What constitutes DNS traffic today?

Changing landscape of DNS

Allen T. Webb
Dept. of ECE
Texas A&M University
allenwebb@tamu.edu

A. L. Narasima Reddy
Dept. of ECE
Texas A&M University
reddy@ece.tamu.edu

ABSTRACT

New technology and new applications are driving rapid change in the services accessed over the Internet. The Domain Name System (DNS) is fundamental to the day-to-day operation of the Internet. As new uses for DNS are introduced and the ways people use the Internet change, the impact on existing network utilities and services need to be continuously studied and evaluated. We present a study of the DNS traffic on a large campus network and investigate the impact of new applications of DNS. We study various aspects of DNS traffic including the network sources, application sources, and especially the various sources of DNS NXDOMAIN responses. In addition, we look into the differences in DNS traffic from wired hosts and Wi-Fi hosts. Our results indicate that non-negligible fraction (10-15%) of DNS queries result in NXDOMAIN responses, many new applications have come to rely on DNS as a side channel through network firewalls and that wireless DNS traffic can exceed the wired DNS traffic.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network monitoring*; C.2.5 [**Communication Networks**]: Local and Wide-Area Networks—*Internet*; C.4 [**Performance of Systems**]: Measurement studies

Keywords

DNS, measurement study, Wi-Fi vs. wired

1. INTRODUCTION

The Domain Name System (DNS) [21, 22] provides a way to map human readable names into Internet Protocol address which can be routed to specific endpoints on the Internet (e.g. www.domain.tld). It serves as a distributed database for records which are identified with a multi-part name. DNS servers have authority over subsets of names delegated to them. In this way the administrative burden of updating records is spread across the organizations which have authority over a particular domain. DNS traffic is typically allowed through firewalls; thus, many new services are using DNS infrastructure as a means of communicating information.

DNS traffic has been widely studied to understand network traffic at large. DNS traffic provides a view into traffic patterns [7], popular domains. DNS responses give a view into the mechanisms of load balancing [29, 23] and the functioning of Content Distribution Networks (CDNs) [27]. DNS traffic has recently been used to provide a view into botnet behavior, by distinguishing the botnet generated DNS query traffic from human generated query traffic [5]. Collecting and analyzing DNS traffic is more scalable than carrying out similar analysis on the entire network traffic.

This study is motivated by two trends in network usage. The first is the growing use of mobile and wireless devices. Users have started replacing desktop computers with laptops and other devices; the widespread adoption of smartphones with Wi-Fi capabilities is changing the network usage patterns. The introduction of mobile devices brings several aspects to the study of DNS traffic. The network traffic can be different because of differences in application usage, network usage, and the different resources in the devices. The application usage on mobile devices may include more frequent accesses to social networking and video sites. The wireless devices on a campus that has wide Wi-Fi coverage allow for more continuous or longer usage of network resources compared to desktop devices. Mobile devices with limited resources (and hence lower caching) may also generate more traffic than desktop devices.

The second trend is the usage of DNS as a side channel by many applications. Most campus and enterprise networks are protected by firewalls at the edge. These firewalls restrict the flow of traffic between the protected network and the outside world. However, most network firewalls typically allow communication on a few designated ports. The ports for HTTP and DNS are commonly left open, and a number of applications have come to rely on these open ports for getting through the firewalls. DNS is being used by a number of applications from spam checkers to anti virus software as a side channel to contact outside servers to provide their service. Other new new services we observed are ENUM over DNS (subsection 5.6), and the ICANN name collision block list. New services which make use of the Domain Name System (DNS) infrastructure change the landscape of DNS queries.

This paper studies the DNS traffic to observe and understand the impact of these two trends. We collect and analyze DNS traffic at an educational institution to understand the nature of network traffic.A better understanding of what is implied by DNS traffic improves decision making in the area of how much monitoring is warranted to stop outbreaks of malicious software and detect intrusion while protecting individual privacy. A better picture of DNS traffic will also help gain an understanding of scalability and other issues. For example, DNS traffic has been recently used as a means to identify botnet C&C (command and control) [33], [34], and [3]. Specifically, these earlier studies have used DNS NXDOMAIN responses as a means of identifying potential botnet traffic. Many new applications such as spam checkers and antivirus tools that use DNS as a side channel generate a significant number of NXDOMAIN responses as part of their legitimate service. Understanding and separating this traffic is important for continuing to provide security of the networks.

The paper makes the following significant contributions:

- Collects and analyzes the DNS traffic over several days at a campus network, separating the traffic from wired networks and wireless networks.
- Shows that wireless networks are starting to dominate the campus traffic, leading to implications on scalability and throughput of these networks.
- Shows that a significant fraction of DNS traffic results in failures, much of this traffic originating from new services exploiting DNS as a side channel. These increased failure traffic increase the processing burden for separating legitimate traffic from malicious traffic.
- Shows that the application usage across wireless and wired networks leads to different network accesses.

2. RELATED WORK

DNS traffic has been widely studied to understand network traffic at large. DNS traffic provides a view into traffic patterns [7], popular domains and through responses a view into the mechanisms of load balancing [29, 23] and the functioning of Content Distribution Networks (CDNs) [27].

DNS traffic in cellular networks has been recently studied [27]. They investigate the impact on CDN load balancing performance from using a single DNS server for a geographically diverse set of mobile devices. DNS has been widely used for load balancing and reducing delays over wide area networks through redirection [29, 23]. DNS has also been used as a way to measure delays across the internet [18]. DNS measurement studies have been conducted in the past from the point of view of a root DNS server [7]. This study identified violations of the DNS standard, reflection attacks, and divided up DNS failure responses by the reasons they failed.

Domain registration has been studied for the possibility of identifying spammers early in [12]. DNS behavior has been studied from several vantages to understand the global DNS query patterns which can be used to detect malicious domain groups [10].

DNS traffic has recently been used to provide a view into botnet behavior, by distinguishing the botnet generated DNS query traffic from human generated query traffic. In [31], [33], [34], and [3] non-existent domain queries are used to identify botnets which use domain generation algorithms (DGA) for locating the C&C channel. The query rates for bots were observed to be high, occurring over a short period of time, and generating NXDOMAIN responses in [31]. DNS error responses were classified using the type of error, query entropy over the number of requests per local resolver and authoritative name server, and query content in [16]. They found a string of malicious domains copying strings from common social network domains. In [2] and [5] rather than focusing on DGA, a classification scheme is developed for malicious domains. [5] makes use of timing information, address record IP related information, reverse record information, TTL statistics, and domain name features to perform the classification.

The possibility of information being leaked through DNS is investigated in [26]. They employ several layers of filtering to identify interesting traffic to be investigated. Their final detection is based on a bound on how much information may be conveyed through the contents and timing of DNS traffic after filtering. Several services now use DNS for purposes beyond looking up the address record of a server by name. We present an exhaustive survey of DNS traffic to identify new areas of usage.

In addition to intentional leakage of information, the architecture of a network can be inferred through DNS traffic. The problem of remotely identifying the type of DNS client infrastructure is addressed in [28]. They distinguish DNS clients into open DNS servers which are forwarders, or recursive resolvers, direct recursive resolvers, and hidden DNS servers. They perform this classification by probing for DNS servers by issuing requests to an authoritative name server they control.

DNS traffic can be analyzed to identify the operating systems of hosts. DNS query traffic has been em-

		Dataset 1	Dataset 2	Dataset 3	Dataset 4
Time Period		Summer	Fall Semester	Spring Sem. Weekend	Spring Sem. Week
Start Date		June 19, 2014	Sep. 9, 2014	Jan. 16, 2015	Jan. 22, 2015
Duration (days)		13.07	10.26	2.26	7.02
Requests	Total	2.02E+09	2.57E+09	2.36E+08	1.82E+09
	% No Error	70.11%	92.02%	86.35%	91.10%
	% Error	29.89%	7.43%	13.00%	7.99%
Unique Domains	Total	9.22E+06	9.08E+06	1.08E+06	4.81E+06
	% Resolved	47.10%	57.72%	76.67%	76.66%
	% Not Res.	52.90%	42.28%	23.33%	23.34%
Unique IPs	Total	1.77E+05	1.66E+05	9.65E+04	1.78E+05
	% Wired	42.96%	29.63%	24.74%	26.59%
	% Wi-Fi	57.04%	70.37%	75.26%	73.41%

Table 1: Volume of traffic present in each datasets. It should be noted that for both Dataset 1 and Dataset 3 and weekend dataset there are fewer students on campus than the other datasets. This provides a variety of different conditions to observe the unique and common features across the datasets.

ployed to carry out passive OS fingerprinting in [20] using OS-specific DNS queries and timing of DNS queries. They also estimated the number of devices generating the queries.

Work has been done to evaluate the performance of DNS caches in [15]. In their study DNS traffic was collected along with TCP SYN, FIN, and RST packets. They investigated the relationship between TTL and DNS cache hit rate and found that most of the DNS cache hits occur within minutes of the initial miss.

Other work has investigated the difference in usage patterns between traditional computers such as desktops and mobile devices [9]. This work makes no mention of DNS but instead relies on other measurements. Our work seeks to gain insights into mobile device usage with only a subset of the network traffic.

3. DATASETS / METHODS

Our measurements were taken from a campus of about 60,000 users. We collected all the DNS traffic from four time periods: one during the summer of 2014, and one during the fall semester of 2014, and two during the spring semester of 2015. Our data includes requests for which the campus DNS servers are the start of authority (SOA) and also local queries which were recursively resolved. The size of the datasets and some basic statistics about their constituents can be found in Table 1.

To collect our data we tapped DNS traffic directed to the campus DNS resolvers. We excluded traffic involved in the process of recursive DNS resolution. Thus, each dataset contains those DNS requests made to the local campus resolvers, as well as the corresponding responses from the resolver back to the requester. These traces were then processed afterward. DNS names longer than the 255 byte limit are included in the count as an empty domain, “”. These made up 0.05% to 0.18% of the total requests.

It is notable that in Table 1 between 7.43% and 29.89% of the DNS requests failed across the datasets. The DNS failure rates were higher during the summer and weekends. The hosts which filter out Spam emails generate a higher ratio of failed to successful DNS responses than typical users. When there are more typical users such as the university students during the semester or week, the relative contribution of the Spam filter traffic is less. In dataset 1 the hosts which accessed Spam blacklists had a 50.7% DNS error rate and were responsible for 37.5% of the total DNS responses. In the other datasets these hosts were responsible for 13-18% of the DNS responses. We found that four forwarding nameservers were responsible for the much larger nonexistent domain percentage in dataset 1. For datasets 2 and 4 the error rates of hosts accessing Spam blacklists were between 16.4% and 19.0%, while for dataset 3 the error rate was 29.4%. The higher error rate for dataset 3 is likely due to the volume of DNS errors remaining relatively flat across the week while the successful traffic is more dependent on the time of day and day of the week, see Figure 4.

Also of note, is the difference in percentages of domains which resolved versus domains which did not resolve. Domains which resolved had at least one successful DNS response while domains which didn't resolve had no successful DNS responses. There were a higher percentage, 70.11% to 92.02%, of successful DNS responses than resolved domains, 47.10% to 76.66%.

The unique IPs in Table 1 show the change in relative percentage of IP addresses between the wired and Wi-Fi networks during the summer when students were away

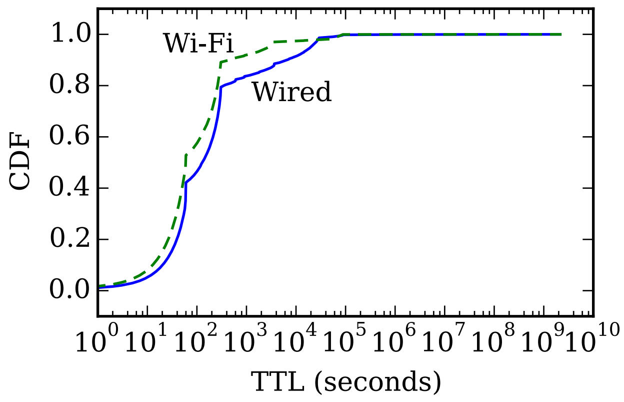


Figure 1: CDF of DNS Address Record TTL Values

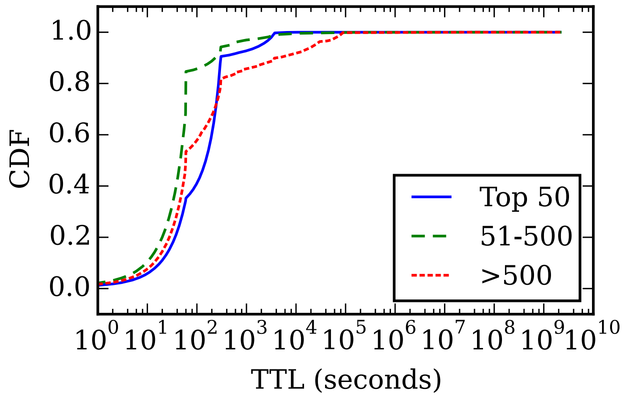


Figure 2: CDF of DNS Record TTLs Grouped by Alexa Rank

versus during the semesters when students were present. Between datasets 1 and 2 this difference was 13.33%.

4. WIRED VS. WI-FI DNS TRAFFIC

The availability of low cost and portable computers such as laptops, smart phones, and tablets has introduced new variation into the way networks are used. One window for observing changes in usage is the DNS requests made by different classes of devices.

Alexa is an analytics company which provides rankings for the most popular websites on the Internet [1]. We compared the ranking of domains on Alexa with the ranking of domains by volume of DNS requests and provide a summary here: The top 50 domains on Alexa were responsible for 34.69% of the successful requests from Wi-Fi IPs and 16.59% on wired IPs. The top 500 domains on Alexa were responsible for 48.63% of the successful requests from Wi-Fi IPs and 27.01% on wired IPs. Of the top 500 domains on Alexa: Four second level domains were responsible for 21.93% of all the successful DNS requests, and of these 83.30% of the requests originated from Wi-Fi IP addresses. 97 second level domains had more successful DNS requests from wired hosts and these accounted for 1.70% of all the successful DNS requests. This shows that a small subset of domains are responsible for a significant bias in the amount of successful DNS queries between wired and Wi-Fi devices.

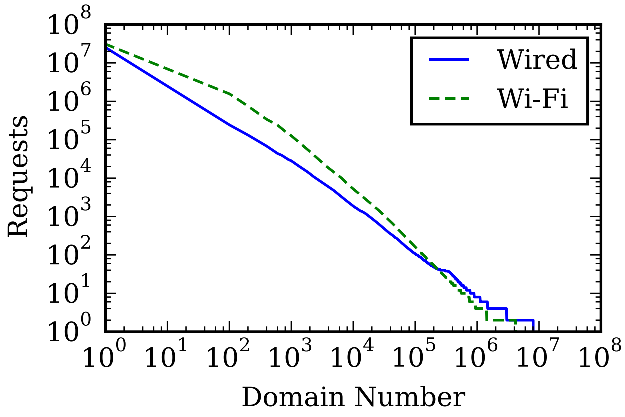


Figure 3: Volume of DNS Requests for each Domain

In Table 2 the top 50 most requested second level domains are listed. Only successful requests are counted in this table. The *% All* column is the percentage of all successful DNS requests which match the specified second level domain. The *Wi-Fi %* column is the fraction of successful DNS requests for the specified second level domain which originate from Wi-Fi IP addresses. The *Rank* column shows the rank of that domain in the top 500 Alexa list.

The Time to Live (TTL) value of a DNS record conveys how long the record should be cached before expiring. More information can be inferred from the TTL values as they reflect a trade off between DNS server load and update response time. A record with a higher TTL is cached longer and is requested less frequently as a result. Web services which depend heavily on load balancing to provide better quality-of-service such as video streaming use small TTLs to allow for faster control over which servers receive new clients. Content distribution networks (CDN) also use small TTL values. Fast-fluxing DNS botnets use small TTL values to make it more difficult to blacklist all their IP addresses and domain names.

The TTL values for different subsets of domains grouped by Alexa rank are shown in Figure 1. More than 80% of the domains ranked 51st through 500th had TTL values of 60 seconds or less. For the top 50 domains almost half of the requests had TTL values between 60 seconds and 5 minutes. The remaining domains had more of a spread over the different TTL values with about half having TTL values 60 seconds or less. Figure 1 shows the CDF of the TTL values of address records for Wi-Fi and wired devices. We saw that more of the TTL

2nd Level Domain	% All	Wi-Fi %	Rank
apple.com	20.76%	92.77%	46
google.com	18.49%	73.22%	1
facebook.com	8.48%	85.52%	2
amazonaws.com	5.15%	77.69%	130
akamaihd.net	4.96%	79.39%	288
instagram.com	4.57%	94.79%	31
yahoo.com	3.18%	74.65%	4
icloud.com	2.96%	89.89%	480
twitter.com	2.79%	77.25%	9
googleapis.com	1.83%	82.32%	495
microsoft.com	1.78%	48.11%	47
bing.com	1.54%	86.59%	25
cloudfront.net	1.33%	65.94%	361
googleusercontent.com	1.29%	68.81%	94
netflix.com	1.22%	76.74%	59
fbcdn.net	1.01%	78.86%	425
amazon.com	0.92%	76.14%	7
live.com	0.73%	62.16%	12
youtube.com	0.73%	72.94%	3
spotify.com	0.69%	91.15%	345
msn.com	0.65%	73.41%	27
twimg.com	0.62%	81.55%	409
tumblr.com	0.62%	79.30%	39
dropbox.com	0.61%	43.60%	83

Table 2: Most Requested Second Level Domains from Alexa Top 500

values of DNS address records requested by Wi-Fi devices were sixty seconds or less than 5 minutes. Other common TTL values include three hours and one day. This suggests that users of Wi-Fi devices requested more CDN hosted content such as online videos than users of wired devices.

The first two second level domains, *apple.com* and *google.com*, both make up a very large percentage of the DNS traffic. This can be explained by synchronization or other mobile device features on devices running iOS and Android. Also, small TTL values introduce positive bias for the number of DNS requests a domain will receive; domains with larger TTL values will be cached longer and have fewer requests than domains with smaller TTL values. In Table 2 only, *microsoft.com* and *dropbox.com*, have more DNS requests from wired hosts than Wi-Fi hosts.

Figure 3 shows that the volume of requests of domains is nearly linear on a log-log plot. This resembles a distribution following a power law [15]. However, there is a slight curve which suggests that it is actually a power law with exponential cutoff or a log-normal distribution. The distribution of requests from wired devices is more linear on the log-log plot than from Wi-Fi devices.

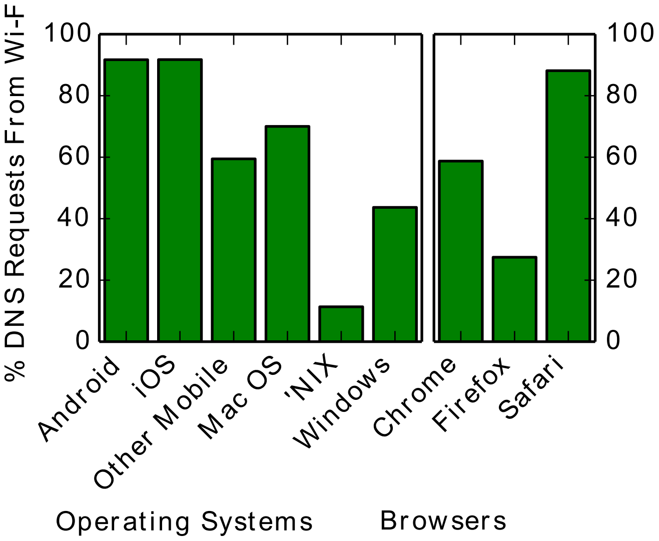


Figure 5: Percent of Total DNS Requests from Wi-Fi hosts for Software Specific Domain Names (Dataset 4)

Figure 4 shows the number of DNS responses with and without errors per five minute interval. The data showed a different diurnal cycle for wired versus wireless especially for DNS error responses. The difference in diurnal cycle is likely due to the Wi-Fi devices being used after work hours for personal use, but the wired devices were mostly accessed during work hours. In a typical day, the wired hosts see a drop-off in activity after 5-6PM and the wireless hosts were active until 10-11PM.

The volume of DNS errors for Wi-Fi hosts more closely follows the volume of DNS successes. The relatively flat error response volume for wired hosts in Figure 4 is due to the requests from Spam filters which are related to the volume of email traffic. The Wi-Fi hosts are responsible for more successful DNS responses than the wired hosts, but the Wi-Fi hosts produce less failed responses than the wired hosts. The spikes in Wi-Fi error responses on Figure 4 result from anomalous traffic which is discussed in subsection 5.7.

One interesting feature in Figure 4 is centered 4.5 days after the start of the trace. There was a football game during this time which resulted in less than half the Wi-Fi DNS responses on each side of the valley. The DNS responses from wired hosts did not have a noticeable change in the volume.

It is possible to predict when students are between classes by changes in the volume of traffic. There are regular peaks on the Wi-Fi DNS responses without errors in Figure 4 during the breaks between classes.

Figure 5 shows the percentages of DNS requests made for software specific domain names. The particular domain chosen where those used for software updates and

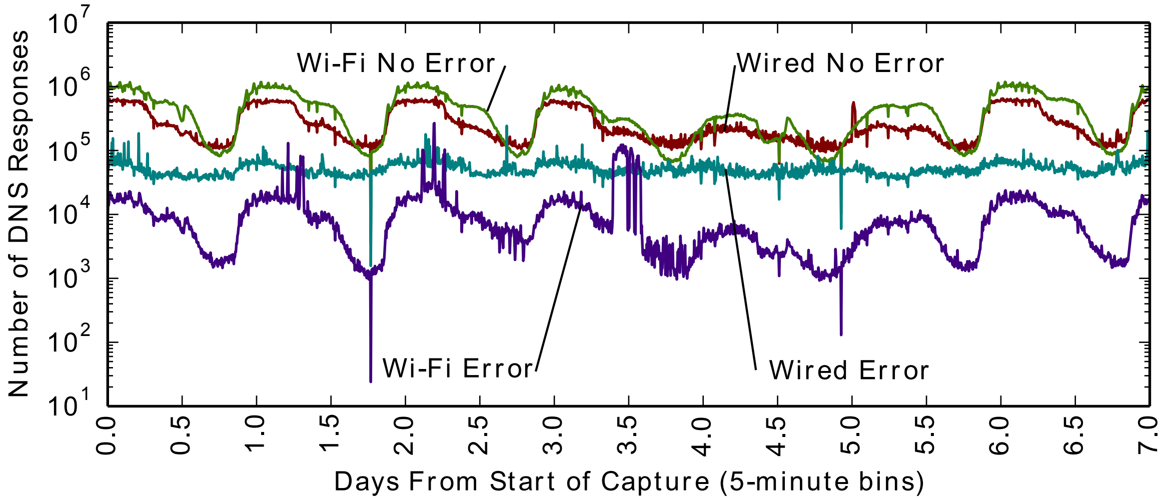


Figure 4: Wired vs. Wi-Fi DNS Response Volume.

account synchronization. An example software specific domain is *android.clients.google.com* which is used by android to download application updates and to access the Google Play API. As expected Android and iOS DNS requests predominately originated from Wi-Fi IP addresses. There is a strong bias toward mobile traffic originating from Wi-Fi IPs, so by observing the Wi-Fi tendencies we can infer mobile usage trends. Windows was split about evenly between the Wi-Fi and wired IP addresses; however, Mac OS had a higher number of requests from Wi-Fi IP addresses. Safari is closely tied to iOS and Mac OS so its percentage lied between the two operating systems. Requests from Firefox were more common from wired hosts; requests from Chrome was more common on Wi-Fi hosts. These percentages are from the raw total number of requests.

Mobile devices such as smart-phones and tablets are increasingly common. These devices typically connect to local networks through Wi-Fi. Separate web sites are provided for these devices because they have touch screens instead of a mouse and keyboard. Also, their screens may be much smaller so that a different layout is preferable than for desktops or laptops. Domains of the following formats are associated with mobile devices:

- ***.mobi** TLD dedicated to mobile websites
- **api.*** subdomain frequently used for web and mobile application programming interface
- **m.***, **mini.***, and **mobile.*** subdomains used for websites formated for mobile devices.
- **palm.***, **pda.*** subdomains used to host websites formatted for personal digital assistants or PalmOS devices.
- **wap.***, **wireless.*** subdomains used for a wireless application protocol browser. This is not used as much now that smart phones can handle HTML which has looser programming requirements but requires more processing power.
- **xhtml.*** subdomain used for an xhtml mobile site which is both HTML and XML compliant requiring less processing power.

The *api.* subdomains made up the largest percentage of the responses with no error with between 0.61% and 1.19% for wired hosts and between 2.46% and 5.71% for Wi-Fi hosts. Although we did observe requests made to the other mobile related domains, these made up less than 0.005% of the total requests.

	Dataset	1	2	3	4
api.*	Wi-Fi	2.46%	3.63%	4.68%	5.71%
	Wired	0.61%	1.19%	0.64%	1.02%
m.*	Wi-Fi	0.42%	0.38%	0.34%	0.30%
	Wired	0.26%	0.29%	0.18%	0.24%
mobile.*	Wi-Fi	0.12%	0.18%	0.22%	0.21%
	Wired	0.03%	0.06%	0.03%	0.05%
*.mobi	Wi-Fi	0.11%	0.18%	0.17%	0.18%
	Wired	0.01%	0.03%	0.01%	0.02%

Table 3: Successful Mobile Related DNS Responses

Table 3 shows that the mobile handsets are not necessarily always visiting mobile domains. The api.* domain group is commonly used with mobile applications instead of websites in a browser. Not all mobile websites use mobile domains; some share the same domain as the website formatted for desktops. Also, many websites are designed using a “responsive design” [19] which is shared across different devices and screen formats. Although there are still hits for older style mobile domains such as palm.*, pda.*, wap.*, and xhtml.*, these

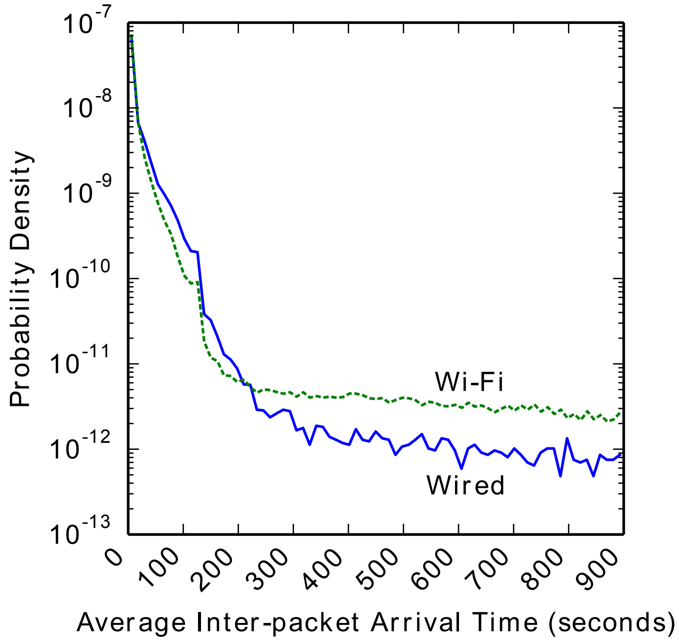


Figure 6: Distribution of Average Delay Between DNS Requests

domains make up a very small percentage of the total responses.

Besides differences in the domains being queried, there were also rate differences between wired and Wi-Fi hosts. We observed higher DNS packet rates on wired hosts vs Wi-Fi hosts based sessions defined by a 15 minute separation between packets for a given IP address shown in Figure 6. To exclude the bias introduced by the edge of the session window we calculated the mean time between packets to be the the time of the last packet minus the time to the first packet divided by the total number of packets minus one. Wi-Fi IP addresses had a higher likelihood of long periods of time between DNS requests than wired IP addresses. Wi-Fi devices may be used more intermittently than wired devices. Wired IP addresses with multiple simultaneous hosts through NAT, and higher demand from DNS dependent services such as Spam filtering contribute to the higher rate on wired IP addresses.

Table 4 shows the percentages of requests linked to applications other than getting the IP addresses of servers by category. Although ARPA is an existing use of DNS, its use in verification for Spam filtering is notable. DNS based blacklists have been created for filtering Spam and identifying viruses for anti-virus tools (AV). WPAD uses DNS to locate a web proxy auto discovery script. Invalid DNS names have been registered as a work around for application bugs. Random top level domains (TLD) are being used to detect DNS hijacking.

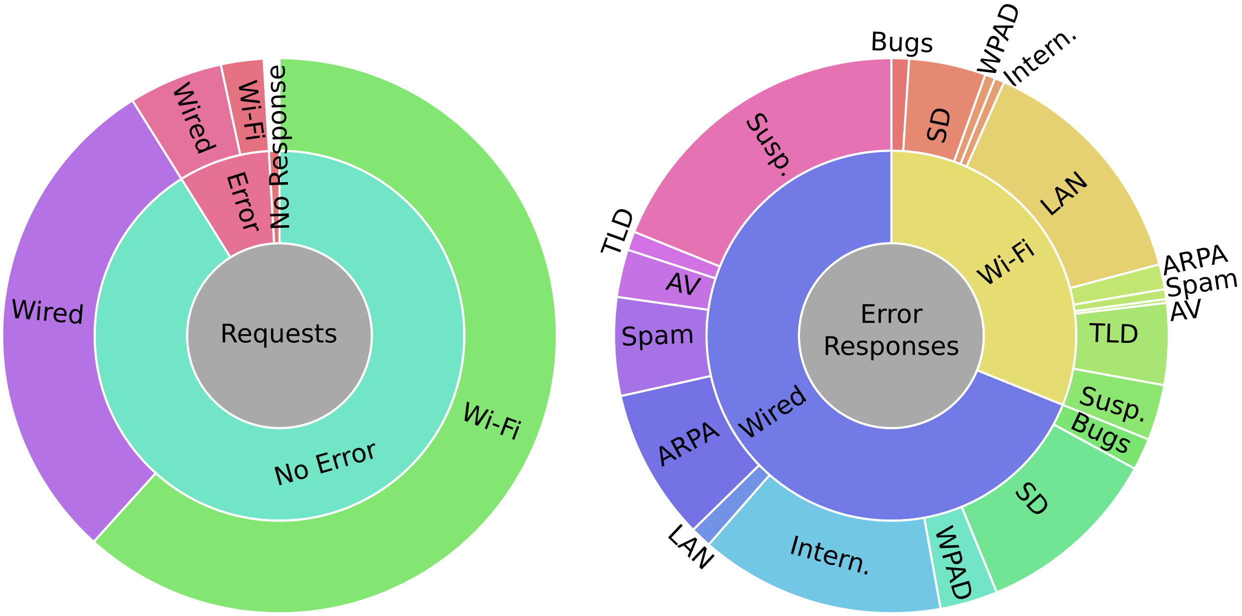
	Wired		Wi-Fi	
Total	12.24%		2.22%	
	Successes	Failures	Successes	Failures
ARPA	3.06%	2.01%	0.34%	0.18%
DNS-SD	0.23%	2.47%	0.03%	0.55%
Spam	0.81%	1.30%	0.07%	0.07%
AV	0.28%	0.63%	0.10%	0.03%
WPAD	0.00%	0.77%	0.00%	0.07%
Bugs	0.01%	0.42%	0.01%	0.13%
TLD	0.01%	0.25%	0.01%	0.59%

Table 4: Different Sources of DNS Traffic

5. CLASSES OF DNS TRAFFIC

This paper looks at the considerable number of DNS query failures in order to understand the source of these failures. The source of failures have been characterized into different groups as shown in Figure 7. In this figure each class of traffic, represented by a color, sums up to 100%. To keep the categories mutually exclusive packets were assigned to the first category they matched according to the order Bugs, SD, WPAD, INTERN., LAN, ARPA, SPAM, AV, TLD. These are descriptions of each category listed in descending order by the percentage of DNS error responses:

- **LAN** includes DNS requests for domains reserved or commonly used for local networks such as .lan, .local, .localdomain, etc. See subsection 5.1.
- **SD**, Service Discovery, is comprised of any domain lookup with one domain level starting with an underscore, _ [8]. These domains are used in the DNS service discovery application of the DNS protocol. See subsection 5.2.
- **Intern.** denotes DNS traffic for which the start of authority (SOA) is internal to the network or belongs to the same organization as the network. A more general description of this category is the internal domains on a business, organization, or campus network. See subsection 5.1.
- **ARPA**, Address and Routing Parameter Area [13], includes reverse IP lookups providing a DNS name record. A DNS lookup to the name returned by this record should but does not necessarily match the IP address with the reverse lookup. It also includes other ARPA services such as lookups for E.164 numbers using *e164.arpa*. See subsection 5.3.
- **Spam and AV** includes DNS traffic for blacklists and whitelists used by Spam filters and anti-virus software. Spam blacklists include those for IP addresses from which Spam originates, unregistered IP addresses (bogons), and URIs (uniform resource



(a) Total Requests

(b) DNS Failures

Figure 7: DNS Traffic Volume by Domain Category

identifiers) commonly included in Spam. Often DNS traffic is allowed through firewalls allowing anti-virus apps with a way to check file hashes online. See subsection 5.4.

- **TLD**, top level domains, are those domains which only have one domain level such as localhost. Randomized TLD requests are used by some browsers to detect DNS hijacking by ISPs (Internet service providers). See subsection 5.5.

- **WPAD**, Web Proxy Auto-Discovery protocol [24], requests are made to either the "wpad" top level domain or the domain a machine belongs to with "wpad" as the last level domain. See subsection 5.2. (e.g. wpad.mydomain.tld).

- **Bugs** represents DNS requests that are either invalid or have clear indicators that they resulted from an application bug or misconfiguration. For example many of these lookups contained an entire URI or parts of a URI such as protocol (e.g. http://), port (:80), path (/index.html). There were DNS requests for domain names of the same form as valid IPv4 and IPv6 addresses. Some resembled file paths and or terminated with common file extensions. Others contained null, or undefined which suggests a software error such as concatenating a null or undefined JavaScript variable to a URI which then is interpreted as part of the domain. Others had invalid characters, spaces, or URL (uniform resource locator) encodings (e.g. %20). See subsection 5.5.

- **Other** lookups which do not fit into the preceding categories are discussed in subsection 5.7. It is noted that DNS query failures seem to be the result of some valid services such as (spam checks, anti virus software), software and protocol bugs, along with malicious and suspicious traffic. With increased use of DNS as a mechanism to get through firewalls for providing services, the DNS failure traffic needs to be filtered appropriately to identify or isolate suspicious and malicious traffic. Filtered traffic should be selected so that it should not be easy for the authors of malicious software to hide their C&C traffic by it being filtered in this step.

In revisiting the DNS usage landscape these datasets contain some familiar categories of traffic. This paper categorizes traffic as being from blacklists or whitelists, reverse lookups, lookups for top level domains, lookups for WPAD domains, lookups for domains which our DNS servers is the SOA for, local domains, and all other DNS lookups. These categories were selected by seeking to explain the bulk of DNS lookup failures and investigate any distinguishing characteristics to provide insight into distinctions between DNS traffic from mobile devices vs servers, desktops, laptops and other devices wired into the network.

There were also groups of random domains which shared some domain levels and had both valid and nonexistent domain responses. In [33] domains of this type were linked to botnet command and control (C&C) traffic. The new traces we collected had domains that matched this profile but had valid uses. Blacklist and whitelist services are hosted using the DNS infrastructure and are discussed in a later section.

5.1 Internal and Local Domains

Internal and local domains are domains whose start of authority is local or that belong to the same organization as the network. This group of traffic was singled out in [26] to remove DNS traffic which was not likely to be a channel for someone trying to get information through the firewall. Although traffic between bots may be local, C&C channels require information from outside the network, so we highlight this as its own category of DNS traffic.

For all four datasets the number of error responses for domains associated with local networks such as *.lan*, *.local*, and *.localdomain* was low for wired hosts 0.40%, 0.71%, 0.76% and 1.79%. For Wi-Fi hosts there was much more variation: 0.80%, 6.09%, 47.37%, and 45.65%. Notice that for Wi-Fi hosts in datasets 3 and 4 these responses are a much larger percentage. The actual volume of responses was about 2.04 million for Dataset 2, but for Dataset 3 was 20.6 million. These datasets were taken over about the same amount of time and both when classes were in session. What is different in this category from the first two datasets from the second two is that there are a large number of requests from Wi-Fi IP addresses for the domain “local.” There are responses with out the error flag set in the later datasets but no IP address is provided in the answers section for local. There was a drastic difference in the number of queries for the domain “local” from Wi-Fi hosts. Mac OS X issues SOA queries for “local” as part of Bonjour [4]. Ubuntu issues these queries as well, but Windows does not by default.

Of the internal DNS failures from wired hosts 12.95% to 33.49% of the internal domain failures were repeated queries which appended the host’s domain to a failed TLD query. For Wi-Fi hosts this number was from 0.53% to 1.94%. For example if the Google Chrome browser queries *lacxspgdze* and the host belongs to *domain.tld* once the first query fails another query will be issued for *lacxspgdze.domain.tld*. These failures can be explained by hosts appending the domain they are configured to belong to failed TLD queries and retrying. In some cases there were more than one parent domain associated with a single IP address. This can be explained by different virtual or physical machines configured with different domains using NAT and the same IP address.

5.2 Auto-configuration Domains and DNS Service Discovery

Auto-configuration domains for which DNS records are used either to point to an auto configuration server or contain a record containing the configuration information. DNS service discovery (DNS-SD) is a technique for enumerating the services available on a network such as printers, media servers, other hosts, etc. It expands the capabilities of zero configuration networking beyond IP address allocation and host resolution to include advertising and querying services on a network. Well known implementations of DNS-SD include *Bonjour*, and *Avahi*.

Examples of auto configuration domains we saw include requests for Active Directory, and WPAD. Active Directory uses domains with *ad* as the last level domain, (i.e. *ad.domain.tld*). It makes use of Kerberos and LDAP. Non-existent domain responses for Active Directory domains point out a misconfiguration or service outage. We observed that only 0.14% of ad.* domain requests failed. Identifying these can help remove potential abuse of any misconfiguration by someone trying to compromise an Active Directory client. The percentages of auto-configuration traffic, out of all the DNS error responses across each of the four datasets, were comparable between the set of wired hosts and the set of Wi-Fi hosts.

Traffic for the *Web Proxy Auto-Discovery* [24] protocol (WPAD) made up between 1.07% and 3.96% of the total DNS errors for each dataset. All the DNS responses in the datasets for WPAD domains were failures. WPAD allows network administrators to distribute a Proxy Auto Config (PAC) file, usually

```
http://wpad/wpad.dat
```

which defines a JavaScript function `FindProxyForURL(url, host)`. This protocol also checks each level of the domain tree which the host machine belongs to in order to automatically determine the proxy settings. Attackers can use WPAD to enable man-in-the-middle attacks [25].

DNS-SD traffic ranged from 4.09% to 17.61% of the total DNS error responses, but only 0.11% to 0.36% of the successful responses. There were also DNS-SD queries which included non-printable characters. Unicode names being used or exploits can cause this behavior. DNS-SD is used to enumerate services. When DNS-SD requests are issued through normal (as opposed to multicast) DNS, it is expected that these requests would fail unless the local DNS server is configured to list these services.

5.3 ARPA

Although the *.arpa* top level domain originally was related to the ARPANET created by DARPA it now has been repurposed to be Address and Routing Parameter Area Domain [13]. This top level domain now provides reverse address resolution name records for IPv4 and IPv6 addresses. An IP version 4 lookup for *127.0.0.1* would have the form, *1.0.0.127.in-addr.arpa*. The result is a *name* record with the value *localhost*. Notice that the order of the octets is reversed so that authority can be delegated by domain level.

The wired hosts had between 12.67% and 28.98% .arpa DNS error responses across all the datasets, and the Wi-Fi hosts had between 3.90% and 6.15%. We included lookups to the deprecated *ip6.int* domain [14] in counting the .arpa traffic. Hosts associated with Spam filtering lookups also had a higher number of reverse resolutions. This can be explained by the forward-confirmed reverse DNS method for verifying associations between IP addresses and domains. Other sources of .arpa traffic include logging services such as rsyslog and packet capture software such as tcpdump and wireshark.

5.4 Blacklists and Whitelists

Blacklists and whitelists over DNS leverage the infrastructure of DNS to provide cached lookups to dynamically changing information. Some Spam detection solutions rely on DNS lookups to determine:

- If the sender of an email is blacklisted.
- If the IP address of the server which the email was originally submitted to reverse resolves to the domain of the email address in the “from” field
- If there are blacklisted URLs in the contents of the message

IP addresses may be blacklisted for a number of reasons such as being a known source of Spam. Spam traps are one way IP addresses of Spam senders are identified. There are blacklists for open proxy servers (HTTP, SOCKS, SMTP) which may be used by Spammers to hide the IP address Spam originates from. Some blacklists contain IP addresses for machines hosting exploits. These can be detected by specialized web crawlers. There are also blacklists for IP addresses exhibiting behavior associated with being part of a botnet. These IP addresses can be identified by listening for attacks with a honeypot. Lastly, there are blacklists for IP address ranges which are not registered, bogons, since there should not be emails originating from these addresses.

This class of traffic has a large ratio of error responses to responses with no error because of lookups which are not listed. Spam related black and white lists made up at least 5.34% to 6.30% of the DNS error responses, and anti-virus related lists made up at least 1.71% to 4.30%. We observed that email servers generate a significant amount of the Spam related black or white list traffic. This explains why in 3 out of the 4 data sets, the wired hosts have more than 2.8 times the percentage of Spam list DNS error responses than Wi-Fi hosts.

Some anti-virus tools use hash based DNS lists. Examples include McAfee and Sophos. It should be noted that although this may allow for a reduction in the size of local databases used by anti-virus tools, the tools should use an authentication technique resilient to man in the middle attacks such as requiring DNSSEC. An example of what this would look like is:

> e9e3f794b69c8707f724f47a48dbb3a7b492a55f6e0e
> 42968c19a3f00f96.57ce.list.domain.tld

where the first two levels of the domain is the SHA 256 or other hash, and the rest of the levels denote the parent domain of the database. Any hash longer than 63 characters will need to be split across domain levels because of the limit in the DNS standard [22]. The maximum length domain name is 253 characters which has a size of 255 data bytes when represented in a DNS packet. Similar to the Spam lists, the number of DNS error responses from anti-virus lists was higher for wired hosts than Wi-Fi hosts, but the ratios varied from 1.3 to 5.8 times.

We took the positive results from URI blacklists and cross checked to see if hosts from other IP addresses were requesting blacklisted domains. We did find hosts accessing domains listed on the URI blacklists. In many cases this is because the domains are for widely used services which Spam emails link to, but which hosts legitimate content or provides a legitimate service (examples: Tiny URL, YouTube, etc). The percentages of blacklisted successful DNS responses for each dataset is 3.79%, 0.08%, 0.10%, and 0.20%.

Those hosts which queried Spam related blacklists had a much higher percentage of the non classified domains. Many nonexistent domains requested by these hosts had a random component along with a valid parent domain. We believe these requests were generated as a result of Spam filters performing forward-confirmed reverse DNS checking. Forward-confirmed reverse DNS is when there is agreement between the reverse resolution and the forward resolution of a domain name. Spammers likely use fake subdomains of valid domains to try to evade Spam filters by taking advantage of Spam exception filters. Emails which claim to originate from an invalid domain or that do not pass the forward-confirmed reverse DNS check are more suspicious than ones which originate from a valid domain. It is noteworthy that if spammers randomize the origin of and URIs contained within their emails, a Spam email may generate a significant number of DNS queries. We observed cases where one URL was checked against three separate blacklists.

Using DNS blacklists for a safe browsing by warning a user when they visit an IP known to host exploits can amplify the amount of DNS traffic made for visiting a single page.

5.5 Application Behavior and Bugs

We observed distinguishing features in some of the DNS Error responses. One category of nonexistent domain requests we observed had the prefix, “https://” or other protocol prefixes such as “http://”. These prefixes shouldn’t occur in DNS requests and indicate the presence of buggy applications. However, we did find examples of valid DNS responses for domains of this format. These domains were likely registered as a work around. Some application bugs included a port specification in the domain name lookup “something.com:8080”, included the relative path, or the entire URI. All of these requests resulted in errors. Javascript bugs which appended an undefined variable to a domain name “something.comundefined” or “something.comnull”. There were also cases where lookups were performed for strings that were likely filenames because they ended with common file extensions (e.g. index.html, image.jpg, script.js). A simple mistake on a web page such as:

```
<iframe src="http://index.html"></iframe>
<img src="http://image.jpg" />
<script src="http://script.js"></script>
```

Loading these HTML tags from a web page results in a DNS query for *index.html*, *image.jpg*, and *script.js*, respectively. Instead of a DNS query the page writer likely intended to make HTTP requests which would either include the domain name along with the path or leave the protocol out of the URI entirely. Some domain lookups had spaces, commas or asterisks. In these cases the lookups appeared to be related to improperly configured servers.

Some domain lookups contained URL encoded strings such as %20. These encodings allow special characters to be included in POST and GET data used in HTTP lookups. These are used heavily in web applications where user submitted content or queries are submitted to the web-server. If a buggy web application neglects to delimit the URL path from the domain, these characters can end up in a query. Also if a user’s browser searches from the address bar, their search query may be interpreted as a domain name and instead of a search parameter. For applications, these bugs triggered repeat failed queries. Bugs which show up in client-side JavaScript may only show up for certain versions of certain browsers or under certain circumstances limiting the rate at which they are repeated. Bugs in mobile applications may trigger lookups periodically.

Dataset	1	2	3	4
Number of blocked domains	1	772	45	427
Number of blocked TLDs	0	24	7	39
TLDs with any DNS record	420	354	132	335

Table 5: ICANN Name Collision Block List Stats

In investigating top level domains which were coming back as having IP addresses we found some with records to *127.0.53.53*. Since, August of 2014 this address has been purposed for ICANN name collision management. Table 5 shows the number of unique domains returning *127.0.53.53* versus the total number of requests for a domain name with only a top level which returned an IP address. Dataset 1 was taken in June and predates the roll out of this feature explaining the lower numbers. Dataset 2 had a large amount of requests to a particular second level domain returning *127.0.53.53*. These requests were present in datasets 1, and 4 but were all nonexistent domains.

One observation we made was queries requesting random top level domains. The source of these queries was found to be the Google Chrome web browser [35]. Some Internet service providers (ISPs) hijack DNS responses and Comcast has submitted an Internet draft on DNS redirection [30]. Although this technique might be used to offer services, ICANN has forbidden NXDOMAIN substitution by gTLD registrars for several reasons including but not limited to privacy and user experience concerns. Properly determining whether domains exist is a dependency of some of the browser’s features. The Google Chrome web browser issues domain requests for several strings of random letters to detect DNS hijacking. These random strings are effectively random TLDs which do not exist. When an ISP hijacks nonexistent domain responses, an address record to an ISP search engine is returned instead of the error response. The Chrome web browser checks to see if the random queries it issued result in an address record instead of nonexistent domain responses. If these requests are directed to the same IP address, it is clear that IP address should be interpreted as an non-existent domain response. Without this detection the address bar search feature would not work properly. These domains are distinguishable from botnet C&C traffic because botnets have not used TLDs as their C&C domains. At first glance this looks like a category of traffic attackers could hide their requests in; however, it isn’t trivial to create a new top level domain as the registration process occurs over a several month period. One disadvantage of using random TLDs for detection is the possibility that DNS hijackers like ISPs may leave TLDs untouched and hijack second level domains and greater. This traffic on some systems causes requests for subdomains of the domain a hosts belongs to (see subsection 5.1).

We found another class of non-existent domain queries generated by the Google Chrome web-browser. Specifically this browser has a DNS prefetching feature which is enabled by default. This class of traffic was observed in [17] and is discussed in [11]. This is different than HTML5 prefetching which requests domains for links on a page that have:

```
<link rel="dns-prefetch"
href="//www.domain.tld/" />
```

When a user is typing a domain in the address bar and the query appears to be valid, this browser will issue a preemptive DNS request. Partial queries that appear to be valid end with a valid TLD. This feature reduces the perceived delay of loading a page to the user at a cost of extra DNS traffic. Some argue this raises privacy issues because it generates traffic even before the user has actually submitted a search query or page lookup.

There were DNS lookups for valid IPv4 and IPv6 addresses. These should have been used directly as the destination address of a service. Requests of this form may originate when an application issues a DNS query for a user specified field without checking if it is an IP address. The `gethostname()` library function does not have this issue.

We observed some domains which appeared to be intentionally used to get nonexistent domain responses. These had the last level domain *none*.

5.6 ENUM For VOIP

One source of a large volume of nonexistent domains was the *nrenum.net* second level domain. Nrenum provides an ENUM database over DNS for participating universities. ENUM provides a way for E.164 numbers to be converted to a URI [6]. Protocols such as SIP and H.323 then provide a way of initiating the voice communication. 99.98% of the nonexistent ENUM domain responses were for a single host. In Dataset 4 for every successful lookup this host had 9.67 failures, but the next most common requester had more than 350 successful lookups for every failure. This appears to be a brute force search of the entire ENUM database. The results of this kind of search could be used to associate IP addresses with VOIP phone numbers, or determine what numbers are active for purposes like robocallers, etc. One technique to mitigate this kind of attack is impose rate limiting per IP by the number of unique ENUMs requested with a higher penalty for nonexistent numbers. Heavy offenders could be blacklisted, and legitimate services which require higher rates could be white-listed.

5.7 Suspicious and Unclassified Traffic

In Figure 4 there are several spikes in the rate of error responses. We found these to be caused by individual hosts which repeatedly queried particular non-existent domains repeatedly. The DNS cache expiration times were larger than the width of the spike, and these responses were answered from the local DNS resolver's cache rather than generating a large volume of DNS requests to the SOA for these domains. If these responses resulted in requests to the SOA, they would be denial of service traffic, but the ones we observed were the result of a software bug or misconfiguration. One bug on one machine caused a more than ten-fold increase in the number of nonexistent domain responses being issued by the DNS server for the entire campus. The average bandwidth of the requests and responses was less than 1 Mbps which is not close enough to the link capacity to cause congestion by itself.

We did observe fast fluxing DNS responses, and suspicious DNS traffic likely related to botnets. Some of the suspicious traffic appeared to be aggregated either by NAT or a DNS forwarder. Of the domains with DNS failures 89.11% to 99.53% of them had an edit distance greater than two to the nearest valid domain. The remaining percentage includes those typos where in one or two instances a letter is omitted, different, or inserted.

Examples of domains we identified which indicate the presence of malware include:

bvkaymxoioe**.www.game499.com.**
2014-12-30**.pw**
bvkcrzkdcsb**.www.gannme499.com**
loooplollokp80**.com**
musechizewrt40**.no-ip.info**

With each example the bold part of the domain name was constant while the rest of the domain had variation. The last two examples were accompanied by several requests to the same prefix with different numbers at the end. This is an attempt to lower the edit distance between requests and avoid detection. The last request was for a DDNS provider.

6. CONCLUSION

How the Internet is used and what services are offered have changed rapidly with the introduction of new technology. DNS is a critical service enabling the Internet to function. We collected DNS traffic from a campus over several days distinguishing wireless from wired traffic. We have investigated network sources, application sources, and sources of NXDOMAIN responses, to provide insight into how the usage of the DNS service is changing. Also, we compared DNS usage between wired and Wi-Fi hosts. The wireless hosts were responsible for close to two thirds of the DNS traffic. Most of the popular domains received more DNS requests from wireless devices. A significant amount of NXDOMAIN responses were measured (10-15%), many of which were requested by applications which use DNS as a side channel through network firewalls. This increase in failures makes it more difficult to identify malicious traffic. DNS traffic analysis has proved useful in identifying several anomalies and infected machines on the network.

7. ACKNOWLEDGMENTS

This publication was made possible by the NPRP award [NPRP 5-648-2-264] from the Qatar National Research Fund.

8. REFERENCES

[1] Alexa Internet, Inc. The top 500 sites on the web. http://www.alexa.com/topsites.

[2] M. Antonakakis, R. Perdisci, W. Lee, N. Vasiloglou, II, and D. Dagon. Detecting malware domains at the upper DNS hierarchy. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, pages 27–27, Berkeley, CA, USA, 2011. USENIX Association.

[3] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon. From throw-away traffic to bots: Detecting the rise of DGA-based malware. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security'12, pages 24–24, Berkeley, CA, USA, 2012. USENIX Association.

[4] Apple Inc. Mac OS X: About Multicast DNS. https://support.apple.com/kb/TA20999?locale=en_US.

[5] L. Bilge, S. Sen, D. Balzarotti, E. Kirda, and C. Kruegel. Exposure: A passive DNS analysis service to detect and report malicious domains. *ACM Trans. Inf. Syst. Secur.*, 16(4):14:1–14:28, Apr. 2014.

[6] S. Bradner, L. Conroy, and K. Fujiwara. The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM). RFC 6116 (Proposed Standard), Mar. 2011.

[7] N. Brownlee, K. Claffy, and E. Nemeth. DNS measurements at a root server. In *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, volume 3, pages 1672–1676 vol.3, 2001.

[8] S. Cheshire and M. Krochmal. DNS-Based Service Discovery. RFC 6763 (Proposed Standard), Feb. 2013.

[9] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in smartphone usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, MobiSys '10, pages 179–194, New York, NY, USA, 2010. ACM.

[10] H. Gao, V. Yegneswaran, Y. Chen, P. Porras, S. Ghosh, J. Jiang, and H. Duan. An empirical reexamination of global dns behavior. *Proc. of ACM SIGCOMM*, Aug. 2013.

[11] Google Inc. DNS Prefetching. http://www.chromium.org/developers/design-documents/dns-prefetching.

[12] S. Hao, M. Thomas, V. Paxson, N. Feamster, C. Kreibich, C. Grier, and S. Hollenbeck. Understanding the domain registration behavior of spammers. *Proc. of Internet Measurement Conf.*, Oct. 2013.

[13] G. Huston. Management Guidelines & Operational Requirements for the Address and Routing Parameter Area Domain ("arpa"). RFC 3172 (Best Current Practice), Sept. 2001.

[14] G. Huston. Deprecation of "ip6.int". RFC 4159 (Best Current Practice), Aug. 2005.

[15] J. Jung, E. Sit, H. Balakrishnan, and R. Morris. Dns performance and the effectiveness of caching. *IEEE/ACM Trans. Netw.*, 10(5):589–603, Oct. 2002.

[16] Y. Kazato, K. Fukuda, and T. Sugawara. Towards classification of DNS erroneous queries. In *Proceedings of the 9th Asian Internet Engineering Conference*, AINTEC '13, pages 25–32, New York, NY, USA, 2013. ACM.

[17] S. Krishnan and F. Monrose. DNS prefetching and its privacy implications: When good things go bad. In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More*, LEET'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

[18] D. Leonard and D. Loguinov. Turbo king: Framework for large-scale internet delay measurements. In *INFOCOM 2008. 27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 13-18 April 2008, Phoenix, AZ, USA*, pages 31–35, 2008.

[19] E. Marcotte. Responsive Web Design. http://alistapart.com/article/responsive-web-design.

[20] T. Matsunaka, A. Yamada, and A. Kubota. Passive OS fingerprinting by DNS traffic analysis. In *Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications*, AINA '13, pages 243–250, Washington, DC, USA, 2013. IEEE Computer Society.

[21] P. Mockapetris. Domain names - concepts and facilities. RFC 1034 (INTERNET STANDARD), Nov. 1987.

[22] P. Mockapetris. Domain names - implementation and specification. RFC 1035 (INTERNET STANDARD), Nov. 1987.

[23] J. S. Otto, M. A. Snchez, J. P. Rula, and F. E. Bustamante. Content delivery and the natural evolution of dns - remote dns trends, performance issues and alternative solutions. *Proc. of Internet Measurement Conf.*, Nov. 2012.

[24] M. D. P. Gauthier, J. Cohen and C. Perkins. Web Proxy Auto-Discovery Protocol. IETF Internet Draft – work in progress 00, IETF, November 2000.

[25] A. Pashalidis. A cautionary note on automatic proxy configuration. In *Proceedings of the IASTED International Conference on Communication, Network, and Information Security*, CNIS, 2003.

[26] V. Paxson, M. Christodorescu, M. Javed, J. Rao, R. Sailer, D. Schales, M. P. Stoecklin, K. Thomas, W. Venema, and N. Weaver. Practical comprehensive bounds on surreptitious communication over DNS. In *Proceedings of the 22nd USENIX Conference on Security*, SEC'13, pages 17–32, Berkeley, CA, USA, 2013. USENIX Association.

[27] J. Rula and F. Bustamante. Behind the curtain –cellular dns and content replica selection. *Proc. of Internet Measurement Conf.*, Nov. 2014.

[28] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman. On measuring the client-side DNS infrastructure. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 77–90, New York, NY, USA, 2013. ACM.

[29] A.-J. Su, D. R. Choffnes, A. Kuzmanovic, and F. E. Bustamante. Drafting behind akamai (travelocity-based detouring). *Proc. of ACM SIGCOMM*, 2006.

[30] J. L. T. Creighton, C. Griffiths and R. Weber. Recommended Configuration and Use of DNS Redirect by Service Providers. IETF Internet Draft – work in progress 00, IETF, July 2009.

[31] R. Villamarin-Salomon and J. Brustoloni. Identifying botnets using anomaly detection techniques applied to DNS traffic. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, pages 476–481, Jan 2008.

[32] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan. Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. *IEEE/ACM Trans. Netw.*, 20(5):1663–1677, Oct. 2012.

[33] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan. Detecting algorithmically generated malicious domain names. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, pages 48–61, New York, NY, USA, 2010. ACM.

[34] S. Yadav and A. L. N. Reddy. Winning with DNS failures: Strategies for faster botnet detection. In *7th International ICST Conference on Security and Privacy in Communication Networks (SecureComm)*, London, United Kingdom, Sept. 2011.

[35] B. Zdrnja. Google Chrome and (weird) DNS requests. `https://isc.sans.edu/diary/Google+Chrome+and+%28weird%29+DNS+requests/10312`.