Evolution and challenges of DNS-based CDNs

Zheng Wang\textsuperscript{a,}\textsuperscript{*}, Jun Huang\textsuperscript{b}, Scott Rose\textsuperscript{a}

\textsuperscript{a}Advanced Network Technologies Division, National Institute of Standards and Technology, USA

\textsuperscript{b}School of Computer Science, Chongqing University of Posts and Telecommunications, China

Abstract

DNS-based server redirecting is considered the most popular means of deploying CDNs. However, with the increasing use of remote DNS, DNS-based CDNs face a great challenge in performance degradation. To address this issue, encouraging progress has been made in both industry and research communities. In this article, state-of-art solutions for the remote DNS problem are discussed at first. Next, privacy concerns about DNS-based CDNs, including client location as well as redirection privacy, are identified and a representative solution is summarized. Finally, the solution is compared to those in prior works under different measures, and a discussion on DNS-based CDN applications is provided. A model is also established to deepen the understanding of CDN performance. We believe that this survey will shed light on the application of DNS-based CDNs, and it is expected to provide design guidelines to CDN service providers.

Keywords

Content delivery network; DNS-Based server redirecting; Remote DNS; DNS privacy

1. Introduction

Over the past three decades, the world has witnessed the rapid growth of the internet and the enormously enriched content delivered over it. The ever-increasing demands from users place a heavy burden on the limited networking and computing resources of content providers. Thus, new approaches or paradigms are needed to address the emerging challenges in delivering content. For example, popular web services are often vulnerable to the so-called flash crowing problem [1,2]. When many users simultaneously access the same web site, the request load may overwhelm the web servers. Flash crowd may cause slowed responsiveness, diminished availability, or even website crashes.

Content Delivery Networks (CDNs) were proposed to solve content delivery bottlenecks, such as scalability, reliability, and performance. In CDNs, content is replicated from the original server to surrogate servers distributed over the internet [3,4]. Surrogate servers are placed at optimal sites, e.g., the edge of the internet infrastructure [39], providing improved...
connectivity to the nearby end users. In this manner, contents are transparently, rapidly, and reliably delivered to end users.

Request routing is a critical issue in CDNs. It directs end users to optimal surrogate servers per specific metrics or policies. Typically, the design of CDN request routing involves:

- **Server selection mechanism.** The mechanism determines the optimal surrogate server for an end user. A server selection algorithm may use a set of metrics, such as network utilization, user perceived latency, network distance, and surrogate server load. Because the nearest surrogate server is commonly considered to best serve end users, end user location is typically used as the decisive parameter in request routing [5–7]. In practice, most CDN servers simply obtain the end user location from the source IP address of the incoming CDN request.

- **Server redirecting mechanism.** The mechanism informs the end user about the optimal surrogate server selected by the server selection mechanism. Among all server redirecting mechanisms, DNS-based server redirecting is the most popular. It makes full use of the existing DNS infrastructure and thus enables quick and easy deployment. Despite its merits, DNS-based server redirecting is increasingly challenged by the remote DNS issue and privacy concerns.

The remote DNS issue arises from the false assumption that a DNS recursive server is in proximity to its clients. When a client queries a remote DNS recursive server, the DNS recursive server contacts the DNS authoritative server for an answer. In determining the optimal response, the DNS authoritative server infers the client location from incoming requests, and therefore uses the source IP address of the DNS recursive server rather than that of the client. Given the location mismatch between the client and the remote DNS recursive server, the DNS recursive server tends to misrepresent the client location to the DNS authoritative server. Thus, the server selection result is likely to deviate from the optimal. Recently, various solutions have been proposed to overcome the remote DNS problem; the most recognized of these is known as a EDNS-Client-Subnet(ECS) DNS extension [21]. ECS explicitly indicates client location information in an option of the DNS message. By inspecting the ECS option rather than the source IP address of the arriving DNS request, the DNS authoritative server can identify the client location.

The cost incurred by adopting ECS is loss of privacy. ECS makes client location information visible to the DNS authoritative server and on-path eavesdroppers. Thus, client location privacy, which is well protected by remote DNS, is almost invalidated by ECS. The other problem with ECS is the vulnerability of redirection enumeration. If ECS is adopted, potential adversaries may enumerate CDN mapping at an affordable cost. As a result, CDN mapping policy, which is often assumed to be private to the CDN provider, is actually exposed to the public.

In this work, we survey the state-of-the-art DNS-based CDN technologies, and provide insights into the emerging challenges and solutions. We expect this work to be the first to systematically discuss DNS-based CDNs and provide design guidelines to CDN service...
providers. The remainder of this paper is structured as follows. Section 2 overviews the existing approaches for CDN server directing with a focus on DNS-based server redirecting. Section 3 discusses the remote DNS problem and presents the state-of-art solutions to it. Section 4 discusses privacy concerns and analyzes possible responses. Based on the discussions above, Section 5 surveys DNS-based server redirecting solutions via comparisons. Section 6 summarizes and concludes the work.

2. Existing approaches

Existing approaches for server directing include HTTP redirection, URL rewriting, anycast, and DNS-based server redirecting.

2.1. HTTP redirection

HTTP redirection allows a web server to propagate the server selection result to the end user via HTTP headers [8,9]. Hence, the end user can be redirected to the optimal server by following the response generated from the Web server. The weakness of HTTP redirection lies in its reliance on support from the server side to the client side. Moreover, HTTP redirection is not a lightweight solution because an extra round-trip delay is introduced in every HTTP session, and the processing overheads of HTTP are non-trivial.

2.2. URL rewriting

This technique mainly targets content delivery with embedded objects in response to client requests [10,11]. In URL rewriting, the origin server rewrites the generated pages’ URL links in order to indicate the best surrogate server. Following the rewritten responses, the client can be optimally redirected. The major cost of URL rewriting is the delay for URL-parsing. Worse yet, the cost is likely to increase because the rewritten URL is non-cacheable for the end user.

2.3. Anycast

Anycast is a network layer technology for transparent server selection and redirecting. In this approach, the same IP address is assigned to multiple surrogate servers located distributively. When the client sends requests to the IP address, the requests will be routed to the nearest surrogate server defined by the routing policy. Note that the server redirecting enabled by anycast is technically not controllable for content providers; therefore, anycast usage is controversial. On the one hand, its transparency to both content providers and end users may be a benefit, as content providers are liberated from request-routing overhead. On the other hand, content providers may lose some server selection flexibility. Consider a scenario in which anycast forwards requests to the nearest (yet overloaded) server, by simply respecting a distance-based routing policy. Additionally, internet routing fluctuations may negatively impact the stability of IP anycast. In a measurement study [12] on the performance of anycast CDN, the simplicity of anycast was found to be gained by sacrificing precise redirection. The measurement showed that the anycast CDN directed roughly 20% of clients to suboptimal front ends. A load-aware anycast CDN architecture [13,14] was proposed as a solution to the problem. However, its efficacy is limited by session disruptions that plague anycast. Another work addressed the session disruption issue.
by dedicating a transport level mechanism [15]. However, both the sophisticated routing
control and the innovated transport protocol are too radical to deploy rapidly and efficiently.

2.4. DNS-based server redirecting

This mechanism piggybacks redirection in responses to DNS queries. It is a flexible and
lightweight solution and fully compatible with the existing DNS infrastructure. Thus, it has
gained significant popularity in major CDN networks such as Akamai [16], Limelight
Networks [17], and Mirror Image [18].

As an indispensable substrate of today’s Internet, the DNS functions as a globally distributed
directory. Its primary role is to map domain names to the corresponding IP addresses. In a
typical DNS session, the client first sends a request to its designated recursive DNS server. A
recursive DNS server is either operated by ISPs (Internet Service Providers) or provided as a
public service for any users around the globe (e.g., the Google DNS service hosted at 8.8.8.8
and 8.8.4.4). The recursive DNS server resolves the request on behalf of the client. It
iteratively traverses relevant authoritative DNS servers following the DNS tree until the final
answer is obtained. The recursive DNS server not only delivers the answer to the client but
also caches it for future queries. In the DNS, a name-to-address mapping is represented by a
DNS record. Each DNS record contains a Time-To-Live (TTL) field that specifies how long
the recursive DNS server caches it.

In DNS-based server redirecting, a domain name associated with the content (e.g., the web
pages available at a web site) offered by the content provider is hosted on the authoritative
servers. Currently, it is common practice to outsource the task of content delivery a CDN
provider. A CDN provider, which disseminates the content on behalf of the content provider,
is likely to operate the authoritative servers as an integral part of its content delivery service.

Many content providers prefer to take care of their brand valued domain names themselves,
and a commercial CDN provider tends to manage its CDN authoritative servers for its
customers below one domain or several domains owned by itself. Here we call a domain
name owned by a content provider a “content domain name” and a domain name owned by a
CDN provider a “CDN domain name.” The separation between a content domain name and
a CDN domain name in some ways decouples the DNS management between a content
provider and a CDN provider, and therefore facilitates independent and stable CDN
operation.

To associate a content domain name with a CDN domain name, a CNAME-type DNS record
can be registered at the content provider to point an alias name, namely a content domain
name, to a canonical name, namely a CDN domain name. When a client sends a request for
a content domain name, the content providers DNS authoritative server answers with the
CNAME record referring to the CDN domain name. The client is then redirected to the CDN
provider’s DNS authoritative server which resolves the CDN domain name.

In DNS-based server redirecting, the server selection result is delivered to the end user by
resolving the domain name to the IP address of the selected surrogate server. To do this, the
CDN authoritative server dynamically updates its DNS record via the server selection
algorithm, always pointing to the optimal surrogate servers. The dynamic name resolution is usually conducted according to the sources of incoming requests. Owing to the caching effects, a small TTL value is usually preferable for the DNS record, to ensure server redirecting promptness [29].

Fig. 1 shows an example of DNS-based server redirecting. First, the client requests the DNS recursive server for the IP address of www.zoo.com (Step (1)). The DNS recursive server is informed by the content provider’s authoritative server that www.zoo.com is an alias of www.cdn.com (Step (2)). The DNS recursive server then obtains the result from the CDN provider’s authoritative server, it shows that www.cdn.com is hosted at an optimal surrogate server whose IP address is 1.1.1.1 (Step (3)). When the DNS recursive server forwards the answer to the client (Step (4)), the client accesses the content delivered from the surrogate server at 1.1.1.1 (Step (5)). All surrogate servers connected by the CDN backbone are synchronized from the original server at the content provider.

DNS-based server redirecting has several advantages over other server redirecting methods:

- **Transparency.** It is fully transparent to the end users. The CDN service hosted on different servers is accessible from one domain name. Moreover, the dynamic mapping between the domain name and different servers is invisible to end users.

- **Simplicity.** It can be seamlessly incorporated into the DNS resolution process. In particular, the DNS infrastructure is so universally available that both content providers and end users are saved from heavy investment.

- **Flexibility.** Server redirecting can be flexibly managed by adjusting the TTL value of the DNS record. A large TTL value is favorable for more static server redirecting and lowered authoritative server load. Small TTL values allow for more dynamic server redirecting. In an extreme case, a zero TTL allows up-to-date direction for every individual DNS request.

3. Remote DNS problem and solutions

3.1. Remote DNS problem

The major limitation of DNS-based server redirecting is a false assumption regarding the recursive DNS server’s proximity to the client. The CDN provider typically determines the optimal server selection based on the source IP address of the DNS request (probably combined with other information). However, the source IP address conveyed to the CDN provider is that of the DNS recursive server, rather than that of the client. This is because the CDN authoritative server is queried by the DNS recursive server, not by the client. The server selection result may be only slightly problematic when the DNS recursive server and client are sufficiently proximate. However, the result may be sub-optimal if the proximity assumption does not hold.

The use of remote DNS recursive servers has increased significantly in recent years. Among the servers, some are public DNS servers offered by major internet enterprises such as Google, OpenDNS, Norton, etc. Many users have switched to those public DNS services because of their better DNS performance in terms of availability, stability, and security.
According to a 2012 study [28], the public DNS user base grew by 27% annually, and 8.6% of users in the sample relied on a public DNS service.

DNS-based server redirecting is likely to perform poorly when the client uses a remote DNS recursive server. For example, an end user located in the US can experience slow responses when accessing the CDN website, if choosing a DNS recursive server located in Europe. The problem is caused by the selected CDN server in Europe. The server selection is optimized based on the location of the DNS recursive server but is essentially suboptimized for the location of the end user. A recent study [19] on Africa’s internet infrastructure reveals that the use of distant DNS servers contributed to in excess of 100 ms of DNS resolution delay for approximately 50% of the measurement probes. In another study, Otto et al. [28] assessed the end-to-end impact of using remote DNS services on CDN performance. To compare performance, server redirections performed by clients, ISP DNS, and public DNS were measured at a set of locations. ISP DNS was shown to have some similarity with clients in at least 80% of locations, and there was no similarity between public DNS and client for 90% of locations. The differences in similarity were explained by the increased distance to the client using public DNS. For HTTP performance, public DNS was found to yield doubled latencies compared with clients and ISP DNS. As one natural explanation, the degraded HTTP performance was correlated with sub-optimal server redirections by remote DNS. Those results showed that remote DNS significantly impacts the client’s perceived CDN performance.

Hidden behind local DNS servers, clusters of hosts are inaccessible to content providers in terms of features such as size and geographical compactness. Thus, without knowledge about the local clusters’ properties, content delivery using local DNS servers is likely to be suboptimized [20].

3.2. Solutions

3.2.1. ECS—A recent solution to the remote DNS problem is the EDNS-Client-Subnet DNS extension (ECS) [21]. It was proposed by Google and agreed to by IETF. The proposed extension allows a recursive server to deliver client location information to an authoritative server.

ECS is based on an EDNS0 DNS extension [22], which is introduced to include optional data in a DNS message. In ECS, an OPT record in the DNS message is used to convey the client’s IP address prefix and the scope. An ECS-enabled client includes that ECS OPT record in its request. Additionally, an ECS-aware DNS recursive server copies the ECS OPT record from the client’s request when sending out its request. The ECS-aware DNS authoritative server can determine the CDN server selection using the client location indicated by the ECS OPT record.

ECS demands substantial deployment costs. It calls for joint efforts by all parties involved in DNS transactions, including the following:
End users should upgrade theirstub resolvers (often embedded in web browsers), email clients, and other applications to support ECS. This should be serviced by software vendors and awareness of end users.

DNS recursive servers should provide ECS support for their clients. Considering the fact that ECS-capable DNS server implementation is barely available, this is a non-trivial task.

DNS authoritative servers should be compatible with ECS in support of content delivery. Even for non-CDN authoritative servers, ECS payloads should not be identified as format errors or malicious data.

Intermediate systems and DNS middle-boxes should be compatible with ECS and at least forward ECS payloads unmodified.

Cache efficiency is another concern with ECS. In the conventional DNS caching model, one DNS question is basically mapped to one corresponding DNS record in cache. This simple caching mechanism has proven to be effective and efficient in DNS practice from the 1980’s through today [30]. ECS expands the one-to-one caching model by introducing another dimension, namely, the ECS scope. In ECS, a set of items in cache are allowed to share one DNS record yet differ in the ECS scope. A diversity of ECS scopes is often needed to ensure good content delivery performance for global end users. Hence, by adopting ECS, the DNS cache will be expanded and complicated by a factor of the number of ECS scopes. The vulnerability may be exploited in DoS attacks, which aim at bypassing caching and flooding authoritative servers [37,38]. A DNS privacy study [32] pinpointed the limitation, which applies to any end-to-end DNS proposal. This limitation is a prohibitive traffic overhead on the name servers, caused by non-interoperability with the caches.

The transition to ECS will be challenging in terms of handling coexisting ECS compliance and ECS non-compliance. In the current ECS extension, an upstream party has no means of signaling its ECS support to a downstream party. For example, ECS compliant clients tend to send ECS queries by default to DNS recursive server that are non-ECS-compliant, causing an unnecessary waste of ECS payload. Even worse an ECS-compliant DNS recursive server contacts an ECS non-compliance DNS authoritative server with ECS payload. In that case, the client’s private data is unnecessarily leaked in the path between the recursive server and the authoritative server, because the latter does not support ECS at all.

3.2.2. Name extension—To bypass the overhead and complexity of handling ECS, an alternative proposal is to encode the client’s location information in the DNS query name [23]. In the proposal, the client sends a request for a specific query name, which is constructed by prefixing the original query name onto the client’s location information. The DNS recursive server handles that modified query name just as it would handle an unmodified one, and it is thus kept transparent. If it supports extended query names, the DNS authoritative server retrieves the client’s location information from that query and thereby returns the CDN server selection. As an end-to-end extension between the client and the DNS authoritative server, the approach does not require support from all intermediate devices and servers. Thus, the obstacles impeding its adoption are greatly reduced compared with ECS.
3.2.3. **Direct resolution**—Considering the low adoption level of ECS, another solution is to use a client-side resolver that directly contacts the CDN authoritative server [24]. The so-called Direct Resolution approach allows a recursive DNS server to translate content domain names to CDN names and then obtain the authoritative CDN server. The final CDN redirection result is then fetched by the client itself rather than by the recursive server. Because redirection by Direct Resolution is based on the client location, it is better optimized compared to using a remote DNS recursive server or even a local DNS recursive server. However, the client-side resolver’s job is complicated in two ways: one is the increased number of queries involved in resolving a CDN name; the other is the caching overheads. Another cost of Direct Resolution is reduced privacy. More private information about the client location is expected to be exposed by Direct Resolution, because the client’s full IP address is visible in a DNS request. In contrast, only the client’s IP address prefix is leaked by the ECS extension.

4. **Privacy concerns and solution**

4.1. **Location privacy**

While ECS improves CDN performance by exposing client location information, it raises concerns over privacy. In the conventional DNS model, the client is well hidden from the DNS authoritative server by the DNS recursive server. For example, DNS authoritative servers for popular domains such as www.google.com, www.facebook.com, and www.amazon.com, have no information regarding which end users actually query them. Additionally, DNS authoritative server operators are unlikely to associate an incoming DNS request to its originating end user. Thus, the end user’s behavior is largely kept private from the DNS authoritative server. In the context of increased DNS privacy concerns in recent years [32–34,36], the conventional DNS model protects end users from being directly monitored, recorded, and analyzed by DNS authoritative servers. However, ECS does reveal private information, as client location information is hardly private. While using an IP address prefix rather than an exact IP address is recommended in the ECS option, a long IP address prefix is generally preferred to ensure optimal CDN server selection. Here a tradeoff exists in that longer IP addresses, and allow better CDN performance, but reduce user privacy. Thus, the end user must trade privacy for CDN performance.

4.2. **Redirection privacy**

To optimize user-perceived latency, some major content providers such as Google invest heavily in building content delivery networks and developing sophisticated CDN mapping algorithms. Individual or small sets of redirection mappings are commonly considered readily available and open to the public. However, mapping out entire content delivery networks by enumerating redirection mapping is likely to infringe on the content provider’s privacy. Complete redirection mapping information may be utilized in orchestrated DDoS attacks against the CDN infrastructure, as well as for other offensive purposes. Therefore, the risks of redirection privacy are either undesired or unexpected by content providers.

Generally, traversing a 32-bit IPv4 address space (excluding private IP addresses) is almost impossible for most attackers, given the enormous number of queries. However, like any
other cluster based redirection mechanism, ECS may make this possible because it greatly reduces the number of queries. Calder et al. [26] used ECS-enabled queries to measure the redirection mapping of the Google web service. Based on the enumeration and geolocation results, they demonstrated the growth of Google’s serving infrastructure and acquired its content serving strategy. By using routable/24 client prefixes, queries against Google were reported as taking about a day to enumerate. The efficiency of ECS-based redirection enumeration highlights the privacy issue with any IP block-based redirection mechanism. Similarly, Streibelt et al. [27] showed measurement opportunities to uncover details about CDN providers’ operational practices with the support of ECS.

4.3. Solution

As a response to the privacy concerns incurred by ECS, a client pseudononymizing scheme was proposed in Ref. [25]. In the proposal, the client does not use its IP address prefix as location information in its DNS requests. Instead, it uses its pseudononymizing identifier to protect its location privacy. The mapping between the IP address and the pseudononymizing identifier is registered, maintained, and resolved at the pseudononymizing registry of a trustworthy third party. Note that all on-path parties, including DNS authoritative servers, have no access to client location (through translating the client’s pseudononymizing identifier) unless they are authorized. A trustworthy third party is tasked with CDN server selection on behalf of the authoritative servers in its pseudononymizing optimizer. When accessing the pseudononymizing service, a client first registers the mapping between its IP address and its pseudononymizing identifier to the pseudononymizing registry in a secure manner (Step (1) in Fig. 2). It can then send DNS requests using the pseudononymizing identifier (Step (2) in Fig. 2). Note that all on-path parties, including DNS authoritative servers, have no access to client location (through translating the client’s pseudononymizing identifier) unless they are authorized. The authoritative server forwards the pseudonymized requests to the third party (Step (3) in Fig. 2), which looks up the mapping to the IP address in a secure manner (Step (4) in Fig. 2) and returns the CDN server selection (Step (5) in Fig. 2). Finally, the authoritative server delivers the CDN server selection to the client (Step (6) in Fig. 2).

Despite its privacy-preserving advantages, client pseudononymizing may suffer from the bottleneck of CDN redirection delay. Compared with common DNS operations, an extra delay is introduced to DNS authoritative servers’ response latency because they ask an external trustworthy third party for the redirection results. Clouding the trustworthy third party’s infrastructure may be an effective way of minimizing the extra DNS lookup delay. Besides investing in the network infrastructure, protocol level optimizations also help. For example, some extra redirection results may be piggybacked on the response if those results are intelligently predicted to be queries that will be used shortly thereafter; the trustworthy third party may set a validity time for each redirection mapping so that the authoritative server can immediately respond without an external request. Additionally, the authoritative server may attach a validity time to its response, leaving the response cacheable by the recursive server and client.
5. Comparison of existing solutions

In this section, we compare all DNS-based solutions in CDN operations, as well as those proposed in recent years. A total of seven solutions are identified. For each solution, we identify its merits and demerits using the following five measures. The comparison is shown in Table 1.

5.1. Metrics

5.1.1. Client complexity—The implementation of the DNS client that accesses the DNS recursive server is usually referred to as the stub resolver. Stub resolvers are installed by default on platforms such as Windows, Linux, and Unix. Stub resolvers are assumed to be simplified enough to be affordable for lightweight and cost-constrained devices or systems such as emerging IoT (Internet of Things) gadgets and mobile devices. Because they rely heavily on recursive servers, stub resolvers’ tasks are limited to sending queries, interpreting responses, and resending them if unanswered. In accordance with the simplicity principle, any design attempting to place complex burdens on stub resolvers may limit their deployment.

5.1.2. Intermediate transparency—To support a CDN solution, any intermediate component located in the path between the querier, namely the DNS client, and the responder, namely the DNS authoritative server, should be able to interpret, process, and forward the messages compliantly whenever necessary. If some middle-box (e.g., a DNS recursive server) must be substantially upgraded or redesigned, the solution’s adoption would be costly. In contrast, a solution with transparency to any intermediate component has the advantage of easy deployment and excellent compatibility.

5.1.3. CDN performance—For location-based CDN server selection, CDN performance is largely determined by the accuracy of the client location information conveyed to the DNS authoritative servers. For simplicity, we merely consider best-effort location exposure for each solution, without factoring in privacy concerns. In particular, ECS and Name Extension are both supposed to use full client IP addresses in our comparison.

5.1.4. Client location privacy—For this privacy concern, we consider the client location information leaked in the path between the DNS recursive server and the authoritative server. Ideally, any in-path eavesdropper or authoritative server can learn a client’s location from DNS messages. Thus, the extent of a client location privacy leak depends on the distance between the exposed client location and the actual client location.

5.1.5. Redirection privacy—This privacy concern is about uncovering a full snapshot of CDN location-to-server mappings. That snapshot reveals private data about the CDN infrastructure and the clustering clients. Compared with conventional DNS, any scope-or prefix-based aggregated redirection mechanism enables easy and fast enumeration opportunities to observe the operational practices of content providers.
5.2. Existing solutions in comparison

First, we identify three solutions that still use conventional DNS but vary in the placement of DNS recursive servers.

5.2.1. Local server—One common practice is that the client uses a local DNS recursive server to access a CDN service. Such local servers are either provided by ISPs (Internet Service Providers) as default DNS servers or manually configured by users for reliable and fast name resolution service. The local server complies with existing clients and intermediate components. Hence, the simplicity of client and intermediate transparency is retained.

In comparison with remote servers, local servers are believed to provide better content delivery performance by maintaining the vicinity of the end-user and its DNS resolver. In a study of DNS performance, the deployments of 50 commercial ISPs are DNS deployment is compared against widely used third-party DNS resolvers, namely GoogleDNS and OpenDNS [31]. It was observed that third-party DNS resolvers did not redirect users toward content available within the ISP, contrary to the local DNS resolvers.

However, the proximity between local servers and clients is not always ensured. For example, in the context of cellular networks, cellular DNS was found to be unsuitable for client localization [35]. As shown by the measurement, even public DNS outperformed cellular DNS in terms of CDN replica performance during 75% of the measurement time. Thus, CDN performance and client location privacy levels in local servers are both medium, in comparison with other solutions.

As a one-to-one mapping based solution, local servers protect redirection privacy well.

5.2.2. Remote server—Remote DNS recursive servers have been increasingly used in recent years. They are mostly utilized as public DNS services. Remote servers provide advantages in client complexity and intermediate transparency. Owing to the location mismatches between clients and remote servers, CDN performance is often considered poor. However, the location mismatches also protect client location privacy. Similar to local server, remote servers ensure redirection privacy.

5.2.3. Client server—Hosting DNS recursive servers at clients is rare in practice, partly because it is uneconomical in resource constrained circumstances. The pros include intermediate transparency, improved CDN performance given by the co-location of client and server, and good redirection privacy. The cons are high client complexity and poor client location privacy caused by the co-located server.

Next, we compare existing proposals that more or less modify the conventional DNS.

5.2.4. ECS—ECS introduces minor extra overhead to clients by adding the ECS option. It violates intermediate transparency by claiming the modified recursive server. It succeeds in optimizing CDN performance but performs poorly in preserving client location privacy and redirection privacy.
5.2.5. **Direct resolution**—Direct Resolution still relies on DNS recursive servers to find authoritative servers, but makes clients solicit the answers on their own. Thus, Direct Resolution has medium client complexity in comparison with local servers and client servers. It gains good intermediate transparency and CDN performance by sacrificing its location privacy. Because it essentially bases redirection on a single IP, it effectively respects redirection privacy.

5.2.6. **Name extension**—Name Extension makes relatively trivial changes to clients; thus, its client complexity is low. Because the client location information is coded in the canonical domain name, Name Extension can be transparent to the DNS recursive server. Similar to ECS, Name Extension exposes client location information and implements scope-based redirection. Hence, it has poor client location privacy and redirection privacy.

5.2.7. **Client pseudononymizing**—Client pseudononymizing maintains low client complexity because the pseudononymizing process at the client is lightweight. However, intermediate transparency is violated because the DNS recursive server requires modifications to accommodate client pseudononymizing. It ensures CDN performance, client location privacy, and redirection privacy.

5.3. **Modeling CDN performance**

To better illustrate CDN performance, we develop a conceptual model to evaluate the impact of location mismatches between clients and DNS recursive servers.

For optimal CDN performance, the provisioning of CDN surrogate servers often utilizes CDN replicas located in proximity to the requestors. Without loss of generality, we can assume that the distribution of distance between a DNS recursive server and a CDN surrogate server, which is denoted by \( f(x) \), follows the following conditions:

- The probability that a DNS recursive server and a CDN surrogate server are co-located is very close to zero. That is
  \[
  \lim_{x \to 0} f(x) = 0 \quad (1)
  \]

- The probability that a DNS recursive server and a CDN surrogate server are infinitely far away from each other is very close to zero. That is
  \[
  \lim_{x \to \infty} f(x) = 0 \quad (2)
  \]

- The probability increases with the distance between a DNS recursive server and a CDN surrogate server, if the distance falls below a point with the highest probability, \( x_{\text{max}} \). That is
  \[
  \frac{df(x)}{dx} > 0 \quad 0 < x < x_{\text{max}} \quad (3)
  \]
The probability decreases with the distance between the DNS recursive server and the CDN surrogate server, if the distance is above a point with the highest probability, \( x_{\text{max}} \). That is

\[
\frac{df(x)}{dx} < 0 \quad x > x_{\text{max}}
\]  

(4)

There is a point with the highest probability, \( x_{\text{max}} \), satisfying

\[
\frac{df(x)}{dx} = 0 \quad x = x_{\text{max}}
\]  

(5)

In the following discussion, we use two distributions satisfying Eqs. (1)–(5) to investigate the expected distance between the client and the CDN surrogate server, which is usually negatively correlated with the user-perceived CDN performance.

A Weibull distribution has the ability to assume the characteristics of many different types of distributions. This has made it popular among engineers. The Probability Density Function (PDF) of the Weibull distribution is given by

\[
f(x) = \frac{k}{\lambda} \left( \frac{x}{\lambda} \right)^{k-1} e^{-\left( \frac{x}{\lambda} \right)^k} \quad x \geq 0
\]  

(6)

where \( \lambda \) and \( k \) are the scale and shape parameters, respectively. Here, we let \( \lambda = 1.09 \) and \( k = 5 \), to ensure that the mean will be 1. Fig. 3 shows the PDF of the Weibull distribution.

Fig. 4 shows the expected distance between a client and a CDN surrogate server, given the distance between a DNS recursive server and a CDN surrogate server according to the Weibull distribution in Fig. 3. We can see that CDN performance degrades slowly when the location mismatch is small. E.g., the expected distance between a client and a CDN surrogate server increases by less than 5% when the DNS recursive server is 0.2 units away from the client. The CDN performance penalty grows rapidly with the increase in location mismatch. E.g., the expected distance between the client and the CDN surrogate server increases by 113% when the DNS recursive server is 2 units away from the client. The results explain the performance issue with the remote server services as well as the medium latency costs of the local server services. They also fully meet previous measurements [24, 28, 31].

The PDF of a Lognormal distribution is given by

\[
f(x) = \frac{1}{\sqrt{2\pi}\sigma x} e^{-\left[ \ln(x) - \mu \right]^2/(2\sigma^2)} \quad x \geq 0
\]  

(7)

We use a lognormal distribution with parameters \( \mu \) and \( \sigma \) as \(-0.5493\) and \(1.0481\), respectively, to ensure that the mean and variance will be 1 and 2, respectively. Fig. 5 shows the PDF of that Lognormal distribution.
Fig. 6 shows the expected distance between a client and a CDN surrogate server, given the distance between the DNS recursive server and the CDN surrogate server according to the Lognormal distribution in Fig. 5. The curve in Fig. 6 is somewhat similar to the curve in Fig. 4, except that Fig. 6 exhibits a steeper slope than Fig. 4. This is due to the “near-field” effects of the Lognormal distribution, which are not exhibited in the Weibull distribution despite the distribution equal mean values.

5.4. Existing solutions in applications

According to previous measurements [28], the three most popular public DNS services (Google, OpenDNS, and Level3) had been adopted by 8.6% of sampled users in December 2011. Also, a 27% annual growth rate in public DNS adoption was found in the survey. In particular, Google’s DNS service claimed a 74% annual increase, which made it the most-used public DNS service. Because remote public DNS services may be greatly penalized in CDN performance, there are growing concerns over it.

By now, it can be inferred that a majority of internet users still use local DNS recursive servers, although there are virtually no direct measurements or surveys on this usage pattern. This is because all other solutions listed in Table 1 are either rarely implemented or seldom deployed, except for the remote server solution.

In the following, we exemplify the usage of each solution in Table 1 using appropriate scenarios:

5.4.1. Local server—For most common internet users, a local server may be their default and probably simplest solution. They do not have to rely on installations on their machines, or external DNS infrastructure upgrades by CDN service providers or network service providers. Moreover, they would not suffer from the probably significant CDN performance costs of remote servers. If the exposure of their locations to the operator of local DNS server is not a significant concern, a local server is often the first choice for most DNS non-professionals.

5.4.2. Remote server—Remote DNS recursive servers may be preferred when local DNS servers are not available, stable, or secure. Often it is recommended that remote servers be configured as secondary or backup servers, because that increases the diversity of DNS servers. Another reason for using remote servers is that the privacy of the client location is considered a priority. Note that the biggest risk with remote servers is the increased latency experienced by CDN users.

5.4.3. Client server—Installing a DNS server on a local machine is not a difficult task for common internet users. However, the greatest issue of that configuration is the diminished cache sharing effects, which benefit shared DNS servers. Cache sharing ensures that a query from one user may be promptly responded to cache if that query was recently issued from another user. In the mechanism, the cache hit rate may be greatly enhanced by a large set of users. However, when a cache is unilaterally used by one user, the cache miss rate will be increased along with the prolonged response latency. Another negative impact is the augmented load on DNS authoritative servers.
5.4.4. **ECS**—The adoption of ECS has different meanings for different stakeholders. For the CDN service provider, ECS is preferred if all the following are met: 1) a large portion of its users rely on remote servers; 2) its CDN mapping is not private; 3) ECS gains support from a large portion of recursive servers. For the end user, ECS is the best choice if all the following are met: 1) the CDN service provider supports ECS; 2) the user uses a remote server; 3) ECS gains support from the user’s recursive server.

5.4.5. **Direct resolution**—Direct Resolution is able to strike a balance between client servers and external servers (which include remote servers and local servers) in terms of client complexity and CDN performance. Thus, its applications should also be neutralized based on our analysis above.

5.4.6. **Name extension**—Name Extension is a good choice if 1) the authoritative side supports it; 2) both client location and redirection are non-confidential.

5.4.7. **Client pseudononymizing**—Client pseudononymizing is best suited to those users who desire the strongest protection of client location privacy and redirection privacy at almost no cost of CDN performance. It is expected to grow into differentiated services, because it places heavy demands on the DNS infrastructure.

6. Conclusions and outlook

This paper has presented a comprehensive survey of DNS-based CDNs. While DNS-based server redirecting demonstrates its merits compared with other approaches, remote DNS poses critical issues in terms of CDN performance. ECS, Name Extension, and Direct Resolution are three promising proposals to address the remote DNS problem. As an increasingly concerned yet probably mostly overlooked issue, privacy is highlighted in the design of DNS-based CDN. Unlike in most prior works, client location privacy concerns and redirection privacy concerns are identified and addressed. A recently proposed client pseudononymizing scheme is the most effective method of addressing both privacy concerns. A comparative study of all exiting DNS-based CDN solutions shows that each model has advantages and disadvantages. In particular, ECS is found to lack intermediate transparency and privacy protection despite its good CDN performance, and Direct Resolution is shown to respect redirection privacy at the expense of increased client complexity.

As a fundamental substrate of today’s internet, DNS has been constantly yet cautiously improved in adapting to emerging technologies and services. It is always risky to roll out a new DNS solution on an Internet scale without fully testing and validating it. Therefore, extensive implementation, experiments, and performance modeling and evaluation of emerging DNS-based CDN solutions will be top priorities for the DNS and CDN community in the next few years.
References

[1]. Yin H, Liu X, Qiu F, Xia N, Lin C, Zhang H, Sekar V, Min G, Inside the bird’s nest: measurements of large-scale live VoD from the 2008 olympics, in: Proc. of the 9th ACM SIGCOMM Conference on Internet Measurement Conference (IMC ’09), 2009, pp. 442–455.

[2]. Wendell P, Freedman MJ, Going viral: flash crowds in an open CDN, in: Proc. Of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference (IMC ’11), 2011, pp. 549–558.

[3]. Vakali A, Pallis G, Content delivery networks: status and trends, IEEE Internet Comput. 7 (no. 6) (2003) 68–74.

[4]. Dilley J, Maggs B, Parikh J, Prokop H, Sitaraman R, Weihi B, Globally distributed content delivery, IEEE Internet Comput. 6 (no. 5) (2002) 50–58.

[5]. Nygren E, Sitaraman RK, Sun J, The akamai network: a platform for high-performance internet applications, ACM SIGOPS Oper. Syst. Rev. 44 (no. 3) (2010) 2–19.

[6]. Wang L, Pai V, Peterson L, The effectiveness of request redirection on cdn robustness, ACM SIGOPS Oper. Syst. Rev. 36 (no. SI) (2002) 345–360.

[7]. Adhikari VK, Guo Y, Hao F, Hilt V, Zhang Z-L, A tale of three cdns: an active measurement study of hulu and its cdns, in: Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference, 2012, pp. 7–12.

[8]. Manfredi S, Oliviero F, Romano SP, Optimized balancing algorithm for content delivery networks, IET Commun. 6 (no. 7) (2012) 733–739.

[9]. Alsum A, Nelson ML, Sanderson R, Sompel HV, Archival HTTP redirection retrieval policies, in: Proc. of the 22nd International Conference on World Wide Web (WWW ’13), 2013, pp. 1051–1058.

[10]. Krishnamurthy B, Willis C, Zhang Y, On the use and performance of content distribution network, in: Proc. of 1st International Internet Measurement Workshop (IMC ’01), 2001, pp. 169–182.

[11]. Yoshimura T, Yonemoto Y, Ohya T, Etoh M, Wee S, Mobile streaming media CDN enabled by dynamic SMIL, in: Pro. of the 11th International Conference on World Wide Web (WWW ’02), 2002, pp. 651–661.

[12]. Calder M, Flavel A, Katz-Bassett E, Mahajan R, Padhye J, Analyzing the performance of an anycast CDN, in: Proc. of the 2015 ACM Conference on Internet Measurement Conference (IMC ’15), 2015, pp. 531–537.

[13]. Alzoubi HA, Lee S, Rabinovich M, Spatscheck O, Merwe JV, Anycast CDNS revisited, in: Proc. of the 17th International Conference on World Wide Web (WWW ’08), 2008, pp. 277–286.

[14]. Alzoubi HA, Lee S, Rabinovich M, Spatscheck O, Merwe JV, A practical architecture for an anycast CDN, ACM Trans. Web 5 (no. 4) (2011). Article 17, 29 pages.

[15]. Qudah ZA, Lee S, Rabinovich M, Spatscheck O, Merwe JV, Anycast-aware transport for content delivery networks, in: Proc. of the 18th International Conference on World Wide Web (WWW ’09), 2009, pp. 301–310.

[16]. Akamai, www.akamai.com.

[17]. Limelight Networks, www.limelight.com.

[18]. Mirror Image, www.mirror-image.com.

[19]. Fanou R, Tyson G, Francois P, Sathiaselvan A, Pushing the frontier: exploring the african web ecosystem, in: Proc of the 25th International Conference on World Wide Web (WWW ’16), 2016, pp. 435–445.

[20]. Alzoubi HA, Rabinovich M, Spatscheck O, The anatomy of LDNS clusters: findings and implications for web content delivery, in: Proc. of the 22nd International Conference on World Wide Web (WWW ’13), 2013, pp. 83–94.

[21]. Contavalli C, Gaast W, Lawrence D, Kumari W, Client subnet in DNS queries, in: IETF RFC 7871, 2016, pp. 1–29.

[22]. Damas J, Graff M, Vixie P, Extension mechanism for DNS (EDNS(0)), in: IETF RFC 6891, 2013, pp. 1–16.

Digit Commun Netw. Author manuscript; available in PMC 2021 June 10.
[23]. Wang Z, Hu AL, A lightweight solution to remote DNS, in: Proc of the IEEE 32nd International Performance Computing and Communications (IPCCC ’13), 2013, pp. 1–3.

[24]. Otto JS, Sanchez MA, Rula JP, Stein T, Bustamante FE, Namehelp: intelligent client-side DNS resolution, in: Proc. of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM ’12), 2012, pp. 287–288.

[25]. Wang Z, POSTER: pseudonymizing client as a privacy-preserving service: a case study of CDN, in: Proc of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS ’15), 2015, pp. 1–3.

[26]. Calder M, Fan X, Hu Z, Katz-Bassett E, Heidemann J, Govindan R, Mapping the expansion of Google’s serving infrastructure, in: Proc. of the 2013 ACM SIGCOMM Conference on Internet Measurement Conference (IMC ’13), 2013, pp. 313–326.

[27]. Streibelt F, Böttger J, Chatzis N, Smaragdakis G, Feldmann A, Exploring EDNS-client-subnet adopters in your free time, in: Proc. of the 2013 ACM SIGCOMM Conference on Internet Measurement Conference (IMC ’13), 2013, pp. 305–312.

[28]. Otto JS, Sanchez MA, Rula JP, Bustamante FE, Content delivery and the natural evolution of DNS: remote dns trends, performance issues and alternative solutions, in: Proc. of the 2012 ACM Conference on Internet Measurement Conference (IMC ’12), 2012, pp. 523–536.

[29]. Benchaita W, Doudane SG, Tixeuil S, Stability and optimization of DNS-based request redirection in CDNs, in: Proc. of the 17th International Conference on Distributed Computing and Networking (ICDCN ’16), Article 11, 2016, 10 pages.

[30]. Jung J, Sit E, Balakrishnan H, Morris R, DNS performance and the effectiveness of caching, IEEE/ACM Trans. Netw. 10 (no. 5) (2002) 589–603.

[31]. Ager B, Mühlbauer W, Smaragdakis G, Uhlig S, Comparing DNS resolvers in the wild, in: Proc. of the 10th ACM SIGCOMM Conference on Internet Measurement (IMC ’10), 2010, pp. 15–21.

[32]. Shulman H, Pretty bad privacy: pitfalls of DNS encryption, in: Proc. of the 13th Workshop on Privacy in the Electronic Society (WPES ’14), 2014, pp. 191–200.

[33]. Jeong SH, Kang AR, Kim J, Kim HK, Mohaisen A, A longitudinal analysis of i2p leakage in the public DNS infrastructure, in: Proc. of the ACM SIGCOMM Conference (SIGCOMM ’16), 2016, pp. 557–558.

[34]. Zhu L, Hu Z, Heidemann J, Wessels D, Mankin A, Somaiya N, T-DNS: connection-oriented DNS to improve privacy and security (poster abstract), SIGCOMM Comput. Commun. Rev. 44 (no. 4) (2014) 379–380.

[35]. Rula JP, Bustamante FE, Behind the curtain: cellular DNS and content replica selection, in: Proc. of the 2014 Conference on Internet Measurement Conference (IMC ’14), 2014, pp. 59–72.

[36]. Bortzmeyer S, DNS privacy considerations, in: IETF RFC 7626, 2015, pp. 1–17.

[37]. Wang Z, Analysis of flooding DoS attacks utilizing DNS name error queries, KSII Trans. Internet & Inf. Syst. 6 (no. 10) (2012) 2750–2763.

[38]. Wang Z, Tseng SS, Impact evaluation of DDoS attacks on DNS cache server using queuing model, KSII Trans. Internet & Inf. Syst. 7 (no. 4) (2013) 859–909.

[39]. Chang X, Li J, Wang G, Zhang Z, Li L, Niu Y, Software defined backpressure mechanism for edge router, in: Proc. of the 2015 IEEE 23rd International Symposium on Quality of Service (IWQoS), 2015, pp. 171–176.
Fig. 1.
DNS-based server redirecting.
Fig. 2.
Client pseduonymizing.
Fig. 3.
PDF of Weibull distribution ($\lambda = 1.09$ and $\kappa = 5$).
Fig. 4.
CDN performance under Weibull distribution in Fig. 3.
Fig. 5.
PDF of Lognormal distribution ($\mu = -0.5493$ and $\sigma = 1.0481$).
Fig. 6.
CDN performance under Lognormal distribution in Fig. 5.
Table 1

Comparison of existing solutions.

| Solution         | Metric               | Client complexity | Intermediate transparency | CDN performance | Client location privacy | Redirection privacy |
|------------------|----------------------|-------------------|---------------------------|-----------------|------------------------|---------------------|
| Local sever      |                      | Low               | Good                      | Medium          | Medium                 | Good                |
| Remote server    |                      | Low               | Good                      | Bad             | Good                   | Good                |
| Client server    |                      | High              | Good                      | Good            | Bad                    | Good                |
| ECS              |                      | Low               | Bad                       | Good            | Bad                    | Bad                 |
| Direct Resolution|                      | Medium            | Good                      | Good            | Bad                    | Good                |
| Name Extension   |                      | Low               | Good                      | Good            | Bad                    | Bad                 |
| Client pseudonymizing |              | Low               | Bad                       | Good            | Good                   | Good                |