University of Amsterdam

CIA Research Project

The Evolution of the Domain Name System: Scaling the Internet at the Cost of Complexity and Security

August 24, 2026

Supervisor:
dr. Marios Avgeris
Assistant Professor, University of Amsterdam
m.avgeris@uva.nl

CoSupervisor:
Willem Toorop
Senior Research Engineer, NLnet Labs
willem@nlnetlabs.nl

Student:
Ilyas Rahimi
12330337

Abstract

The Domain Name System (DNS), one of the core Internet protocols, has developed into a sophisticated and security-critical infrastructure from a straightforward distributed name service. This study combines two comprehensive empirical measures of encrypted DNS performance with historical analysis to investigate that progression. In a multi-interval experiment including 12,000 DNS requests, Traditional DNS, DNS over TLS (DoT), and DNS over HTTPS (DoH) are examined over a number of time periods. The results demonstrate that, in cold-start scenarios, encrypted transports cause a detectable latency overhead. By reevaluating all three procedures under consistent settings, utilizing a single resolver, a single measurement tool, and randomized subdomain searches to reduce resolver-side cache effects, a subsequent ablation study improves these results. The ablation experiment shows that although absolute delay varies by environment, methodological heterogeneity significantly reduces the DoH–DoT performance disparity.

When combined, these results measure the baseline performance costs of contemporary encrypted DNS protocols: DoH and DoT experience mean latency overheads of 31.6% and 25.6%, respectively, in the controlled ablation scenario when compared to Traditional DNS settings. These findings provide insight on the larger conflict in DNS evolution, where security and privacy additions increase the protocol's resilience at the expense of increased operational complexity and performance costs. This work explains the trade-offs that come with the protocol's shift toward more robust secrecy assurances by recording both historical and empirical aspects of DNS evolution.

Contents

1 Introduction

In the early 1980s, the Domain Name System (DNS) was introduced to address the scalability problems of the centrally managed `HOSTS.TXT` file. Instead of relying on a single, manually distributed list of hostnames, DNS adopted a distributed, hierarchical naming structure. That move from a central file to a federated namespace is one of the earliest examples of the Internet shifting away from tight, central coordination toward an infrastructure that could scale and tolerate failures. It also forms the starting point for understanding how DNS has evolved since: design decisions that initially favoured simplicity and speed were later confronted with a much larger, more diverse, and increasingly hostile Internet, which in turn drove the introduction of additional security mechanisms and operational extensions.

This tension is not unique to DNS. The conflict between performance optimization and security assurance, the technical debt incurred from retrofitting security onto existing protocols, and the difficulty of maintaining backward compatibility while adjusting to new requirements are all examples of fundamental networking principles that are demonstrated in the DNS evolution narrative. The original DNS design assumptions of a cooperative academic environment, as mentioned in [1], proved insufficient for the current Internet environment, resulting in vulnerabilities that required sophisticated security additions like DNSSEC [2] and encrypted transports [3, 4]. Much of the evolution of Internet infrastructure follows this pattern of progressive adaptation, where each generation responds to new challenges while preserving operational continuity.

The privacy implications of plain text queries discussed in [5], the pressures of centralization discussed in [6], and the operational complexity brought about by security extensions are some of the major issues facing modern DNS that are representative of larger Internet concerns. The most recent evolutionary stage is represented by the recent standardization of DNS over HTTPS (DoH) and DNS over TLS (DoT), which address secrecy issues while bringing new trade-offs in terms of performance, manageability, and network visibility. These changes demonstrate how priorities in Internet infrastructure are always being reevaluated to strike a balance between the efficiency and simplicity of the original architecture and the modern needs for security, privacy, and resilience.

In order to comprehend the tangible costs and effects of these adjustments, this study methodically examines DNS evolution using historical analysis and empirical measurement. We discuss four distinct subquestions in addition to the main question.

Main Research Question: *What complexity and security issues have resulted from the DNS's design decisions, and how has it evolved to accommodate the expansion of the Internet?*

Subquestions:

1. *What were the drawbacks of early Internet naming, such as the HOSTS.TXT file, and how were those scalability concerns resolved with the development of DNS in the 1980s?*
2. *Considering the extensive evolution of DNS specifications via several RFCs over the years, how has DNS increased in complexity and scope, and what implications does this have for administrators and implementers?*
3. *Which DNS security mechanisms or proposed modifications were considered in the past but never widely adopted, and what consequences might their deployment have had for today's Internet?*
4. *When compared to the original DNS design, how much do the security improvements brought about by contemporary encrypted DNS protocols (DoT and DoH) increase latency and system complexity?*

Through a methodical inquiry that blends historical interpretation and empirical measurement, we attempt to answer these questions. Section 3 explores the trajectory from HOSTS.TXT to modern encrypted DNS, addressing subquestions 1–3. To empirically address subquestion 4, this work combines a controlled ablation experiment in Section 5 with a complementary multi-interval measurement study that provides a broader empirical context. Section 4 discusses the primary experiment and the methodological lessons that motivated this structure, Section 6 synthesizes the empirical findings through a structured discussion of architectural and performance implications,

and Section 7 concludes by reflecting on the broader evolutionary trajectory of DNS and the trade-offs inherent in its modern security mechanisms.

The study combines quantitative evaluation of encrypted DNS performance overhead with qualitative examination of DNS evolution patterns.The empirical evaluation is centered on a controlled ablation study, complemented by a large-scale multi-interval measurement study that provides broader contextual insight into observed performance behavior.Together, these measurements quantify current security–performance trade-offs while clarifying how methodological choices influence observed protocol behavior. In doing so, the work offers both a historical analysis of DNS design decisions and baseline performance measurements that help to characterize the overhead properties of encrypted DNS protocols and to highlight methodological considerations for future measurement studies.

2 Related Work

A fundamental part of the worldwide Internet, the Domain Name System (DNS) has a long history of security issues that have been well documented in academic literature and technical standards. The original philosophical and architectural justification for a distributed hierarchical naming system to replace the centralized HOSTS.TXT file is given in [7], and the initial protocol specifications are described in [8]. These works establish delegation and caching as the key mechanisms that enabled DNS to scale, while deliberately prioritizing performance and simplicity over security guarantees. These sources provide a canonical historical account. Understanding these design goals and limitations is essential for interpreting the subsequent security-driven evolution of DNS.

The security flaws in the initial design have been the focus of later investigation. The study by [1] was among the first to systematically identify fundamental weaknesses in the DNS protocol, including the absence of integrity and authentication mechanisms that enabled cache poisoning attacks. The lengthy and intricate development of DNSSEC, which is formally described in [9], [10], and [2], was directly prompted by this. Numerous studies have examined the operational and adoption problems of DNSSEC, with guidance such as [11] that details operational practices including key management and rollover procedures.

More recent scholarly work has shifted its focus from integrity and authenticity toward confidentiality and privacy. The development of encrypted DNS protocols was motivated by privacy risks of plaintext DNS queries discussed in [5], including exposure to on-path observation and interference. The most recent significant step is the standardization of DNS over TLS (DoT) in [3] and DNS over HTTPS (DoH) in [4]. More recently, DNS over QUIC (DoQ) has been specified in [12], introducing an encrypted, UDP-based transport for DNS that leverages the QUIC protocol. This has brought about new trade-offs, though, work such as [6] discusses how DoH adoption can contribute to centralization of DNS resolution toward a small number of providers, and [4] discusses the implications for enterprise network management and troubleshooting. DNS over QUIC (DoQ) [12] extends the use of encrypted transports for DNS by operating over the QUIC protocol, providing confidentiality and integrity properties comparable to those of DoT and DoH. However, its performance behavior is shaped by transport-level characteristics specific to QUIC, including connection establishment and loss recovery. In this work, DoQ is therefore considered as part of the broader evolutionary context of encrypted DNS, but is not included in the empirical evaluation, which concentrates on measuring the baseline latency overhead introduced by established encrypted DNS transports under cold-start conditions. Taken together, these works describe how DNS evolved from a system focused mainly on scalability and fault tolerance into one that is now shaped to a growing extent by security and privacy concerns in an increasingly hostile Internet environment.

2.1 The Design Gap of Counterfactual DNS

A systematic analysis of how DNS might be designed today under modern technology and threat models, while maintaining backward compatibility with the existing hierarchical namespace, remains largely absent from the current body of research. While the evolution of DNS through successive security and privacy extensions has been well documented, this raises unresolved questions about how the protocol might be developed today to meet modern requirements without

inheriting technical debt from earlier design decisions. Despite offering fundamental direction, the Internet architecture concepts presented in [13] do not address the specific counterfactual rebuilding of DNS given contemporary restrictions.

The existing literature exhibits a clear dichotomy: either research focuses solely on small-scale enhancements to the current DNS infrastructure, or it suggests radical architectural departures that avoid compatibility, such as blockchain-based alternatives [14] or information-centric networking paradigms [15]. By revamping delegation procedures while preserving compatibility, recent initiatives such as the IETF Deleg working group seek to overcome DNS deficiencies [16, 17]. A detailed analysis that brings these different approaches together is still lacking, particularly one that examines how fundamental design choices might differ if DNS were designed today while taking its evolutionary history into account. Although many studies focus on individual extensions or alternatives, they rarely consider how earlier design decisions continue to constrain present-day development.

This gap is especially visible when considering the operational complexity introduced by encrypted DNS transports and the technical debt that has accumulated through add-on security mechanisms such as DNSSEC, this disparity is especially noticeable. Whether alternative trust models, native cryptographic integration, or different hierarchical structures would result from a ground-up design process guided by modern demands for mobile computing, IoT scalability, and privacy-by-design principles could be clarified by a methodical retrospective analysis.

The lack of RFCs or peer-reviewed publications explicitly addressing this counterfactual design paradigm points to a research opportunity that could guide the development of next-generation naming systems as well as the ongoing evolution of DNS. It also offers important insights into the path dependencies that limit protocol evolution in long-lived Internet infrastructure.

3 Current State: The Evolution of DNS from HOSTS.TXT to a Complex Ecosystem

Due to the shortcomings of the HOSTS.TXT file system, the Domain Name System was created out of necessity. To appreciate the latter evolution and the difficulties that arose, a thorough understanding of this original design is essential.

3.1 A Distributed Hierarchy from HOSTS.TXT

The early Internet was based on a single centrally controlled file called HOSTS.TXT, which was transferred over FTP (File Transfer Protocol) from the Network Information Center (NIC), the central authority in charge of keeping track of network information. [7] claims that this model has several scale issues, such as uncontrollably growing file sizes, difficulties maintaining consistency, and a key single point of failure that jeopardizes the dependability of the network as a whole.

- **Scalability Barrier:** The file size became uncontrollable as the number of hosts increased. Every current host had to periodically download the latest version and every new host had to be manually updated.
- **Overhead Traffic and Consistency:** It was not feasible to maintain uniformity across all machines. Increasing amounts of network bandwidth were used by the traffic that resulted from frequent downloads.
- **One Potential Point of Failure:** The central NIC was crucial because it was the single point of failure. The entire network mapping system would stall if it became inaccessible.

According to [7] and [8], these problems were resolved with the introduction of DNS in the 1980s by a series of design decisions:

- **Delegation:** Top-level domains (TLDs), such as '.com' and '.org', could be given responsibility by the root of the hierarchy to individual domain owners. The major bottleneck was removed as a result.

- **Caching:** For a Time-To-Live (TTL) duration, resolvers were made to cache replies. A crucial component in managing the expansion of the Internet, this realizes decreased query load and enhanced response times for commonly visited names.

The initial architecture was quite effective in accomplishing its primary objective: scalable name resolution. Nonetheless, its fundamental assumptions, primarily a cooperative and trustworthy ecosystem, established the foundation for subsequent security difficulties.

3.2 Expansion in Scope and Complexity: The Implied Burden

Continuous development has propelled DNS from its initial distributed naming system with multiple record types to a multifunctional infrastructure. After decades of gradual standardization, the protocol's scope has significantly grown beyond its initial goal, adding substantial complexity. Numerous fundamental RFCs addressing basic protocols, security extensions, operational procedures, and various technological improvements are now included in the DNS specification [7, 2].

The initial protocol defined in [8] introduced a basic query-response mechanism. Over time, this mechanism was extended through a series of incremental enhancements, driven by evolving operational, scalability, and security requirements:

- **New Record Types:** Although A (address), MX (mail routing), and TXT (arbitrary text) records were among the various record types included in the original DNS specification [8], the system has since grown to include additional record types. These include AAAA records for IPv6 addresses [18] and others introduced to support an increasing range of Internet services.
- **Extension Mechanisms for DNS (EDNS(0)):** An enhancement to DNS that must be implemented and maintained is EDNS(0) [19], which facilitates increased message length and optional features while maintaining backward compatibility with the original protocol.
- **Security Extensions (DNSSEC):** as a direct reaction to the vulnerabilities found in [1], DNSSEC [9, 10, 2] adds data with additional cryptographic signatures. Perhaps the biggest increase in complexity was this one: It brought with it complicated key management processes, new record types (RRSIG, DNSKEY, DS, NSEC/NSEC3), and a chain-of-trust concept that needs to be upheld from the root zone to the leaf.
- **Internationalization:** Internationalized Domain Names (IDNs) addressed a major usability issue by enabling non-ASCII characters, but they also made name canonicalization more difficult and exposed to homograph attacks.
- **Encrypted DNS Protocols:** The implementation of DNS over TLS (DoT) [3], DNS over HTTPS (DoH) [4] and DNS over QUIC (DoQ) [12] mitigated privacy issues by encrypting DNS traffic; however, it also introduced additional complexity in transport layers, trust models, and network management, potentially leading to the centralization of DNS resolution [6].

Implications of Growing Complexity:

- **For Implementers:** It is presently extremely difficult to write a DNS resolver or authoritative server that is completely compliant, safe, and efficient. Numerous record kinds, intricate feature interactions, and delicate edge cases must all be handled by the codebase. This makes flaws and vulnerabilities more likely, exhibiting the "DNS Camel" phenomena, wherein each additional feature exacerbates the strain until the system approaches failure [20].
- **For Administrators:** The operating load has increased dramatically. It is now necessary to comprehend zone serialization and TTL techniques only to manage a domain. An additional layer of crucial operational security is added by implementing DNSSEC, which includes key generation, safe key storage, rollover protocols, and communication with registrars. The DNS is a high-maintenance part of contemporary infrastructure since incorrect setting can result in service interruptions or security lapses representing the "DNS Camel" metaphor, which states that incremental complexity has the potential to overwhelm a resilient system [20].

The cumulative impact of these developments has resulted in a protocol that has deviated considerably from its original simplicity, imposing substantial operational and implementation constraints that exemplify the 'DNS Camel' phenomena of complexity overload.

3.3 Unfollowed Routes: Insights from Unaccepted Proposals

The development of DNS was influenced by ideas that were never widely accepted, providing insights into alternate Internet histories, in addition to the established complications. Although they were discussed, a number of suggested modifications or security measures were not widely adopted. As a result, the present security posture of DNS reflects the long-term consequences of these unadopted proposals [7].

- **DNS as the Primary Protocol over TCP:** Performance and simplicity were given top priority in the original DNS specification [8], which required UDP as the primary transport and TCP as a backup for large responses. Concerns about the increased overhead of connection setup and server state maintenance led to discussions about transport choices during DNS development, with UDP being selected for its lightweight characteristics in the original design [7].
- **First Authentication Systems:** There was no cryptographic authentication in the original DNS design [8]. Although ideas for incorporating basic cryptographic challenges were considered, they were judged to be too sophisticated and computationally costly for the hardware available at the time, as noted in retrospective analyses of early Internet security [1].
- **DNSCurve:** Daniel Bernstein proposed DNSCurve, a solution that may have offered DNS encryption (confidentiality) and authentication [21]. This method, which used elliptic curve cryptography to provide end-to-end security between resolvers and authoritative servers, was not extensively used in favor of the incremental DNSSEC method, which prioritized authentication over confidentiality.
- **Hypothetical Security Benefits:** Had TCP been adopted as the primary transport from the outset, certain classes of cache poisoning attacks that exploit UDP statelessness, as described in later analyses [1], may have been mitigated. Such a design choice could also have simplified the integration of later extensions such as DNSSEC, which introduce larger response sizes [2]. In this counterfactual scenario, some of the operational complexity associated with add-on DNSSEC deployment [11], as well as several DNS spoofing and phishing-based man-in-the-middle attack vectors identified in subsequent work [1], might not have emerged in their observed form.

A recurrent theme in the history of the Internet is the trade-off between short-term usefulness and long-term security and resilience, which is exemplified by the inability to embrace these more resilient initial designs. Through intricate add-ons like DNSSEC and DoH/DoT, the community is still paying back the technical debt caused by the decisions made in the 1980s regarding scalability and performance.

The dismissal of these more resilient early designs resulted in enduring technological debt, requiring the intricate retrofitting security enhancements that define the contemporary DNS environment.

3.4 Advanced Features and Their Consequences: DNS vs. HTTPS

The previously overlooked deficiency in the original design-insufficient confidentiality is now rectified in the latest major phase with encrypted DNS protocols such as DNS over TLS (DoT), DNS over QUIC (DoQ), and DNS over HTTPS (DoH). Although DoT [3] and DoQ ensure encryption and integrity protection, DNS over HTTPS (DoH) [4] also facilitates the integration with other web traffic through the utilization of HTTPS port 443.

The Way DoH Handles Privacy Issues:

- **Encryption:** DoH stops network eavesdroppers from knowing which domains a user is requesting by encrypting the entire DNS transaction. This guards against profiling based on surfing patterns and spying.
- **Respect for integrity:** By prohibiting on-path attackers from altering DNS answers, HTTPS offers integrity protection.

Introducing Trade-offs:

- **DNS Traffic Centralization:** DoH makes it easy for applications to submit queries directly to a third-party DoH provider that has already been configured, bypassing the operating system and local network resolvers. As a result, DNS query data becomes concentrated at a limited number of providers, introducing new privacy and control concerns that differ in form, but not in nature, from the issues DoH was designed to mitigate [6].
- **Complication of Policy and Network Troubleshooting:** DNS is an essential tool for enterprise and institutional networks because it provides internal name resolution, enforces access controls, and keeps an eye on network health. DoH can complete circumvention of these local DNS servers. Because DNS requests are concealed behind encrypted HTTPS streams, a technical support staff may not be able to identify a connectivity problem or an employee using a DoH-enabled browser may access a prohibited website without the corporate firewall's intervention [4].

To sum up, DNS's development is a tale of iterative adaptation. Every design decision, from the encrypted channels of DoH that handle contemporary privacy concerns to the hierarchical namespace that resolved the scalability issue of HOSTS.TXT, has brought new capabilities along with new trade-offs and complications. Although the system's continued performance is evidence of the soundness of its fundamental architecture, its increasing complexity and the cyclical nature of its security issues draw attention to the ongoing expense of upholding a foundational protocol in a quickly evolving Internet.

4 Primary Experiment and Lessons Learned

We first executed the multi-interval experiment to quantify the latency overhead of encrypted DNS under cold-start conditions. The measurements provide baseline results, but they also make clear that tooling, experimental controls, and the chosen vantage point can strongly influence the observed differences if they are not held constant.

In particular, the multi-interval experiment was conducted from a well-provisioned server environment with a dedicated 1 Gbps network link. While this setup enables stable and repeatable measurements, it represents an optimized infrastructure perspective rather than the conditions under which most end users issue DNS queries. Since the objective of this work is to understand user-perceived DNS resolution latency, a complementary ablation study was performed from a consumer-grade environment. This second vantage point captures DNS behavior under more typical access conditions and allows protocol overhead to be examined independently of infrastructure-specific performance advantages.

4.1 What the Primary Experiment Can and Cannot Claim

The multi-interval experiment captures end-to-end, user-perceived resolution latency using common system utilities and public resolvers. The resulting dataset is accurate and reproducible, and it supports the conclusion that encrypted DNS introduces measurable overhead relative to Traditional DNS.

However, because DNS over HTTPS was measured using `curl` while Traditional DNS and DNS over TLS were measured using `dig`, the primary experiment should not be interpreted as a strict protocol-to-protocol comparison between DoH and DoT. In this setting, any observed DoH–DoT gap may reflect tool and implementation effects in addition to transport mechanics, and therefore does not, by itself, justify definitive claims about the relative performance ordering of DoH and DoT.

4.2 Methodological Take-away

To obtain defensible protocol-level comparisons, performance evaluations should use a uniform toolchain, a consistent resolver configuration, and controls that limit caching and other confounders. In this work, those conditions are explicitly enforced in the ablation study, which is therefore the basis for the DoH–DoT comparative conclusions. Full procedural details of the multi-interval experiment are retained in Appendix A for reproducibility.

5 Ablation Study: Performance of DNS Protocol Under Uniform Measurement Conditions

5.1 Experimental Methodology

This ablation experiment aimed to isolate protocol performance by mitigating tool and provider variability through stringent methodological controls, while also addressing shortcomings found in our initial study. The study utilized a standardized measurement technique, evaluating all DNS protocols only using the `dig` utility (BIND 9.18+) under uniform conditions, therefore rectifying the tool inconsistency (mixed `dig` and `curl`) seen in our initial methodology.

Measurements were performed on an Apple MacBook Pro (Model MRX33N/A) including an M3 Pro CPU (11 cores: 5 performance, 6 efficiency), 18 GB of unified memory, and operating on macOS 26.1 (Build 25B78). This system functioned in a standard end-user environment with simultaneous background processes and applications, in contrast to the specialized Dell PowerEdge R240 server (16 GB ECC RAM) utilized in our primary research. The system was linked over Gigabit Ethernet to a home broadband connection with a throughput of 100 Mbps, in contrast to the enterprise-grade dedicated internet connection of 1 Gbps accessible to the server infrastructure.

The experimental design preserved the fundamental framework of our primary study while incorporating methodological enhancements: five representative domains (`example.com`, `google.com`, `github.com`, `nytimes.com`, and `icann.org`) were examined across three protocols (Traditional DNS, DNS over TLS, and DNS over HTTPS) with five repetitions for each domain-protocol pairing, resulting in 75 measurements per experimental iteration. To mitigate resolver side cache effects and guarantee that each query exhibited authentic cold start behavior, all measurements were conducted using randomly generated subdomains prefixed to the base domain (e.g., `a1b2c3d4.google.com`). This method compels the resolver to execute a complete resolution route for every request, decreasing the probability of cache reutilization and thereby isolating inherent protocol overhead. All queries were solely forwarded to Google's public DNS resolver (8.8.8.8), hence reducing the unpredictability associated with providers in our previous analysis that utilized both Google and Cloudflare resolvers. A total of 30 runs were executed within a single temporal frame, grounded in empirical findings from our multi-interval research indicating negligible diurnal performance change (Appendix A). All measurements were conducted under cold-start conditions without connection reuse to capture the baseline protocol overhead.

5.2 Results and Performance Analysis

Table 1: Protocol Performance in Ablation Study (2,250 Measurements)

Protocol	Mean (ms)	Median (ms)	Std Dev	Success Rate	Overhead
Traditional DNS	74.7	59.9	258.7	100.0%	Baseline
DNS over HTTPS	98.3	95.1	27.6	100.0%	+31.6%
DNS over TLS	93.8	90.9	21.1	100.0%	+25.6%

Under the controlled measurement conditions of the ablation study, clear latency differences between the evaluated DNS protocols are observed. Traditional DNS shows the lowest mean latency, with an average resolution time of 74.7 ms. DNS over HTTPS shows a mean latency overhead of 31.6% compared to this baseline, while DNS over TLS shows an overhead of 25.6%. All 2,250

measurements completed successfully for all protocols, and no failed queries were observed during the experiment.

5.3 Methodological Impact and Performance Consistency

The methodological standardization in this ablation study elucidates significant insights regarding DNS performance assessment. Although absolute latency values escalated across all protocols in the consumer environment, Traditional DNS increased from 26.7 ms to 74.7 ms (+179.8%), DNS over HTTPS from 45.1 ms to 98.3 ms (+117.9%), and DNS over TLS from 92.1 ms to 93.8 ms (+1.8%) this augmentation predominantly signifies environmental variances rather than constraints of the protocols.

The most notable observation concerns the comparative efficacy of encrypted methods. In our initial study, DNS over HTTPS (45.1 ms) demonstrated a significant speed advantage over DNS over TLS (92.1 ms), indicating a 104.2% performance superiority for DoH. Nonetheless, under uniform measurement circumstances, this discrepancy practically vanishes: DoH at 98.3 ms compared to DoT at 93.8 ms indicates just a 4.8% variation. This significant decrease suggests that the previously noted disparity stemmed mostly from tool variations (`dig` versus `curl`) rather than intrinsic protocol attributes.

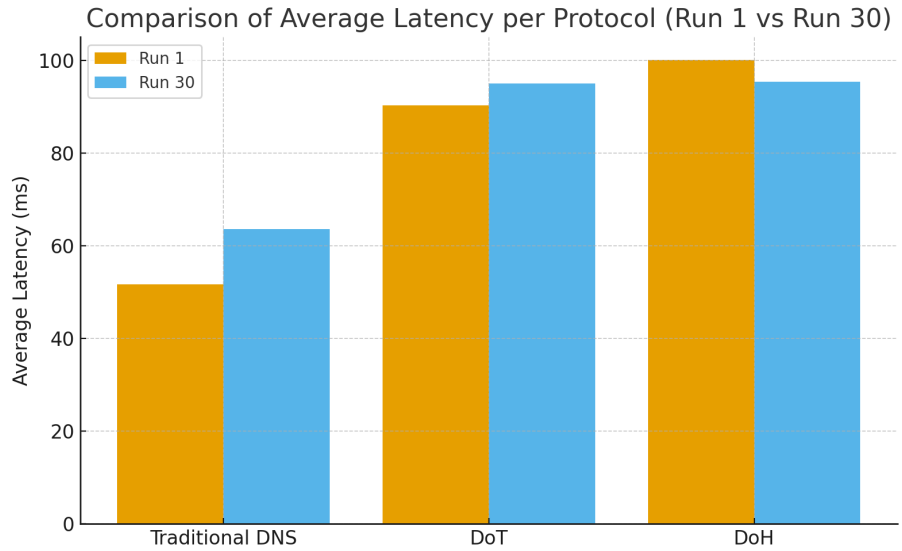

Figure 1: Comparison of mean delay per protocol between the initial and 30th ablation runs under consistent measurement circumstances. The data illustrates that subsequent runs do not gain from systematic latency decreases, suggesting that resolver side caching does not significantly affect the outcomes.

To ensure that resolver side caching did not skew the ablation data, we analyzed the average latency of the initial and 30th runs for each protocol (Figure 1). The findings indicate that Traditional DNS rises from 51.6 ms to 63.6 ms, DNS over TLS grows from 90.2 ms to 95.0 ms, and DNS over HTTPS has a modest decline from 100.1 ms to 95.3 ms. Instead of demonstrating a systematic decrease in latency, subsequent tests are equivalent to or slightly slower than the original run, which aligns with the utilization of randomized subdomains that inhibit efficient resolver caching. This comparison substantiates the notion that the documented overheads reflect authentic cold-start behavior rather than artifacts of cache warm-up.

5.4 Comparative Analysis with Primary Results

Table 2: Performance Comparison: Primary vs. Ablation Study

Protocol	Primary Mean	Ablation Mean	Difference	Relative Performance
Traditional DNS	26.7 ms	74.7 ms	+179.8%	Baseline in both studies
DNS over HTTPS	45.1 ms	98.3 ms	+117.9%	4.8% slower than DoT in ablation
DNS over TLS	92.1 ms	93.8 ms	+1.8%	4.8% faster than DoH in ablation

The comparison study uncovers environmental effects and methodological insights. Traditional DNS exhibits the greatest environmental sensitivity, demonstrating a delay increase of 179.8% in consumer circumstances. This significant enhancement arises from many factors: the Dell PowerEdge R240 server's exclusive 1 Gbps enterprise connection compared to 100 Mbps home internet, the server's refined network stack vs consumer-grade equipment, and dedicated versus shared computational capabilities.

DNS over TLS has exceptional environmental resilience with an only 1.8% increase in latency. The uniformity seen across the two measured infrastructures indicates that encrypted protocols, especially DoT, provide reliable performance across the environments studied.

The primary methodological discovery pertains to DNS over HTTPS. Under uniform measuring settings, DoH's perceived performance superiority over DoT decreases from 104.2% in our primary trial to only 4.8% in the ablation study. This convergence suggests that a significant portion of the performance disparity formerly ascribed to protocol features was, in reality, due to measurement artifacts.

5.5 Environmental Context and Performance Analysis

The infrastructural disparities among trials offer essential background for analyzing performance outcomes. The primary study's server setup (Dell PowerEdge R240, 16GB ECC memory, 1 Gbps dedicated connection, devoid of competing processes) exemplifies optimum circumstances seldom observed in real-world scenarios. The consumer setting of the ablation research (Apple MacBook Pro, 18GB unified memory, 100 Mbps home internet, standard background processes) more accurately reflects real-world usage conditions.

This ambient setting elucidates the general delay increase seen in our ablation investigation. Conventional DNS exhibits significant sensitivity to these settings, with performance deteriorating markedly when transitioning from optimized to standard surroundings. Conversely, encrypted protocols exhibit far more stable performance, with DNS over TLS demonstrating notable resilience.

The methodological enhancements in this ablation investigation, including singular tool deployment, consistent provider selection, and controlled experimental circumstances, rectify the inadequacies of our first experimental design. Eliminating heterogeneity in tools and providers yields greater insights into the authentic qualities of protocols, free from measuring artifacts.

5.6 Implications for DNS Protocol Evaluation

This ablation research illustrates that the measuring approach substantially influences DNS protocol assessment. Three key implications arise:

Initially, environmental conditions adversely impact the performance of Traditional DNS. Encrypted protocols exhibit uniform performance across diverse infrastructures, but Traditional DNS performance fluctuates significantly based on network quality and processing capabilities. This discovery indicates that encrypted protocols may offer a more consistent user experience in diverse deployment contexts.

The selection of tools significantly impacts performance evaluations of encrypted protocols. The significant alignment of DoH and DoT performance under uniform measurement settings suggests that previously noted discrepancies were mostly due to implementation artifacts rather than inherent protocol traits.

Third, uniform measuring methodologies produce more dependable protocol comparisons. Future DNS performance studies must emphasize uniform tools and environmental controls to reduce measurement artifacts and achieve precise protocol characterizations.

These findings indicate to network architects and protocol developers that encrypted DNS protocols, especially DNS over TLS, provide reliable performance in many contexts. Although Traditional DNS may attain reduced latency in optimum conditions, its performance significantly deteriorates in standard user situations, diminishing its practical benefits.

This study's methodological modification enhances confidence in performance connections by removing measurement variability. This methodology lays the groundwork for enhanced DNS protocol assessment in further studies and provides pragmatic guidance for implementation choices across various infrastructure contexts.

6 Discussion

Our multi-interval experiment, in conjunction with the controlled ablation experiment, provides an extensive analysis of the performance attributes of contemporary DNS protocols and the emerging trade-offs associated with encrypted transports. The combined data illustrate that design decisions described in [7] continue to influence protocol behaviour, and that observed performance is strongly shaped by measurement methodology and experimental environment.

The large-scale multi-interval experiment shows the real-world costs and effects of encrypted DNS, while the ablation study shows which differences are due to the way the protocol works and which are due to the tools or infrastructure used. An analysis of these numbers reveals the impact of encryption on performance, the influence of implementation choices, and the overarching trend of DNS evolving towards enhanced privacy amidst increasing complexity.

6.1 Overhead in Encryption as an Evolutionary Cost

The performance overhead associated with incorporating privacy measures into a protocol originally designed for scalability and fault tolerance is evident in our testing, with Traditional DNS exhibiting the lowest latency. The performance disparity between DNS over HTTPS and DNS over TLS may be affected by the distinct measurement tools employed (`curl` for DoH and `dig` for DoT), since previous research has indicated more comparable performance traits between both protocols. This interpretation is corroborated by the ablation experiment, in which all protocols were assessed using a consistent tool and resolver under standardized conditions. In that controlled environment, the disparity between DNS over HTTPS and DNS over TLS reduced to a minimal 4–5% variation in mean latency, indicating that the significant gap noted in the multi-interval study was primarily due to methodological discrepancies rather than intrinsic protocol characteristics. The first DNS specification [8] emphasized minimal overhead through UDP-based queries, a premise that has been deemed inadequate for the contemporary adversarial Internet landscape, as indicated in [1].

The initial DNS architecture emphasized low overhead due to the bandwidth limitations and latency issues prevalent in 1980s network settings. In the current age of gigabit Ethernet, ubiquitous broadband, and enhanced TCP implementations, the trade-offs have evolved. The performance costs of encrypted DNS, albeit quantifiable, are frequently alleviated by contemporary network capabilities, while offering crucial privacy safeguards and the inherent stability advantages of TCP that were lacking in the original UDP-based architecture. The ablation measurements further demonstrate that environmental context plays a substantial role in shaping perceived protocol overheads. While Traditional DNS exhibited significantly higher latency in the consumer environment of the ablation setup, the encrypted protocols showed far more stable behaviour, with DNS over TLS increasing by only 1.8% relative to its performance in the server-based measurements of the primary study. As a result, the relative cost of encryption becomes less pronounced when baseline latency is elevated, narrowing the observable performance difference between encrypted and unencrypted transports.

6.2 Cold-Start Conditions and the Basics of Protocol

In order to capture the essential performance features of each protocol without the confounding effects of connection reuse, both the multi-interval measurement study and the ablation experiment were deliberately conducted under cold-start conditions. This method provides important insights for comprehending the baseline performance costs of DNS encryption by exposing the intrinsic overhead of encrypted DNS protocols in situations when persistent connections cannot be maintained.

For the following worst-case scenarios, the assessed overheads are relevant:

- **Intermittent and Mobile Connectivity**: Devices with often-changing network conditions
- **First Service Contact**: Scenarios requiring first engagement with services that require sporadic DNS access.
- **Low-Frequency Applications**: Applications or services that perform infrequent DNS requests, where sustaining persistent connections would be inefficient due to minimal consumption.

The basic analysis methodology in [1], which looked at DNS vulnerabilities in baseline settings to understand the underlying problems of the protocol, is consistent with this methodological decision. The ablation investigation supports this view by suppressing resolver-side caching by using the same cold-start methodology while executing queries to randomized subdomains under consistent tool and provider settings. DNS over HTTPS and DNS over TLS show roughly aligned delay under these controlled conditions, suggesting that both encrypted transports have similar baseline overheads when permanent connections are not available. This behavior demonstrates that inherent protocol mechanics, not artifacts of connection reuse or cache warm-up, are the primary cause of the measured encryption costs.

6.3 Future Directions and Evolutionary Pathways

Our study shows how DNS has continued to evolve from a basic distributed database to a sophisticated multifunctional infrastructure with important security obligations. The trajectory set by DNSSEC [11], where security extensions added operational complexity but were required as the Internet environment became more hostile, is followed by this evolutionary trend.

Each protocol's proper deployment environment is determined by the measured trade-offs:

- **Traditional DNS**: Upholds the efficiency priorities of the original architecture and is best suited for performance-critical applications where local threat models allow.
- **DNSSEC**: Is a fundamental integrity technique that authenticates DNS data via cryptographic signatures, thereby augmenting security against spoofing and cache poisoning, while also introducing increased operational and administration complexity.
- **DNS over TLS**: A security-first choice for settings where performance considerations are subordinated to encryption needs.
- **DNS over HTTPS**: A balanced strategy for widespread deployment that leverages existing web infrastructure, simplifying deployment at the cost of a modest performance overhead.
- **DNS over QUIC (DoQ)**: An innovative encrypted transport that enhances the DNS privacy framework over QUIC, providing advantages such as diminished connection establishment latency and enhanced loss recovery in settings where QUIC is extensively utilized.

This viewpoint shows how, albeit at the expense of greater complexity and performance overhead, Internet infrastructure can adjust to new needs while preserving backward compatibility. The results provide network architects with empirical support for well-informed protocol selection based on particular security and performance needs by quantifying the tangible costs of this adaptation.

Our study's documentation of the DNS evolution serves as an example of the larger trend in the development of Internet infrastructure, wherein initial designs that prioritize simplicity and

scalability later necessitate intricate adjustments to tackle emerging security issues. From the initial replacement of HOSTS.TXT to modern encrypted transports, this pattern demonstrates how long-standing Internet protocols continue to strike a balance between security, complexity, and performance.

6.4 Quantum Computing and Prospective Cryptographic Challenges

Reflecting on the growth of DNS from a basic distributed name system to its current advanced architecture, it is evident that each security improvement has responded to modern threats while also bringing additional complexity. The emergence of quantum computing introduces unparalleled challenges that could render existing cryptography safeguards ineffective. DNSSEC's dependence on Traditional public-key encryption, specifically RSA and elliptic curve algorithms as outlined in [9], will be susceptible to Shor's algorithm [22] once sufficiently advanced quantum computers are developed. This prompts alarming inquiries on the potential for today's encrypted DNS traffic to be decrypted retrospectively as quantum technology advances, presenting a "harvest now, decrypt later" scenario that jeopardizes the long-term confidentiality assurances of DoH and DoT [4, 3]. The shift to post-quantum cryptography signifies the forthcoming significant advancement in DNS security. Incorporating quantum-resistant algorithms into DNSSEC and encrypted DNS transports necessitates extensive coordination among resolvers, authoritative servers, software implementations, and global operational practices, while maintaining backward compatibility with the current DNS infrastructure. This forthcoming change highlights the ongoing evolution of DNS in response to growing risks, reflecting previous instances where the protocol adapted to new security requirements, resulting in increased complexity.

6.5 Limitations

Methodological constraints may affect the interpretation of this study's results. The single regional vantage point allows for constant and controlled assessment, but limits generalizability to global network conditions. Our multi-interval experiment showed little variation between peak and off-peak periods, but the ablation study was conducted within a single temporal window and is therefore exposed to external factors like transient congestion, resolver load, and routing fluctuations. These factors may introduce additional latency that cannot be directly observed or quantified within our framework. While the available evidence does not indicate pronounced diurnal sensitivity, repeating the ablation procedure across multiple, independently sampled intervals would yield a more comprehensive estimate of central tendencies and reduce the influence of unknown environmental variation. Moreover, the exclusive emphasis on A-record lookups and cold-start behavior isolates baseline protocol characteristics but does not capture performance dynamics that emerge under connection reuse, persistent sessions, or operational workloads typical of high-throughput resolvers.

A central limitation of the multi-interval experiment was the use of different measurement tools for encrypted and unencrypted protocols (`dig` for Traditional DNS and DoT, `curl` for DoH), which introduced potential implementation-specific effects. As noted in previous research, direct protocol comparisons require a standardized measurement framework to avoid implementation-driven bias [23]. The ablation study adhered to this guidance by employing a uniform toolchain and resolver across all protocols, thereby eliminating the methodological inconsistency present in the primary experiment and enabling a more accurate assessment of intrinsic protocol behaviour.

The measurement window, although producing a substantial dataset, may not capture rare network events or longer-term temporal trends. Additionally, the experiments do not evaluate protocol behaviour under persistent connections, which are common in operational resolvers and may reduce the relative overhead of encrypted transports. These factors constrain the extent to which the measured cold-start overheads can be generalized to all real-world deployments.

Taken together, these limitations emphasize that while the experiments provide reliable baseline measurements of protocol behaviour, the interpretation of encrypted DNS performance must account for environmental, methodological, and operational variability. The two-part experimental design partially addresses these issues, and future work should extend this approach through multi-vantage measurements, additional record types, and evaluations of connection reuse.

6.6 Threats to Validity

Although this study used strict methodological controls to guarantee the quality of the data, a number of possible validity threats should be taken into account for a thorough interpretation of the results.

Internal validity: The primary experiment was affected by tool heterogeneity (`dig` for Traditional DNS and DoT, `curl` for DoH), which introduced implementation-specific variability. The ablation study mitigated this issue by evaluating all protocols through a uniform measurement tool and resolver, thereby reducing tool-driven bias. Nonetheless, uncontrollable background factors persist—such as transient resolver load, anycast routing adjustments, and local system scheduling jitter—which may contribute minor variability that cannot be fully isolated.

External validity: The main issue is generalizability to various geographic locations and network infrastructures. Although the Netherlands' singular vantage point offers reliable comparative data, it could not accurately reflect performance traits in areas with differing infrastructure. Although it restricts application to customized DNS implementations, the emphasis on major public resolvers guarantees relevance to widely used services.

Construct validity: The operationalization of "performance" solely through latency measurements is one of the factors to be taken into account. While latency serves as an adequate proxy for user-perceived speed, it fails to include energy consumption, computational expense, or the privacy benefits of encrypted DNS, as outlined in [5]. The cold-start measurement technique may exaggerate typical operational overhead in connection-reuse situations, although it precisely quantifies protocol overhead.

7 Conclusion

This work indicates that the security extensions added to DNS for confidentiality and integrity result in a quantifiable performance cost, which contrasts with the efficiency assumptions inherent in the original protocol design. The ablation study, performed under consistent measurement conditions with a single toolchain and resolver, quantifies this overhead. DNS over HTTPS results in a 31.6% increase in mean latency compared to Traditional DNS, whereas DNS over TLS leads to a 25.6% increase. Contemporary network architectures, enhanced transport-layer implementations, and the prevalent adoption of high-speed consumer connectivity mitigate the perceptual effects of delays in routine usage; however, the fundamental costs associated with cryptography and connection establishment persist in encrypted transport protocols. The additional latency for each individual query is relatively modest, generally a few tens of milliseconds. However, the cumulative impact is substantial when accounting for the large number of DNS requests generated by a single user monthly, as well as the trillions issued worldwide. This illustrates that the incremental integration of security mechanisms into DNS, although essential, imposes a structural performance cost that diverges from the lightweight, stateless functionality intended by the original architecture.

7.1 Future Work

A complementary direction for future work concerns the operational and efficiency implications of maintaining a local copy of the DNS root zone at recursive resolvers. This practice has been documented in the DNS Operations working group and described as a Best Current Practice [24]. While the primary motivation is improved robustness in environments where connectivity to the root server system is intermittent or policy-restricted, the system-level efficiency trade-offs of this approach remain less clearly characterized.

Under the conventional hierarchical resolution model, recursive resolvers interact with the root layer whenever cached delegation information expires or when resolving previously unseen names. Although individual root responses are small, the aggregate volume of root-directed traffic may become substantial at Internet scale. A systematic measurement of how frequently resolvers reach the root layer under representative workloads, together with an estimate of the resulting data exchange, would provide a quantitative baseline for evaluating whether local root operation offers measurable efficiency gains in addition to its resilience benefits.

Maintaining the root zone locally changes this interaction pattern by replacing frequent query traffic with periodic synchronization of root zone contents. In that setting, the dominant cost shifts from request volume to update behavior: how often the zone is refreshed, how much data is transferred per refresh, and what operational complexity is introduced by update verification and failure handling. Quantifying these costs would enable a direct comparison between the traditional query-driven model and a local-root deployment model under varying network conditions, including constrained or lossy access links.

An additional aspect concerns the different distribution and update mechanisms described in [24]. The draft documents both DNS-based zone acquisition and HTTPS-based distribution approaches, each of which implies different assumptions about trust anchors, transport behavior, and operational workflows. A comparative evaluation of these mechanisms across widely deployed resolver software would therefore be valuable, particularly in terms of bandwidth consumption, update latency, and operational complexity. Such an evaluation could also clarify how implementation choices affect deployment feasibility in practice, given that resolvers may differ in how they validate, store, and refresh locally maintained root data.

Finally, extending this line of work beyond a single query type and beyond a single network environment would improve external validity. Measuring the same effects under weaker access connectivity and with additional query patterns (e.g., DNSSEC-related lookups) would clarify whether the observed efficiency and robustness properties generalize across realistic deployment contexts.

7.2 Research Questions Addressed

Main Question: What complexity and security issues have resulted from the DNS's design decisions, and how has it evolved to accommodate Internet expansion?

- **Answered in Section 3:** The evolution from HOSTS.TXT to distributed hierarchy and subsequent feature expansion demonstrates how initial design choices created both scalability solutions and long term security challenges.
- **Answered in Section 5:** Empirical measurements quantify the complexity costs of encrypted DNS protocols, showing a 31.6% mean latency overhead for DNS over HTTPS (DoH) and 25.6% for DNS over TLS (DoT) relative to Traditional DNS, while maintaining 100% reliability in the controlled ablation study.

Subquestion 1: What were the drawbacks of early Internet naming, such as the HOSTS.TXT file, and how were those scalability concerns resolved with the development of DNS in the 1980s?

- **Answered in Section 3.1:** HOSTS.TXT limitations encompassed single point of failure, consistency challenges, and scaling constraints, addressed by DNS's distributed hierarchy, delegation, and caching strategies.

Subquestion 2: Considering the extensive evolution of DNS specifications via several RFCs over the years, how has DNS increased in complexity and scope, and what implications does this have for administrators and implementers?

- **Answered in Section 3.2:** The expansion from simple name resolution to multi-purpose infrastructure through decades of RFC standardization introduced record type proliferation, DNSSEC complexity, and operational burdens for implementers and administrators.

Subquestion 3: Which DNS security mechanisms or proposed modifications were considered in the past but never widely adopted, and what consequences might their deployment have had for today's Internet?

- **Answered in Section 3.3:** TCP-as-primary and early cryptographic authentication were considered but rejected, creating technical debt that required later complex additions like DNSSEC and encrypted transports.

Subquestion 4: When compared to the original DNS design, how much do the security improvements brought about by contemporary encrypted DNS protocols (DoT and DoH) increase latency and system complexity?

- **Answered in Section 6:** The ablation study offers a systematic assessment of the baseline overhead. Introduced through encrypted transports, it is demonstrated that DNS over HTTPS enhances the mean latency increased by 31.6% and DNS over TLS improved by 25.6% compared to Traditional DNS. The results indicate that encryption introduces quantifiable performance costs, although the relative differences between DoH and DoT are significant less than indicated by the initial experiment and primarily due to methodological factors were adjusted in the ablation setup. The analysis continues illustrates that encrypted DNS protocols introduce structural complexity through establishment of connections, processing of cryptographic functions, and transport-layer protocols requirements, in contrast to the lightweight, stateless model of the original Architecture of DNS.

The DNS is still evolving, striking a balance between the simplicity and scalability of its initial architecture and the needs of modern security and privacy. The empirical basis for comprehending these evolutionary trade-offs and guiding the creation of future protocols is provided by this study.

A Multi-interval Experiment Details

A.1 Methodology

We planned and carried out a multi-interval measurement study in order to thoroughly assess the operational consistency and performance traits of encrypted DNS protocols under various network circumstances. This method preserved strict experimental controls while allowing for a methodical examination of protocol behavior across diurnal traffic patterns.

A.2 Experimental Design

Our study employed a factorial approach with repeated assessments over critical temporal intervals to measure protocol performance under expected changes in network congestion:

- **DNS Protocols**: Incorporate classical DNS (Do53 over UDP port 53), DNS over HTTPS (DoH over HTTPS port 443), and DNS over TLS (DoT over TCP port 853).
- **Temporal Sampling**: According to traffic trends from AMS-IX, one of the largest Internet exchange point in Europe [25], four daily measurement intervals were designated: 04:00 UTC (minimal traffic), 10:00 UTC (morning business hours), 13:00 UTC (post-lunch moderate traffic), and 20:00 UTC (evening peak traffic).
- **Test Domains**: Five domains exemplifying distinct sorts of Internet services: `example.com` (documentation), `google.com` (search and CDN), `github.com` (development platform), `nytimes.com` (news media), and `icann.org` (Internet governance). This variability reflects performance across various DNS hosting infrastructures and operating scales, offering a representative sample of actual usage patterns.
- **Measurement Scale**: 40 experimental runs, comprising 10 runs per time slot, generated 12,000 unique DNS measurements across four strategically scheduled intervals. Each whole measurement session necessitated around five minutes for sequential execution.

A.3 Measurement Infrastructure

In order to eliminate hardware variability and ensure measurement consistency, all experiments were carried out from a consistent perspective utilizing an Ubuntu server equipped with an Intel Xeon E-2224 processor and 15GB of RAM. We used common Linux utilities with particular setups:

- `dig` for Google Public DNS Traditional DNS queries (8.8.8.8)
- `dig +tls` For DoT queries to Cloudflare DNS (1.1.1.1)
- `curl` for DoH queries to Cloudflare DNS API (`https://cloudflare-dns.com/dns-query`)

Timing measurements utilized high-resolution system timers using Python's `time.time()` function, recording the total end-to-end latency from command initiation to response finalization. The timer commenced immediately upon executing the query command through Python's `subprocess.run()` and concluded upon the complete reception and processing of the response. This method includes all operational overhead encountered by an end user, such as process initiation, network transmission, connection establishment, cryptographic handshakes for secure protocols, and response processing. By quantifying the overall command execution duration across all protocols with this uniform methodology, we accurately capture the genuine user-perceived latency that represents real-world DNS resolution using normal system utilities.

Provider Selection Rationale The deliberate use of multiple DNS providers (Google for Traditional DNS and Cloudflare for encrypted DNS protocols) was intended to mitigate provider-specific optimizations and to establish a performance baseline that is more representative of real-world deployment practices.

Measurement Tool Considerations: Due to implementation availability, different tools were used for protocol measurement (`curl` for DoH, `dig` for Traditional DNS/DoT). This represents standard operating procedures for every technique, even if it may introduce tool-specific variation.

A.4 Data Collection Procedure

A methodical multi-interval approach was used in the conduct of the experiment:

- **Session Structure**: To avoid interference, each temporal session consisted of ten consecutive measurement runs spaced 30 seconds apart.
- **Run Composition**: 75 measurements were made for each run (5 domains × 3 methods × 5 repeats).
- **Quality Assurance**: Automated error reporting, data integrity checks, and thorough success validation via response content analysis.
- **Total Dataset**: 40 runs × 75 measurements = 3,000 measurements per time slot × 4 time slots = 12,000 validated DNS queries.

No failed queries were detected during the experiment across any of the protocols or measurement intervals.

A.5 Results

The empirical results of our multi-interval DNS performance investigation, which included 12,000 measurements taken over several time periods, are shown in this section. To describe protocol performance, distribution properties, temporal stability, and encryption overhead, the research makes use of visuals and quantitative tables.

A.6 An overview of performance

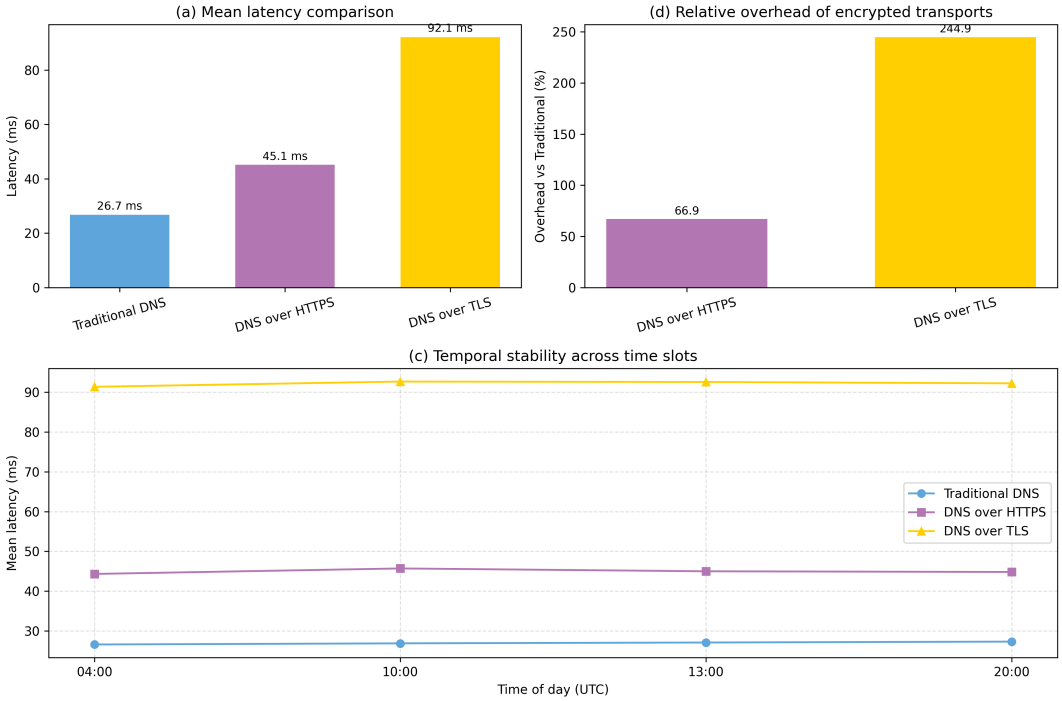

Figure 2: Detailed multi-interval performance view of the DNS protocols: (a) mean latency comparison showing Traditional DNS (26.7 ms), DNS over HTTPS (45.1 ms), and DNS over TLS (92.1 ms); (c) temporal stability across the four measurement intervals, indicating limited diurnal variation; and (d) relative latency overhead of the encrypted transports, expressing the performance cost of DNS over HTTPS and DNS over TLS compared to Traditional DNS.

Figure 2 summarizes the multi-interval measurements in a compact form. Panel (a) highlights the absolute mean latency gap between the three protocols, panel (c) shows that these averages remain broadly stable across the four time slots, and panel (d) expresses the cost of encryption as percentage overhead relative to Traditional DNS. The patterns observed in these panels reflect both protocol-level mechanics and measurement-tool differences (Traditional DNS and DoT measured via `dig`, DoH via `curl`), as discussed further in the Limitations (Section 6.5).

A.7 Features of the Distribution and Analysis of Reliability

Traditional DNS shows a low median latency of 23.3 ms and a relatively narrow spread across the measurements. DNS over HTTPS has a higher median latency of 34.9 ms and a wider distribution, and DNS over TLS is again slower, with a median of 93.9 ms.

Within the DNS over HTTPS measurements we also see one extreme outlier with a latency above 1400 ms. This single value is about 0.02% of all samples and therefore does not change the main comparison between the protocols, but we still include it here for completeness. In contrast, Traditional DNS remains bounded in its worst-case behaviour, with a maximum observed latency

of 156.8 ms, while DNS over TLS shows consistently higher but more predictable latency without similar extreme values.

A.8 Trade-offs between encryption overhead and performance

The encryption overhead and overall performance trade-offs in the DNS protocol's transition to encrypted transports are quantified in Figure 2. A limited maximum latency, Traditional DNS provides the best performance and dependability. A balanced strategy with minimal performance costs is offered by DNS over HTTPS, however it occasionally displays extreme outliers that raise questions about reliability under particular network circumstances. DNS over TLS preserves predictable behavior while prioritizing security at the expense of considerable performance reduction.

The existence of severe outliers in DNS over HTTPS, notwithstanding its statistical rarity (0.02% incidence), suggests that while the protocol functions well for most requests, there may be significant performance deterioration in extreme circumstances. TCP-based protocols such as DoT and DoH offer intrinsic stability via packet loss detection and recovery techniques, a feature absent in classic DNS via UDP. The trade-off between UDP's reduced latency and TCP's reliability highlights the necessity of assessing encrypted DNS protocols for critical applications, taking into account both standard performance and extreme cases.

This study reveals that under cold-start conditions, the transition of the DNS protocol to encrypted transports incurs measurable performance costs, with each protocol presenting unique trade-offs among security, performance consistency, and outlier characteristics. It is essential to recognize that real-world stub resolvers employing persistent connections and query aggregation techniques, such as Nagle's method [26], may enhance performance in environments with elevated query volumes. Moreover, although encrypted DNS queries are larger and demand increased bandwidth, advancements in network infrastructure such as gigabit Ethernet, enhanced TCP stack implementations, and the widespread adoption of broadband may alleviate some of these performance effects in contemporary network settings.

Taken together, the measurements show that encrypted DNS introduces a measurable latency overhead. This reflects the broader trade-off in which additional privacy and TCP-based reliability are obtained at the expense of lower raw resolution performance.

References

[1] S. Bellovin. "Using the Domain Name System for System Break-ins". In: *Proceedings of the 5th USENIX UNIX Security Symposium*. 1995.

[2] R. Arends et al. *DNS Security Introduction and Requirements*. RFC 4033. Mar. 2005.

[3] Z. Hu et al. *Specification for DNS over Transport Layer Security (TLS)*. RFC 7858. May 2016.

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

[5] T. Wicinski. *DNS Privacy Considerations*. RFC 9076. July 2021.

[6] APNIC Labs. *The prevalence of DNS over HTTPS*. `https://blog.apnic.net/2021/09/13/the-prevalence-of-dns-over-https/`. Accessed: 2026-01-12. Sept. 2021.

[7] P.V. Mockapetris and K.J. Dunlap. "Development of the Domain Name System". In: *ACM SIGCOMM Computer Communication Review* 18.4 (Aug. 1988), pp. 123–133.

[8] P. Mockapetris. *Domain names - implementation and specification*. RFC 1035. Nov. 1987.

[9] R. Arends et al. *Resource Records for the DNS Security Extensions*. RFC 4034. Mar. 2005.

[10] R. Arends et al. *Protocol Modifications for the DNS Security Extensions*. RFC 4035. Mar. 2005.

[11] O. Kolkman and R. Gieben. *DNSSEC Operational Practices*. RFC 4641. Sept. 2006.

[12] C. Huitema et al. *DNS over Dedicated QUIC Connections*. RFC 9250. May 2022.

[13] B. Carpenter. *Architectural Principles of the Internet*. RFC 1958. June 1996.

[14] J. Bonneau et al. "SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies". In: *IEEE Symposium on Security and Privacy*. 2015.

[15] L. Zhang et al. "Named Data Networking". In: *ACM SIGCOMM Computer Communication Review* 44.3 (2014), pp. 66–73.

[16] IETF. *Deleg DNS/Name Resolution Evolution Working Group*. https://datatracker.ietf.org/wg/deleg/about/. 2025.

[17] IETF Deleg Working Group. *Requirements for a New Delegation and Name Resolution Protocol*. https://www.ietf.org/archive/id/draft-ietf-deleg-requirements-04.html. 2023.

[18] S. Thomson et al. *DNS Extensions to Support IP Version 6*. RFC 3596. Oct. 2003.

[19] J. Damas et al. *Extension Mechanisms for DNS (EDNS(0))*. RFC 6891. Apr. 2013.

[20] Bert Hubert. *The DNS Camel, or the rise of DNS complexity*. https://www.youtube.com/watch?v=8N_PO3s_Z24&t=4756s. IETF 101. 2018.

[21] Daniel J. Bernstein. *DNSCurve: Usable security for DNS*. https://dnscurve.org/. 2009.

[22] Peter W. Shor. *Algorithms for quantum computation: discrete logarithms and factoring*. 1994.

[23] Mike Kosek et al. *One to Rule them All? A First Look at DNS over QUIC*. arXiv:2202.02987 (PAM 2022). https://arxiv.org/abs/2202.02987. 2022.

[24] Warren Kumari et al. *Making Local Root a Best Current Practice*. Internet-Draft draft-wkumari-dnsop-localroot-bcp. Work in progress. IETF, 2026.

[25] AMS-IX. *AMS-IX Traffic Statistics*. https://www.ams-ix.net/ams/documentation/total-stats. Accessed: 2024. 2024.

[26] John Nagle. *Congestion Control in IP/TCP Internetworks*. RFC 896. Jan. 1984.