MR-DNS: Multi-Resolution Domain Name System

Saidur Rahman[1] and Mike P. Wittie[2]

[1] Montana State University, Bozeman MT 59715, USA
sr.rifat@gmail.com
[2] Montana State University, Bozeman MT 59715, USA
mike.wittie@montana.edu

Abstract. Users want websites to deliver rich content quickly. However, rich content often comes from separate subdomains and requires additional DNS lookups, which negatively impact web performance metrics such as First Meaningful Paint Time, Page Load Time, and the Speed Index. In this paper we investigate the impact of DNS lookups on web performance and propose Multi-Resolution DNS (MR-DNS) to reduce DNS resolutions through response batching. Our results show that MR-DNS has the potential to improve Page Load Time around 14% on average, Speed Index around 10% on average and reduce DNS traffic around 50%. We also discuss how these gains may be realized in practice through incremental changes to DNS infrastructure.

Keywords: DNS · Web Performance

1 Introduction

Modern websites support a broad range of complex and interactive services such as online social networks, e-commerce, streaming video, or gaming. Consumers want websites to deliver full-featured content and to do so seamlessly. However as web developers create richer online experiences, their websites fetch content from a growing number of subdomains, or other URLs embedded in the base HTML, which decreases site responsiveness [10].

To quantify the impact of website responsiveness on user experience the web performance community proposed several web performance metrics, such as First Meaningful Paint (FMP) time, Above The Fold (ATF) time, Page Load Time (PLT) and a composite metric called the Speed Index (SI) among others [7, 11, 12, 18, 20]. Subsequently, these metrics have been used as a measuring stick against which to further improve webpage performance. As webpage content grows richer, the number of Domain Name System (DNS) lookups needed to resolve distinctive subdomains increases and negatively impacts web performance metrics and user experience [14].

We propose an experimental method to reduce the number of DNS lookups of page's subdomains. Our method, Multi-Resolution DNS (MR-DNS), reduces the number of DNS request-response round trips through an assumed collaboration between the local DNS (LDNS) and authoritative DNS (ADNS) servers. The two pillars of our approach are the creation of a mapping between a base page and its subdomain by the LDNS and the batching of DNS responses by the ADNS. Although what we propose makes the DNS system more stateful, a significant reimagining of the DNS, the implementation relies on existing DNS extensions and does require new packet formats. The proposed system is also backwards compatible and permits incremental deployment.

We evaluate the impact of MR-DNS on 13 pages from an unmodified Chrome browser (version-75.0.3770.80) on a computer pointed to a custom LDNS server. Our results show that MR-DNS has the potential to reduce PLT by up to 31% and 14.45% on average. We also show reductions in SI of up to 14.67% and 10.15% on average.

The rest of this paper is organized as follows. Section 2 presents the background on DNS and web performance metrics. In Section 3 we analyze the structure of subdomains in modern webpages. Section 4 details the design and implementation of MR-DNS. In Section 5 we present the results from MR-DNS performance measurement. Section 6 outlines related work in reducing DNS resolution time during webpage loads. Finally, we conclude in Section 7.

2 Background

2.1 DNS Resolution Process

Figure 1 illustrates the process of a DNS resolution [13]. To resolve a new URL such as `https://example.com` to an IP of its web server, a browser sends a DNS request to a local DNS (LDNS) server operated by the Internet provider (step 1). Assuming the LDNS does not have a valid DNS entry for the requested domain, the LDNS contacts the root DNS server to find the IP of a top-level domain (TLD) server responsible for the `com` portion of the domain (steps 2 and 3). A request to the TLD server returns the IP of the authoritative DNS (ADNS) server, maintained by the organization hosting `https://example.com` (steps 4 and 5). A subsequent request to the ADNS returns the mapping between the URL and the IP of the server hosting the site (steps 6 and 7). Finally, the LDNS returns the resolution to the browser (step 8), which issues an HTTP `GET` request for site content.

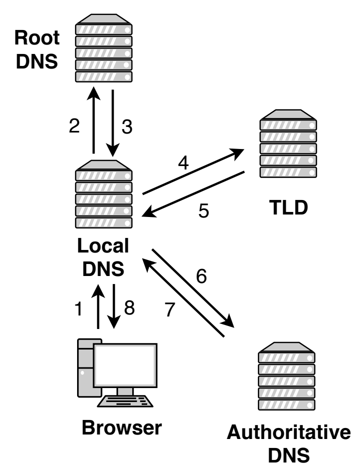


Fig. 1: DNS query path

The obtained base page often contains many subdomains that download various assets, such as `img.example.com`. A subdomain may share a portion of the address with a domain already resolved and so may use an ADNS already discovered by the LDNS. Although a subdomain lookup in this case requires only steps 1, 6, 7, 8, the number of subdomain lookups may still be significant. We investigate the number of subdomains in Section 3 and the impact of their lookups on web performance metrics in Section 5.

2.2 Web Performance Metrics

The web performance community has proposed a number of web performance metrics to objectively measure website responsiveness. The need for multiple metrics comes from the fact that websites provide functionality not only when fully loaded, but also when different aspects of their rendering finish. Figure 2 shows the process of loading and rendering a website along with the labels of the various stages of process completion.

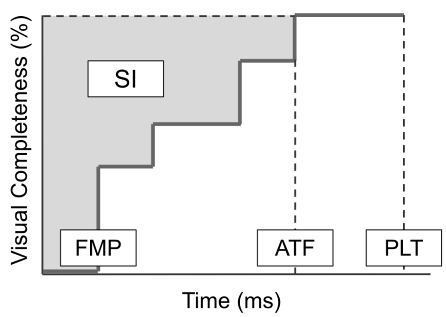


Fig. 2: Web Performance Metrics

The First Paint is the time when the first pixel is painted onto the screen for instance, a background color of the page and First Meaningful Paint (FMP) time occurs when a page's primary content, such as text, image, or a form field, first appears on the screen [7]. That means the time when the browser paints the content that users are interested in.

Browsers measure FMP from the time of the initial page load request to the rendering of the first visible object above the fold. The metric is useful, because it reflects users' perception of page responsiveness in starting to show content.

Above the Fold (ATF) time occurs when the browser finishes rendering page elements visible in the original browser window, before scrolling down [18]. The metric is useful, because it represents users' perception of then the page finishes to load and is fully visible.

Page Load Time (PLT) occurs at the `window.onLoad` event when the browser finishes receiving data for all the requests issued as part of the page load request [15, 22]. PLT may occur after all the page content is visible, for example as data visible below the fold arrives.

Since different users may assign different importance to FMP vs. ATF vs. PLT, the Web performance community developed a composite metric dubbed the Speed Index (SI) that captures them jointly [23]. SI calculates the area above the curve of visual completion to ATF, as illustrated in Figure 2, based on a series of screenshots of a loading page taken every 100 ms. Several studies have shown a correlation between the SI and the quality of user experience (QoE) [2, 12]. To understand the potential of speeding up the resolution of subdomain addresses, in Section 5 we will look at the impact on FMP, PLT, and SI.

3 Analysis of Webpage Subdomain Structure

To deliver rich content, modern webpages rely on a large number of subdomains. We wanted to know exactly how many subdomains different webpages contain including 3rd party subdomains and whether browsers use the same ADNS to resolve a large number of these subdomains.

To obtain the number of subdomains we loaded 13 pages with the Chrome browser and tracked the issued DNS requests using Tshark. We observed the each page requests as slightly different set of subdomains on each load. To capture all subdomains, we loaded each page a 100 times to collect a super set of subdomains for each page. We show number of subdomains and the overall size of the page in Figure 3 on the left and right y-axes, with the address of each page on the x-axis.

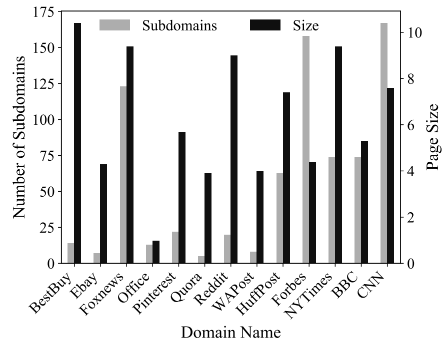


Fig. 3: Subdomain counts.

Based on the figure we observe that web pages have a large number of subdomains. For example, six of the pages have over 100 subdomains each. The number of subdomains is not tightly correlated with page size, which means that even for small pages that the user might expect to load quickly the time of DNS resolution of its subdomains might impact performance.

Next, we wanted to understand how many CDNs support the load of these pages. Since each CDN maintains their own ADNS servers, our goal was to understand the potential of speeding DNS resolutions by modifying a ADNS of one or more CDNs.

Peters and Kayan developed CDN Planet [17], a tool for monitoring performance of DNS and CDNs. One of the CDN Planet tools, CDN Finder, finds the list of subdomains in a website and extracts the Canonical Name Record (CNAME) of each subdomain. CDN Finder then extracts the server portion of the CNAME URL, matches it to a name of known CDNs, and assigns the subdomain to the CDN subdomain count.

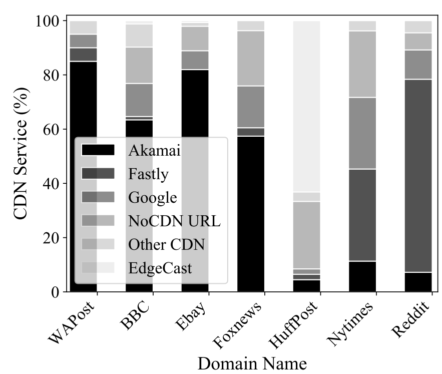


Fig. 4: Subdomain percent by CDN.

In Figure 4 we show the mapping of subdomains to CDNs for several webpages using CDN Finder. The x-axis shows the different websites, while the y-axis show the percentage of subdomains served by each CDN. The figure shows that most of the subdomains on a given page are served by the CDN that also hosts the base domain. The one exception is `bbc.com`, where Fastly serves the base page, but most of the subdomains reside on Akamai. We conclude that by improving the DNS lookup process for only one ADNS hosted by one CDN, we might be able to affect a large number of subdomain lookups thereby significantly reducing the impact of DNS resolutions on web performance metrics.

4 Multi-Resolution DNS System

To speed up the resolution of subdomains we propose MR-DNS – a DNS extension that enables LDNS to resolve subdomains contained within a webpage in bulk. The system implements a collaboration between the LDNS and ADNS servers that builds a mapping between the base page and the contained subdomains. The ADNS server then uses that mapping to send resolutions for subdomains when it receives a request to resolve the base page.

We illustrate the operations of MR-DNS in Figure 5. When the browser requests the resolution of a base page domain, the LDNS forwards the DNS request to the ADNS. Instead of sending a single resolution, the ADNS replies with the mapping for the base page domain as well as the mappings for subdomains in the base page known to the ADNS. To form this response, the ADNS needs to know which subdomains belong to a base page – we discuss how to create this mapping momentarily.

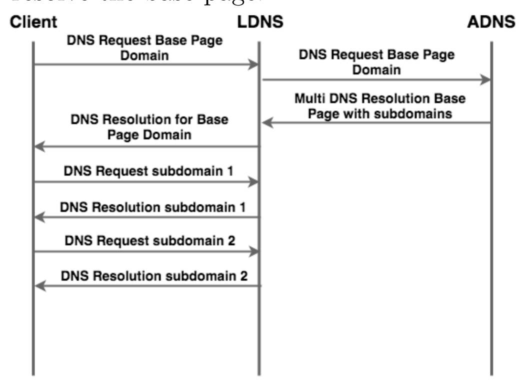


Fig. 5: MR-DNS resolution process.

The ADNS response containing the bulk resolution of subdomains may be to large to fit in a single DNS response message contained within a UDP packet, even when using EDNS0. To communicate multiple resolutions, the ADNS may send multiple DNS responses. The LDNS implementation needs to be modified to, first, cache multiple mappings and, second, accept multiple response packets. Upon receiving ADNS responses the LDNS caches the mappings and responds to the browser with the mapping for the base domain. When the browser requests resolutions for subdomains contained in the base page, the LDNS does not need forward these requests to the ADNS, but simply serves them from its cache. As a result, subdomain resolutions take less time and communications with the ADNS do not hold up web page rendering, which as we show in Section 5 speeds up web performance metrics.

To implement MR-DNS the ADNS needs to have a mapping between base page domains and subdomains. Regularly this mapping is not available to the ADNS and is contained within the HTML of pages hosted by a CDN. There are a number of possible approaches to create this mapping. Since most ADNSs are hosted by the same CDN that serves the base page, it is possible for a

CDN to conduct a static analysis of page subdomains. That approach, however, might miss subdomains revealed by executing JavaScript, or those hidden under multiple resolutions of CNAMEs [1].

Instead, we propose the approach we take to generate Figure 3, where the set of page subdomains comes from runtime analysis of DNS requests triggered by a page load. We observe that for a single client the resolution of a base page precedes resolutions for its subdomains. The LDNS may observe DNS requests coming from the same IP for an amount of time to generate a probable mapping between the first request for the base page and the subsequent requests for its subdomains. ADNS stores the mapping information of the domain using a data structure format shown in Figure 6.

TTL	Subdomain	IP List	Last-Use

Fig. 6: Data Structure for ADNS mapping

Different LDNS servers would then report that mapping to the ADNS, which could not only combine them into a more accurate mapping, but also create a mapping for different geographic regions based on the similarity of reported subdomain sets and the IP prefix of each LDNS.

Figure 7 shows a measurement of the number of subdomains observed on the first load of a page vs. the super set of subdomains on a page over many loads. The x-axis shows the different base page domains, while the y-axis shows the normalized number of subdomains. We found that the super set of subdomains can be quite a bit larger than the set of domains observed in any given load. For example only 60% of subdomains on Quora are visible on the first load. As a result the LDNS should repeatedly observe and report page subdomains and let the ADNS comprise the super set of the mapping.

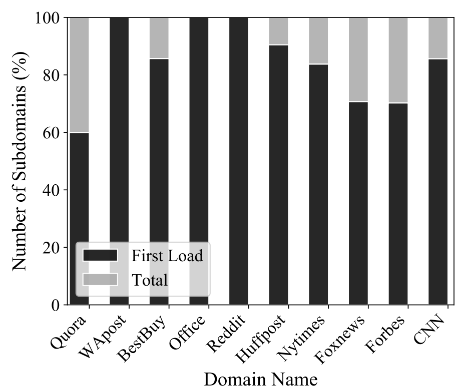


Fig. 7: Subdomain load number.

We explore mapping creation in two modes. In the *CacheCDN* mode the ADNS creates a mapping for all the subdomains hosted by the same CDN as the base page. We obtain the set of subdomains for CacheCDN using the data from Figure 4. In the *CacheAll* mode the ADNS maintains a mapping of all subdomains contained on a page. There are practical challenges to implementing the CacheAll approach in that the CDN ADNS provides resolutions for domains maintained by other ADNS servers. While there could be cooperative approaches among ADNS servers to implement CacheAll, we do not explore them in this paper. Instead, we consider CacheAll as a theoretical upper limit of the benefit of bulk domain resolution. We observe, however, that as more of page subdomains are hosted by the same CDN, the benefit of the CacheCDN mode will approach the CacheAll mode.

Finally, we need to consider the results of an inaccurate mapping between the base page and its subdomains. In the case where the bulk resolution contains more subdomains then needed, the ADNS will deliver unnecessary mappings to the LDNS. While that may create additional DNS traffic by sending unneeded mappings, we show in Section 5 that MR-DNS reduces DNS traffic overall. It may also be possible for the ADNS to deliver mappings that the LDNS already has in its cache, thereby sending unnecessary information. This may be avoided in many cases by setting the time to live (TTL) of subdomains to be the same as the TTL of the base page. The added benefit of synchronized TTLs is that any time a client loads a base page the LDNS will have up-to-date subdomain mappings and make it faster for the ADNS to update LDNS mappings. In the case where the mapping lacks certain subdomains, the LDNS will not be able to satisfy a browser request for these subdomains from its cache and will instead forward the DNS request as it does currently. In either case, the bulk resolution of subdomains does not affect the correctness of the DNS.

5 Evaluation

We demonstrate the effectiveness of MR-DNS to improve web performance metrics by resolving page subdomains in bulk. We also measure MR-DNS resource usage in terms of DNS traffic and LDNS memory usage.

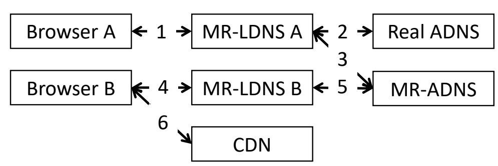


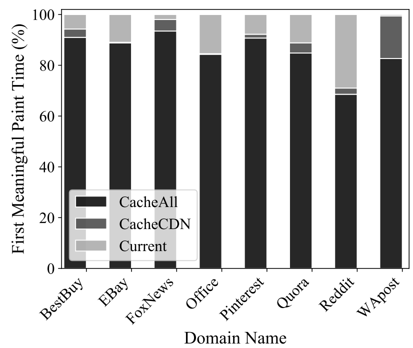
Fig. 8: Experimental Setup

To conduct our measurements we set up customized LDNS and ADNS servers on dedicated Ubuntu 18.04 servers implementing MR-DNS bulk resolution, illustrated in Figure 8. The first stage in the evaluation process is to create the mapping between the base page and the subdomains for a given page. We load the page using Browser A connected to LDNS A (step 1), which uses the real ADNS servers to resolve all page subdomains (step 2). LDNS A creates the mapping and sends it to MR-ADNS (step 3). In the second stage we collect collect web performance metrics. We load each page on Browser B connected to MR-LDNS B (step 4), which contacts MR-ADNS to obtain the resolution of the base page and the subdomains in bulk (step 5). We configure the network latency between Browser B, MR-LDNS B, and MR-ADNS using `netem` to reflect the latency between the browser and the real LDNS and ADNS servers for each page. For example, for `www.bbc.com`, we found the average latency between LDNS and ADNS is 42.113 ms. During the load of a page, Browser B loads each page from the real CDN servers over a network with actual delays and bandwidth limitations from our lab in Bozeman, MT (step 6). For our measurements we used several popular pages from the Alexa list of most popular pages [5]. We load each page 100 times in new Chrome Incognito window each time, to prevent caching, using the Chrome Headless mode. To collect web performance metrics we use Google Lighthouse [8].

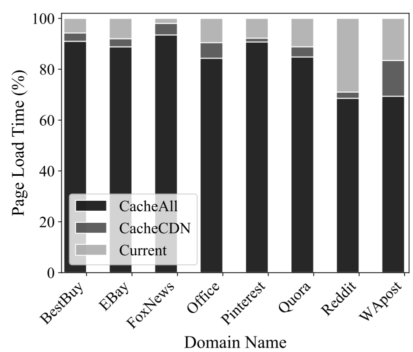
5.1 Web Performance Metrics

We show the measurement of web performance metrics in Figures 9a-9c. The x-axis shows the different base page domains, while the y-axis shows FMP, PLT, and SI respectively. The 100% mark of each bar shows the metric value of the page loaded through the unmodified DNS system. The CacheCDN and CacheAll modes show the reduction in each metric value as a percent.

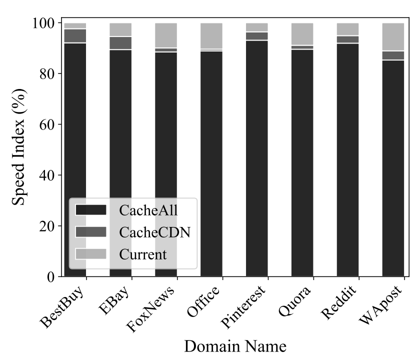
In Figure 9a we observe that the CacheCDN mode reduces FMP time by as much as 16.76% for WAPost and 4.12% on average. The theoretical improvement of the CacheAll mode reduces FMP time by as much as 31.42% for Reddit and 14.45% on average. In Figure 9b we observe that the CacheCDN mode reduces PLT by as much as 14.02% for WAPost and 4.89% on average. The theoretical improvement of the CacheAll mode reduces PLT by as much as 31% for Reddit and 16.11% on average. In Figure 9c we observe that the CacheCDN mode reduces SI by as much as 5.57% for BestBuy and 3.05% on average. The theoretical improvement of the CacheAll mode reduces SI by as much as 14.67% for WAPost and 10.15% on average. Overall, we observe the the CacheCDN mode has a significant potential to reduce both individual and composite web performance metrics thereby improving user QoE. As more of the page content is hosted by a single CDN, the gains of CacheCDN mode approach those of CacheAll mode as for example in the case of Reddit.



(a) FMP time vs. page domain.



(b) PLT vs. page domain.



(c) SI vs. page domain.

Fig. 9: Measured web performance.

5.2 Resource Usage

Figure 10 shows the memory on the LDNS needed to store different numbers of subdomains. The x-axis shows the number of subdomains, while the y-axis shows the cache memory usage, including the domain name, IP address, TTL, and last used time obtained with the `object-sizeof` function on the dictionary data structure. We observe that the memory footprint is highly linear and therefore predictable in the number of subdomains.

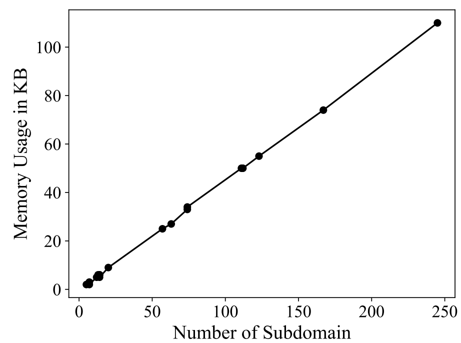


Fig. 10: Subdomain load number.

Domain Name	NV	MT	CA	MT ∩ CA	NV ∩ CA	NV ∩ MT	NV ∩ CA ∩ MT
www.bestbuy.com	18	12	19	12	16	12	12
www.ebay.com	12	7	73	7	11	7	7
www.foxnews.com	185	110	181	104	155	104	103
www.office.com	13	10	17	10	11	9	9
www.pinterest.com	18	14	19	13	16	14	13
www.quora.com	4	4	12	3	4	3	3
www.reddit.com	38	20	44	20	38	20	20
www.washingtonpost.com	11	7	14	7	10	6	6
www.forbes.com	198	118	243	105	169	102	102
www.usatoday.com	356	179	329	160	261	159	155
www.weather.com	11	11	13	10	10	8	7
www.walmart.com	76	76	82	59	68	58	58

Table 1: Subdomains in different regions

This information allows us to predict how much memory would be required on the ADNS to store the information reported to LDNS servers in different regions. We measure the number of subdomains observed by LDNS servers in the CacheAll mode in different regions by deploying MR-DNS on AWS servers in N. California (CA), N. Virginia (NV), and on our lab servers in Montana (MT). In Table 1 we show the number of subdomains for each page in the different regions as well as the intersections to show how many domains the regions have in common. We observe that there is a high degree of subdomain overlap between the different regions. As a result the requirement for the ADNS to store subdomains from the different regions creates a relatively low and predictable memory overhead, even in the CacheAll mode.

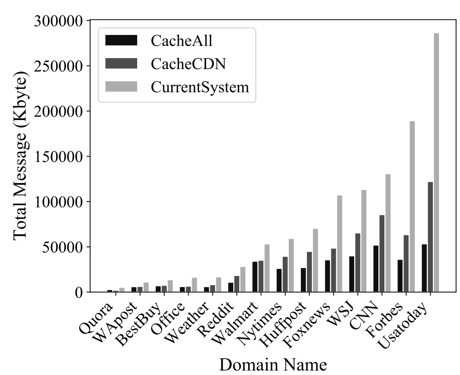


Fig. 11: Subdomain load number.

Finally, in Figure 11, we show the comparison of three different system CacheAll, CacheCDN, and Current system in terms of total message volume for DNS resolutions. The x-axis shows the base page domain, while the y-axis shows the total DNS traffic in the current, CacheCDN, and CacheAll modes measured using Tshark. We observe that while MR-DNS delivers more domains that may be strictly needed for an individual page load, see Figure 7, the total volume of DNS traffic actually decreases, especially for the CacheCDN mode.

6 Related Work

Several research projects have proposed solutions to reduce the delay of DNS lookups. Park *et al.* proposed CoDNS to replicate DNS queries to different DNS servers to improve lookup reliability and the latency [16]. A similar tool also replicates DNS queries to different servers to reduce lookup delay by accepting the first response [21]. However, the first responding DNS server might not necessarily offer the best CDN mapping for the client. Goel *et al.* solve this problem by probing the resolved CDN servers before issuing requests for web objects [9].

DNS Pre-Resolve eliminates DNS lookup delay by resolving domain names during page rendering. This pre-resolution, proposed by Google, embedded requests in HTML header which reduces DNS lookup time by giving hints to the browser at the beginning of the HTML and the browser pre-resolve DNS for resources on the page [6]. Shang *et al.* proposed a method to decrease the DNS cache drop rate by utilizing the correlation of requested Web domains to the domains that have a DNS resolution cached [19]. Though these two techniques decrease the impact of DNS on webpage load time, for both cases, the ADNS gets all the requests for all subdomains.

Most similar to our work is the proposal by Almeida *et al.* discuss potential improvement to PLT from reducing DNS lookup delay [1]. They find that for top 10k Alexa websites DNS lookup is responsible for 9.2% of PLT on average, which collaborates an earlier result by Wang *et al.* [22]. Almeida *et al.* also show that pre-resolving webpage sub-domains and pre-staging these resolutions in browser cache, via an oracle, before the page is opened, has the potential to reduce PLT by 8.5%. Their experiment, however, does not pre-resolve all the sub-domains, but only those available through static analysis of page HTML, which does not include domains embedded through JavaScript execution [1].

Chowdaiah has proposed, in an IETF experimental internet draft, for HTTP proxies to intercept base page HTML, parse out web addresses, resolve them proactively, and push the results into client DNS cache [4]. This approach is complimentary to LDNS-ADNS cooperation. The downside of the proxy approach is that the proxy does not know the state of the client DNS cache and so might perform unnecessary resolutions. This problem could be solved by communicating the state of client DNS cache through a shared dictionary approach, but is not considered in the internet draft [3]. Almeida *et al.* found that the speed up of this scheme is limited to 4% reduction in PLT on average [1].

7 Conclusions

DNS lookups affect Web application performance due to the increasing number of webpage subdomains. Although the current DNS may provide fast service, we showed that current techniques could be improved to speed up DNS resolutions. We proposed MR-DNS, a multi-resolution technique that reduce the number of DNS request-response round trips through a collaboration between the LDNS and the ADNS servers. Our measurement study showed that MR-DNS reduces webpage load time by 14% and Speed Index by 10%. MR-DNS also reduces the average traffic of ADNS by 50%. We want to note that we measured these improvements in wired networks. In cellular networks the distance between LDNS and ADNS may be smaller, if the ADNS is hosted by a CDN in the edge network. For those networks, it maybe be practical to implement MR-DNS in the browser between the device and the LDNS to eliminate radio latency in DNS lookups. Overall, we believe that MR-DNS enables a more effective use of existing DNS infrastructure and represents a strategy to improve user experience by reducing DNS lookups for rich content websites.

Acknowledgements

We would like to thank Utkarsh Goel for his helpful insights on the ideas and mechanisms presented in this paper.

References

1. Almeida, M., Finamore, A., Perino, D., Vallina-Rodriguez, N., Varvello, M.: Dissecting DNS stakeholders in mobile networks. CoNEXT (December 2017)
2. Bocchi, E., De Cicco, L., Rossi, D.: Measuring the quality of experience of web users. SIGCOMM Comput. Commun. Rev. **46**, 8–13 (Dec 2016)
3. Butler, J., Lee, W.H., McQuade, B., Mixter, K.: A proposal for shared dictionary compression over HTTP (September 2008)
4. Chowdaiah, P.: Method to pre-fetch domain names at HTTP proxy servers (September 2018)
5. Developers, A.: The top 500 sites on the web. https://www.alexa.com/topsites
6. Developers, G.: Pre-resolve dns (December 2016), https://developers.google.com/speed/pagespeed/service/PreResolveDns
7. Developers, G.: First Meaningful Paint. https://developers.google.com/web/tools/lighthouse/audits/first-meaningful-paint (May 2019)
8. Developers, G.: Tools for web Developers: Lighthouse. https://developers.google.com/web/tools/lighthouse/ (May 2019)
9. Goel, U., Wittie, M.P., Steiner, M.: Faster web through client-assisted CDN server selection. In: ICCCN (Aug 2015)
10. Goel, U., Steiner, M., Na, W., Wittie, M.P., Flack, M., Ludin, S.: Are 3rd parties slowing down the mobile web? In: S3@MobiCom (2016)
11. da Hora, D., Rossi, D., Christophides, V., Teixeira, R.: A practical method for measuring web above-the-fold time. SIGCOMM (August 2018)

12. Hofeld, T., Metzger, F., Rossi, D.: Speed index: Relating the industrial standard for user perceived web performance to web qoe. In: QoMEX (May 2018)
13. Kurose, J.F., Ross, K.W.: Computer Networking: A Top-Down Approach (6th Edition) (2012)
14. Meenan, P.: How fast is your web site? Queue (Mar 2013)
15. Netravali, R.A.: Understanding and improving web page load times on modern networks (Feb 2015)
16. Park, K., Pai, V.S., Peterson, L., Wang, Z.: CoDNS: Improving DNS performance and reliability via cooperative lookups. OSDI (December 2004)
17. Peters, A., Kayan, S.: CDN Finder tool. https://www.cdnplanet.com/tools/cdnfinder/ (2019), [Online; accessed 28-April-2019]
18. Saverimoutou, A., Mathieu, B., Vaton, S.: Web browsing measurements: An above-the-fold browser-based technique (July 2018)
19. Shang, H., Wills, C.E.: Piggybacking related domain names to improve DNS performance. Computer Networks **50** (August 2006)
20. Shroff, P.H., Chaudhary, S.R.: Critical rendering path optimizations to reduce the web page loading time. In: I2CT (April 2017)
21. Vulimiri, A., Godfrey, P.B., Mittal, R., Sherry, J., Ratnasamy, S., Shenker, S.: Low latency via redundancy. CoNEXT (December 2013)
22. Wang, X.S., Balasubramanian, A., Krishnamurthy, A., Wetherall, D.: Demystifying page load performance with WProf. In: NSDI (April 2013)
23. WebPagetest: WebPagetest Documentation: Speed Index. https://sites.google.com/a/webpagetest.org/docs/using-webpagetest/metrics/speed-index (April 2012)