represented in our findings.

Lastly, our measurements are limited to DNS resolvers accessible from the public Internet. For example, we do not include (recursive) resolvers that are hidden from the public, such as ISP resolvers solely serving their customers' DNS requests.

8.2 Future Work

The limitations of our study provide valuable opportunities for future work. Our measurement architecture offers a solid foundation to implement distributed measurements from multiple VPs, enabling a more comprehensive view of the (encrypted) resolver landscape from the perspective of DDR. This capability is supported by our architecture's distributed message broker system (see Section 4.1). We have already initiated collaborations with VPN providers, with *ProtonVPN* granting us access to their infrastructure for conducting measurements. Unfortunately, *RIPE Atlas* [6] does not permit scans for DDR-enabled resolvers. However, projects like *VPNGate* [118] or *OONI* [49] could provide insights into measurements from residential Internet connections. Additionally, integrating the data enrichment process directly into the measurement architecture could allow real-time enrichment, mitigating biases caused by outdated data in sources like *GeoLite2* [111].

Our measurements also lack information about the actual usage of the DDR protocol. While DDR may be enabled on a resolver, it is unclear whether DNS libraries for stub resolvers actually perform DDR discovery. Analyzing passively recorded DNS data (e.g., data from *DNSDB* [39, 138] or *DNS Observatory* [50]) could provide insights into the real-world usage rate of DDR. Furthermore, it remains unknown whether stub resolver libraries implement DDR and verify DDR responses correctly. Our findings indicate that most DDR-enabled resolvers point to resolvers outside their AS, often failing to comply with the DDR's response verification requirements (see Section 6.4).

This raises security concerns, as clients that do not verify DDR responses may become vulnerable. Investigating DDR verification methods and compliance in DNS libraries lies beyond the scope of this work.

While DDR is one method for discovering DoE resolvers, the IETF has standardized Discovery of Network-designated Resolvers (DNR) [10] for local discovery of designated network resolvers via DHCP. To the best of our knowledge, no studies have yet investigated the adoption of *EDR*. Future work could explore the adoption of *EDR* and compare it with DDR adoption rates.

Our study further reveals limited details about the TLS layer on which DoE resolvers rely. While we provide an overview of the most frequently negotiated TLS versions and cipher suites (see Section 7.1.5), we lack insights into whether these servers support any insecure TLS versions or cipher suites. Addressing this would require a probing mechanism like *SSLabs* [129], which our architecture does not currently support.

Specifically, the certificates exchanged during DoQ sessions merit further investigation. According to the QUIC standard [81], resolvers offering DoQ are subject to an amplification limit, restricting response sizes during the initial handshake to three times the client's message size. While other studies show that this limit is often disregarded [116], our dataset allows us to analyze certificate lengths and compliance with amplification limits across a broader range of DoQ resolvers. As our dataset includes the largest collection of unique DoQ resolvers by ADNs, combining it with other datasets [116] could provide a more comprehensive view of the DoQ landscape.

During the development of our measurement architecture, we frequently questioned appropriate timeouts for DNS and DoE protocols. To the best of our knowledge, no detailed study empirically investigates timeouts for Do53 or DoE connection attempts. While DNS libraries and their default timeout values (used in this study, see Section 4.1) offer a basis, these values are rarely substantiated in documentation. Our experiments also show that responses are lost even with standard library values (see Section 4.2). A study empirically analyzing timeouts for DoE and Do53 could identify optimal values, conserving resources and improving protocol performance. Such research could extend beyond DNS to other Internet protocols.

Our analysis of DoE probes also revealed an unusually high number of replayed queries on our name servers (see Section 7.3.1). While this behavior has been described in the literature [165] as *traffic shadowing*, it has not been explored in the context of DoE resolvers. Xing et al. [165] assume that DNS requests sent over Do53 are intercepted by unknown parties during transit

and then replayed. In the DoE context, interception is less likely due to encryption between clients and recursive resolvers. Additionally, the adoption of QNAME Min. has increased in recent years [108]. We suspect that recursive resolvers themselves may record and replay these DNS requests. A detailed study could provide further insights into *traffic shadowing* behavior in DoE resolvers.

Finally, we collected data during this study that remains without any analysis. This includes fingerprinting of DNS servers, which could theoretically track resolvers beyond IP churn. For instance, we could observe configuration changes in DDR and DoE resolvers at the server level instead of the global scale. Additionally, we collected data on the EDSR protocol (see Section 2.4.4), scheduled scans for *iphints* and glue records, and scans bypassing certificate validation, which have yet to be analyzed.

9 Conclusion

In this study, we present a comprehensive empirical investigation of the DDR protocol, focusing on its adoption, configuration, and the operational challenges associated with enabling automated transitions to encrypted DNS communication via DoE protocols. By addressing our research questions, we conclude this study.

RQ1: What are the adoption rates and trends of DDR-enabled resolvers in IPv4 and IPv6, and how do they vary across geographical regions and network types over time?

Among the approximately 4M IPv4 DNS servers discovered on average, 7.59% are DDR-enabled, i.e., return a DDR configuration, compared to only 2.65% of the 287K IPv6 DNS servers. During the four-month measurement period, IPv4 DDR-enabled resolvers increased by 3.5K, but DDR density declined slightly by -0.14%, indicating slower relative growth compared to the overall IPv4 DNS population. Conversely, IPv6 experienced a 1K decrease in DDR-enabled resolvers but showed a positive trend in DDR density of 2.8%, reflecting proportional growth. These trends suggest differing dynamics between IPv4 and IPv6 adoption, though the short measurement period limits the robustness of these conclusions.

Geographically, Asia hosts the largest number of IPv4 DDR-enabled resolvers (152K, 50.05%), while Africa leads in DDR density, with 34.46% of all DNS servers supporting DDR. While Asia, Africa and Europe saw increases of DDR-enabled servers throughout our measurement period (>3%), South America and North America experienced declines of -9.05% and -4.56%, respectively. The IPv6 space exhibits different patterns. South America dominates both the number and density of DDR-enabled resolvers, with Bolivia alone contributing 22.31% of the global total through 1.8K servers. Further, it achieves an exceptional DDR density of 98.11%, highlighting its unique role as a leader in IPv6 DDR adoption. By contrast, Europe, despite hosting the largest number of IPv6 DNS servers, lags in DDR adoption, with IPv4 and IPv6 DDR densities of only 3.47% and 0.95%, respectively. On a country level, Bangladesh achieves the highest IPv4 DDR density with 81.89%.

From a network perspective, DDR-enabled servers are primarily hosted

within "Network Services" (e.g., ISP networks), accounting for 58.60% of IPv4 and 77.37% of IPv6 DDR-enabled servers. While IPv4 DDR density remains relatively low and stable across ASes, IPv6 networks show a trend towards centralization meaning IPv6 DDR-enabled servers concentrate within fewer ASes. This trend is particularly pronounced in South America, where the top-performing ASes achieve DDR densities approaching 100%.

Overall, DDR adoption remains uneven across regions, countries, and network types. While some areas stagnate or decline, our findings confirm that DDR adoption is concentrated within networks of ISPs, consistent with the protocol's design focus on *stub-to-recursive* communication.

RQ2: What configuration patterns are observed in DDR-enabled resolvers, and how do these patterns differ across networks and over time?

During the measurement period, the configurations of DDR-enabled resolvers exhibited little change, remaining relatively stable over time. However, a key finding is the limited diversity in DDR configurations, with over 97% of DDR-enabled resolvers delegating their clients to just four major providers: *Google*, *Cloudflare*, *Cisco*, and *Quad9*. *Google*'s dominance is particularly striking, as 79.3% of IPv4 and 82.54% of IPv6 DDR-enabled resolvers delegate to its DoE resolver. In contrast, only 0.69% of IPv4 and 1.60% of IPv6 DDR-enabled resolvers delegate within their own AS. This overwhelming reliance on a few dominant providers raises concerns regarding DDR's contribution to DNS resolver centralization and its implications for user privacy and governance.

The distribution of advertised DoE protocols reveals that DoH/2, DoT and DoH/3 are the most commonly supported. The legacy protocol DoH/1.1 remains in use, often associated with delegations to *AdGuard*'s resolvers. Conversely, DoQ adoption remains notably low (<7%), primarily due to limited support from major cloud DNS providers. However, outside these dominant providers, DoQ shows a stronger presence in specific network types, particularly in non-profit, content, and enterprise networks (>87%). In these contexts, DoQ frequently surpasses DoH/3 and DoT in DDR configurations, especially in enterprise and content-oriented networks.

In fact, the resulting low DDR configuration diversity results from resolvers replicating the exact configurations used by major DNS cloud providers. This raises concerns that operators may be copying these configurations without adapting them to their specific needs (e.g., to their own DoE resolvers). Such practices could inadvertently contribute to DNS resolver centralization, although DDR provides valuable methods to counteract centralization if properly applied.

RQ3: What observable challenges hinder clients from successfully transitioning from plain DNS to DoE protocols in real-world DDR deployments?

Real-world DDR deployments reveal severe challenges that impede clients from transitioning seamlessly from unencrypted DNS (Do53) to DoE protocols. One of the primary hurdles lies in DDR's *IP-based Verified Discovery*, which requires clients to validate TLS certificates and ensure that the DDR-enabled resolver's IP address is listed in the certificate's SAN field. Our analysis shows that this method succeeds in only 75 IPv4 (0.0002%) and 40 IPv6 (0.0048%) DDR-to-DoE resolver combinations. Large DNS providers such as *Google* and *Cloudflare* comply with these verification requirements as they delegate to their own DoE resolvers, but the majority of other resolvers, particularly those managed by ISPs, fail to meet these requirements. Consequently, DDR-compliant clients cannot upgrade to advertised DoE protocols in over 99% of cases, leaving users vulnerable to privacy risks associated with unencrypted DNS.

The complexity of DoE protocols introduces additional operational challenges. Across the 44 distinct error types observed, protocols such as DoE/1.1, DoH/3, and DoT exhibited high success rates of 93%-98% during probing, while DoH/2 and DoQ showed elevated error rates of 38.6% and 42.2%, respectively. Timeouts were the predominant cause of failure, accounting for over 50% of DoH/2 errors and 66% of DoQ errors. HTTP errors in DoH/2 (15%) and DoH/3 (34%) were often linked to invalid URI paths, while query refusals (87% of non-zero RCODEs) reflected misalignments in DDR configurations. Such errors undermine the intended security and privacy benefits of DDR and DoE and showcase discrepancies between DDR configurations and real-world deployments of DoE protocols. Addressing these challenges requires concerted efforts from operators to improve DDR configurations, stricter adherence to protocol specifications, and further research into robust mechanisms for secure and reliable upgrades to encrypted DNS communication.

Bibliography

[1] Gautam Akiwate, Raffaele Sommese, Mattijs Jonker, Zakir Durumeric, kc claffy, Geoffrey M. Voelker, and Stefan Savage. **Retroactive identification of targeted DNS infrastructure hijacking**. In: *Proceedings of the 22nd ACM Internet Measurement Conference, IMC 2022, Nice, France, October 25-27, 2022*. Ed. by Chadi Barakat, Cristel Pelsser, Theophilus A. Benson, and David R. Choffnes. ACM, 2022, 14–32. DOI: 10.1145/3517745.3561425. URL: https://doi.org/10.1145/3517745.3561425 (see page 97).

[2] Fatemah Alharbi, Jie Chang, Yuchen Zhou, Feng Qian, Zhiyun Qian, and Nael B. Abu-Ghazaleh. **Collaborative Client-Side DNS Cache Poisoning Attack**. In: *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*. IEEE, 2019, 1153–1161. DOI: 10.1109/INFOCOM.2019.8737514. URL: https://doi.org/10.1109/INFOCOM.2019.8737514 (see page 14).

[3] Mark Allman and Vern Paxson. **Issues and etiquette concerning use of shared measurement data**. In: *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference, IMC 2007, San Diego, California, USA, October 24-26, 2007*. Ed. by Constantine Dovrolis and Matthew Roughan. ACM, 2007, 135–140. DOI: 10.1145/1298306.1298327. URL: https://doi.org/10.1145/1298306.1298327 (see pages 29, 58).

[4] Marios Anagnostopoulos, Georgios Kambourakis, Elisavet Konstantinou, and Stefanos Gritzalis, 201–220. In: *Situational Awareness in Computer Network Defense*. IGI Global, 2012. DOI: 10.4018/978-1-4666-0104-8.ch012. URL: http://dx.doi.org/10.4018/978-1-4666-0104-8.ch012 (see page 15).

[5] Alfred Arouna, Mattijs Jonker, and Ioana Livadariu. **On unifying diverse DNS data sources**. In: *Proceedings of the 22nd ACM Internet Measurement Conference, IMC 2022, Nice, France, October 25-27, 2022*. Ed. by Chadi Barakat, Cristel Pelsser, Theophilus A. Benson, and David R. Choffnes. ACM, 2022, 752–753. DOI: 10.1145/3517745.3563022. URL: https://doi.org/10.1145/3517745.3563022 (see page 31).

[6] Vaibhav Bajpai, Steffie Jacob Eravuchira, and Jürgen Schönwälder. **Lessons Learned From Using the RIPE Atlas Platform for Measurement Research**. *Comput. Commun. Rev.* 45:3 (2015), 35–42. DOI: 10.1145/2805789.2805796. URL: https://doi.org/10.1145/2805789.2805796 (see pages 30, 118).

[7] Birk Blechschmidt. *GitHub — blechschmidt/massdns: A high-performance DNS stub resolver for bulk lookups and reconnaissance (subdomain enumeration)*. (visited on October 4, 2024). 2016. URL: https://github.com/blechschmidt/massdns/ (see pages 33, 43, 47, 59).

[8] Stephane Bortzmeyer. **DNS Query Name Minimisation to Improve Privacy**. *RFC* 7816 (2016), 1–11. DOI: 10.17487/RFC7816. URL: https://doi.org/10.17487/RFC7816 (see pages 11, 16).

[9] Stephane Bortzmeyer and Shumon Huque. **NXDOMAIN: There Really Is Nothing Underneath**. *RFC* 8020 (2016), 1–10. DOI: 10.17487/RFC8020. URL: https://doi.org/10.17487/RFC8020 (see page 9).

[10] Mohamed Boucadair, T. Reddy K., Dan Wing, Neil Cook, and Tommy Jensen. **DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)**. *RFC* 9463 (2023), 1–23. DOI: 10.17487/RFC9463. URL: https://doi.org/10.17487/RFC9463 (see page 119).

[11] J.A. Braakhuis. *Design of a Reactive DNS Measurement System*. July 2021. URL: http://essay.utwente.nl/87863/ (see page 35).

[12] Canonical. *dockerhub - ubuntu/bind9*. (visited on October 10, 2024). URL: https://hub.docker.com/r/ubuntu/bind9/ (see pages 47, 159).

[13] Vinton G. Cerf. **Guidelines for Internet Measurement Activities**. *RFC* 1262 (1991), 1–3. DOI: 10.17487/RFC1262. URL: https://doi.org/10.17487/RFC1262 (see page 58).

[14] Stuart Cheshire and Marc Krochmal. **Special-Use Domain Names**. *RFC* 6761 (2013), 1–13. DOI: 10.17487/RFC6761. URL: https://doi.org/10.17487/RFC6761 (see page 24).

[15] Rishabh Chhabra, Paul Murley, Deepak Kumar, Michael D. Bailey, and Gang Wang. **Measuring DNS-over-HTTPS performance around the world**. In: *IMC '21: ACM Internet Measurement Conference, Virtual Event, USA, November 2-4, 2021*. Ed. by Dave Levin, Alan Mislove, Johanna Amann, and Matthew Luckie. ACM, 2021, 351–365. DOI: 10.1145/3487552.3487849. URL: https://doi.org/10.1145/3487552.3487849 (see page 37).

[16] Brent N. Chun, David E. Culler, Timothy Roscoe, Andy C. Bavier, Larry L. Peterson, Mike Wawrzoniak, and Mic Bowman. **PlanetLab: an overlay testbed for broad-coverage services**. *Comput. Commun. Rev.* 33:3 (2003), 3–12. DOI: 10.1145/956993.956995. URL: https://doi.org/10.1145/956993.956995 (see page 35).

[17] Taejoong Chung, Roland van Rijswijk-Deij, Balakrishnan Chandrasekaran, David R. Choffnes, Dave Levin, Bruce M. Maggs, Alan Mislove, and Christo Wilson. **A Longitudinal, End-to-End View of the DNSSEC Ecosystem**. In: *26th USENIX Security Symposium, USENIX Security 2017, Vancouver, BC, Canada, August 16-18, 2017*. Ed. by Engin Kirda and Thomas Ristenpart. USENIX Association, 2017, 1307–1322. URL: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/chung (see page 110).

[18] Cloudflare, Inc. *Announcing experimental DDR for 1.1.1.1*. (visited on December 13, 2024). Aug. 2022. URL: https://blog.cloudflare.com/announcing-ddr-support/ (see page 1).

[19] Cloudflare, Inc. *GitHub - cloudflare/odoh-go: Oblivious DoH library in Go*. (visited November 26, 2024). Dec. 2023. URL: https://github.com/cloudflare/odoh-go/ (see page 95).

[20] Confluent Inc. *confluentinc/kafka-images - Confluent Docker images for Apache Kafka*. (visited on October 10, 2024). URL: https://github.com/confluentinc/kafka-images (see page 49).

[21] David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and W. Timothy Polk. **Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**. *RFC* 5280 (2008), 1–151. DOI: 10.17487/RFC5280. URL: https://doi.org/10.17487/RFC5280 (see page 17).

[22] Michelle Cotton and Leo Vegoda. **Special Use IPv4 Addresses**. *RFC* 5735 (2010), 1–10. DOI: 10.17487/RFC5735. URL: https://doi.org/10.17487/RFC5735 (see page 47).

[23] João Damas, Michael Graff, and Paul Vixie. **Extension Mechanisms for DNS (EDNS(0))**. *RFC* 6891 (2013), 1–16. DOI: 10.17487/RFC6891. URL: https://doi.org/10.17487/RFC6891 (see page 10).

[24] Casey T. Deccio and Jacob Davis. **DNS privacy in practice and preparation**. In: *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies, CoNEXT 2019, Orlando, FL, USA, December 09-12, 2019*. Ed. by Aziz Mohaisen and Zhi-Li Zhang. ACM, 2019, 138–143. DOI: 10.1145/3359989.3365435. URL: https://doi.org/10.1145/3359989.3365435 (see page 38).

[25] Matthew Dempsky. **DNSCurve: Link-Level Security for the Domain Name System**. Internet-Draft draft-dempsky-dnscurve-01. Work in Progress. Internet Engineering Task Force, Feb. 2010. 10 pp. URL: https://datatracker.ietf.org/doc/draft-dempsky-dnscurve/01/ (see pages 15, 16).

[26] Frank Denis. **The DNSCrypt protocol**. Internet-Draft draft-denis-dprive-dnscrypt-04. Work in Progress. Internet Engineering Task Force, Aug. 2024. 17 pp. URL: https://datatracker.ietf.org/doc/draft-denis-dprive-dnscrypt/04/ (see page 15).

[27] Amogh Dhamdhere and Constantine Dovrolis. **Twelve Years in the Evolution of the Internet Ecosystem**. *IEEE/ACM Trans. Netw.* 19:5 (2011), 1420–1433. DOI: 10.1109/TNET.2011.2119327. URL: https://doi.org/10.1109/TNET.2011.2119327 (see page 118).

[28] John Dickinson, Sara Dickinson, Ray Bellis, Allison Mankin, and Duane Wessels. **DNS Transport over TCP - Implementation Requirements**. *RFC* 7766 (2016), 1–19. DOI: 10.17487/RFC7766. URL: https://doi.org/10.17487/RFC7766 (see page 51).

[29] Sara Dickinson, Daniel Kahn Gillmor, and Tirumaleswar Reddy. **Usage Profiles for DNS over TLS and DNS over DTLS**. *RFC* 8310 (2018), 1–27. DOI: 10.17487/RFC8310. URL: https://doi.org/10.17487/RFC8310 (see pages 22, 39).

[30] Tim Dierks and Christopher Allen. **The TLS Protocol Version 1.0**. *RFC* 2246 (1999), 1–80. DOI: 10.17487/RFC2246. URL: https://doi.org/10.17487/RFC2246 (see page 108).

[31] Tim Dierks and Eric Rescorla. **The Transport Layer Security (TLS) Protocol Version 1.1**. *RFC* 4346 (2006), 1–87. DOI: 10.17487/RFC4346. URL: https://doi.org/10.17487/RFC4346 (see page 108).

[32] Tim Dierks and Eric Rescorla. **The Transport Layer Security (TLS) Protocol Version 1.2**. *RFC* 5246 (2008), 1–104. DOI: 10.17487/RFC5246. URL: https://doi.org/10.17487/RFC5246 (see pages 17, 108, 109).

[33] Pratyush Dikshit, Jayasree Sengupta, and Vaibhav Bajpai. **Recent Trends on Privacy-Preserving Technologies under Standardization at the IETF**. *Comput. Commun. Rev.* 53:2 (2023), 22–30. DOI: 10.1145/3610381.3610385. URL: https://doi.org/10.1145/3610381.3610385 (see page 38).

[34] Jeremy Dix, Patrick Sattler, and Johannes Zirngibl. **ZDNS vs MassDNS: A Comparison of DNS Measurement Tools**. In: URL: https://api.semanticscholar.org/CorpusID:269808174 (see page 33).

[35] DNS0.EU. *The European public DNS that makes your Internet safer.* (visited on November 29, 2024). URL: https://www.dns0.eu/ (see page 94).

[36] DNSCrypt. *A protocol to improve DNS security and privacy.* (visited on September 24, 2024). 2011. URL: https://github.com/DNSCrypt/ (see pages 15, 16).

[37] Trinh Viet Doan, Justus Fries, and Vaibhav Bajpai. **Evaluating Public DNS Services in the Wake of Increasing Centralization of DNS**. In: *IFIP Networking Conference, IFIP Networking 2021, Espoo and Helsinki, Finland, June 21-24, 2021*. Ed. by Zheng Yan, Gareth Tyson, and Dimitrios Koutsonikolas. IEEE, 2021, 1–9. DOI: 10.23919/IFIPNETWORKING52078.2021.9472831. URL: https://doi.org/10.23919/IFIPNetworking52078.2021.9472831 (see pages 41, 87).

[38] Docker Inc. *Docker Engine*. (visited on October 14, 2024). URL: https://docs.docker.com/engine/ (see page 54).

[39] DomainTools. *Introducing DNSDB 2.0 - Passive DNS*. (visited on September 26, 2024). URL: https://domaintools.com/products/farsight-dnsdb/ (see pages 31, 43, 118).

[40] Zakir Durumeric. *ZMap - Sending Multiple Probes*. (visited on November 08, 2024). May 2017. URL: https://github.com/zmap/zmap/wiki/Sending-Multiple-Probes/ (see page 58).

[41] Zakir Durumeric, David Adrian, Ariana Mirian, Michael D. Bailey, and J. Alex Halderman. **A Search Engine Backed by Internet-Wide Scanning**. In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*. Ed. by Indrajit Ray, Ninghui Li, and Christopher Kruegel. ACM, 2015, 542–553. DOI: 10.1145/2810103.2813703. URL: https://doi.org/10.1145/2810103.2813703 (see pages 29, 55, 57).

[42] Zakir Durumeric, David Adrian, Phillip Stephens, Eric Wustrow, and J. Alex Halderman. **Ten Years of ZMap**. In: *Proceedings of the 2024 ACM on Internet Measurement Conference, IMC 2024, Madrid, Spain, November 4-6, 2024*. Ed. by Narseo Vallina-Rodriguez, Guillermo Suarez-Tangil, Dave Levin, and Cristel Pelsser. ACM, 2024, 139–148. DOI: 10.1145/3646547.3689012. URL: https://doi.org/10.1145/3646547.3689012 (see page 33).

[43] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. **ZMap: Fast Internet-wide Scanning and Its Security Applications**. In: *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*. Ed. by Samuel T. King. USENIX Association, 2013, 605–620. URL: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/durumeric (see pages 29, 30, 32, 43, 44, 47, 59).

[44] Electronic Frontier Foundation. *Certbot*. (visited on October 10, 2024). URL: https://certbot.eff.org/ (see pages 47, 158, 159).

[45] Robert Elz and Randy Bush. **Clarifications to the DNS Specification**. *RFC* 2181 (1997), 1–14. DOI: 10.17487/RFC2181. URL: https://doi.org/10.17487/RFC2181 (see page 51).

[46] Let's Encrypt. *Free SSL/TLS Certificates*. (visited on October 10, 2024). URL: https://letsencrypt.org/de/ (see pages 47, 158, 159).

[47] Chris Evans, Chris Palmer, and Ryan Sleevi. **Public Key Pinning Extension for HTTP**. *RFC* 7469 (2015), 1–28. DOI: 10.17487/RFC7469. URL: https://doi.org/10.17487/RFC7469 (see page 22).

[48] Stephen Farrell and Hannes Tschofenig. **Pervasive Monitoring Is an Attack**. *RFC* 7258 (2014), 1–6. DOI: 10.17487/RFC7258. URL: https://doi.org/10.17487/RFC7258 (see pages 14, 15, 151).

[49] Arturo Filastò and Jacob Appelbaum. **OONI: Open Observatory of Network Interference**. In: *2nd USENIX Workshop on Free and Open Communications on the Internet, FOCI '12, Bellevue, WA, USA, August 6, 2012*. Ed. by Roger Dingledine and Joss Wright. USENIX Association, 2012. URL: https://www.usenix.org/conference/foci12/workshop-program/presentation/filast%5C%C3%5C%B2 (see page 118).

[50] Pawel Foremski, Oliver Gasser, and Giovane C. M. Moura. **DNS Observatory: The Big Picture of the DNS**. In: *Proceedings of the Internet Measurement Conference, IMC 2019, Amsterdam, The Netherlands, October 21-23, 2019*. ACM, 2019, 87–100. DOI: 10.1145/3355369.3355566. URL: https://doi.org/10.1145/3355369.3355566 (see pages 1, 7, 31, 118).

[51] Ian Foster. *DNS Coffee Presentation*. (visited on September 26, 2024). URL: https://caida.org/workshops/kismet/1912/slides/kismet1912_ifoster.pdf (see page 31).

[52] Oliver Gasser, Quirin Scheitle, Sebastian Gebhard, and Georg Carle. **Scanning the IPv6 Internet: Towards a Comprehensive Hitlist**. In: *Traffic Monitoring and Analysis - 8th International Workshop, TMA 2016, Louvain la Neuve, Belgium, April 7-8, 2016*. Ed. by Alessio Botta, Ramin Sadre, and Fabian E. Bustamante. IFIP, 2016. URL: http://dl.ifip.org/db/conf/tma/tma2016/tma2016-final51.pdf (see pages 32, 43, 44, 47, 59, 117, 157).

[53] Miek Gieben. *miekg/dns - DNS library in Go*. (visited on October 8, 2024). 2012. URL: https://github.com/miekg/dns/ (see pages 4, 43, 44, 53).

[54] Daniel Kahn Gillmor, Joey Salazar, and Paul Hoffman. **Unilateral Opportunistic Deployment of Encrypted Recursive-to-Authoritative DNS**. *RFC* 9539 (2024), 1–24. DOI: 10.17487/RFC9539. URL: https://doi.org/10.17487/RFC9539 (see pages 4, 16, 23, 46, 52, 114).

[55] Google LLC. *google/cadvisor: Analyzes resource usage and performance characteristics of running containers*. (visited on October 14, 2024). URL: https://github.com/google/cadvisor/ (see page 54).

[56] Grafana Labs. *Grafana - The open observability platform*. (visited on October 14, 2024). URL: https://grafana.com/ (see page 54).

[57] Christian Grothoff, Matthias Wachs, Monika Ermert, and Jacob Appelbaum. **NSA's MORECOWBELL: knell for DNS**. *Unpublished technical report* (2017) (see page 14).

[58] Jeroen van der Ham. **Ethics and Internet Measurements**. In: *2017 IEEE Security and Privacy Workshops, SP Workshops 2017, San Jose, CA, USA, May 25, 2017*. IEEE Computer Society, 2017, 247–251. DOI: 10.1109/SPW.2017.17. URL: https://doi.org/10.1109/SPW.2017.17 (see page 58).

[59] Luuk Hendriks, Ricardo de Oliveira Schmidt, Roland van Rijswijk-Deij, and Aiko Pras. **On the Potential of IPv6 Open Resolvers for DDoS Attacks**. In: *Passive and Active Measurement - 18th International Conference, PAM 2017, Sydney, NSW, Australia, March 30-31, 2017, Proceedings*. Ed. by Mohamed Ali Kâafar, Steve Uhlig, and Johanna Amann. Vol. 10176. Lecture Notes in Computer Science. Springer, 2017, 17–29. DOI: 10.1007/978-3-319-54328-4_2. URL: https://doi.org/10.1007/978-3-319-54328-4%5C_2 (see page 36).

[60] Dominik Herrmann, Christoph Gerber, Christian Banse, and Hannes Federrath. **Analyzing Characteristic Host Access Patterns for Re-identification of Web User Sessions**. In: *Information Security Technology for Applications - 15th Nordic Conference on Secure IT Systems, NordSec 2010, Espoo, Finland, October 27-29, 2010, Revised Selected Papers*. Ed. by Tuomas Aura, Kimmo Järvinen, and Kaisa Nyberg. Vol. 7127. Lecture Notes in Computer Science. Springer, 2010, 136–154. DOI: 10.1007/978-3-642-27937-9_10. URL: https://doi.org/10.1007/978-3-642-27937-9%5C_10 (see pages 1, 14).

[61] Nguyen Phong Hoang, Ivan Lin, Seyedhamed Ghavamnia, and Michalis Polychronakis. **K-resolver: Towards Decentralizing Encrypted DNS Resolution**. In: *Proceedings 2020 Workshop on Measurements, Attacks, and Defenses for the Web*. MADWeb 2020. Internet Society, 2020. DOI: 10.14722/madweb.2020.23009. URL: http://dx.doi.org/10.14722/madweb.2020.23009 (see page 41).

[62] Nguyen Phong Hoang, Michalis Polychronakis, and Phillipa Gill. **Measuring the Accessibility of Domain Name Encryption and Its Impact on Internet Filtering**. In: *Passive and Active Measurement - 23rd International Conference, PAM 2022, Virtual Event, March 28-30, 2022, Proceedings*. Ed. by Oliver Hohlfeld, Giovane Cesar Moreira Moura, and Cristel Pelsser. Vol. 13210. Lecture Notes in Computer Science. Springer, 2022, 518–536. DOI: 10.1007/978-3-030-98785-5_23. URL: https://doi.org/10.1007/978-3-030-98785-5%5C_23 (see page 38).

[63] Paul Hoffman. **DNS Security Extensions (DNSSEC)**. *RFC* 9364 (2023), 1–10. DOI: 10.17487/RFC9364. URL: https://doi.org/10.17487/RFC9364 (see page 15).

[64] Paul E. Hoffman and Patrick McManus. **DNS Queries over HTTPS (DoH)**. *RFC* 8484 (2018), 1–21. DOI: 10.17487/RFC8484. URL: https://doi.org/10.17487/RFC8484 (see pages 1, 17, 18, 23, 94).

[65] Oliver Hohlfeld. **Operating a DNS-based Active Internet Observatory**. In: *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos, SIGCOMM 2018, Budapest, Hungary, August 20-25, 2018*. ACM, 2018, 60–62. DOI: 10.1145/3234200.3234239. URL: https://doi.org/10.1145/3234200.3234239 (see page 35).

[66] Austin Hounsel, Kevin Borgolte, Paul Schmitt, Jordan Holland, and Nick Feamster. **Comparing the Effects of DNS, DoT, and DoH on Web Performance**. In: *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. Ed. by Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen. ACM / IW3C2, 2020, 562–572. DOI: 10.1145/3366423.3380139. URL: https://doi.org/10.1145/3366423.3380139 (see page 38).

[67] Austin Hounsel, Paul Schmitt, Kevin Borgolte, and Nick Feamster. **Encryption without centralization: distributing DNS queries across recursive resolvers**. In: *ANRW '21: Applied Networking Research Workshop, Virtual Event, USA, July 24-30, 2021*. ACM, 2021, 62–68. DOI: 10.1145/3472305.3472318. URL: https://doi.org/10.1145/3472305.3472318 (see page 41).

[68] Rebekah Houser, Zhou Li, Chase Cotton, and Haining Wang. **An investigation on information leakage of DNS over TLS**. In: *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies, CoNEXT 2019, Orlando, FL, USA, December 09-12, 2019*. Ed. by Aziz Mohaisen and Zhi-Li Zhang. ACM, 2019, 123–137. DOI: 10.1145/3359989.3365429. URL: https://doi.org/10.1145/3359989.3365429 (see page 39).

[69] Russell Housley, W. Timothy Polk, Warwick Ford, and David Solo. **Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile**. *RFC* 3280 (2002), 1–129. DOI: 10.17487/RFC3280. URL: https://doi.org/10.17487/RFC3280 (see page 25).

[70] Zi Hu, Liang Zhu, John S. Heidemann, Allison Mankin, Duane Wessels, and Paul E. Hoffman. **Specification for DNS over Transport Layer Security (TLS)**. *RFC* 7858 (2016), 1–19. DOI: 10.17487/RFC7858. URL: https://doi.org/10.17487/RFC7858 (see pages 1, 13, 17, 18).

[71] Qing Huang, Deliang Chang, and Zhou Li. **A Comprehensive Study of DNS-over-HTTPS Downgrade Attack**. In: *10th USENIX Workshop on Free and Open Communications on the Internet, FOCI 2020, August 11, 2020*. Ed. by Roya Ensafi and Hans Klein. USENIX Association, 2020. URL: https://www.usenix.org/conference/foci20/presentation/huang (see page 39).

[72] Christian Huitema, Sara Dickinson, and Allison Mankin. **DNS over Dedicated QUIC Connections**. *RFC* 9250 (2022), 1–27. DOI: 10.17487/RFC9250. URL: https://doi.org/10.17487/RFC9250 (see pages 1, 17, 19, 90, 93).

[73] Geoff Huston. *DNS Privacy - APNIC Blog*. (visited on November 10, 2024). May 2016. URL: https://blog.apnic.net/2016/05/27/dns-privacy/ (see pages 1, 14).

[74] Geoff Huston. *DNS resolver centrality — APNIC Blog*. (visited on October 2, 2024). Sept. 2019. URL: https://blog.apnic.net/2019/09/23/dns-resolver-centrality/ (see pages 41, 68, 70, 87).

[75] Geoff Huston. *DNS Zombies - APNIC Blog*. (visited on November 11, 2024). Apr. 2016. URL: https://blog.apnic.net/2016/04/04/dns-zombies/ (see page 113).

[76] Geoff Huston. *The resolvers we use - APNIC Blog*. (visited on November 10, 2024). Nov. 2013. URL: https://blog.apnic.net/2014/11/28/the-resolvers-we-use/ (see pages 1, 7).

[77] IANA. *DNS Service Bindings (SVCB)*. (visited on November 26, 2024). Oct. 2024. URL: https://www.iana.org/assignments/dns-svcb/dns-svcb.xhtml (see pages 89, 95).

[78] IANA. *Special-Use Domain Names*. (visited on December 12, 2024). Sept. 2024. URL: https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml (see page 11).

[79] Internet Systems Consortium Inc. *BIND 9 - isc.org*. (visited on October 10, 2024). URL: https://www.isc.org/bind/ (see pages 46, 159).

[80] Internet Systems Consortium Inc. *dig(1)*. (visited on October 9, 2024). 2010. URL: https://linux.die.net/man/1/dig/ (see page 50).

[81] Jana Iyengar and Martin Thomson. **QUIC: A UDP-Based Multiplexed and Secure Transport**. *RFC* 9000 (2021), 1–151. DOI: 10.17487/RFC9000. URL: https://doi.org/10.17487/RFC9000 (see pages 17, 19, 90, 108, 119).

[82] Liz Izhikevich, Gautam Akiwate, Briana Berger, Spencer Drakontaidis, Anna Ascheman, Paul Pearce, David Adrian, and Zakir Durumeric. **ZDNS: a fast DNS toolkit for internet measurement**. In: *Proceedings of the 22nd ACM Internet Measurement Conference, IMC 2022, Nice, France, October 25-27, 2022*. Ed. by Chadi Barakat, Cristel Pelsser, Theophilus A. Benson, and David R. Choffnes. ACM, 2022, 33–43. DOI: 10.1145/3517745.3561434. URL: https://doi.org/10.1145/3517745.3561434 (see pages 30, 33, 43).

[83] Tommy Jensen, Jessica Krynitsky, Jeffrey Damick, Matt Engskow, and Joe Abley. **Client Authentication Recommendations for Encrypted DNS**. Internet-Draft draft-jaked-cared-00. Work in Progress. Internet Engineering Task Force, Oct. 2024. 12 pp. URL: https://datatracker.ietf.org/doc/draft-jaked-cared/00/ (see pages 5, 107, 108).

[84] Liang Jiao, Yujia Zhu, Baiyang Li, and Qingyun Liu. **Measuring DNS-over-Encryption Performance Over IPv6**. In: *22nd IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2024, Exeter, UK, November 1-3, 2023*. Ed. by Jia Hu, Geyong Min, Guojun Wang, and Nektarios Georgalas. IEEE, 2023, 444–451. DOI: 10.1109/TRUSTCOM60117.2023.00075. URL: https://doi.org/10.1109/TrustCom60117.2023.00075 (see page 38).

[85] Lin Jin, Shuai Hao, Haining Wang, and Chase Cotton. **Understanding the Impact of Encrypted DNS on Internet Censorship**. In: *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. Ed. by Jure Leskovec, Marko Grobelnik, Marc Najork, Jie Tang, and Leila Zia. ACM / IW3C2, 2021, 484–495. DOI: 10.1145/3442381.3450084. URL: https://doi.org/10.1145/3442381.3450084 (see page 39).

[86] Vineet John and Xia Liu. *A Survey of Distributed Message Broker Queues*. 2017. arXiv: 1704.00411 [cs.DC]. URL: https://arxiv.org/abs/1704.00411 (see page 50).

[87] Simon Josefsson. **The Base16, Base32, and Base64 Data Encodings**. *RFC* 4648 (2006), 1–18. DOI: 10.17487/RFC4648. URL: https://doi.org/10.17487/RFC4648 (see page 19).

[88] Eric Kinnear, Patrick McManus, Tommy Pauly, Tanya Verma, and Christopher A. Wood. **Oblivious DNS over HTTPS**. *RFC* 9230 (2022), 1–19. DOI: 10.17487/RFC9230. URL: https://doi.org/10.17487/RFC9230 (see pages 20, 95).

[89] Mike Kosek, Luca Schumann, Robin Marx, Trinh Viet Doan, and Vaibhav Bajpai. **DNS privacy with speed?: evaluating DNS over QUIC and its impact on web performance**. In: *Proceedings of the 22nd ACM Internet Measurement Conference, IMC 2022, Nice, France, October 25-27, 2022*. Ed. by Chadi

Barakat, Cristel Pelsser, Theophilus A. Benson, and David R. Choffnes. ACM, 2022, 44–50. DOI: 10.1145/3517745.3561445. URL: https://doi.org/10.1145/3517745.3561445 (see page 38).

[90] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. **Going Wild: Large-Scale Classification of Open DNS Resolvers**. In: *Proceedings of the 2015 ACM Internet Measurement Conference, IMC 2015, Tokyo, Japan, October 28-30, 2015*. Ed. by Kenjiro Cho, Kensuke Fukuda, Vivek S. Pai, and Neil Spring. ACM, 2015, 355–368. DOI: 10.1145/2815675.2815683. URL: https://doi.org/10.1145/2815675.2815683 (see pages 34, 35, 43, 49, 55).

[91] Rashna Kumar and Fabián E. Bustamante. **Decentralization, privacy and performance for DNS**. In: *SIGCOMM '21: ACM SIGCOMM 2021 Conference, Virtual Event, August 23-27, 2021, Poster and Demo Sessions*. Ed. by Marco Chiesa, David R. Choffnes, Athina Markopoulou, and Marinho P. Barcellos. ACM, 2021, 56–58. DOI: 10.1145/3472716.3472869. URL: https://doi.org/10.1145/3472716.3472869 (see pages 41, 68, 70).

[92] Franck Le, Jorge Ortiz, Dinesh C. Verma, and Dilip D. Kandlur. **Policy-Based Identification of IoT Devices' Vendor and Type by DNS Traffic Analysis**. In: *Policy-Based Autonomic Data Governance [extended papers from the Second International Workshop on Policy-based Autonomic Data Governance, PADG@ESORICS 2018, September 6, 2018, Barcelona, Spain]*. Ed. by Seraphin B. Calo, Elisa Bertino, and Dinesh C. Verma. Vol. 11550. Lecture Notes in Computer Science. Springer, 2018, 180–201. DOI: 10.1007/978-3-030-17277-0_10. URL: https://doi.org/10.1007/978-3-030-17277-0%5C_10 (see page 1).

[93] Iain R. Learmonth, Mallory Knodel, and Gurshabad Grover. **Guidelines for Performing Safe Measurement on the Internet**. Internet-Draft draft-irtf-pearg-safe-internet-measurement-10. Work in Progress. Internet Engineering Task Force, July 2024. 15 pp. URL: https://datatracker.ietf.org/doc/draft-irtf-pearg-safe-internet-measurement/10/ (see page 58).

[94] Jianfeng Li, Xiaobo Ma, Guodong Li, Xiapu Luo, Junjie Zhang, Wei Li, and Xiaohong Guan. **Can We Learn what People are Doing from Raw DNS Queries?** In: *2018 IEEE Conference on Computer Communications, INFOCOM 2018, Honolulu, HI, USA, April 16-19, 2018*. IEEE, 2018, 2240–2248. DOI: 10.1109/INFOCOM.2018.8486210. URL: https://doi.org/10.1109/INFOCOM.2018.8486210 (see pages 1, 14).

[95] Ruixuan Li, Xiaofeng Jia, Zhenyong Zhang, Jun Shao, Rongxing Lu, Jingqiang Lin, Xiaoqi Jia, and Guiyi Wei. **A Longitudinal and Comprehensive Measurement of DNS Strict Privacy**. *IEEE/ACM Trans. Netw.* 31:6 (2023),

2793–2808. DOI: 10.1109/TNET.2023.3262651. URL: https://doi.org/10.1109/TNET.2023.3262651 (see pages 39–41).

[96] Ruixuan Li, Baojun Liu, Chaoyi Lu, Haixin Duan, and Jun Shao. **A Worldwide View on the Reachability of Encrypted DNS Services**. In: *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*. Ed. by Tat-Seng Chua, Chong-Wah Ngo, Ravi Kumar, Hady W. Lauw, and Roy Ka-Wei Lee. ACM, 2024, 1193–1202. DOI: 10.1145/3589334.3645539. URL: https://doi.org/10.1145/3589334.3645539 (see pages 40, 41, 87).

[97] Wilson Lian, Eric Rescorla, Hovav Shacham, and Stefan Savage. **Measuring the Practical Impact of DNSSEC Deployment**. In: *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013*. Ed. by Samuel T. King. USENIX Association, 2013, 573–588. URL: https://www.usenix.org/conference/usenixsecurity13/technical-sessions/paper/lian (see page 15).

[98] Franziska Lichtblau. **From the edge to the core: towards informed vantage point selection for internet measurement studies**. PhD thesis. Saarland University, Saarbrücken, Germany, 2021. URL: https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/34810 (see page 117).

[99] Linux Man Pages. *resolv.conf(5)*. (visited on October 10, 2024). May 2024. URL: https://man7.org/linux/man-pages/man5/resolv.conf.5.html (see pages 50, 51).

[100] Zhenyan Liu, Yifei Zeng, Pengfei Zhang, Jingfeng Xue, Ji Zhang, and Jiangtao Liu. **An Imbalanced Malicious Domains Detection Method Based on Passive DNS Traffic Analysis**. *Secur. Commun. Networks* 2018 (2018), 6510381:1–6510381:7. DOI: 10.1155/2018/6510381. URL: https://doi.org/10.1155/2018/6510381 (see page 31).

[101] Jason Livingood, Manos Antonakakis, Bob Sleigh, and Alister Winfield. **Centralized DNS over HTTPS (DoH) Implementation Issues and Risks**. Internet-Draft draft-livingood-doh-implementation-risks-issues-04. Work in Progress. Internet Engineering Task Force, Sept. 2019. 24 pp. URL: https://datatracker.ietf.org/doc/draft-livingood-doh-implementation-risks-issues/04/ (see pages 41, 87).

[102] Vorsk LLC. *DNS Coffee*. (visited on September 26, 2024). 2016. URL: https://dns.coffee/ (see pages 31, 43).

[103] Similarweb LTD. *Top Websites Ranking*. (visited on October 7, 2024). Dec. 2024. URL: https://www.similarweb.com/top-websites/ (see page 47).

[104] Chaoyi Lu, Baojun Liu, Zhou Li, Shuang Hao, Hai-Xin Duan, Mingming Zhang, Chunying Leng, Ying Liu, Zaifeng Zhang, and Jianping Wu. **An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come?** In: *Proceedings of the Internet Measurement Conference, IMC 2019, Amsterdam, The Netherlands, October 21-23, 2019*. ACM, 2019, 22–35. DOI: 10.1145/3355369.3355580. URL: https://doi.org/10.1145/3355369.3355580 (see pages 13, 14, 16, 37, 52).

[105] Meng Luo, Liling Xin, Yepeng Yao, Zhengwei Jiang, Qiuyun Wang, and Wenchang Shi. **Who Are Querying For Me? Egress Measurement For Open DNS Resolvers**. In: *26th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2023, Rio de Janeiro, Brazil, May 24-26, 2023*. Ed. by Weiming Shen, Jean-Paul A. Barthès, Junzhou Luo, Adriana S. Vivacqua, Daniel Schneider, Cheng Xie, Jinghui Zhang, Haibin Zhu, Kunkun Peng, and Cláudia Lage Rebello da Motta. IEEE, 2023, 1544–1550. DOI: 10.1109/CSCWD57460.2023.10152616. URL: https://doi.org/10.1109/CSCWD57460.2023.10152616 (see page 35).

[106] Gordon Lyon. *Nmap: the Network Mapper*. (visited on September 26, 2024). 1997. URL: https://nmap.org/ (see page 32).

[107] Minzhao Lyu, Hassan Habibi Gharakheili, and Vijay Sivaraman. **A Survey on DNS Encryption: Current Development, Malware Misuse, and Inference Techniques**. *ACM Comput. Surv.* 55:8 (2023), 162:1–162:28. DOI: 10.1145/3547331. URL: https://doi.org/10.1145/3547331 (see pages 17, 37).

[108] Jonathan Magnusson, Moritz Müller, Anna Brunström, and Tobias Pulls. **A Second Look at DNS QNAME Minimization**. In: *Passive and Active Measurement - 24th International Conference, PAM 2023, Virtual Event, March 21-23, 2023, Proceedings*. Ed. by Anna Brunström, Marcel Flores, and Marco Fiore. Vol. 13882. Lecture Notes in Computer Science. Springer, 2023, 496–521. DOI: 10.1007/978-3-031-28486-1_21. URL: https://doi.org/10.1007/978-3-031-28486-1%5C_21 (see pages 113, 120).

[109] Keyu Man, Zhiyun Qian, Zhongjie Wang, Xiaofeng Zheng, Youjun Huang, and Haixin Duan. **DNS Cache Poisoning Attack Reloaded: Revolutions with Side Channels**. In: *CCS '20: 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, USA, November 9-13, 2020*. Ed. by Jay Ligatti, Xinming Ou, Jonathan Katz, and Giovanni Vigna. ACM, 2020, 1337–1350. DOI: 10.1145/3372297.3417280. URL: https://doi.org/10.1145/3372297.3417280 (see page 12).

[110] Jiarun Mao, Michael Rabinovich, and Kyle Schomp. **Assessing Support for DNS-over-TCP in the Wild**. In: *Passive and Active Measurement - 23rd International Conference, PAM 2022, Virtual Event, March 28-30, 2022,*

Proceedings. Ed. by Oliver Hohlfeld, Giovane Cesar Moreira Moura, and Cristel Pelsser. Vol. 13210. Lecture Notes in Computer Science. Springer, 2022, 487–517. DOI: 10.1007/978-3-030-98785-5_22. URL: https://doi.org/10.1007/978-3-030-98785-5%5C_22 (see pages 36, 49, 57).

[111] MaxMind Inc. *GeoLite2 Free Geolocation Data*. (visited on October 11, 2024). URL: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data/ (see pages 53, 64, 67, 118).

[112] Paul V. Mockapetris. **Domain names - concepts and facilities**. *RFC* 1034 (1987), 1–55. DOI: 10.17487/RFC1034. URL: https://doi.org/10.17487/RFC1034 (see pages 1, 7, 11, 12, 20).

[113] Paul V. Mockapetris. **Domain names - implementation and specification**. *RFC* 1035 (1987), 1–55. DOI: 10.17487/RFC1035. URL: https://doi.org/10.17487/RFC1035 (see pages 1, 7, 9, 10, 12, 18, 105).

[114] Giovane C. M. Moura, Sebastian Castro, Wes Hardaker, Maarten Wullink, and Cristian Hesselman. **Clouding up the Internet: how centralized is DNS traffic becoming?** In: *IMC '20: ACM Internet Measurement Conference, Virtual Event, USA, October 27-29, 2020*. ACM, 2020, 42–49. DOI: 10.1145/3419394.3423625. URL: https://doi.org/10.1145/3419394.3423625 (see page 1).

[115] Arvind Narayanan and Bendert Zevenbergen. **No Encore for Encore? Ethical Questions for Web-Based Censorship Measurement**. *SSRN Electronic Journal* (2015). ISSN: 1556-5068. DOI: 10.2139/ssrn.2665148. URL: http://dx.doi.org/10.2139/ssrn.2665148 (see page 58).

[116] Marcin Nawrocki, Pouyan Fotouhi Tehrani, Raphael Hiesgen, Jonas Mücke, Thomas C. Schmidt, and Matthias Wählisch. **On the interplay between TLS certificates and QUIC performance**. In: *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies, CoNEXT 2022, Roma, Italy, December 6-9, 2022*. Ed. by Giuseppe Bianchi and Alessandro Mei. ACM, 2022, 204–213. DOI: 10.1145/3555050.3569123. URL: https://doi.org/10.1145/3555050.3569123 (see pages 38, 119).

[117] Alexandra Nisenoff, Ranya Sharma, and Nick Feamster. **User Awareness and Behaviors Concerning Encrypted DNS Settings in Web Browsers**. In: *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*. Ed. by Joseph A. Calandrino and Carmela Troncoso. USENIX Association, 2023, 3117–3133. URL: https://www.usenix.org/conference/usenixsecurity23/presentation/nisenoff-awareness (see page 39).

[118] Daiyuu Nobori and Yasushi Shinjo. **VPN Gate: A Volunteer-Organized Public VPN Relay System with Blocking Resistance for Bypassing Government Censorship Firewalls**. In: *Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2014, Seattle, WA, USA, April 2-4, 2014*. Ed. by Ratul Mahajan and Ion Stoica. USENIX Association, 2014, 229–241. URL: https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/nobori (see page 118).

[119] OpenINTEL. *Active DNS Measurement Project*. (visited on September 26, 2024). URL: https://openintel.nl/ (see pages 31, 35, 43).

[120] John S. Otto, Mario A. Sánchez, John P. Rula, and Fabián E. Bustamante. **Content delivery and the natural evolution of DNS: remote dns trends, performance issues and alternative solutions**. In: *Proceedings of the 12th ACM SIGCOMM Internet Measurement Conference, IMC '12, Boston, MA, USA, November 14-16, 2012*. Ed. by John W. Byers, Jim Kurose, Ratul Mahajan, and Alex C. Snoeren. ACM, 2012, 523–536. DOI: 10.1145/2398776.2398831. URL: https://doi.org/10.1145/2398776.2398831 (see page 41).

[121] Linux Man Pages. *pipe(7)*. (visited on October 7, 2024). May 2024. URL: https://man7.org/linux/man-pages/man7/pipe.7.html (see page 48).

[122] Jeman Park, Aminollah Khormali, Manar Mohaisen, and Aziz Mohaisen. **Where Are You Taking Me? Behavioral Analysis of Open DNS Resolvers**. In: *49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2019, Portland, OR, USA, June 24-27, 2019*. IEEE, 2019, 493–504. DOI: 10.1109/DSN.2019.00057. URL: https://doi.org/10.1109/DSN.2019.00057 (see pages 35, 55).

[123] T. Reddy K. Tommy Pauly. **Discovery of Oblivious Services via Service Binding Records**. *RFC* 9540 (2024), 1–10. DOI: 10.17487/RFC9540. URL: https://doi.org/10.17487/RFC9540 (see pages 4, 25, 53).

[124] Tommy Pauly, Eric Kinnear, Christopher A. Wood, Patrick McManus, and Tommy Jensen. **Discovery of Designated Resolvers**. *RFC* 9462 (2023), 1–16. DOI: 10.17487/RFC9462. URL: https://doi.org/10.17487/RFC9462 (see pages 1, 2, 23–26, 63, 84, 87–89, 93, 96, 109).

[125] PeeringDB. *PeeringDB - The Interconnection Database*. (visited on October 11, 2024). URL: www.peeringdb.com (see pages 53, 67).

[126] Carlos Perez. *GitHub — darkoperator/dnsrecon: DNS Enumeration Script*. (visited on October 4, 2024). 2010. URL: https://github.com/darkoperator/dnsrecon/ (see pages 33, 43).

[127] David Plonka and Arthur W. Berger. **Temporal and Spatial Classification of Active IPv6 Addresses**. In: *Proceedings of the 2015 ACM Internet Measurement Conference, IMC 2015, Tokyo, Japan, October 28-30, 2015*. Ed. by Kenjiro Cho, Kensuke Fukuda, Vivek S. Pai, and Neil Spring. ACM, 2015, 509–522. DOI: 10.1145/2815675.2815678. URL: https://doi.org/10.1145/2815675.2815678 (see pages 48, 117).

[128] Jon Postel. **Domain Name System Structure and Delegation**. *RFC* 1591 (1994), 1–7. DOI: 10.17487/RFC1591. URL: https://doi.org/10.17487/RFC1591 (see page 7).

[129] Qualys, Inc. *Qualys SSL Labs - SSL Pulse*. (visited on September 12, 2024). URL: https://www.ssllabs.com/ssl-pulse/ (see pages 17, 119).

[130] Rapid7. *Rapid7 Open Data*. (visited on September 26, 2024). URL: https://opendata.rapid7.com/ (see page 30).

[131] Rapid7. *Rapid7 Research - Project Sonar*. (visited on September 26, 2024). URL: https://www.rapid7.com/research/project-sonar/ (see page 30).

[132] Yakov Rekhter, Bogert G. Moskowitz, Daniel Karrenberg, Geert Jan de Groot, and Eliot Lear. **Address Allocation for Private Internets**. *RFC* 1918 (1996), 1–9. DOI: 10.17487/RFC1918. URL: https://doi.org/10.17487/RFC1918 (see pages 25, 47).

[133] Eric Rescorla. **HTTP Over TLS**. *RFC* 2818 (2000), 1–7. DOI: 10.17487/RFC2818. URL: https://doi.org/10.17487/RFC2818 (see pages 17, 19).

[134] Eric Rescorla. **The Transport Layer Security (TLS) Protocol Version 1.3**. *RFC* 8446 (2018), 1–160. DOI: 10.17487/RFC8446. URL: https://doi.org/10.17487/RFC8446 (see pages 17, 108, 109).

[135] Eric Rescorla, Kazuho Oku, Nick Sullivan, and Christopher A. Wood. **TLS Encrypted Client Hello**. Internet-Draft draft-ietf-tls-esni-20. Work in Progress. Internet Engineering Task Force, Aug. 2024. 53 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-tls-esni/20/ (see page 12).

[136] Philipp Richter, Georgios Smaragdakis, David Plonka, and Arthur W. Berger. **Beyond Counting: New Perspectives on the Active IPv4 Address Space**. In: *Proceedings of the 2016 ACM on Internet Measurement Conference, IMC 2016, Santa Monica, CA, USA, November 14-16, 2016*. Ed. by Phillipa Gill, John S. Heidemann, John W. Byers, and Ramesh Govindan. ACM, 2016, 135–149. URL: http://dl.acm.org/citation.cfm?id=2987473 (see page 49).

[137] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. **A High-Performance, Scalable Infrastructure for Large-Scale Active DNS Measurements**. *IEEE J. Sel. Areas Commun.* 34:6 (2016), 1877–1888. DOI: 10.1109/JSAC.2016.2558918. URL: https://doi.org/10.1109/JSAC.2016.2558918 (see page 31).

[138] Roland van Rijswijk-Deij, Mattijs Jonker, Anna Sperotto, and Aiko Pras. **The Internet of Names: A DNS Big Dataset**. In: *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM 2015, London, United Kingdom, August 17-21, 2015*. Ed. by Steve Uhlig, Olaf Maennel, Brad Karp, and Jitendra Padhye. ACM, 2015, 91–92. DOI: 10.1145/2785956.2789996. URL: https://doi.org/10.1145/2785956.2789996 (see pages 31, 118).

[139] RIPE Network Coordination Centre. *RIPE Atlas*. (visited on September 27, 2024). URL: https://atlas.ripe.net/ (see page 30).

[140] Hans Christian Rudolph and Nils Grundmann. *Ciphersuite Info*. (visited on September 6, 2024). URL: https://ciphersuite.info/ (see page 108).

[141] Erik C. Rye and Dave Levin. **IPv6 Hitlists at Scale: Be Careful What You Wish For**. In: *Proceedings of the ACM SIGCOMM 2023 Conference, ACM SIGCOMM 2023, New York, NY, USA, 10-14 September 2023*. Ed. by Henning Schulzrinne, Vishal Misra, Eddie Kohler, and David A. Maltz. ACM, 2023, 904–916. DOI: 10.1145/3603269.3604829. URL: https://doi.org/10.1145/3603269.3604829 (see page 32).

[142] Stefan Santesson. **Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name**. *RFC* 4985 (2007), 1–10. DOI: 10.17487/RFC4985. URL: https://doi.org/10.17487/RFC4985 (see page 25).

[143] Steffen Sassalla. *DoE-Hunter*. (visited on October 8, 2024). June 2024. URL: https://github.com/steffsas/doe-hunter/ (see pages 4, 44, 47, 55, 157, 158).

[144] Steffen Sassalla. *hitlist-downloader*. (visited on October 8, 2024). June 2024. URL: https://github.com/steffsas/hitlist-downloader/ (see pages 48, 157).

[145] Steffen Sassalla. *miegk/dns - Pull Request - Add RFC 9540 to SVCBs to indicate Oblivious HTTPS is available*. (visited on October 22, 2024). June 2024. URL: https://github.com/miekg/dns/pull/1567/ (see pages 4, 53).

[146] Quirin Scheitle, Oliver Hohlfeld, Julien Gamba, Jonas Jelten, Torsten Zimmermann, Stephen D. Strowes, and Narseo Vallina-Rodriguez. **A Long Way to the Top: Significance, Structure, and Stability of Internet Top Lists**. In: *Proceedings of the Internet Measurement Conference 2018, IMC 2018, Boston, MA, USA, October 31 - November 02, 2018*. ACM, 2018, 478–493.

URL: https://dl.acm.org/citation.cfm?id=3278574 (see pages 32, 43, 44, 47, 59).

[147] Giovanni Schmid. **Thirty Years of DNS Insecurity: Current Issues and Perspectives**. *IEEE Commun. Surv. Tutorials* 23:4 (2021), 2429–2459. DOI: 10.1109/COMST.2021.3105741. URL: https://doi.org/10.1109/COMST.2021.3105741 (see pages 13, 15).

[148] Kyle Schomp, Tom Callahan, Michael Rabinovich, and Mark Allman. **On measuring the client-side DNS infrastructure**. In: *Proceedings of the 2013 Internet Measurement Conference, IMC 2013, Barcelona, Spain, October 23-25, 2013*. Ed. by Konstantina Papagiannaki, P. Krishna Gummadi, and Craig Partridge. ACM, 2013, 77–90. DOI: 10.1145/2504730.2504734. URL: https://doi.org/10.1145/2504730.2504734 (see pages 34, 35, 55).

[149] Haya Schulmann. **Pretty Bad Privacy: Pitfalls of DNS Encryption**. In: *Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES 2014, Scottsdale, AZ, USA, November 3, 2014*. Ed. by Gail-Joon Ahn and Anupam Datta. ACM, 2014, 191–200. DOI: 10.1145/2665943.2665959. URL: https://doi.org/10.1145/2665943.2665959 (see pages 15, 16).

[150] Ben Schwartz, Mike Bishop, and Erik Nygren. **Service Binding and Parameter Specification via the DNS (SVCB and HTTPS Resource Records)**. *RFC* 9460 (2023), 1–47. DOI: 10.17487/RFC9460. URL: https://doi.org/10.17487/RFC9460 (see pages 11, 12, 24, 36, 87–89, 93, 95).

[151] Benjamin Schwartz. **Service Binding Mapping for DNS Servers**. *RFC* 9461 (2023), 1–10. DOI: 10.17487/RFC9461. URL: https://doi.org/10.17487/RFC9461 (see pages 11, 87, 89, 94).

[152] Ranya Sharma, Nick Feamster, and Austin Hounsel. **Measuring the Availability and Response Times of Public Encrypted DNS Resolvers**. *CoRR* abs/2208.04999 (2022). DOI: 10.48550/ARXIV.2208.04999. arXiv: 2208.04999. URL: https://doi.org/10.48550/arXiv.2208.04999 (see page 37).

[153] Yaron Sheffer, Ralph Holz, and Peter Saint-Andre. **Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS)**. *RFC* 7457 (2015), 1–13. DOI: 10.17487/RFC7457. URL: https://doi.org/10.17487/RFC7457 (see page 21).

[154] Yaron Sheffer, Peter Saint-Andre, and Thomas Fossati. **Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)**. *RFC* 9325 (2022), 1–34. DOI: 10.17487/RFC9325. URL: https://doi.org/10.17487/RFC9325 (see page 18).

[155] Shodan. *Shodan*. (visited on September 26, 2024). URL: https://www.shodan.io/ (see pages 30, 57).

[156] Yuuki Takano, Ruo Ando, Takeshi Takahashi, Satoshi Uda, and Tomoya Inoue. **A measurement study of open resolvers and DNS server version**. In: *Internet Conference (IEICE)*. Vol. 21. 2013 (see pages 34, 35, 55).

[157] The Linux Foundation. *GitHub - prometheus/node_exporter: Exporter for machine metrics*. (visited on October 14, 2024). URL: https://github.com/prometheus/node_exporter/ (see page 54).

[158] The Linux Foundation. *GitHub - prometheus/prometheus: The Prometheus monitoring system and time series database*. (visited on October 14, 2024). URL: https://github.com/prometheus/prometheus (see page 54).

[159] Martin Thomson and Sean Turner. **Using TLS to Secure QUIC**. *RFC* 9001 (2021), 1–52. DOI: 10.17487/RFC9001. URL: https://doi.org/10.17487/RFC9001 (see pages 17, 19, 108).

[160] John Todd, Tommy Jensen, and Corey Mosher. **Encrypted DNS Server Redirection**. Internet-Draft draft-ietf-add-encrypted-dns-server-redirection-00. Work in Progress. Internet Engineering Task Force, July 2024. 13 pp. URL: https://datatracker.ietf.org/doc/draft-ietf-add-encrypted-dns-server-redirection/00/ (see pages 26, 27, 51).

[161] Olivier van der Toorn, Moritz Müller, Sara Dickinson, Cristian Hesselman, Anna Sperotto, and Roland van Rijswijk-Deij. **Addressing the challenges of modern DNS a comprehensive tutorial**. *Comput. Sci. Rev.* 45 (2022), 100469. DOI: 10.1016/J.COSREV.2022.100469. URL: https://doi.org/10.1016/j.cosrev.2022.100469 (see page 31).

[162] Andrea Tundis, Wojciech Mazurczyk, and Max Mühlhäuser. **A review of network vulnerabilities scanning tools: types, capabilities and functioning**. In: *Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Hamburg, Germany, August 27-30, 2018*. Ed. by Sebastian Doerr, Mathias Fischer, Sebastian Schrittwieser, and Dominik Herrmann. ACM, 2018, 65:1–65:10. DOI: 10.1145/3230833.3233287. URL: https://doi.org/10.1145/3230833.3233287 (see page 30).

[163] Andrea Tundis, Eric Marc Modo Nga, and Max Mühlhäuser. **An exploratory analysis on the impact of Shodan scanning tool on the network attacks**. In: *ARES 2021: The 16th International Conference on Availability, Reliability and Security, Vienna, Austria, August 17-20, 2021*. Ed. by Delphine Reinhardt and Tilo Müller. ACM, 2021, 39:1–39:10. DOI: 10.1145/3465481.3469197. URL: https://doi.org/10.1145/3465481.3469197 (see page 30).

[164] Tim Wicinski. **DNS Privacy Considerations**. *RFC* 9076 (2021), 1–22. DOI: 10.17487/RFC9076. URL: https://doi.org/10.17487/RFC9076 (see pages 1, 13, 14, 16, 22, 98).

[165] Yunpeng Xing, Chaoyi Lu, Baojun Liu, Haixin Duan, Junzhe Sun, and Zhou Li. **Yesterday Once More: Global Measurement of Internet Traffic Shadowing Behaviors**. In: *Proceedings of the 2024 ACM on Internet Measurement Conference, IMC 2024, Madrid, Spain, November 4-6, 2024*. Ed. by Narseo Vallina-Rodriguez, Guillermo Suarez-Tangil, Dave Levin, and Cristel Pelsser. ACM, 2024, 230–240. DOI: 10.1145/3646547.3689023. URL: https://doi.org/10.1145/3646547.3689023 (see pages 4, 113, 119).

[166] Luciano Zembruzki, Arthur Selle Jacobs, and Lisandro Zambenedetti Granville. **On the Consolidation of the Internet Domain Name System**. In: *IEEE Global Communications Conference, GLOBECOM 2022, Rio de Janeiro, Brazil, December 4-8, 2022*. IEEE, 2022, 2122–2127. DOI: 10.1109/GLOBECOM48099.2022.10001425. URL: https://doi.org/10.1109/GLOBECOM48099.2022.10001425 (see page 31).

[167] Johannes Zirngibl, Patrick Sattler, and Georg Carle. **A First Look at SVCB and HTTPS DNS Resource Records in the Wild**. In: *IEEE European Symposium on Security and Privacy, EuroS&P 2023 - Workshops, Delft, Netherlands, July 3-7, 2023*. IEEE, 2023, 470–474. DOI: 10.1109/EUROSPW59978.2023.00058. URL: https://doi.org/10.1109/EuroSPW59978.2023.00058 (see page 36).

[168] zoomeye.ai. *ZoomEye*. (visited on September 26, 2024). URL: https://www.zoomeye.ai/ (see pages 30, 57).

Acronyms

ADN An Authentication Domain Name (ADN) is a domain name that is used to authenticate endpoints, particularly in the context of DoE protocols, where the ADN is associated with the validation of the server's identity using the DNS server's TLS certificate. 4, 100, 119

AEAD Authenticated Encryption with Associated Data (AEAD) is an encryption method that provides both confidentiality and integrity in one operation. 21

AS An Autonomous System (AS) is a collection of IP networks and routers under the control of a single organization that presents a common routing policy to the Internet. 2, 5, 32, 35, 37, 44, 47, 53, 54, 61, 67–70, 77–81, 83–85, 88, 89, 95, 97, 101–104, 106, 110–115, 117, 118, 122, 158, 159, 175–177

CA A Certificate Authority (CA) is a trusted entity in a PKI responsible for issuing, validating, and revoking digital certificates that verify the identities of entities (such as individuals, organizations, or devices) and enable secure communication over networks. 17, 18, 47, 104, 107

CDN A Content Delivery Network (CDN) is a distributed network of servers that work together to deliver web content to users from geographically closer locations, improving load times, performance, and reliability. 37, 41, 95

CT The Certificate Transparency (CT) is an open framework designed to monitor and audit TLS certificates, enabling the detection of misissued or malicious certificates by logging them in publicly accessible, tamper-evident logs. 32, 35

DDoS A Distributed Denial of Service (DDoS) attack is a type of DoS attack where multiple compromised or controlled systems (often part of a botnet) are used to flood a target with overwhelming traffic, disrupting normal service and making it unavailable to legitimate users. 36

DDR Discovery of Designated Resolvers (DDR) is a protocol that enables clients to automatically discover and securely connect to encrypted DNS resolvers via DoE, improving security and privacy for DNS queries and responses. 1–5, 7, 12, 23–27, 31, 33, 40, 41, 43–46, 48–59, 61–64, 71–101, 104–110, 112, 114, 115, 117–123, 157, 161–163, 165–168, 170–179

DHCP The Dynamic Host Configuration Protocol (DHCP) is a network management protocol that automatically assigns IP addresses and other network configuration settings like DNS recursive resolvers to devices on a network, allowing them to communicate efficiently. 8, 22, 41, 97, 119

DNR Discovery of Network-designated Resolvers (DNR) refers to the discovery of encrypted DNS resolvers in local networks using DHCP and IPv6 Router Advertisement options to provide hosts with resolver configuration information. 119

DNS The Domain Name System is a hierarchical and distributed database to map domain names to IP addresses among other resources. 1–5, 7–16, 18–24, 26, 27, 29–41, 43, 44, 46–48, 50–53, 55, 57–59, 61–82, 84, 85, 87–92, 94–99, 103–114, 117–123, 145–155, 161–163, 172–179

DNSKEY A DNS Key (DNSKEY) is a DNSSEC record that contains a public cryptographic key used to verify the authenticity of DNS responses and validate RRSIG signatures within a DNS zone. 13, 148

DNSSEC DNS Security Extensions (DNSSEC) is a suite of security protocols that add authentication and data integrity to DNS, helping to protect against threats like DNS spoofing and cache poisoning. 13–15, 17, 26, 44, 45, 51, 52, 110, 147, 148, 151, 153, 173

Do53 DNS over UDP 53 refers to the traditional method of transmitting DNS queries and responses over the User Datagram Protocol (UDP) using port 53. 9, 19, 20, 22, 26, 37, 38, 47, 89, 94, 98, 108, 113, 119, 123, 148, 159

DoE DNS over Encryption (DoE) is a broad term that refers to the practice of transmitting DNS queries over a secure channel, such as HTTPS, TLS or QUIC. Secure in this context means that the communication ensures authenticity, integrity, and confidentiality of the DNS queries and responses. 1–5, 7, 16, 18, 20–24, 27, 37–41, 43–46, 50–54, 58, 59, 71, 82, 84, 85, 87–115, 117, 119–123, 145–148, 157, 170–172, 174–176, 178, 179

DoH DNS over HTTPS is a DoE protocol that securely transmits DNS queries and responses using the HTTPS transport protocol. 1, 5, 17–21, 24–26, 37–40, 46, 47, 52, 90–95, 98, 100–104, 106–108, 114, 122, 123, 151, 159, 170, 173

DoQ DNS over QUIC is a protocol DoE that securely transmits DNS queries and responses using the QUIC transport protocol on top of UDP. 1, 4, 5, 17, 19, 20, 38, 40, 90–94, 100–104, 106–108, 119, 122, 123, 170

DoS Denial of Service (DoS) is an attempt to disrupt the normal functioning of a service or system by making it unavailable to legitimate users, either by overwhelming it with excessive traffic or resource demands, or by exploiting vulnerabilities — such as downgrading secure connections — which force clients into a non-operational state. 22, 146

DoT DNS over TLS is a DoE protocol that securely transmits DNS queries and responses using TLS. 1, 5, 17–20, 22, 24, 37–40, 46, 47, 52, 90, 92, 93, 100–102, 104, 106–108, 114, 122, 123, 159, 170

DoTCP53 DNS over TCP refers to the method of transmitting DNS queries and responses over DNS using port 53, usually in the case as a fallback mechanism for Do53. 9, 36

DS A Delegation Signer (DS) is a DNSSEC record in the parent DNS zone, containing a cryptographic hash of the child zone's DNSKEY record. This helps to establish a chain of trust between the parent and child zones by verifying the authenticity of the child zone's DNSSEC keys. 13

ECH Encrypted ClientHello (ECH) is a TLS extension that enhances privacy by encrypting the "ClientHello" message in TLS handshakes, preventing network observers from seeing the domain name a client is trying to connect to, improving security compared to the previous ESNI (Encrypted Server Name Indication) mechanism. 12

EDNS0 The Extension Mechanisms for DNS (EDNS0) are extensions of DNS that enhance its capabilities by allowing larger message sizes than 512 bytes, additional features and extensions, and better performance without changing the basic DNS protocol. 10, 51

EDSR Encrypted DNS Server Redirection (EDSR) is an IETF draft that defines a mechanism enabling an encrypted DNS server to redirect clients to a more suitable encrypted DNS server, optimizing user privacy and performance by facilitating server transitions within encrypted DNS services. 26, 27, 51, 120, 173

FQDN A Fully-Qualified Domain Name (FQDN) is a complete and specific domain name that uniquely identifies a host or server on the internet. An FQDN includes both the hostname and the domain name, and it specifies the exact location of the entity within the DNS hierarchy. 7, 8, 11, 14, 24, 25, 34

HPKE Hybrid Public Key Encryption (HPKE) is a cryptographic scheme that combines both public key encryption and symmetric encryption to provide confidentiality and integrity protection of encrypted data. It has performance benefits than just using public-key encryption. 20

IANA The Internet Assigned Numbers Authority (IANA) is responsible for coordinating key elements of the Internet's infrastructure, including the allocation of IP addresses, management of the DNS root zone, and assignment of protocol parameters. 11, 95

IETF The Internet Engineering Task Force (IETF) is an international organization that develops and promotes voluntary internet standards, particularly those related to TCP/IP protocols, through open collaboration and consensus among industry professionals, researchers, and engineers. 1, 5, 14–16, 20, 23, 26, 87, 108, 114, 119, 151

IoT Internet-of-Things (IoT) refers to a network of interconnected devices that collect, exchange, and act on data through sensors and software, enabling automation and remote control over the internet. 1, 9, 30

ISP An Internet Service Provider (ISP) is a company that provides individuals and organizations with access to the Internet and related services such as web hosting and email. 22, 41, 53, 68, 70, 85, 87, 95, 98, 118, 122, 123, 177

MAC A Message Authentication Code (MAC) is a cryptographic construct used to ensure data integrity and authenticity by combining a secret key with the message to produce a unique tag. This tag is verified by the recipient using the same secret key, ensuring that the message has not been tampered with and originates from a legitimate source. 104

mTLS Mutual Transport Layer Security (mTLS) is a protocol that ensures both the client and server authenticate each other's identities during a TLS handshake. 5, 99, 107, 108

NRR A Non-Recursive Resolver (NRR) is a DNS server that provides responses only from cached or authoritative data without performing recursive resolving tasks. 11, 62, 174

NSEC A Next Secure (NSEC) record is a DNSSEC record used to prove the non-existence of a DNS name or type by linking one domain name to the next in a canonical order, providing authenticated denial of existence and preventing certain types of attacks such as zone enumeration. 13

ODoH Oblivious DNS over HTTPS (ODoH) is an extension of DoH that enhances privacy by separating the client issuing the DNS query from the server resolving it. It introduces an additional layer of encryption and routing through a proxy, ensuring that the DNS resolver cannot identify the client sending the query. 4, 5, 20, 21, 25, 53, 95, 96, 173

PKI The Public Key Infrastructure (PKI) is a framework of policies, procedures, and technologies used to manage, distribute, and validate digital certificates and public-private key pairs, providing a foundation for secure communication, authentication, and data encryption over insecure networks. 15, 17, 145

PM Pervasive Monitoring (PM) is the large-scale, often hidden, surveillance that involves collecting protocol data such as application content or metadata (e.g., headers) through techniques like wiretapping and traffic analysis. The IETF views PM as a threat to the privacy of internet users and organizations [48]. 14, 15

PTR A Pointer Record (PTR) in DNS maps an IP address to a domain name, enabling reverse DNS lookups (rDNS) to identify the domain associated with a given IP. 44

QNAME A Query Name (QNAME) is the requested name in a DNS request, representing the domain name for which the client is seeking resolution, such as converting it into an IP address or other DNS record. 11, 16, 24, 34, 46, 111, 112, 152

QNAME Min. QNAME Minimization is a DNS privacy technique that reduces the amount of information sent in DNS queries by only revealing the minimum necessary part of the query to each upstream DNS server in the resolution chain, thus limiting exposure of full domain names and enhancing user privacy. 11, 16, 113, 120

QUIC QUIC is a transport protocol developed by Google, designed to improve the speed, reliability, and security of internet communications by combining features of TCP, TLS, and HTTP/2 while operating over UDP. It reduces latency and enhances performance, especially on unreliable networks. 17, 19, 20, 35, 38, 45, 90, 103, 108, 119, 147, 148

RCODE A DNS Return Code (RCODE) in the DNS message header indicates the outcome of a DNS query, specifying whether the request was successful or if an error occurred. These codes are essential for diagnosing and troubleshooting DNS response issues in network communication. 11, 20, 36, 51, 58, 61–66, 71, 104, 105, 107, 108, 114, 123, 162–164, 174, 178

ResR A DNS Resource Record (ResR) is a database entry in the DNS that provides information about a domain, mail server, or other domain-specific data. 11–13, 23, 24, 30–34, 36, 43, 46, 47, 50–52, 59, 85, 96, 114, 155, 173

RR A recursive DNS resolver (RR) is a server that receives DNS queries from clients and resolves them by querying other DNS servers to obtain the requested information. 8–10, 12, 15, 16, 20, 21, 24, 31, 33, 36, 40, 41, 57, 62, 63, 67, 70, 73, 74, 77, 87–89, 92, 94–96, 105, 174

RRSIG A Resource Record Signature (RRSIG) is a DNSSEC record used to store digital signatures that authenticate DNS data, ensuring its integrity and authenticity. 13, 147

RTT The Round-Trip Time (RTT) is the time it takes for a signal to travel from a source to a destination and back, commonly used to measure network latency. 25, 38, 56–58, 103, 104, 174, 176

SAN The subjectAltName TLS extension (SAN) is an extension field in X.509 certificates used in TLS that allows a certificate to specify additional hostnames, IP addresses, or email addresses that the certificate applies to, enabling the use of multiple identities. 25, 26, 96–98, 123

SLD A Sub-Level Domain (SLD) is the portion of a domain name that is directly to the left of the TLD in the DNS hierarchy. It represents the specific name of an entity or organization within the broader domain space. 8, 158

SNI Server Name Indication (SNI) is an extension to the TLS protocol that allows a client to specify the hostname it is trying to connect to during the handshake, enabling the server to present the correct certificate for the requested domain. 45

SUDN A Special Use Domain Name (SUDN) is a domain name reserved for specific technical purposes, which is not intended for general use in the global DNS. For example, one such SUDN is `localhost` which resolves to the loopback IP address (`127.0.0.1`). 11, 24, 51

SVCB A Service Binding and Parameter (SVCB) record is a DNS record that provides clients with information on how to connect to a service, including protocol preferences, alternative endpoints, and security features, improving performance, privacy, and flexibility over traditional DNS records. 4, 11, 12, 23, 24, 26, 30, 31, 36, 43, 50, 51, 53, 85, 87–89, 92–96, 105, 177

TCP The Transmission Control Protocol (TCP) is a reliable, connection-oriented communication protocol that ensures the accurate and sequential delivery of data between devices on a network by providing error-checking, retransmission, and flow control mechanisms. 10, 18, 19, 36, 40, 58, 150

TFO TCP Fast Open (TFO) is a TCP extension that reduces connection setup latency by allowing data to be sent during the initial handshake phase, improving the performance of short-lived connections. 38

TLD A Top-Level Domain (TLD) is the highest level of the domain name hierarchy in the DNS. It appears at the end of a domain name, following the final (optional) dot. 8, 16, 31, 153, 158

TLS Transport Layer Security (TLS) is a cryptography protocol designed to provide secure communication over a computer network by ensuring confidentiality, integrity, and authenticity of data exchanged between two endpoints, commonly used in securing web traffic through HTTPS. 5, 15, 17–19, 21, 22, 25, 29, 30, 35, 37, 40, 43, 45, 47, 96–99, 104, 106–110, 113, 119, 123, 145, 147, 148, 150, 153, 158

TTL The Time-To-Live (TTL) is the amount of time (in seconds) that a DNS ResR is cached by a resolver before it must be refreshed from the authoritative server. 11, 13

UDP The User Datagram Protocol (UDP) is a lightweight communication protocol used in network systems that allows fast, connectionless data transmission without error-checking or re-transmission mechanisms, making it ideal for time-sensitive applications and protocols like DNS. 9, 10, 13, 19, 20, 30, 40, 43, 44, 47, 51, 55, 57, 58, 117, 157, 174

v4NRR An IPv4 Non-Recursive Resolver (v4NRR) is a non-recursive DNS resolver that supports the IPv4 protocol. 63, 64, 67, 68, 73, 177

v4RR An IPv4 Recursive Resolver (v4RR) is a recursive DNS resolver that supports the IPv4 protocol. 63, 64, 67, 68, 70, 73, 177

v6NRR An IPv6 Non-Recursive Resolver (v6NRR) is a non-recursive DNS resolver that supports the IPv6 protocol. 63, 64, 68, 69, 73, 177

v6RR An IPv6 Recursive Resolver (v6RR) is a recursive DNS resolver that supports the IPv6 protocol. 63, 68–70, 73, 177

VP A Vantage Point (VP) refers to a specific location or node in a network from which data is collected or measurements are performed, offering a particular perspective on network behavior, performance, or traffic. 4, 30, 35, 37–40, 47, 50, 52, 54, 57, 103, 104, 117, 118

A Reproducibility

A.1 Measurement Architecture

In this chapter, we describe the network infrastructure, hardware, and software used to conduct our measurements.

Virtualization Except for *ZMap*, which runs natively on one of the worker nodes, all other applications are containerized using *Docker*, version *27.4.0*. We orchestrate the containers with *Docker Compose* (*v2.31.0*). The public repository of `DoE-Hunter` [143] includes four Docker compose files: two for the worker nodes, one for the backend, which hosts *MongoDB* and the message broker system *Apache Kafka*, and one for the monitoring node running *Prometheus* and *Grafana*.

Worker Nodes Each worker node has 8 vCPUs and 16 GB of memory. Although CPU utilization is low, memory usage is high due to the large volume of data stored in memory. Both nodes connect to the Internet via a 1 Gbps up/downlink, have IPv6 enabled, and possess two public IP addresses each, for IPv4 and IPv6. The firewalls are configured to be stateless for UDP, which prevents buffer overflow issues within the firewall when sending datagrams at a high rate (approximately 3.7B packets within one day).

One worker node runs *ZMap* natively to scan the IPv4 address space semi-weekly and download the latest data from the *IPv6 Hitlist Service* [52] using *hitlist-downloader* [144]. This node then produces scans to the *Apache Kafka* message broker. The second worker node handles scans from the second and third measurement stages (see Section 4.1.4 and Section 4.1.5), such as DDR discoveries, DoE probes, and other scans. These tasks run fully inside *Docker* containers, whose images are available in the *DoE-Hunter* repository [143]. Additionally, both nodes are connected to the same local network as the backend node.

Measurement Hint To provide transparency for other network operators, both worker nodes host a measurement hint website. This site informs

network operators about our measurements and can be accessed via the public IPv4 and IPv6 addresses attached to the worker nodes. The site is hosted within a dockerized *nginx* web server. Additionally, we host the same site on both authoritative name servers, accessible through a domain name (`www.measurement.raiun.de`) and through the IP addresses of the authoritative name servers.

The website is a plain *HTML* page supporting TLS to adhere to current web standards. We use *certbot* [44] and *Let's Encrypt* [46] to deploy the necessary certificates.

Network We conducted the scans from an educational network within AS *8881* (*Versatel, DE*), using dedicated public IPv4 and IPv6 addresses assigned to each worker node. The network provides a 1 Gbps up/downlink.

Backend Node The backend node is a virtual machine with 32 vCPUs and 62 GB of memory. It stores and processes measurement data. This node is not connected to the Internet but resides on the same local network as the worker nodes and monitoring node.

On the backend node, we run *MongoDB* (image `mongo:8.0.4`) and *Apache Kafka* (image `confluentinc/cp-kafka:7.8.0`). We also deploy *Zookeeper* (image `confluentinc/cp-zookeeper:7.8.0`) to coordinate *Kafka* operations and broker nodes. For visualizing topics and messages in Kafka, we use *KafkaUI* using the image `provectuslabs/kafka-ui:v0.7.2`.

Monitoring Node To monitor the network and nodes, we use a dedicated monitoring node with 4 vCPUs and 8 GB of memory. This node is connected to the same local network as the worker and backend nodes.

We deploy *Prometheus* and *Grafana* as *Docker* containers, using the images `prom/prometheus:v3.0.1` and `grafana/grafana-enterprise:11.4.0`. To collect hardware, container, and network metrics, we run *cadvisor* and *node_exporter* on every node, using the images `gcr.io/cadvisor/cadvisor:v0.47.2` and `quay.io/prometheus/node-exporter:v1.8.2`, respectively. We created a custom *Grafana* dashboard for visualization, which is available in the *DoE-Hunter* repository [143].

Authoritative Name Servers To operate an authoritative name server as an SLD under the *.de* TLD, the *DENIC* requires two topologically separate

networks attached to the name servers. We use two virtual machines provided by *Netcup GmbH*: one in Nuremberg, Germany, and the other in Vienna, Austria (AS *197540*, *NETCUP-AS netcup GmbH, DE*).

Both virtual machines have 6 vCPUs, 8 GB of memory, and a 1 Gbps up/downlink. We use *bind9* [79] within *Docker* containers, running on *Ubuntu* with the image version `9.18-22.04_beta` [12]. As described in Section 4.1.2, we run for each protocol (Do53, DoT, and DoH/2) its own docker container, to separate query and error logs. IPv6 is enabled, and the *Docker* daemon and the firewall are configured to forward IPv6 traffic to the containers on the ports 53, 853 and 443. Additionally, we enable full logging in *bind9* to collect detailed logs.

We deploy certificates via *certbot* [44] and *Let's Encrypt* [46] to support DoT and DoH/2. The `NS` records pointing from the parent zone to ours include IPv4 and IPv6 addresses for both name servers.

To enhance measurement visibility and traceability, we deploy the measurement hint on the two name servers (see `www.ns1.raiun.de` and `www.ns2.raiun.de`).

Scheduling We use *cron* to schedule and initiate *ZMap* scans semi-weekly. For IPv6, the *hitlist-downloader* includes its own scheduler, which automatically attempts to download the latest data daily from the *IPv6 Hitlist Service*.

A.2 Analysis

For the analysis of the collected data, we use *Python* (version *3.12.3*) and *Jupyter Notebooks*. Data manipulation is performed using the *pandas* library, while visualizations are created with the *matplotlib* library. The complete analysis code, including a list of all required libraries (`requirements.txt`), is available in the Data-Intensive Internet Computing Chair's GitLab repository[7].

We distinguish between two types of notebooks for analysis: those used for data retrieval from the *MongoDB* and data enrichment (file names ending with `_analysis`), and those dedicated to generating plots and statistics (file names ending with `_plots` or `_statistics`). We cache the data from the data retrieval notebooks in *pickle* files to speed up the analysis process.

For data analysis, we recommend a system with around 64 GB memory, as the uncompressed data within the *MongoDB* exceeds 1.5 TB.

7 See https://gitlab.hpi.de/data-intensive-internet-computing/master-students

B Additional Material

B.1 Discovered DNS Server Figures

Table B.1: Discovered IPv4 (left) and IPv6 (right) DNS server during the second methodology stage (DDR discovery).

Date	# IPs Scanned	# DNS Servers	# Timeouts
11.11.24	31,929,925	4,235,660 (13.27%)	27,694,265 (86.73%)
07.11.24	32,341,002	4,217,935 (13.04%)	28,123,067 (86.96%)
06.11.24	32,462,201	4,249,871 (13.09%)	28,212,330 (86.91%)
31.10.24	31,674,069	4,179,010 (13.19%)	27,495,059 (86.81%)
28.10.24	31,206,556	4,395,140 (14.08%)	26,811,416 (85.92%)
26.10.24	31,459,269	4,358,404 (13.85%)	27,100,865 (86.15%)
21.10.24	28,954,127	4,187,568 (14.46%)	24,766,559 (85.54%)
17.10.24	32,605,171	4,409,239 (13.52%)	28,195,932 (86.48%)
10.10.24	32,024,480	4,119,012 (12.86%)	27,905,468 (87.14%)
07.10.24	31,122,416	4,026,622 (12.94%)	27,095,794 (87.06%)
03.10.24	31,467,690	4,094,823 (13.01%)	27,372,867 (86.99%)
30.09.24	30,741,945	4,060,793 (13.21%)	26,681,152 (86.79%)
26.09.24	30,973,173	3,911,408 (12.63%)	27,061,765 (87.37%)
23.09.24	30,898,243	3,902,188 (12.63%)	26,996,055 (87.37%)
20.09.24	31,335,512	3,966,671 (12.66%)	27,368,841 (87.34%)
16.09.24	31,136,880	3,966,855 (12.74%)	27,170,025 (87.26%)
11.09.24	31,032,449	3,908,456 (12.59%)	27,123,993 (87.41%)
05.09.24	26,759,342	3,905,359 (14.59%)	22,853,983 (85.41%)
29.08.24	26,961,658	3,956,014 (14.67%)	23,005,644 (85.33%)
26.08.24	26,615,529	3,956,879 (14.87%)	22,658,650 (85.13%)
22.08.24	26,105,411	3,938,641 (15.09%)	22,166,770 (84.91%)
19.08.24	26,392,421	3,934,748 (14.91%)	22,457,673 (85.09%)
12.08.24	26,629,148	3,940,854 (14.80%)	22,688,294 (85.20%)
08.08.24	26,755,866	4,004,361 (14.97%)	22,751,505 (85.03%)
12.07.24	27,060,938	4,012,658 (14.83%)	23,048,280 (85.17%)

Date	# IPs Scanned	# DNS Servers	# Timeouts
30.10.24	350,558	248,500 (70.89%)	102,058 (29.11%)
29.10.24	352,008	262,633 (74.61%)	89,375 (25.39%)
09.10.24	338,814	278,504 (82.20%)	60,310 (17.80%)
30.09.24	341,692	279,472 (81.79%)	62,220 (18.21%)
24.09.24	314,592	265,587 (84.42%)	49,005 (15.58%)
16.09.24	314,661	272,896 (86.73%)	41,765 (13.27%)
01.09.24	297,494	255,501 (85.88%)	41,993 (14.12%)
27.08.24	301,646	257,474 (85.36%)	44,172 (14.64%)
21.07.24	419,064	388,788 (92.78%)	30,276 (7.22%)
14.07.24	473,976	356,598 (75.24%)	117,378 (24.76%)

Table B.2: Breakdown of IPv4 (DDR) DNS response RCODEs of *recursive resolvers* for each run. The percentage shows the proportion of individual RCODEs in relation to the others from the same run.

Date	Total	NOERR	SERVFAIL	NXDOMAIN	REFUSED	NOAUTH	Others
11.11.24	1,175,984	373,044 (31.72%)	115,604 (9.83%)	587,382 (49.95%)	68,407 (5.82%)	1 (0.00%)	31,546 (2.68%)
07.11.24	1,114,468	387,461 (34.77%)	116,565 (10.46%)	510,793 (45.83%)	67,470 (6.05%)	0 (0.00%)	32,179 (2.89%)
06.11.24	1,119,621	387,413 (34.60%)	116,361 (10.39%)	515,619 (46.05%)	67,775 (6.05%)	0 (0.00%)	32,453 (2.90%)
31.10.24	1,096,029	398,513 (36.36%)	92,430 (8.43%)	508,299 (46.38%)	65,675 (5.99%)	1 (0.00%)	31,111 (2.84%)
28.10.24	1,126,967	386,063 (34.26%)	125,513 (11.14%)	517,636 (45.93%)	66,828 (5.93%)	0 (0.00%)	30,927 (2.74%)
26.10.24	1,112,219	385,801 (34.69%)	120,813 (10.86%)	509,026 (45.77%)	65,701 (5.91%)	1 (0.00%)	30,877 (2.78%)
21.10.24	1,126,646	385,661 (34.23%)	115,259 (10.23%)	526,884 (46.77%)	66,441 (5.90%)	3 (0.00%)	32,398 (2.88%)
17.10.24	1,126,137	385,425 (34.23%)	115,567 (10.26%)	526,674 (46.77%)	66,100 (5.87%)	2 (0.00%)	32,369 (2.87%)
10.10.24	1,112,087	385,460 (34.66%)	110,859 (9.97%)	518,921 (46.66%)	64,537 (5.80%)	0 (0.00%)	32,310 (2.91%)
07.10.24	1,080,375	383,836 (35.53%)	102,985 (9.53%)	498,174 (46.11%)	63,660 (5.89%)	0 (0.00%)	31,720 (2.94%)
03.10.24	1,100,461	377,292 (34.28%)	110,369 (10.03%)	521,382 (47.38%)	60,004 (5.45%)	0 (0.00%)	31,414 (2.85%)
30.09.24	1,099,167	371,315 (33.78%)	111,253 (10.12%)	522,102 (47.50%)	63,902 (5.81%)	1 (0.00%)	30,594 (2.78%)
26.09.24	1,086,012	367,623 (33.85%)	109,368 (10.07%)	514,871 (47.41%)	63,587 (5.86%)	3 (0.00%)	30,560 (2.81%)
23.09.24	1,076,270	372,211 (34.58%)	108,846 (10.11%)	503,227 (46.76%)	61,533 (5.72%)	0 (0.00%)	30,453 (2.83%)
20.09.24	1,091,229	376,358 (34.49%)	110,165 (10.10%)	509,412 (46.68%)	64,126 (5.88%)	1 (0.00%)	31,167 (2.86%)
16.09.24	1,082,543	386,055 (35.66%)	97,204 (8.98%)	504,706 (46.62%)	63,299 (5.85%)	1 (0.00%)	31,278 (2.89%)
11.09.24	1,086,401	371,629 (34.21%)	109,264 (10.06%)	512,157 (47.14%)	61,995 (5.71%)	1 (0.00%)	31,355 (2.89%)
05.09.24	1,089,877	377,281 (34.62%)	109,766 (10.07%)	507,688 (46.58%)	64,122 (5.88%)	2 (0.00%)	31,018 (2.85%)
29.08.24	1,091,641	370,259 (33.92%)	108,242 (9.92%)	515,370 (47.21%)	66,907 (6.13%)	2 (0.00%)	30,861 (2.83%)
26.08.24	1,083,356	375,324 (34.64%)	103,350 (9.54%)	507,155 (46.81%)	66,765 (6.16%)	2 (0.00%)	30,760 (2.84%)
22.08.24	1,086,202	377,820 (34.78%)	107,039 (9.85%)	503,612 (46.36%)	67,060 (6.17%)	1 (0.00%)	30,670 (2.82%)
19.08.24	1,056,534	371,322 (35.15%)	105,966 (10.03%)	484,657 (45.87%)	64,183 (6.07%)	2 (0.00%)	30,404 (2.88%)
12.08.24	1,048,212	365,737 (34.89%)	106,578 (10.17%)	485,216 (46.29%)	59,817 (5.71%)	1 (0.00%)	30,863 (2.94%)
08.08.24	1,089,228	385,359 (35.38%)	92,862 (8.53%)	524,105 (48.12%)	55,677 (5.11%)	1 (0.00%)	31,224 (2.87%)
12.07.24	1,072,766	365,501 (34.07%)	104,552 (9.75%)	516,104 (48.11%)	55,274 (5.15%)	2 (0.00%)	31,333 (2.92%)

Table B.3: Breakdown of IPv6 (DDR) DNS response RCODEs of *recursive resolvers* for each run. The percentage shows the proportion of individual RCODEs in relation to the others from the same run.

Date	Total	NOERR	SERVFAIL	NXDOMAIN	REFUSED	NOAUTH	Others
30.10.24	60,368	9,863 (16.34%)	922 (1.53%)	42,934 (71.12%)	6,560 (10.87%)	0 (0.00%)	89 (0.15%)
29.10.24	46,357	9,113 (19.66%)	873 (1.88%)	30,535 (65.87%)	5,765 (12.44%)	0 (0.00%)	71 (0.15%)
09.10.24	65,534	9,218 (14.07%)	904 (1.38%)	48,880 (74.59%)	6,451 (9.84%)	0 (0.00%)	81 (0.12%)
30.09.24	62,987	9,498 (15.08%)	914 (1.45%)	46,013 (73.05%)	6,466 (10.27%)	0 (0.00%)	96 (0.15%)
24.09.24	42,963	8,929 (20.78%)	921 (2.14%)	26,965 (62.76%)	6,059 (14.10%)	0 (0.00%)	89 (0.21%)
16.09.24	44,639	8,333 (18.67%)	947 (2.12%)	28,921 (64.79%)	6,359 (14.25%)	0 (0.00%)	79 (0.18%)
01.09.24	36,973	9,215 (24.92%)	675 (1.83%)	22,809 (61.69%)	4,192 (11.34%)	0 (0.00%)	82 (0.22%)
27.08.24	39,410	9,161 (23.25%)	918 (2.33%)	24,967 (63.35%)	4,270 (10.83%)	0 (0.00%)	94 (0.24%)
21.07.24	104,838	12,561 (11.98%)	1,213 (1.16%)	82,620 (78.81%)	8,232 (7.85%)	0 (0.00%)	212 (0.20%)
14.07.24	104,265	11,601 (11.13%)	1,126 (1.08%)	83,488 (80.07%)	7,801 (7.48%)	0 (0.00%)	249 (0.24%)

Table B.4: Breakdown of IPv4 (DDR) DNS response RCODEs of *non-recursive resolvers* for each run. The percentage shows the proportion of individual RCODEs in relation to the others from the same run.

Date	Total	NOERR	SERVFAIL	NXDOMAIN	REFUSED	NOAUTH	Others
11.11.24	3,059,676	162,941 (5.33%)	159,444 (5.21%)	406,457 (13.28%)	2,162,556 (70.68%)	164,333 (5.37%)	3,945 (0.13%)
07.11.24	3,103,467	163,158 (5.26%)	159,634 (5.14%)	406,578 (13.10%)	2,205,799 (71.08%)	164,339 (5.30%)	3,959 (0.13%)
06.11.24	3,130,250	163,451 (5.22%)	159,884 (5.11%)	407,944 (13.03%)	2,230,597 (71.26%)	164,406 (5.25%)	3,968 (0.13%)
31.10.24	3,082,981	164,027 (5.32%)	159,067 (5.16%)	398,251 (12.92%)	2,194,649 (71.19%)	162,957 (5.29%)	4,030 (0.13%)
28.10.24	3,268,173	169,151 (5.18%)	158,318 (4.84%)	398,909 (12.21%)	2,375,290 (72.68%)	162,442 (4.97%)	4,063 (0.12%)
26.10.24	3,246,185	167,879 (5.17%)	157,444 (4.85%)	397,980 (12.26%)	2,357,483 (72.62%)	161,371 (4.97%)	4,028 (0.12%)
21.10.24	3,060,922	163,277 (5.33%)	158,979 (5.19%)	378,626 (12.37%)	2,191,844 (71.61%)	164,486 (5.37%)	3,710 (0.12%)
17.10.24	3,283,102	170,897 (5.21%)	159,189 (4.85%)	386,053 (11.76%)	2,399,479 (73.09%)	163,551 (4.98%)	3,933 (0.12%)
10.10.24	3,006,925	161,535 (5.37%)	157,066 (5.22%)	357,961 (11.90%)	2,163,898 (71.96%)	162,804 (5.41%)	3,661 (0.12%)
07.10.24	2,946,247	156,017 (5.30%)	151,959 (5.16%)	340,913 (11.57%)	2,136,380 (72.51%)	157,192 (5.34%)	3,786 (0.13%)
03.10.24	2,994,362	158,366 (5.29%)	154,263 (5.15%)	341,906 (11.42%)	2,177,690 (72.73%)	158,328 (5.29%)	3,809 (0.13%)
30.09.24	2,961,626	163,911 (5.53%)	111,907 (3.78%)	348,997 (11.78%)	2,176,166 (73.48%)	156,591 (5.29%)	4,054 (0.14%)
26.09.24	2,825,396	159,025 (5.63%)	32,988 (1.17%)	335,191 (11.86%)	2,137,934 (75.67%)	156,402 (5.54%)	3,856 (0.14%)
23.09.24	2,825,918	158,377 (5.60%)	32,897 (1.16%)	335,996 (11.89%)	2,137,778 (75.65%)	157,054 (5.56%)	3,816 (0.14%)
20.09.24	2,875,442	160,832 (5.59%)	33,011 (1.15%)	341,053 (11.86%)	2,177,688 (75.73%)	159,003 (5.53%)	3,855 (0.13%)
16.09.24	2,884,312	162,423 (5.63%)	33,182 (1.15%)	336,103 (11.65%)	2,190,786 (75.96%)	157,844 (5.47%)	3,974 (0.14%)
11.09.24	2,822,055	158,607 (5.62%)	31,988 (1.13%)	323,144 (11.45%)	2,147,341 (76.09%)	157,112 (5.57%)	3,863 (0.14%)
05.09.24	2,815,482	159,666 (5.67%)	32,530 (1.16%)	319,401 (11.34%)	2,143,194 (76.12%)	156,972 (5.58%)	3,719 (0.13%)
29.08.24	2,864,373	163,236 (5.70%)	33,527 (1.17%)	318,350 (11.11%)	2,188,177 (76.39%)	157,049 (5.48%)	4,034 (0.14%)
26.08.24	2,873,523	163,566 (5.69%)	33,525 (1.17%)	307,699 (10.71%)	2,207,789 (76.83%)	156,910 (5.46%)	4,034 (0.14%)
22.08.24	2,852,439	162,145 (5.68%)	33,661 (1.18%)	280,985 (9.85%)	2,215,601 (77.67%)	156,054 (5.47%)	3,993 (0.14%)
19.08.24	2,878,214	163,531 (5.68%)	33,721 (1.17%)	235,771 (8.19%)	2,285,141 (79.39%)	156,027 (5.42%)	4,023 (0.14%)
12.08.24	2,892,642	163,134 (5.64%)	33,779 (1.17%)	230,161 (7.96%)	2,313,010 (79.96%)	148,747 (5.14%)	3,811 (0.13%)
08.08.24	2,915,133	163,544 (5.61%)	33,923 (1.16%)	227,008 (7.79%)	2,328,983 (79.89%)	157,825 (5.41%)	3,850 (0.13%)
12.07.24	2,939,892	164,698 (5.60%)	34,889 (1.19%)	34,619 (1.18%)	2,546,087 (86.60%)	155,835 (5.30%)	3,764 (0.13%)

Table B.5: Breakdown of IPv6 (DDR) DNS response RCODEs of *non-recursive resolvers* for each run. The percentage shows the proportion of individual RCODEs in relation to the others from the same run.

Date	Total	NOERR	SERVFAIL	NXDOMAIN	REFUSED	NOAUTH	Others
30.10.24	188,132	1,760 (0.94%)	1,628 (0.87%)	28,005 (14.89%)	151,883 (80.73%)	4,753 (2.53%)	103 (0.05%)
29.10.24	216,276	1,746 (0.81%)	1,210 (0.56%)	27,905 (12.90%)	180,576 (83.49%)	4,736 (2.19%)	103 (0.05%)
09.10.24	212,970	1,750 (0.82%)	1,499 (0.70%)	27,956 (13.13%)	176,834 (83.03%)	4,827 (2.27%)	104 (0.05%)
30.09.24	216,485	1,802 (0.83%)	1,498 (0.69%)	27,531 (12.72%)	180,805 (83.52%)	4,752 (2.20%)	97 (0.04%)
24.09.24	222,624	1,800 (0.81%)	1,395 (0.63%)	27,523 (12.36%)	186,987 (83.99%)	4,820 (2.17%)	99 (0.04%)
16.09.24	228,257	1,802 (0.79%)	506 (0.22%)	27,610 (12.10%)	193,416 (84.74%)	4,828 (2.12%)	95 (0.04%)
01.09.24	218,528	1,863 (0.85%)	518 (0.24%)	27,090 (12.40%)	184,089 (84.24%)	4,879 (2.23%)	89 (0.04%)
27.08.24	218,064	1,902 (0.87%)	541 (0.25%)	26,912 (12.34%)	183,706 (84.24%)	4,911 (2.25%)	92 (0.04%)
21.07.24	283,950	2,470 (0.87%)	1,098 (0.39%)	4,013 (1.41%)	270,558 (95.28%)	5,703 (2.01%)	108 (0.04%)
14.07.24	252,333	2,422 (0.96%)	629 (0.25%)	3,969 (1.57%)	239,481 (94.91%)	5,725 (2.27%)	107 (0.04%)

Table B.6: IPv4 addresses scanned and mapped to continents from the first scan (12.07.24).

Continent	Total	Timed Out	RCODE != 0	RCODE = 0
Asia	23,530,732 (87.02%)	22,385,446.0 (95.13%)	950,088 (4.04%)	195,198 (0.83%)
Europe	1,517,936 (5.61%)	303,505.0 (19.99%)	1,115,997 (73.52%)	98,434 (6.48%)
North America	1,511,982 (5.59%)	156,642.0 (10.36%)	1,201,083 (79.44%)	154,257 (10.20%)
South America	240,755 (0.89%)	70,535.0 (29.30%)	118,182 (49.09%)	52,038 (21.61%)
Africa	178,158 (0.66%)	120,569.0 (67.68%)	34,815 (19.54%)	22,774 (12.78%)
Oceania	59,500 (0.22%)	10,797.0 (18.15%)	43,270 (72.72%)	5,433 (9.13%)
Antarctica	6 (0.00%)	0.0 (0.00%)	5 (83.33%)	1 (16.67%)

Table B.7: IPv4 Addresses scanned and mapped to continents from the last scan (11.11.24).

Continent	Total	Timed Out	RCODE != 0	RCODE = 0
Asia	28,298,937 (88.69%)	27,082,914.0 (95.70%)	1,010,596 (3.57%)	205,427 (0.73%)
Europe	1,518,997 (4.76%)	277,687.0 (18.28%)	1,139,832 (75.04%)	101,478 (6.68%)
North America	1,505,253 (4.72%)	152,181.0 (10.11%)	1,202,161 (79.86%)	150,911 (10.03%)
South America	353,996 (1.11%)	60,196.0 (17.00%)	245,757 (69.42%)	48,043 (13.57%)
Africa	170,350 (0.53%)	108,298.0 (63.57%)	37,925 (22.26%)	24,127 (14.16%)
Oceania	59,876 (0.19%)	10,900.0 (18.20%)	43,840 (73.22%)	5,136 (8.58%)
Antarctica	7 (0.00%)	0.0 (0.00%)	6 (85.71%)	1 (14.29%)

Table B.8: IPv6 Addresses scanned and mapped to continents from the first scan (14.07.24).

Continent	Total	Timed Out	RCODE != 0	RCODE = 0
Europe	192,730 (42.01%)	32,162 (16.69%)	155,922 (80.90%)	4,646.0 (2.41%)
Asia	165,052 (35.98%)	62,783 (38.04%)	100,473 (60.87%)	1,796.0 (1.09%)
North America	77,786 (16.96%)	17,968 (23.10%)	57,605 (74.06%)	2,213.0 (2.84%)
South America	16,943 (3.69%)	3,879 (22.89%)	7,973 (47.06%)	5,091.0 (30.05%)
Oceania	5,075 (1.11%)	385 (7.59%)	4,521 (89.08%)	169.0 (3.33%)
Africa	1,161 (0.25%)	73 (6.29%)	1,009 (86.91%)	79.0 (6.80%)
Antarctica	2 (0.00%)	1 (50.00%)	1 (50.00%)	0.0 (0.00%)

Table B.9: IPv6 Addresses scanned and mapped to continents from the first scan (30.10.24).

Continent	Total	Timed Out	RCODE != 0	RCODE = 0
Europe	156,716 (46.03%)	40,367 (25.76%)	112,766 (71.96%)	3,583 (2.29%)
Asia	98,632 (28.97%)	42,412 (43.00%)	54,666 (55.42%)	1,554 (1.58%)
North America	64,373 (18.91%)	16,316 (25.35%)	45,711 (71.01%)	2,346 (3.64%)
South America	15,477 (4.55%)	2,638 (17.04%)	8,899 (57.50%)	3,940 (25.46%)
Oceania	4,417 (1.30%)	276 (6.25%)	4,017 (90.94%)	124 (2.81%)
Africa	883 (0.26%)	45 (5.10%)	769 (87.09%)	69 (7.81%)

B.2 Discovered DDR Discovery Figures

Table B.10: Discovered IPv4 (left) and IPv6 (right) DDR-enabled resolvers. Based on the `RA` bit in the DDR discovery's response, we distinguish whether the server is a recursive resolver.

Date	# Non-Recursive Resolvers	# Recursive Resolvers	Total
12.07.24	2,558 (0.85%)	298,055 (99.15%)	300,613
08.08.24	2,459 (0.82%)	297,822 (99.18%)	300,281
12.08.24	2,309 (0.78%)	295,133 (99.22%)	297,442
19.08.24	2,539 (0.84%)	300,851 (99.16%)	303,390
22.08.24	2,525 (0.82%)	306,761 (99.18%)	309,286
26.08.24	2,328 (0.77%)	301,430 (99.23%)	303,758
29.08.24	2,228 (0.74%)	299,869 (99.26%)	302,097
05.09.24	2,323 (0.75%)	307,882 (99.25%)	310,205
11.09.24	2,417 (0.79%)	303,431 (99.21%)	305,848
16.09.24	2,830 (0.92%)	304,257 (99.08%)	307,087
20.09.24	2,985 (0.97%)	305,345 (99.03%)	308,330
23.09.24	2,622 (0.85%)	304,446 (99.15%)	307,068
26.09.24	2,503 (0.83%)	299,962 (99.17%)	302,465
30.09.24	2,441 (0.80%)	302,618 (99.20%)	305,059
03.10.24	2,279 (0.73%)	308,090 (99.27%)	310,369
07.10.24	2,228 (0.72%)	309,129 (99.28%)	311,357
10.10.24	2,369 (0.75%)	314,804 (99.25%)	317,173
17.10.24	2,525 (0.79%)	315,430 (99.21%)	317,955
21.10.24	2,291 (0.72%)	316,753 (99.28%)	319,044
26.10.24	2,450 (0.76%)	318,996 (99.24%)	321,446
28.10.24	2,457 (0.77%)	316,386 (99.23%)	318,843
31.10.24	2,429 (0.78%)	307,234 (99.22%)	309,663
06.11.24	2,530 (0.80%)	315,688 (99.20%)	318,218
07.11.24	2,487 (0.78%)	316,426 (99.22%)	318,913
11.11.24	2,481 (0.82%)	301,605 (99.18%)	304,086

Date	# Non-Recursive Resolvers	# Recursive Resolvers	Total
14.07.24	72 (0.81%)	8,852 (99.19%)	8,924
21.07.24	90 (0.94%)	9,496 (99.06%)	9,586
27.08.24	77 (1.08%)	7,048 (98.92%)	7,125
01.09.24	72 (1.03%)	6,892 (98.97%)	6,964
16.09.24	69 (1.08%)	6,321 (98.92%)	6,390
24.09.24	65 (0.92%)	6,990 (99.08%)	7,055
30.09.24	68 (0.90%)	7,446 (99.10%)	7,514
09.10.24	62 (0.86%)	7,150 (99.14%)	7,212
29.10.24	60 (0.82%)	7,260 (99.18%)	7,320
30.10.24	63 (0.80%)	7,861 (99.20%)	7,924

Table B.11: Discovered IPv4 DDR-enabled resolvers by continent, based on data from November 11, 2024.

Continent	# Discovered IPs	# Timed Out	# DNS Servers	# DDR-enabled Servers	ρ_{DDR}
Asia	28,298,937 (88.69%)	27,082,914 (97.80%)	1,216,023 (28.85%)	152,046 (50.05%)	12.50
Europe	1,518,997 (4.76%)	277,687 (1.00%)	1,241,310 (29.45%)	52,665 (17.34%)	4.24
North America	1,505,253 (4.72%)	152,181 (0.55%)	1,353,072 (32.10%)	33,238 (10.94%)	2.46
South America	353,996 (1.11%)	60,196 (0.22%)	293,800 (6.97%)	41,838 (13.77%)	14.24
Africa	170,350 (0.53%)	108,298 (0.39%)	62,052 (1.47%)	21,381 (7.04%)	34.46
Oceania	59,876 (0.19%)	10,900 (0.04%)	48,976 (1.16%)	2,634 (0.87%)	5.38
Antarctica	7 (0.00%)	0 (0.00%)	7 (0.00%)	0 (0.00%)	0.00

Table B.12: Discovered IPv6 DDR-enabled resolvers by continent, based on data October 30, 2024.

Continent	# Discovered IPs	# Timed Out	# DNS Servers	# DDR-enabled Servers	ρ_{DDR}
Europe	192,730 (42.01%)	32,162 (27.43%)	160,568 (47.02%)	1,631 (18.33%)	1.02
Asia	165,052 (35.98%)	62,783 (53.55%)	102,269 (29.95%)	838 (9.42%)	0.82
North America	77,786 (16.96%)	17,968 (15.32%)	59,818 (17.52%)	1,319 (14.83%)	2.21
South America	16,943 (3.69%)	3,879 (3.31%)	13,064 (3.83%)	4,932 (55.43%)	37.75
Oceania	5,075 (1.11%)	385 (0.33%)	4,690 (1.37%)	105 (1.18%)	2.24
Africa	1,161 (0.25%)	73 (0.06%)	1,088 (0.32%)	72 (0.81%)	6.62
Antarctica	2 (0.00%)	1 (0.00%)	1 (0.00%)	0 (0.00%)	0.00

Table B.13: Top 10 IPv4 DDR-enabled resolvers by country, based on data from November 11, 2024.

Country	# DDR-enabled Servers	# DNS Servers	ρ_{DDR}
Bangladesh	49,863 (16.40%)	61,287 (0.19%)	81.36
Indonesia	26,672 (8.77%)	93,540 (0.29%)	28.51
United States	21,459 (7.06%)	1,346,605 (4.22%)	1.59
India	18,864 (6.20%)	218,843 (0.69%)	8.62
Brazil	18,459 (6.07%)	240,864 (0.75%)	7.66
South Africa	13,708 (4.51%)	34,382 (0.11%)	39.87
Russia	13,064 (4.30%)	188,185 (0.59%)	6.94
Iran	11,575 (3.81%)	6,477,568 (20.29%)	0.18
China	6,333 (2.08%)	20,651,053 (64.68%)	0.03
Colombia	6,065 (1.99%)	16,304 (0.05%)	37.20

Table B.14: Top 10 IPv6 DDR-enabled resolvers by country, based on data from October 30, 2024.

Country	# DDR-enabled Servers	# DNS Servers	ρ_{DDR}
Bolivia	1,768 (22.31%)	2,793 (0.80%)	63.30
Brazil	853 (10.76%)	7,827 (2.24%)	10.90
United States	837 (10.56%)	58,053 (16.59%)	1.44
Peru	822 (10.37%)	945 (0.27%)	86.98
Costa Rica	693 (8.75%)	979 (0.28%)	70.79
Czechia	283 (3.57%)	5,525 (1.58%)	5.12
Spain	197 (2.49%)	15,897 (4.54%)	1.24
France	182 (2.30%)	33,510 (9.58%)	0.54
Japan	170 (2.15%)	6,372 (1.82%)	2.67
Italy	154 (1.94%)	1,568 (0.45%)	9.82

B.3 Most Advertised DDR Configurations

Listing B.1: The most used DDR configuration, redirecting to Google.

```
1  dns.google.
       alpn="dot"
2  dns.google.
       alpn="h2,h3"
       dohpath="/dns-query{?dns}"
```

Listing B.2: The second most used DDR configuration, redirecting to *Cloudflare*.

```
1  one.one.one.one.
       alpn="h2,h3"
       port=443
       ipv4hint=1.1.1.1,1.0.0.1
       ipv6hint=2606:4700:4700::1111,2606:4700:4700::1001
       dohpath="/dns-query{?dns}"
2  one.one.one.one.
       alpn="dot"
       port=853
       ipv4hint=1.1.1.1,1.0.0.1
       ipv6hint=2606:4700:4700::1111,2606:4700:4700::1001
```

Listing B.3: The third most used DDR configuration, redirecting to *Cisco Umbrella (OpenDNS)*.

```
5  dns.opendns.com.
       alpn="dot"
       port=853
       ipv4hint=208.67.220.220,208.67.222.222
       ipv6hint=2620:119:35::35,2620:119:53::53
5  dns.umbrella.com.
       alpn="dot"
       port=853
       ipv4hint=208.67.220.220,208.67.222.222
       ipv6hint=2620:119:35::35,2620:119:53::53
10 dns.opendns.com.
       alpn="h2"
       ipv4hint=208.67.220.220,208.67.222.222
       ipv6hint=2620:119:35::35,2620:119:53::53
```

```
      dohpath="/dns-query{?dns}"
10  dns.umbrella.com.
      alpn="h2"
      ipv4hint=208.67.220.220,208.67.222.222
      ipv6hint=2620:119:35::35,2620:119:53::53
      dohpath="/dns-query{?dns}"
20  doh.opendns.com.
      alpn="h2"
      ipv4hint=146.112.41.2
      ipv6hint=2620:119:fc::2
      dohpath="/dns-query{?dns}"
20  doh.umbrella.com.
      alpn="h2"
      ipv4hint=146.112.41.2
      ipv6hint=2620:119:fc::2
      dohpath="/dns-query{?dns}"
```

Listing B.4: The fourth most used DDR configuration, redirecting to *Cisco Umbrella (OpenDNS)*.

```
5  familyshield.opendns.com.
      alpn="dot"
      port=853
      ipv4hint=208.67.220.123,208.67.222.123
      ipv6hint=2620:119:35::123,2620:119:53::123
10  familyshield.opendns.com.
      alpn="h2"
      ipv4hint=208.67.220.123,208.67.222.123
      ipv6hint=2620:119:35::123,2620:119:53::123
      dohpath="/dns-query{?dns}"
20  doh.familyshield.opendns.com.
      alpn="h2"
      ipv4hint=146.112.41.3
      ipv6hint=2620:119:fc::3
      dohpath="/dns-query{?dns}"
```

Listing B.5: The fifth most used DDR configuration, redirecting to Quad9.

```
1  dns.quad9.net.
      alpn="dot"
      port=853
```

```
    ipv4hint=9.9.9.9,149.112.112.112
    ipv6hint=2620:fe::fe
2 dns.quad9.net.
    alpn="h2"
    port=443
    ipv4hint=9.9.9.9,149.112.112.112
    ipv6hint=2620:fe::fe
    dohpath="/dns-query{?dns}"
```

B.4 DDR Configurations and their advertised Protocols and Priorities

Table B.15: List of most-advertised alternative domains (DoE resolvers) and their offered protocols.

a IPv4 DDR-enabled resolvers' most advertised alternative domains and their protocols (November 11, 2024).

	Alt. Domain	# Total Configs	DoH/1.1	DoH/2	DoH/3	DoT	DoQ
1	dns.google.	233,461 (72.58%)	0	233,458	233,458	233,460	0
2	one.one.one.one.	40,140 (12.48%)	0	40,140	40,140	40,140	0
3	dns.opendns.com.	8,111 (2.52%)	0	8,110	0	8,111	0
4	dns.umbrella.com.	8,111 (2.52%)	0	8,110	0	8,110	0
5	doh.umbrella.com.	8,006 (2.49%)	0	8,006	0	0	0
6	doh.opendns.com.	8,005 (2.49%)	0	8,005	0	0	0
7	dns.quad9.net.	3,462 (1.08%)	0	3,462	0	3,462	0
8	familyshield.opendns.com.	2,534 (0.79%)	0	2,534	0	2,534	0
9	family.cloudflare-dns.com.	1,233 (0.38%)	0	1,233	1,233	1,233	0
10	security.cloudflare-dns.com.	902 (0.28%)	0	902	902	902	0

b IPv6 DDR-enabled resolvers' most advertised alternative domains and their protocols (October 30, 2024).

	Alt. Domain	# Total Configs	DoH/1.1	DoH/2	DoH/3	DoT	DoQ
1	dns.google.	233,461 (72.58%)	0	233,458	233,458	233,460	0
2	one.one.one.one.	40,140 (12.48%)	0	40,140	40,140	40,140	0
3	dns.opendns.com.	8,111 (2.52%)	0	8,110	0	8,111	0
4	dns.umbrella.com.	8,111 (2.52%)	0	8,110	0	8,110	0
5	doh.umbrella.com.	8,006 (2.49%)	0	8,006	0	0	0
6	doh.opendns.com.	8,005 (2.49%)	0	8,005	0	0	0
7	dns.quad9.net.	3,462 (1.08%)	0	3,462	0	3,462	0
8	familyshield.opendns.com.	2,534 (0.79%)	0	2,534	0	2,534	0
9	family.cloudflare-dns.com.	1,233 (0.38%)	0	1,233	1,233	1,233	0
10	security.cloudflare-dns.com.	902 (0.28%)	0	902	902	902	0

Table B.16: List of DoE protocols and their priority combinations. The column "# Records" indicates the total count of observed protocol-priority combinations, whereas "# Records within Same Protocol" represents how many times the specific protocol was advertised across all observed configurations

a IPv4 DDR configurations and their protocol-priority combinations advertised (November 11, 2024).

Protocol	Priority	# Records	# Records within same DoE Protocol
DoT	1	237,150 (79.12%)	299,745 (26.29%)
DoH/2	2	236,950 (73.73%)	321,374 (26.27%)
DoH/3	2	233,460 (84.28%)	276,991 (25.88%)
DoH/2	1	47,093 (14.65%)	321,374 (5.22%)
DoT	2	43,817 (14.62%)	299,745 (4.86%)
DoH/3	1	43,528 (15.71%)	276,991 (4.83%)
DoT	5	18,768 (6.26%)	299,745 (2.08%)
DoH/2	10	18,767 (5.84%)	321,374 (2.08%)
DoH/2	20	18,558 (5.77%)	321,374 (2.06%)
DoQ	1	1,472 (54.52%)	2,700 (0.16%)
DoQ	3	1,218 (45.11%)	2,700 (0.14%)
DoH/1.1	1	1,218 (100.00%)	1,218 (0.14%)
DoT	10	6 (0.00%)	299,745 (0.00%)
DoQ	30	6 (0.22%)	2,700 (0.00%)
DoH/2	3	5 (0.00%)	321,374 (0.00%)
DoQ	2	4 (0.15%)	2,700 (0.00%)
DoH/3	3	3 (0.00%)	276,991 (0.00%)
DoT	4	2 (0.00%)	299,745 (0.00%)
DoT	0	2 (0.00%)	299,745 (0.00%)
DoH/2	5	1 (0.00%)	321,374 (0.00%)

b IPv4 DDR configurations and their protocol-priority combinations advertised (October 30, 2024).

Protocol	Priority	# Records	# Records within same DoE Protocol
DoT	1	6,729 (85.57%)	7,864 (28.26%)
DoH/2	2	6,718 (80.52%)	8,343 (28.22%)
DoH/3	2	6,608 (89.12%)	7,415 (27.75%)
DoH/2	1	989 (11.85%)	8,343 (4.15%)
DoT	2	810 (10.30%)	7,864 (3.40%)
DoH/3	1	806 (10.87%)	7,415 (3.39%)
DoT	5	317 (4.03%)	7,864 (1.33%)
DoH/2	10	317 (3.80%)	8,343 (1.33%)
DoH/2	20	315 (3.78%)	8,343 (1.32%)
DoQ	1	138 (83.13%)	166 (0.58%)
DoQ	3	22 (13.25%)	166 (0.09%)
DoH/1.1	1	22 (100.00%)	22 (0.09%)
DoT	10	4 (0.05%)	7,864 (0.02%)
DoQ	30	4 (2.41%)	166 (0.02%)
DoH/2	3	3 (0.04%)	8,343 (0.01%)
DoQ	2	2 (1.20%)	166 (0.01%)
DoT	0	2 (0.03%)	7,864 (0.01%)
DoT	4	2 (0.03%)	7,864 (0.01%)
DoH/2	5	1 (0.01%)	8,343 (0.00%)
DoH/3	3	1 (0.01%)	7,415 (0.00%)

B.5 DDR-enabled DoE Servers passing DDR's name-based Verification

Table B.17: DDR-enabled DoE resolvers and their advertised targets theoretically passing DDR's verification method for *Name-based Discoveries* excluding the major cloud DNS providers *Google*, *Cisco*, and *AdGuard* (November 11, 2024).

	DoE Server	DoE Target(s)
1	ns1.cryonet.de.	ns1.cryonet.de.
2	ns1.raiun.de.	ns1.raiun.de.
3	ns2.cryonet.de.	ns2.cryonet.de.
4	ns2.raiun.de.	ns2.raiun.de.
5	nsec.arnor.org.	nsec.arnor.org.
6	resolver.cybernetwork.ltd.	resolver.cybernetwork.ltd.
7	resolver.dns4all.eu.	resolver.dns4all.eu., dot.dns4all.eu., doh.dns4all.eu., doq.dns4all.eu.
8	resolver.somaf.de.	resolver.somaf.de.
9	resolver64.dns4all.eu.	doh64.dns4all.eu., dot64.dns4all.eu., resolver64.dns4all.eu., doq64.dns4all.eu.

List of Figures

List of Tables