Dissecting DNS Stakeholders in Mobile Networks

Mario Almeida, Alessandro Finamore★, Diego Perino★, Narseo Vallina-Rodriguez†, Matteo Varvello‡*
Universitat Politècnica de Catalunya, ★Telefónica Research, †IMDEA Networks/ICSI, ‡AT&T

ABSTRACT

The functioning of mobile apps involves a large number of protocols and entities, with the Domain Name System (DNS) acting as a predominant one. Despite being one of the oldest Internet systems, DNS still operates with semi-obscure interactions among its stakeholders: domain owners, network operators, operating systems, and app developers. The goal of this work is to holistically understand the dynamics of DNS in mobile traffic along with the role of each of its stakeholders. We use two complementary (anonymized) datasets: traffic logs provided by a European mobile network operator (MNO) with 19M customers, and traffic logs from 5,000 users of Lumen, a traffic monitoring app for Android. We complement such passive traffic analysis with active measurements at four European MNOs. Our study reveals that 10k domains (out of 198M) account for 87% of total network flows. The time to live (TTL) values for such domains are mostly short (< 1min), despite domain-to-IPs mapping tends to change on a longer time-scale. Further, depending on the operators recursive resolver architecture, end-user devices receive even smaller TTL values leading to suboptimal effectiveness of the on-device DNS cache. Despite a number of on-device and in-network optimizations available to minimize DNS overhead, which we find corresponding to 10% of page load time (PLT) on average, we have not found wide evidence of their adoption in the wild.

CCS CONCEPTS

• **Networks → Network measurement**; **Naming and addressing**; *Network performance analysis*;

KEYWORDS

Mobile DNS traffic; page load time (PLT); ephemeral domains; time to live (TTL); caching

ACM Reference Format:
Mario Almeida, Alessandro Finamore★, Diego Perino★, Narseo Vallina-Rodriguez†, Matteo Varvello‡. 2017. Dissecting DNS Stakeholders in Mobile Networks. In *CoNEXT '17: CoNEXT '17: The 13th International Conference on emerging Networking EXperiments and Technologies, December 12–15, 2017, Incheon, Republic of Korea.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3143361.3143375

1 INTRODUCTION

According to recent estimates, mobile traffic is expected to have a sevenfold increase by 2021 [9]. At such growing rate, improving network architectures and understanding quality of experience (QoE) are fundamental steps towards shaping and optimizing current and future mobile networks. A recent study [20] revealed that, for some mobile operators, the Domain Name System (DNS) accounts for the highest fraction of flows and up to 50% of the total traffic. However, it is unclear *who* is responsible for such traffic load, and its impact on the end-user QoE.

Many previous studies investigated QoE in mobile networks focusing on inefficiencies of the access technology and network path [5, 7, 17, 19, 22, 25, 31], and in-path proxies [1, 14, 27]. DNS has been considered as part of general traffic performance studies [7, 17, 19], and (small scale) active experiments [23]. Less attention was instead given to DNS in mobile networks and, most importantly, its impact on users QoE.

Despite recent evolutions such as 4G, mobile networks last mile is still a shared access medium suffering from high latency. Hence communications should be optimized (or avoided) when possible. Considering DNS, on-device DNS caches and app-specific optimizations such as pre-fetching [15] are adopted to minimize DNS lookups. However, DNS *stakeholders*—domain owners, mobile network operators (MNOs), operating systems (OSes), and app developers—operate autonomously, leading to suboptimal conditions, and hidden inefficiencies. Therefore, we argue that an in-depth understanding of their interactions is a fundamental step towards identifying how to optimize mobile networks performance.

This paper presents a holistic analysis of real world mobile DNS traffic where we dissect the role of the DNS stakeholders. We leverage three complementary datasets: (i) a month-long dataset collected from a large European mobile network operator (MNO) serving about 19M customers; (ii) a 1.5 year-long dataset containing over 5M flows provided by Lumen, an Android traffic monitor app with over 5,000 users; and (iii) active measurements from a testing device connected to four European MNOs.

We demonstrate that each DNS stakeholder can indeed impact on traffic load and performance. In addition, the study identifies new venues for mobile traffic and QoE optimizations at the DNS level. A detailed summary of our results is as follows.

- The top-10k most popular domains (based on the number of users connecting to them) account for 87% of the global flows. A detailed inspection of those domains reveals that most of them belong to third-party services used by multiple app developers for advertising, user tracking, and social network integration.
- 82% of domains are *ephemeral*, *i.e.,* they are requested only once during the whole month and across 19M users. These domains, owned by a small number of companies, are possibly used to deliver personalized content or for signaling/tracking purposes.

[*] Work done while at Telefonica Research

CoNEXT '17, December 12–15, 2017, Incheon, Republic of Korea
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5422-6/17/12…$15.00
https://doi.org/10.1145/3143361.3143375

- Authoritative name servers assign low time to live (TTL) values to domains as a mean to enable load balancing: 52% of the top-10k domains have TTL < 1min. However, we observe a limited variability of domain-to-IPs mapping over time.
- Half of the users require a DNS request for 65% of their flows. This implies low hit rates at local DNS caches, probably due to domain owners assigning (low) TTL values without fully understanding mobile apps traffic patterns.
- Explicit in-path proxies and ad-hoc app/OS optimizations can reduce PLT up to 10%. However, neither of the solutions are widely adopted by mobile apps.

2 THE DNS ECOSYSTEM

Every DNS lookup involves the interactions of three main stakeholders: (1) the OS via the on-device client resolver (cDNS); (2) the MNO via the local DNS resolver (LDNS);[1] (3) the domain owner via the authoritative name server (ADNS).

Mobile OSes cDNS offers domain resolution mechanisms to apps with basic caching functionalities. If a cache miss occurs (no entry is found for the requested domain, or the entry is too stale) the OS propagates the query towards the LDNS. In turn, the LDNS implements its own cache which, as opposed to the cDNS cache, is shared across all the operator's subscribers. In case of a cache miss, the LDNS recursively contacts the ADNS of the domain components (top-level domain, 2nd-level domain, and so on) until the domain owner ADNS is found. Finally, the domain owner maps (via its ADNS) the domain to a set of IP addresses, and defines a TTL, an elapsing timer controlling for how long the mapping is valid. Domain-to-IPs mapping and TTL are carried in DNS response queries.

Notice that this mechanism allows domain owners to configure responsive and dynamic load balancing policies by using low TTL values, at the cost of enforcing more DNS lookups. Moreover, only LDNS sees ADNS queries response, hence cDNS receives TTL values smaller than authoritative ones, as they reflect LDNS caching. Despite the fact that the authoritative TTL value is supposed to be honored, both cDNS and LDNS can "override" DNS response query values by proxying or re-writing them [6]. Finally, applications like Chrome develop their own customized DNS clients to bypass the OS DNS cache and gain control over DNS queries [1, 15, 18].

3 METHODOLOGY AND DATASETS

To address our research goals, we gather and combine the datasets summarized in Table 1.

Table 1: Datasets characteristics.

Dataset	Users	Apps	Flows	Domains	IPs
MNO	19M	-	250M	198M	4.2M
Lumen	5,000	8,279	5.3M	35,135	99,685
NexusTTL	1	host	104k	10k	20,074
NexusPLT	1	chrome	46k	6,926	7,921

MNO weblogs. *Weblogs* are one of the richest data sources for telcos: they report on the traffic handled by in-path web proxies deployed for performance enhancement (e.g., image transcoding) [11]. In this study we use a 1 month-long Weblogs dataset collected during May 2017 from a major European MNO. Each Weblogs entry provides details about HTTP/HTTPS flows including full IP/TCP-port tuples, handset OS,[2] timestamp, duration, bytes delivered, and associated domain name.[3] The entries also contain an anonymized user-id which allows investigating per-user traffic patterns while preserving subscribers' privacy. Overall, the dataset provides an operator-scale view of mobile traffic: it contains about 250M flows generated by 19M customers, contacting 198M different domains.

On-device measurements. We gather on-device passive mobile traffic logs using Lumen; an Android privacy-enhancing tool that leverages Android's VPN permission to intercept and analyze apps' traffic in user-space [21]. Lumen uploads anonymized generic per-flow data (e.g., IP-port tuples, domain, bytes exchanged) enriched with OS-provided app metadata (e.g., process name originating the flow).[4] We used a 1.5 year-long dataset starting from November 2015 that provides detailed traffic logs for 8,279 apps. Compared to the MNO dataset, Lumen dataset allows us to accurately map flows to apps and obtain traffic logs with a wider geographical diversity (5k users from 94 countries). However, it is constrained to Android devices and its scale, both in terms of users and apps, is smaller.

External datasets. To characterize and validate the above datasets, we gather two external datasets comprising 1M domains provided by Alexa [4] and Cisco Umbrella [10]. The Cisco rank is based on DNS queries (as seen by OpenDNS) performed by users from all over the world. Hence, it provides a global view of domains popularity regardless of applications, OSes, and platforms. Conversely, Alexa dataset includes only landing web pages. For this reason we extend it by crawling the top-1k webpages to include all related domains (e.g., ad networks, and CDN domains).

Active measurements. We perform two different active measurement campaigns using a Nexus 5 shipped with four LTE-enabled SIM cards, each one provided by a different European MNO. In NexusTTL we run the `host` command using the device terminal towards the LDNS and ADNS resolvers. This allows us to investigate TTL values, and detect their manipulations, for the top-10k most popular domains according to the MNO dataset (see Sec. 4). We run the experiments twice a day for 1 month, querying each domain 5 consecutive times. In NexusPLT we measure page load time (PLT) for the top-1k Alexa web pages via the `chrome-har-capturer` [2] tool to study the impact of DNS requests on mobile web performance (see Sec. 5.1). We flush both Chrome and OS DNS cache before each run. We run the experiments 5 times for each webpage, averaging the results.

4 MEASURING DOMAINS RELEVANCE

We start our investigation from an aggregate view of mobile network traffic, aiming to find a manageable and representative set of mobile domains to be later measured in our active measurements. For that, we compute the *popularity* (*i.e.,* the unique number of

[1] LDNS such as Google DNS or OpenDNS do not typically apply to cellular traffic as neither Android nor iOS allow users to configure their DNS configuration (unless devices are rooted, connected over WiFi or VPN-based apps are used).

[2] The proxy classifies handsets based on IMEI values and a GSMA commercial database.

[3] For HTTPS flows, we leverage the TLS SNI extension.

[4] Lumen has been considered as a non-human research study by ICSI/UC Berkeley's IRB as no payload and personal data is collected. Our previous tech report discusses in depth Lumen's ethical and technical aspects [21].

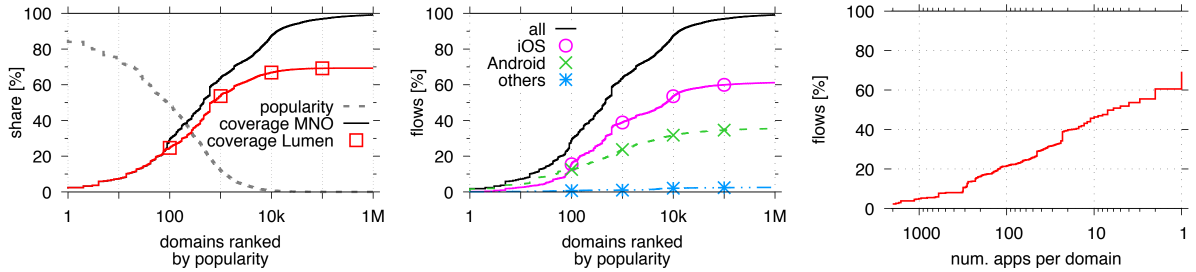

Figure 1: Domains footprint: popular domains and overall traffic (left), OS impact (center), multi-app domains impact (right).

users contacting it) and *coverage* (*i.e.,* the fraction of flows associated to it) for each one of the 198M domains in the MNO dataset. Figure 1 shows both the coverage and popularity for the top-1M domains. As expected, mobile domains present a clear power-law distribution for both metrics: the top-10k are responsible for 87% of traffic flows (solid line), while less than 1% of users contact domains outside the top-10k (dashed line).

Comparing ranks. To understand if the identified top-10k domains (top-1M in brackets) are specific to the MNO dataset, we measure its overlap with the Lumen, Alexa, and Cisco datasets separately. Considering the Lumen dataset, we find 4,589 (22,730) of its domains intersecting MNO with a 64.8% (69.3%) traffic coverage. This overlap is quite high considering the Lumen dataset scale and its Android-specific nature (squares pattern line in Figure 1 (left)). Instead, only 30.2% (31.3%) of the MNO flows are found in the Alexa rank domains. This suggests that, despite being widely adopted by research studies for benchmarking, the Alexa rank does not provide a good picture of mobile traffic. Conversely, nearly 65% out of the top-10k domains in Cisco's rank are also present in the top-10k MNO domains. Those domains are responsible for 86.1% MNO traffic.

Figure 1 (center) shows, for the MNO dataset, the coverage as a function of domains popularity per mobile OS. We would like to stress how Lumen's domain coverage (Figure 1 (left)) is higher than the MNO coverage when considering traffic generated only from Android devices. This is because the top-10k domains are found both in iOS and Android apps. Nevertheless, the two OSes present differences. For instance, 805 (1,089) domains have 50x (20x) more iOS users than Android ones. Moreover, iOS devices generate almost twice as many flows compared to Android ones, thus generating a higher DNS traffic load too.

Multi-app domains. We leverage Lumen's ability to identify the app responsible for a given network flow in order to identify the set of domains relevant for a given mobile app, and the interactions between mobile apps and online services. We find that 75% (31%) of the apps in Lumen dataset connect to <10 (<2) second-level domains (SLDs). Conversely, over 32% of SLDs are reached by at least 2 apps. A careful inspection of these domains reveals a combination of third-party advertising and analytics services (e.g., DoubleClick, Flurry, and Crashlytics) and social network APIs (e.g., Facebook Graph API) [26, 28], common to both iOS and Android apps.

Figure 1 (right) shows the CDF of the number of MNO flows with respect to the number of apps per domain according to Lumen. This analysis shows that mobile traffic is dominated by third-party services, enriching the interpretation of Figure 1 (left): 4,954 MNO top-10k domains are contacted by at least 2 apps in the Lumen dataset, accounting for 60.1% of the total MNO flows.

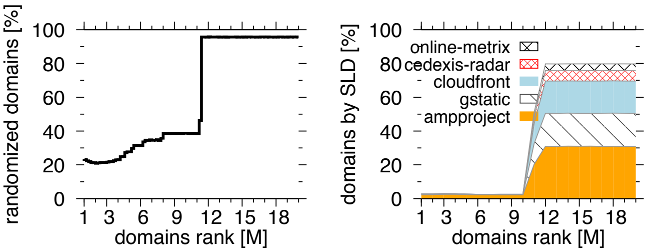

Figure 2: Dissecting ephemeral domains: fraction of randomized domains (left); percentage of domains used by top-5 most popular SLD (right).

Per-user traffic. Even if the top-10k are relevant at a global scale, we verify if they also dominate at a per-user level. Using the MNO logs, we observe that mobile users generally connect to a small number of domains: 50% users reach less then 200 domains, and only 20% users connect to more than 900 domains during a month-period. More importantly, 86% of users have more than 80% of their flows directed to the top-10k domains. Combining the results seen so far, the top-10k most popular domains are not only representative in terms of total volume, but also at a per-user and per-app scale.

Ephemeral domains. Despite the relevance of the top-10k domains, 82% of all domains were reached only once within a month across all 19M users, *i.e.,* they are *ephemeral*. This phenomena has been already reported in the literature for fixed access network [8]: 88% of all unique domain names in a major US cable provider were ephemeral, corresponding to 32% of DNS queries. The authors of this study associated most of these domains with CDNs, finding that they may be used for some sort of application-level signaling. Understanding their nature is critical as such a large number of ephemeral domains can "pollute" both cDNS and LDNS caches, thus reducing their effectiveness [16].

To better understand the nature of ephemeral domains, we focus on the left-most component of the domains which we find to be commonly "randomized".[5] We use basic natural language processing (NLP) rules and the English dictionary to classify domains as random or not, further validating the classification manually.

[5] One example of ephemeral domain is *d-22947172434039033539*.**ampproject**.net

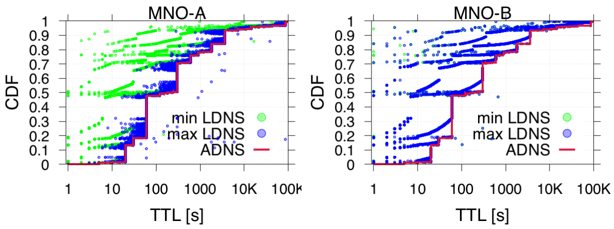

Figure 3: LDNS and ADNS TTL analysis.

Figure 2 (left) shows the fraction of randomized domains found in the MNO dataset as a function of their popularity rank grouped as 1M-bins. For visibility we limit the plot to 20M domains, but results hold for the remaining 160M domains. While the first 10M domains contain 60-80% of the most unpopular website or online services, more than 95% of the remaining ones are classified as randomized.

To understand who is responsible for those, we further group ephemeral domains by their SLD. Figure 2 (right) dissects the domains to highlight the top-5 SLDs with highest fraction of ephemeral domains; as above, we organize the domains in 1M-bins ordered by their popularity rank. The top-5 SLDs are mostly related to content delivery services, which confirms previous literature [8]. However, we find surprising the very strong concentration: 80% (160M) of the domains are handled by just 5 services. We also find unexpected TTL configurations: despite mostly being used once, both *ampproject*, and *gstatic* domains present a high TTL value (300s), while *cedexis-radar* adopts a very small value (20s). We further analyze TTL values and implications in the following sections.

5 DISSECTING THE STAKEHOLDERS ROLE

We now investigate how each DNS stakeholder handles the MNO top-10k domains. Specifically, we focus on TTL values and domain-to-IPs mapping diversity. We analyze the role of ADNS and LDNS resolvers first, and then users' device OS and apps.

5.1 ADNS Domain Configuration

Domain owners define both domain-to-IPs mapping and TTL values in the ADNS. Accordingly, they represent the "ground truth" for our analysis. We leverage the NexusTTL dataset which performs DNS queries for the top-10k domains towards the ADNS.

Figure 3 shows the CDF of the obtained TTL values assigned by the ADNS for the top-10k domains (solid line). To ease visualization we report only on 2 of the 4 operators in the NexusTTL dataset, and for a single day. Nevertheless, the observations hold across the tested operators and over the whole 1-month dataset. As we can see in Figure 3, low TTL values are common in mobile traffic: 52% (15%) of domains have TTL < 1min (20s) while only 15% of them have a TTL higher than 30min. We find limited TTL variability over time, suggesting that this is a *static* property of mobile domains.

We highlight that these observations have been reported by previous research efforts for fixed access networks [6]. However, the implication for mobile traffic can be significantly different as the DNS resolution time in mobile networks can be 2-3 orders or magnitude larger than in fixed networks [23].

5.2 LDNS Manipulations

When LDNS resolves a domain, it can cache the domain-to-IPs mapping of the response for a period of time regulated by the TTL. In practice, however, LDNS can "override" TTL values creating *TTL violations* [6]. To investigate these dynamics, we resort to the NexusTTL dataset to compare the TTL values found in the LDNS responses with those provided by ADNS.

LDNS caching. The y-axis in Figure 3 represents the top-10k domains, one per line, sorted by the ADNS TTL value. The x-axis maps TTL variability across queries. The solid line represents the expected value as provided by the domain owner (ADNS), while dots correspond to TTL values obtained querying LDNS. Different colors highlight *min* and *max* LDNS TTL values across different lookups. As we can see in the figure, LDNS provides lower TTL values due to caching: even if 52% of domains are expected to have a TTL < 1min according to the ADNS, this percentage is higher in practice for the end-user due to presence of the LDNS. However, this value varies across MNOs: MNO-A's LDNS provides (on average) TTL values which are 81% of the values provided by the ADNS, while this value gets reduced to 50% for MNO-B's LDNS. In other words, MNO-B's subscribers naturally flush their local DNS caches more often, so they perform more DNS queries.

As we actively run queries in rapid succession (and not in sync with TTL expiration), we expect to see small differences between *min* and *max* TTL values returned by LDNS servers. However, MNO-A TTL values (left plot) largely depart from each other as opposed to MNO-B (right plot). We conjecture that MNO-A may use different LDNS servers with load balance and (possibly) not sync-ed with each other. For instance, this would be possible via IP anycast, a fairly common technology for DNS [12]. Conversely, MNO-B may use a more centralized DNS architecture.

TTL violations. Figure 3 also shows instances of TTL violations, *i.e.,* cases in which the end-client receives DNS answers with TTL values exceeding the ones set by the ADNS. Those are occasional, although systematically happening for 40 domains in one of the 4 operators considered. Unfortunately, our data does not allow us to identify if this is the result of an intentional or suboptimal LDNS configuration. Nevertheless, we highlight that this finding contradicts the results from previous studies on wired access networks, where such violations have been shown to be more frequent [6].

5.3 TTL values and On-Device DNS Cache

Domain owners may not tune TTL values to end-users/apps traffic patterns: the lower the TTL value received by the client, the higher the chance to have a cache miss if the traffic is not "bursty", *i.e.,* subsequent flows are not generated right after the DNS request.

Our datasets do not allow us to analyze on-device DNS cache performance. However, we can simulate its behavior on a per-user basis with the MNO dataset. For that, given a user and a domain, we order the flows by their starting time, assuming that the first flow generates a local cache miss while subsequent flows generated within a time window as large as the domain TTL value assigned by the ADNS generate a cache hit. Then, the first connection outside the TTL window generates a new local cache miss, triggering a new DNS lookup and defining a new time window to check, and so on. Finally, we compute the fraction of cache misses for each

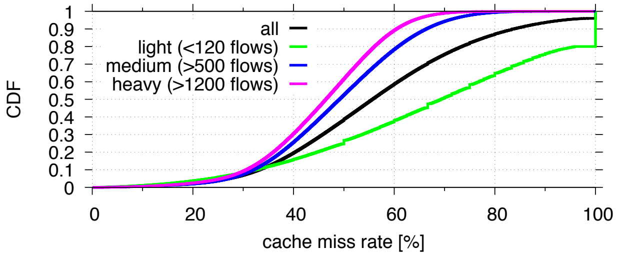

Figure 4: CDF of on-device DNS cache miss rate.

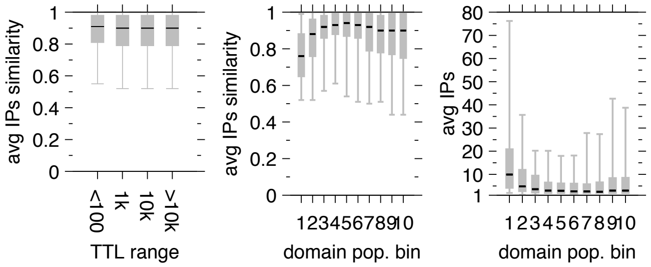

Figure 5: IP addresses variability: comparison with TTL (left), domain popularity (center), and average number of IPs (right).

user, and then compute its distribution across all MNO users. We consider a 4-hour period (peak time) for each day during a full week. We group MNO users as light, medium, or heavy, according to the quartiles of the distribution of the number of flows that they generate during the considered period of time.

As we can see in Figure 4, on-device caching is not very effective: 50% of users trigger DNS lookups for more than 65% of their flows. However, caching performance improves for user generating more traffic, with the median cache miss rate reducing to 44% for heavy users. The improvement is due to the fact that mobile traffic has indeed a "bursty" nature, but the still high cache miss rate highlights suboptimal policies. For instance, by artificially doubling the TTL duration, in our simulation the cache miss rate is reduced by 10%.

We stress that our methodology does not consider the presence of LDNS, which as seen in Sec. 5.2 likely serves lower TTL values to end-users. Consequently, our results are a low bound, and in practice users should experience a higher number of cache misses.

5.4 Measuring IPs Diversity

A possible explanation to the aggressive TTL values defined by ADNS resolvers is the need for load balancing [13]. If our assumption is correct, the lower the TTL, the larger set of IPs mapped to a given domain.

Correlation with TTL. To verify our hypothesis, we order the flows by their creation time for each of the top-10k domains in the MNO dataset. Then, we split the flows into consecutive time windows of twice the size of the domain ADNS TTL value.[6] For each time interval, we extract the set of IPs contacted by each MNO (user, domain) tuple, which we later compare across consecutive time windows with the *cosine similarity* index, using a binary vector to indicate the presence or absence of a given IP in a set. For each domain we measure the similarity across pairs of consecutive windows, averaging pair-wise similarities at the end. This allows us to obtain an index between 0 and 1 to measure IPs diversity for a given domain: the higher the value, the less variable the set of IPs.

As TTL values follow a step function (see Fig. 3), we bin per-domain similarity indexes based on 4 TTL ranges (<100s, 100-1,000s, 1,000-10,000s, >10,000s). For each bin, we compute the distribution of cosine similarities which we report in Figure 5 (left) as boxplots (3 quartiles, 5th, and 95th percentiles). Differently from what we initially expected, we find very high similarity values, and a weak correlation between domain-to-IPs mappings and ADNS TTL values. We conclude that it is arguable that aggressive TTL values are mostly used for dynamic mapping policies.

Correlation with popularity. Intuitively, popular domains should handle higher traffic load, hence they should use a wider range of IP addresses. To verify if this is the case, we split the top-10k into ten groups of 1,000 domains each. For each group, we compute the IPs cosine similarity that we report in Figure 5 (center) as boxplots. As in the previous experiment, IPs cosine similarity across domain bins is high. However, we see that IPs variability is proportional to domains popularity as we hypothesized. Figure 5 further details the actual number of IPs for each bin. We would like to stress how each domain is served by a median value of 2 IPs, which further corroborates on the presence of aggressive TTL policies. However, for some domains (namely very popular ones such as Google services) the configuration reflects the larger set of IPs used.

5.5 ISP Proxies and Developer Choices

Application developers and mobile platform idiosyncrasies can also have an impact on DNS traffic.

APN settings. Mobile operators provide an Access Point Name (APN) configuration to end-devices [27] in order to connect to the Internet. MNOs can use APN configurations to force user's traffic to go through an explicit proxy which could perform DNS queries on behalf of the device.

Android and iOS have a different way to control and enforce APN settings on the devices. Android has a hardcoded APN configuration file[7] containing APN setting for 1,175 MNOs. An inspection of the xml file reveals that 27% of the operators force users to go through an explicit proxy. In the case of iOS, none of the 598 MNOs found in the 3k configuration files hosted on a central repository [3] force traffic to go through an in-path proxy. Overall, we can conclude that explicit proxy configurations are not a popular solution across MNOs.

App developer. Despite the APN configuration, developers are still left with the final choice to use proxies or not depending on the socket libraries they use. Indeed, not all libraries honor the system proxy specified by the APN configuration. To investigate the role of the app developer on the DNS ecosystem, we follow a two-step approach. First, we use static analysis to identify the most popular

[6] We selected this time window to stress that the larger the time window, the higher the chance to see different IPs.

[7] `apns-full-conf.xml`

socket libraries used by the 20k popular Android apps. Then, we actively test them to verify if the APN configuration is honored or not. We find that over 80% of apps use the *HTTP(S)URLConnection* APIs to send/receive HTTP(S) traffic which indeed honors the system proxy as specified in the APN configuration. Moreover, we find that 69.5% of the apps also make use of Java Sockets which bypasses the explicit proxy setting. However, only 5.5% use them directly in their main package. Despite the fact that app developers can define their own optimizations, the majority of the apps stick to the default OS configuration. This implies using cDNS, and being exposed to the TTL values reduction (see Sec. 5.2), and suboptimal caching performance (see Sec. 5.3).

6 DNS MANAGEMENT AND QOE

We finish our study by quantifying the impact of DNS on user QoE along with the benefits of a few simple optimizations. We consider a Web browsing scenario and study page load time (PLT) as it is a well established performance metric [29, 30].

We process the HAR files provided by our `chrome-har-capturer` instrumentation in NexusPLT dataset to identify the *critical path*, *i.e.,* HTTP(S) transactions (*i.e.,* request/response pairs) which content is partially or entirely downloaded without other transactions in parallel. Those are the transactions that (if shortened) can potentially reduce the PLT. Figure 6 shows the CDF of the fraction of critical path spent doing DNS resolutions for Alexa's top-1k websites. Results refer to MNO-A, but identical observations have been found for the other three operators considered.

Overall, Figure 6 shows that the cost of DNS is inversely proportional to the complexity of a webpage, e.g., the median grows from 10% in webpages containing up to 50 objects up to about 20% in webpages containing only up to 5 objects. This happens because the number of DNS resolutions is not linear with the number of objects as many objects within a page reside at the same domain. On average, the DNS lookup accounts for 9.2% of the PLT (7.1s). A similar fraction was also observed previously when investigating the cost of DNS when loading webpages over fixed networks [30]. However, its impact on user QoE is quite different in mobile networks where DNS lookups can be two orders of magnitude larger, e.g., 100ms (mobile access) [23] versus 1-5ms (fixed access) [13, 24].

A potential solution to reduce DNS overhead can be to adopt an explicit proxy (enforced via APN configuration). With this setup the DNS resolution happens at the proxy, alleviating radio resources utilization. However, testing this scenario we find only a 4% reduction of PLT in average.

To further reduce PLT, the optimal scenario would be to have an "oracle" which *pre-stages* the resolution of the required domains into the device DNS cache. This approach could be implemented by forcing the device to resolve domains before their usage as done for instance by modern browsers. However, this design would not reduce the overall network overhead. Conversely, an in-network component could potentially resolve a predefined set of domains and push them to the devices cache. Independently from the actual implementation, we tested this oracle scenario by resolving all domains found in the benchmarking webpages considered, and loading them into the device cache before running an experiment. In this optimistic scenario, the average gain is of about 8.5%, very close to the expected value. However, DNS operations are not fully eliminated due to ephemeral domains (about 10% in the considered benchmark) that partially nullify the positive effect of basic pre-staging. We underline that, being Alexa's rank not entirely representative of mobile traffic (see Sec. 4), it would be more interesting to assess DNS impact on generic mobile apps traffic. However, doing so would require defining a new performance metric as PLT no longer applies. We leave this investigation for a future work.

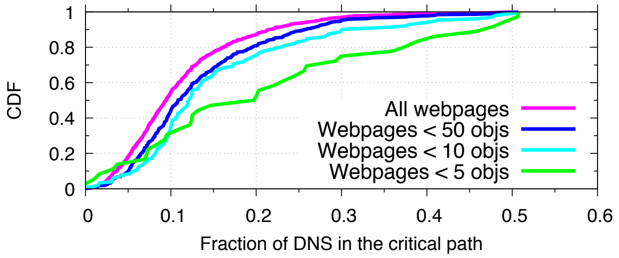

Figure 6: CDF of the fraction of time spent doing DNS resolutions on webpage critical path.

7 CONCLUSIONS

This paper has presented a holistic analysis of the Domain Name System (DNS) and its *stakeholders* in mobile networks. Our methodology consists in analyzing real traffic passively collected from a large European mobile network operator and from Lumen, a traffic monitoring app for Android with 5,000 users. When needed, the analysis was complemented with active measurements across four MNOs. Among the DNS stakeholders, domain owners are the main responsible for high DNS traffic load by setting low TTL values—not entirely justified by load-balancing arguments considering the limited domain-IPs mapping variability. Further, specific LDNS architectures can increase DNS traffic by forcing clients to perform new lookups. Adopting less aggressive TTL values would alleviate part of the load, but it requires cooperation from domain owners or the CDNs hosting them. An alternative, but unpopular, solution consists in adopting explicit proxies (enforced via APN configuration) where DNS operations are offloaded to the proxy saving radio resources. DNS pre-fetching is another solution, already implemented in Chrome, but this improves performance only for the apps adopting it, and does not reduce the network traffic. The optimal solution should instead target multiple apps, while reducing network traffic. Hence, we speculate that implementing pre-staging of DNS lookups on the device can be a more transparent, effective, and network-friendly solution. An in-network component can passively monitor the traffic to study and predict relevant domains (for each user, and across all apps), and push the domain-to-IPs mapping to the devices. The challenge ahead to make this a reality is identifying the right set of domains to pre-stage considering: (i) IPs mapping variability, (ii) global network load, (iii) app characteristics, and (iv) user needs.

ACKNOWLEDGEMENTS

This paper is partially funded by the NSF grant CNS-1564329.

REFERENCES

[1] Victor Agababov, Michael Buettner, Victor Chudnovsky, Mark Cogan, Ben Greenstein, Shane McDaniel, Michael Piatek, Colin Scott, Matt Welsh, and Bolian Yin. 2015. Flywheel: Google's Data Compression Proxy for the Mobile Web. In *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

[2] Andrea Cardaci. 2017. Chrome Har Capturer. (2017). https://github.com/cyrus-and/chrome-har-capturer.

[3] Apple. 2017. iOS APNs default configuration file. (2017). http://phobos.apple.com/version.

[4] HTTP Archive. 2016. Alexa 1M. (2016). http://httparchive.org/about.php#listofurls.

[5] Fabian Bustamante and John Rula. 2014. Alice - A Lightweight Interface for Controlled Experiments. (2014). http://aqualab.cs.northwestern.edu/262-details-alice.

[6] Thomas Callahan, Mark Allman, and Michael Rabinovich. 2013. On Modern DNS Behavior and Properties. *ACM SIGCOMM Computer Communication Review* 43, 3 (July 2013), 7–15.

[7] Qi Alfred Chen, Haokun Luo, Sanae Rosen, Z. Morley Mao, Karthik Iyer, Jie Hui, Kranthi Sontineni, and Kevin Lau. 2014. QoE Doctor: Diagnosing Mobile App QoE with Automated UI Control and Cross-layer Analysis. In *Proc. ACM Internet Measurement Conference (IMC)*. ACM.

[8] Yizheng Chen Chen, Manos Antonakakis, Roberto Perdisci, Yacin Nadji, David Dagon, and Wenke Lee. 2014. DNS Noise: Measuring the Pervasiveness of Disposable Domains in Modern DNS Traffic. In *Proc. IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*.

[9] Cisco. 2017. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016—2021 White Paper. (2017). http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html.

[10] Dan Hubbard CISCO. 2016. Cisco Umbrella 1M. (2016). https://umbrella.cisco.com/blog/blog/2016/12/14/cisco-umbrella-1-million.

[11] Jeffrey Erman, Alexandre Gerber, Mohammad Hajiaghayi, Dan Pei, Subhabrata Sen, and Oliver Spatscheck. 2011. To cache or not to cache: The 3g case. *Internet Computing, IEEE* 15, 2 (March 2011), 27–34.

[12] Xun Fan, John S. Heidemann, and Ramesh Govindan. 2013. Evaluating anycast in the domain name system.. In *Proc. IEEE International Conference on Computer Communications (INFOCOM)*.

[13] Alessandro Finamore, Ignacio Bermudez Bermudez, and Marco Mellia. 2013. Public DNS Resolvers: Friends or Foes? (2013). http://www.retitlc.polito.it/finamore/papers/dns-techreport.pdf.

[14] Utkarsh Goel, Moritz Steiner, Mike P. Wittie, Martin Flack, and Stephen Ludin. 2016. Detecting Cellular Middleboxes Using Passive Measurement Techniques. In *Proc. Passive and Active Measurement (PAM)*. Springer.

[15] Google. 2016. Pre-resolve DNS. (2016). https://developers.google.com/speed/pagespeed/service/PreResolveDns.

[16] Haining Hao, Shuai an Wang. 2017. Exploring Domain Name Based Features on the Effectiveness of DNS Caching. *ACM SIGCOMM Computer Communication Review* 47, 1 (2017), 36–42.

[17] Junxian Huang, Qiang Xu, Birjodh Tiwana, Z. Morley Mao, Ming Zhang, and Paramvir Bahl. 2010. Anatomizing Application Performance Differences on Smartphones. In *Proc. ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[18] Ashish Kumar. 2016. Using Prefetch as a Proactive Approach. (2016). http://blog.catchpoint.com/2017/04/28/prefetching-proactive-approach/.

[19] Ashkan Nikravesh, Hongyi Yao, Shichang Xu, David Choffnes, and Z. Morley Mao. 2015. Mobilyzer: An Open Platform for Controllable Mobile Network Measurements. In *Proc. ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*.

[20] David Pariag and Tim Brecht. 2017. Application Bandwidth and Flow Rates from 3 Trillion Flows Across 45 Carrier Networks. In *Proc. Passive and Active Measurement (PAM)*.

[21] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, and V. Paxson. 2015. Haystack: In Situ Mobile Traffic Analysis in User Space. *ArXiv e-prints* (2015).

[22] Sanae Rosen, Haokun Luo, Qi Alfred Chen, Z. Morley Mao, Jie Hui, Aaron Drake, and Kevin Lau. 2014. Discovering Fine-grained RRC State Dynamics and Performance Impacts in Cellular Networks. In *Proc. ACM International Conference on Mobile Computing and Networking (MobiCom)*.

[23] John P. Rula and Fabian E. Bustamante. 2014. Behind the Curtain: Cellular DNS and Content Replica Selection. In *Proc. ACM Internet Measurement Conference (IMC)*. Vancouver, BC, Canada.

[24] Kyle Schomp, Mark Allman, and Michael Rabinovich. 2014. DNS Resolvers Considered Harmful. In *Proc. ACM HotNets*. Los Angeles, CA, USA.

[25] Narseo Vallina-Rodriguez, Andrius Auçinas, Mario Almeida, Yan Grunenberger, Konstantina Papagiannaki, and Jon Crowcroft. 2013. RILAnalyzer: a comprehensive 3G monitor on your phone. In *Proceedings of the ACM IMC*.

[26] N. Vallina-Rodriguez, J. Shah, A. Finamore, Y. Grunenberger, K. Papagiannaki, H. Haddadi, and J. Crowcroft. 2012. Breaking for commercials: characterizing mobile advertising. In *ACM IMC*.

[27] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Christian Kreibich, Nicholas Weaver, and Vern Paxson. 2015. Beyond the radio: Illuminating the higher layers of mobile networks. In *ACM MobiSys*.

[28] Narseo Vallina-Rodriguez, Srikanth Sundaresan, Abbas Razaghpanah, Rishab Nithyanand, Mark Allman, Christian Kreibich, and Phillipa Gill. 2016. Tracking the Trackers: Towards Understanding the Mobile Advertising and Tracking Ecosystem. *DAT Workshop* (2016).

[29] Matteo Varvello, Jeremy Blackburn, David Naylor, and Konstantina Papagiannaki. 2016. EYEORG: A Platform For Crowdsourcing Web Quality Of Experience Measurements. In *CONEXT*.

[30] Xiao Sophia Wang, Aruna Balasubramanian, Arvind Krishnamurthy, and David Wetherall. 2013. Demystifying Page Load Performance with WProf. In *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*.

[31] Kyriakos Zarifis, Tobias Flach, Srikanth Nori, David R. Choffnes, Ramesh Govindan, Ethan Katz-Bassett, Zhuoqing Morley Mao, and Matt Welsh. 2014. Diagnosing Path Inflation of Mobile Client Traffic. In *Proc. Passive and Active Measurement (PAM)*.