A First Look at Web Browsing Predictions using DNS Logs

Utkarsh Goel, Clint Cooper, Brittany Terese Fasy, and Mike P. Wittie
Department of Computer Science, Montana State University, Bozeman, MT
{utkarsh.goel, clint.cooper, brittany, mwittie}@cs.montana.edu

Abstract

Despite several decades of research towards improving Web experience, users remain dissatisfied. Content Providers (CPs) make contracts with Content Delivery Networks (CDNs) to speed up webpage load times. CPs also make contracts with Online Advertisement providers (OAPs) to improve relevancy of Web content shown on the website. However, both CDNs and OAPs remain unaware of what Web content users will request next. As a result, CDNs cannot prefetch data from servers hosted by CPs, which prevents reductions in webpage load times. Also, OAPs display advertisements based on stale and incomplete Web browsing histories.

In this paper, we present `WebNext`, a technique to analyze passively collected Domain Name System (DNS) logs for predicting users' Web browsing behaviors. In our experience with `WebNext` on a large scale DNS dataset, we identified several sequences of websites that many users often tend to follow. Therefore, we suggest CDNs and OAPs to predict users' Web browsing interests for improved quality of Web experience.

keywords: Web, behavior, prediction, DNS, `WebNext`

1 Introduction

Content providers (CPs) such as Google, Facebook, and Amazon care about high quality of user experience on their websites. CPs desire that their users remain engaged with their websites because long user engagement times result in higher number of purchases on the website, as well as enable online advertisers to improve content relevancy on the website [12]. However, several factors influence user engagement, such as 1) the responsiveness of the website, that is, how fast the website loads after the user clicks a link; and 2) Whether the website provides content that the user is interested in at the time website is loaded. As a result, CPs employ several sophisticated techniques to speed up the webpage load times and offer the content users are interested [3, 8, 21]. However, despite several years of research to improve user experience with the Web, users remain dissatisfied with current Web performance [22]. We consider the lack of knowledge about users' next Web browsing behaviors as the major challenge to improve Web experience.

Content Delivery Networks (CDNs) such as Akamai and Level 3 are effective in speeding up webpage load times as they distribute application content to geographically diverse Web servers [20]. A data distribution technique allows users to download website data from a server in close proximity with low network latency – resulting in faster page loads. However, current content delivery techniques only allow for content to be distributed to a server proximal to the user's location **only after** the user requests the webpage. Therefore, every time the requested content is not available on a nearby Web server, the websites load slower as the content is fetched from a more distant server.

Online Advertisement Providers (OAPs) such as Google Ads and Amazon Advertising are effective in approximating content relevance on websites based on users' previous Web browsing histories. However, OAPs face two challenges to ensuring that Web browsing histories are up-to-date and complete. First, users care about their privacy and tend to install browser-based plugins that prevent OAPs from tracking their Web activity. Second, when the user exits the website on which OAP's JavaScript is running, OAPs do not keep track of what next websites the users visit. Therefore, OAPs remain with an undesirable choice of displaying advertisements based on **stale and incomplete** Web browsing histories of users, resulting is degraded click-through rates of advertisements [2].

One of the major challenges to improve user engagement with websites is the lack of knowledge about users' next Web browsing interests. Specifically, if CDNs could predict users browsing interests, website data could be prefetched from CPs before it is needed by the user, resulting in even faster webpage load times [1, 24]. Further, if OAPs could predict users' Web browsing interests, they could make well-informed decisions as to which content more suited for the user. In this paper, we investigate Web browsing behaviors of a large group of users and present techniques that allow for approximate predictions of Web browsing behaviors.

We classify the five major contributions of our work as follows:

Novelty: In this paper, we offer a *novel understanding* of Web browsing behaviors using from a large university campus network, with the goal of improving Web experience for users. Our investigation includes detailed analysis of network traffic generated mostly by a large student population. Specifically, our work *predicts* Web browsing behaviors of users using only passively collected DNS logs from within the university network, as opposed to using user privacy-sensitive HTTP logs.

Dataset richness: We perform a large scale comprehensive analysis of Web browsing behaviors on a dataset comprising over 228 million DNS records collected during a 24-hour period in August 2014. Our sanitized dataset (as described in Section 3) comprises of Web browsing sessions from about 12,000 unique users resolving a total of about 100,000 unique domain names.

Implementation: We developed `WebNext`, a technique to perform a thorough search of relationships among websites using Domain Name System (DNS) logs.[1] `WebNext` identifies sequences of websites that are requested more often by users. Our results suggests that `WebNext` is effective in predicting Web browsing behaviors even after users leaves websites, or install browser plugins to prevent websites from tracking browsing history.

Results: Based on our experiences with `WebNext`, we make the following three observations:

- Web browsing behaviors tend to differ at different times of a day. For example, users visit social networking, video streaming, and e-commerce websites more frequently during the day and evening hours than during the morning hours.
- Based on our analysis of DNS logs, we identify several popular sequences of websites that users tend to follow. Our results show that many users frequently visit *Google.com* and *Facebook.com* in both forward and reverse orders.
- Finally, we observe that when users are at a Web search engine website, such as *Google.com*, they tend to request a diverse set of websites, thus making the prediction of user's next interest more challenging. However, when users are at an online social network website, such as *Facebook.com*, they tend to visit other similar websites such as *Twitter.com* and *LinkedIn.com*.

Inferences drawn: Based on our results, we make recommendations for CDNs and OAPs that enable them to more effectively speed up webpages, as well as improve content relevance on webpages by using DNS logs. Specifically, we suggest CDNs to investigate Web browsing behaviors of their users to prefetch any content that is not available in the cache at the time it is requested by users. Finally, we suggest OAPs to predict user's interests to deliver more relevant advertisements on the websites for improved click-through rates.

[1] We make `WebNext`'s source code available at `https://github.com/msu-netlab/WebNext`.

The rest of the paper is organized as follows. In the next section, we discuss related work that investigates Web browsing behaviors. In Section 3, we discuss our approach to collect large scale Web browsing data from a university campus network. In Section 4, we offer a discussion on our technique to analyse the Web access patterns and predict user's interests on the Web, followed by results in Section 5. Finally, we conclude in Section 6.

2 Related Work

A number of previous studies investigate popularity, type, and amount of time spent on websites visited by users [7, 13, 16, 19]. Other studies estimate gender, age, and identity of users by analysing Web browsing behaviors [5, 9, 15, 17]. Several studies have developed algorithms to predict Web browsing behaviors but have not evaluated their effectiveness on real world data [6, 10, 14, 23, 25, 26]. Finally, recent initiative by Akamai Technologies allows predictive content delivery solutions for streaming videos on mobile devices [4].

Our work, in contrast to these studies, identifies actual sequences of websites often visited by users using a real world dataset. Our work allows prediction of Web requests across different websites, instead of requesting content on the same webpage. Finally, instead of using HTTP logs containing users' privacy-sensitive information, we use DNS logs that do not contain any personally identifiable information about the users.

3 Data Collection Methodology

Our goal is to discover sequences in which users access different websites. To accomplish this goal, we require a metric to uniquely identify users, a chronological list of websites that each user visits, as well as the timestamps of when the websites are visited.

In Figure 1, consider the sequence of Internet communication that takes place when a user enters a website URL in the Web browser. In Step 1, the user's Web browser sends a DNS request to a DNS server to resolve the website name into an IP address. The DNS server replies to the user with an IP address (Step 2). Next, the user's Web browser creates a connection with the server hosting the IP address returned in Step 2 and sends an Hypertext Transfer Protocol (HTTP) request to download the website content (Step 3). Finally, in Step 4, the Web server sends the website content to user's browser.

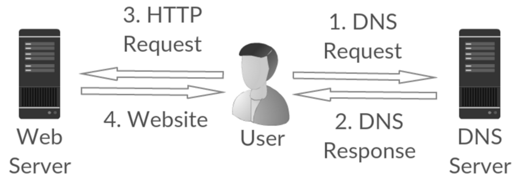

Figure 1: Sequence of DNS and HTTP requests being made when visiting a website.

For our dataset, we choose to collect DNS logs using `TCPDump` on Montana State University's (MSU) centralized DNS server for a period of 24 hours [18]. Our decision to use DNS logs, as opposed to using HTTP logs collected by Web servers [11], is based on the fact that DNS offers a broader view of Web browsing behaviors than the HTTP requests alone. Specifically, several websites use encrypted connections with users' Web browsers. Therefore, any encrypted HTTP request that the user makes cannot be recorded by monitoring just the HTTP requests. This is because the content of such HTTP requests is not in plain text and instead is encrypted using public keys. The DNS logs are recorded in plain text and contains all website names that the user requests regardless of whether the actual content on the website is downloaded over an encrypted or unencrypted connection. DNS logs are easily accessible because every user connected to the network would likely use network's default DNS server for resolving domain names.

Our collected DNS logs consist of 1) Users IP addresses, which we use to uniquely identify each user; 2) DNS requests containing domain names of websites visited by the users; and 3) Epoch timestamps of when DNS requests were sent. Our current captured DNS logs consists of `pcap` files totally about 16 GB in size. The `pcap` files contains over 228 million DNS records generated during our data collection period.[2]

In order to process DNS logs, we first convert the `pcap` files to relational database storage. We next sanitize our data to eliminate DNS responses from the recorded data, which results in the data to only consists of DNS requests. We only need to consider DNS requests because every DNS requests contains the domain name of the website that the user requests. Next, we delete DNS requests which were not destined for our university's centralized DNS server. Our goal here is to investigate Web browsing behavior for users that use the university's DNS server for requesting all the websites. Finally, we delete DNS requests for IPv6 addresses. We argue that we only require DNS requests that resolve to an IPv4 address, because Web browsing behaviors are irrespective of which IP protocol version users' networks supports. Our sanitized data consists

[2]A `pcap` (packet capture) file is a binary file that contains information about network requests and traffic.

Algorithm 1: `WebNext`

Data: (T, S) where T is a time and S is tuple size.
Result: $H = (K, V)$ where H is a HashMap with keys K as a sequence (seq) of domains and values V as the number of times that sequence occurs

$IPList = DataAtTime(T)$
for $ip \in IPList$**:**
  $Domains = DomainsWithIP(ip)$
  for $i \in \texttt{range}(0, len(Domains) - S)$**:**
    $seq = (domain_i, domain_{i+1}, ..., domain_S)$
    if $seq \in H$**:**
      $H[seq] = H[seq] + 1$
    else:
      $H[seq] = 1$

of over 12 million DNS requests for about 12,000 unique IP addresses accessing over 100,000 domains in a 24-hour period.

4 Implementation Details

We developed `WebNext` to analyze the DNS data and identify popular sequences of websites that users often visit. `WebNext` accepts user defined arguments, such as the time of day T and a desired sequence size S, as inputs to calculate popularities of different sequences of websites visited by users.

As shown in Algorithm 1, `WebNext` allows generation of HashMaps, or key-value pairs of a sequence of different domain names with the number of times the sequence is requested. $DataAtTime()$ is an SQL statement that returns a list of IP addresses for a given time window. $DomainsWithIP()$ is an SQL statement that returns a chronologically ordered list of domains associated with the specified IP address. `WebNext` then generates a list of sequences and sequence request counts. As shown in Algorithm 1, we use $DataAtTime$ and $DomainsWithIP$ to obtain IP addresses of user devices and domain names requested by each IP address, respectively. For every IP address extracted with $DataAtTime$ query, we fetch domain names in chronological order and generate sequences of website visits that are of length S. These sequences represent keys in the HashMap H. The count value associated to a sequence is initialized to 1 and incremented for each occurrence of the same sequence. The resulting HashMap is returned to the user after all IP addresses within $IPList$ are iterated and added as sequences to the HashMap.

5 Putting `WebNext` in Context

We now employ `WebNext` to investigate and understand Web browsing behaviors using our collected DNS data. Our first goal is to understand whether

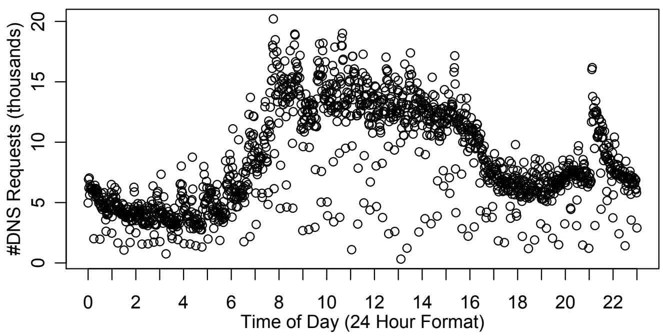

Figure 2: Distribution of the total number of DNS requests over a period of 24 hours.

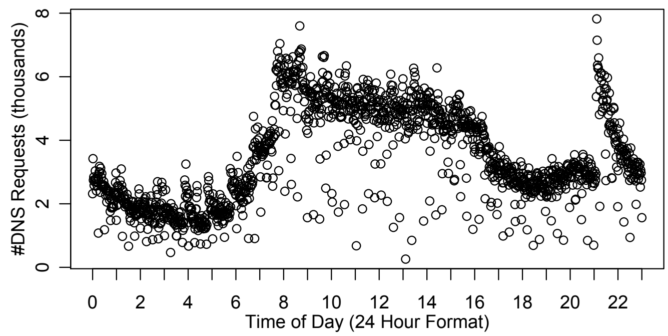

Figure 3: Distribution of the number of unique DNS requests over a period of 24 hours.

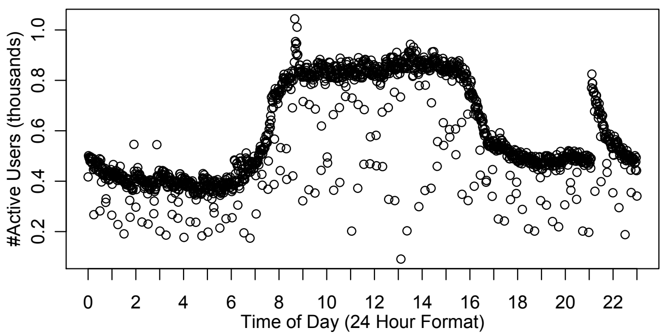

Figure 4: Distribution of the unique number of active users over a period of 24 hours.

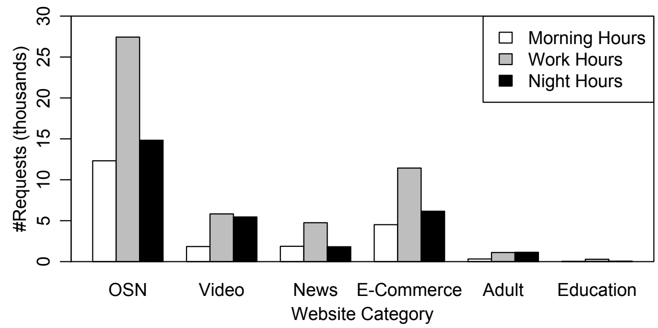

Figure 5: Popularity of websites classified under various categories of online services.

or not our data constitutes a statistically significant number of DNS records at any point in time, allowing us to reliably predict Web browsing behaviors.

5.1 Data Significance

In Figure 2, we show the total number of sanitized DNS requests that the university's DNS servers received over a period of 24 hours. The x-axis represents time duration in a 24-hour format. The y-axis represents the number of DNS requests received by the DNS server. From the figure, we see that at any given point in time during the day, our dataset consists of about 4,000 DNS requests, especially during 8 AM to 5 PM where we see about 15,000 requests every minute. Moreover, when analysing DNS requests for unique domain names in Figure 3, we observe that our dataset consists of about 2,000 DNS requests at any given point in time.

Finally, we investigate whether our dataset consists of DNS requests from substantial number of unique users at any given time of the day. In Figure 4, we show the distribution of active users at different times of a day. Similarly to previous figures, the x-axis represents time of day in a 24-hour format. The y-axis represents the total number of active users or unique IP addresses in our dataset. From the figure, we observe that at any given time of a day our dataset consists of DNS requests generated by 400 unique users on average, especially during 8 AM to 5 PM where we see about 850 active users. Similarly to Figure 2, we observe that during the peak hours (8AM to 5PM) the number of active users are also at a maximum. Therefore, based on the data we collected, we argue that such a large dataset is substantial to allow us to make reliable observations about Web browsing behaviors of different users.

5.2 Data Classification

Next, we are interested in understanding the popularity of different types of online services across different times of a day. Our goal here is to understand whether or not some online services are visited more often than others. In Figure 5, we show that different online services tend to exhibit different popularities, especially at different times of a day. The x-axis on the figure represents six different popular online service groups. The y-axis represents the popularity of each service, where we define popularity as the number of DNS requests served by the DNS server. The white, gray, and black bar graphs represent the popularity in morning (5 AM to 8 AM), work (8 AM to 5 PM), and night(5 PM to 5 AM) hours respectively.

From the figure, we see that the number of websites visited by people in the morning and evening are about one-third of the number of websites visited during the work hours. For example, Online Social Networks (OSNs) such as Facebook, LinkedIn, Google+, and Twitter, are visited almost twice as

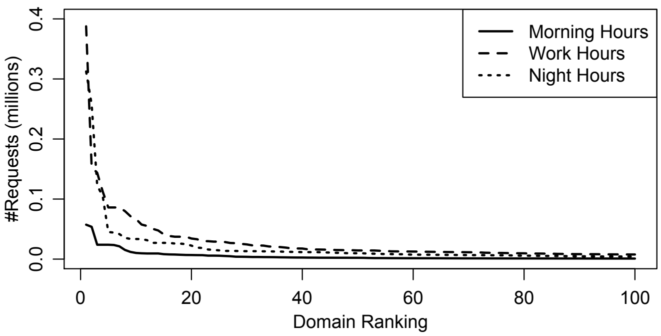

Figure 6: Distribution of visits to top 100 websites.

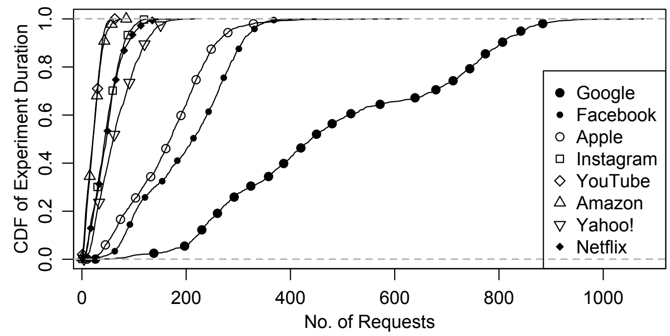

Figure 7: Distribution of visits to top eight websites.

Category	Websites
OSN	G+, FaceBook, Twitter, LinkedIn
Video	YouTube, Netflix, Hulu, Vimeo, DailyMotion, HBO
News	CNN, Fox, HuffingtonPost, Reddit, NYTimes, Yahoo News
E-Commerce	Walmart, Target, Ebay, BestBuy, Amazon, Costco, Craigslist
Adult	Top adult websites
Education	D2L, ECats

Table 1: List of websites classified in different categories based on the online service offered.

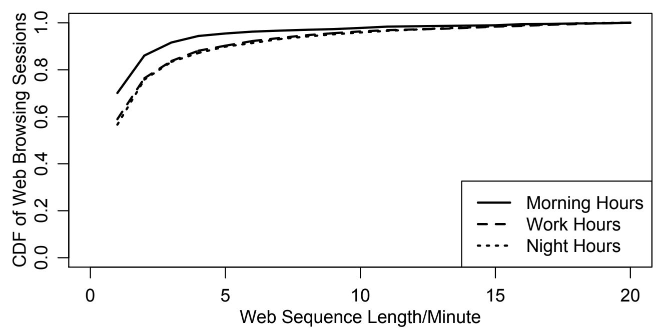

Figure 8: Distribution of the number of websites visited every minute.

much during the work hours as they are during the morning and night hours. Similarly to OSN websites, online services that offer e-commerce and news tend to be visited more during work hours. However, video streaming and adult websites show similar popularity during both the work and night hours. Surprisingly for a university network, both *D2L* and *ECats* websites used by students and faculty throughout MSU colleges to access study materials, are not visited as much as other online services at any time in the day. In Table 1, we list the various websites which we selected to classify the online services described in Figure 5.

Next, to understand whether there exist websites that users visit more often than the other websites, we investigate the popularity of all the websites. In Figure 6, we show the difference in how many times different websites are requested by users. The x-axis shows the rank of the top 100 websites, based on how often they were requested. The y-axis shows the number of times the top 100 websites are requested, out of a total of about 13,000, 55,000, and 33,000 unique websites requested during morning, work, and evening hours respectively. The different lines represent the requests at different times of the day. From the figure, we observe that users tend to visit only about 8 websites frequently, regardless of time of the day.

Therefore, in Figure 7, we show the popularity of the top eight websites in terms of how often users visit these websites throughout the day. The x-axis in the figure represents number of times a given website is requested. The y-axis represents a CDF of experiment duration during the day (24 hours). From the figure, we observe that for about 60% of the day, *Google.com* is requested upto 500 times every minute. Whereas, users visit other websites such as *Facebook.com* or *Apple.com* only about 200 times a minute for 60% of the day. We argue that because some websites are visited less often than other websites, there may exist some scenarios when websites tend to be visited more than in other scenarios. Specifically, what remains unclear is whether some websites are requested more often after a specific website is opened, or the requests for websites do not follow a discernible pattern. Therefore, in the next section we investigate whether users tend to have a sequence in which websites are requested.

5.3 Identifying Website Sequences

Interactive websites today consist of content that lead users from one website to another website with related content. While it is possible that users sequentially visit a number websites when looking for specific information online, it would be practically challenging to prefetch the data pertaining to all the websites that the users visit. Therefore, we first investigate the number of websites users often visit. Our immediate goal here is to identify the most popular length of sequences of websites that users visit. In Figure 8, we show a distribution of length of sequences that users visit in order. The x-axis represents the number of websites users visits every

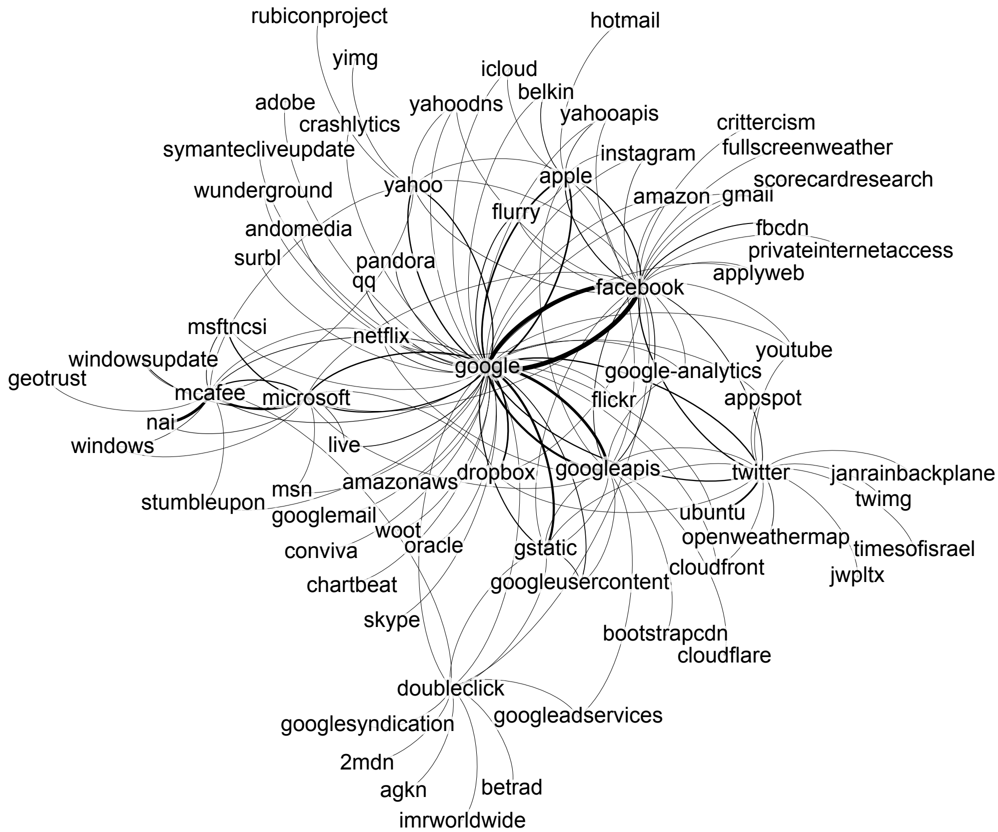

Figure 9: Web browsing behavior in the morning hours.

minute. The y-axis represents a CDF of Web browsing sessions. From the figure we observe that about 80% of the browsing sessions only consists of visits to 2 websites. Therefore, we argue that investigating popular sequences of only length two would enable us to understand which Web content should be prefetched.

Finally, we use `WebNext` to identify popular sequences of website visits that are of length two. In Figures 9, 10, and 11, we plot node graphs to represent Web browsing behaviors in the morning, working, and evening hours respectively. The labels represent names of websites visited by users.[3] The edges represent a causal relationship between the two connected website names. Specifically, the clockwise directionality of edges represents users' interests from visiting a `Source website` followed by visiting a `Destination website`. The thickness of edges represents the number of visits from one website to another.

In general, we observe that the node graphs representing Web browsing behaviors during work (Figure 10) and evening (Figure 11) hours consists of diverse set of websites than the node graph for the morning hours (Figure 9). We argue that this is because, as shown in Figure 2, the number of websites visited by users during work and evening hours are significantly larger than the number of websites visited during the morning hours.

Next, from the figures we observe that most of the Web browsing traffic is among a few websites, such as *Google.com*, *Facebook.com*, *Twitter*, *Yahoo.com*, *DoubleClick.com*. For example, based on the edge thickness in the node graphs, we observe that the user visits between *Facebook* and *Google* websites is the most popular browsing behavior across all three different times of the day.

We also observe that when a user visits a Web search engine website, such as *Google.com*, there exists a large number of websites that the user may choose to visit next. Such a behavior makes predicting of user's next Web request more challenging. However, when a user visits an online social network (OSN) website, such as *Facebook.com*, the next visit is likely another OSN website such as *Twitter.com* and *LinkedIn.com*.

5.4 Accuracy of `WebNext`

In Figures 9-11, we observe that website such as *Facebook*, *Twitter*, and *Yahoo* are connected with website names such as *Fbcdn*, *Twing*, and *Yimg* respectively. Websites such as *Fbcdn*, *Twimg*, and *Yimg* hosts images for *Facebook*, *Twitter*, and *Yahoo* respectively. For example, whenever a page pertaining to *Facebook*

[3]For readability, we only display website names without the domain name extensions, such as *.com* or *.net*.

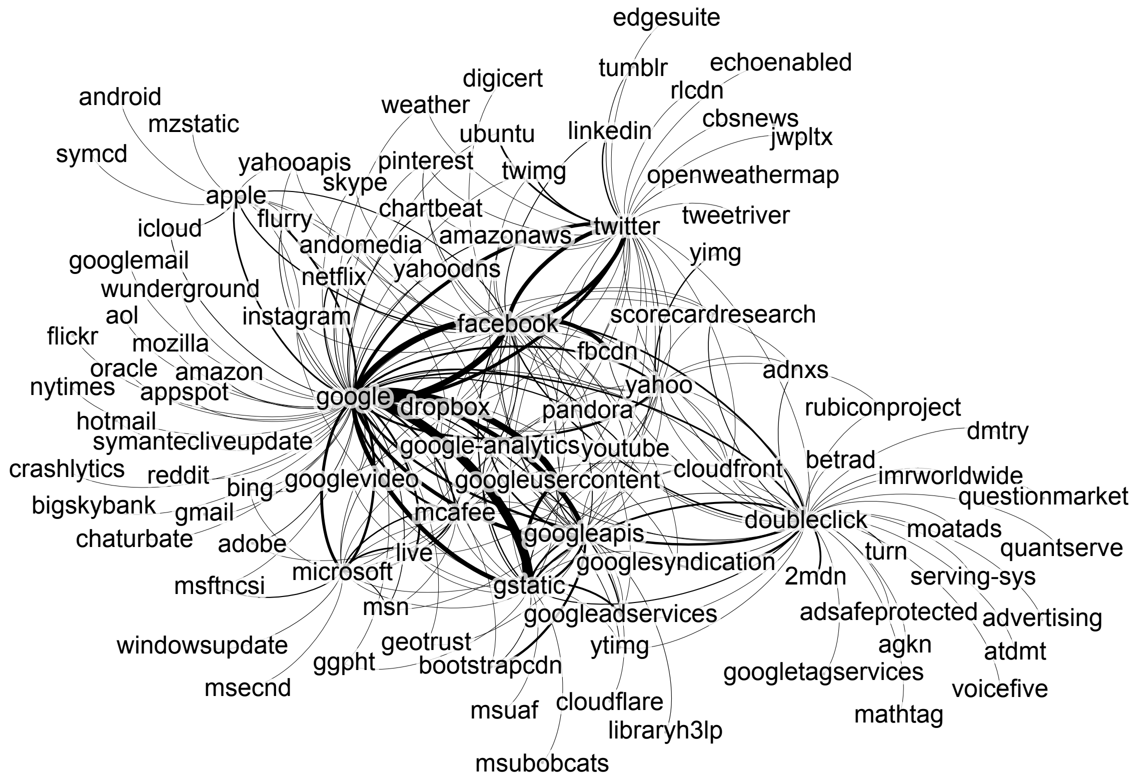

Figure 10: Web browsing behavior in the working hours.

is visited, the Web browser fetches images hosted on *Fbcdn* domain. Since our data analysis technique identified well-known connections between such websites, we argue that `WebNext` is accurate for predicting Web browsing behaviors on large scale DNS logs.

6 Conclusions

Prediction of users' Web browsing interests remains one of the major challenges to high quality of Web experience. We present `WebNext`, a technique to predict Web browsing behaviors by analyzing passively collected DNS logs, as opposed to user privacy-sensitive HTTP logs. Based on our extensive analysis of large scale DNS logs collected from a university campus network, we demonstrate that `WebNext` can accurately predict users' Web browsing behaviors, thus enabling content providers to improve Web experience for their users.

Acknowledgments

The authors would like to thank Mike Hitch at the IT Center of Montana State University for helping us in the data collection process. We also thank Kanika Shah and the anonymous reviewers for providing us constructive feedback. Finally, we thank National Science Foundation (NSF) for supporting this work through grants NSF CNS-1555591 and NSF CNS-1527097.

References

[1] Pre-Resolve DNS. `https://goo.gl/mXsW96`, Apr. 2015.

[2] Clickthrough Rate. `https://support.google.com/adwords/answer/2615875?hl=en`, Apr. 2016.

[3] Front End Optimization. `https://goo.gl/pKFHa2`, Apr. 2016.

[4] Predictive Content Delivery for Mobile Video. `https://goo.gl/aBASUW`, Apr. 2016.

[5] M. Abramson and D.W. Aha. User Authentication from Web Browsing Behavior. In *Florida Artificial Intelligence Research Society Conference*, May 2013.

[6] M.A. Awad and I. Khalil. Prediction of User's Web-Browsing Behavior: Application of Markov Model. *IEEE Transactions on Systems, Man, and Cybernetics*, 42(4), 2012.

[7] J.P. Bigham, A. Cavender, J.T. Brudvik, J.O. Wobbrock, and R.E. Ladner. WebinSitu: A Comparative Analysis of Blind and Sighted Browsing Behavior. In *ACM ASSETS*, Oct. 2007.

[8] J. Butler, W. Lee, B. McQuade, and K. Mixter. A Proposal for Shared Dictionary Compression over HTTP. `https://goo.gl/VC2v1X`, Sept. 2008.

[9] D. Canali, L. Bilge, and D. Balzarotti. On the Effectiveness of Risk Prediction Based on Users Browsing Behavior. In *ACM Symposium on Information, Computer and Communications Security*, Jun. 2014.

[10] V. Dongre and J. Raikwal. An Improved User Browsing Behavior Prediction using Regression Analysis on Web Logs. *Journal of Computer Applications*, Jan. 2015.

[11] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. `http://www.rfc-base.org/txt/rfc-2616.txt`, Jun. 1999.

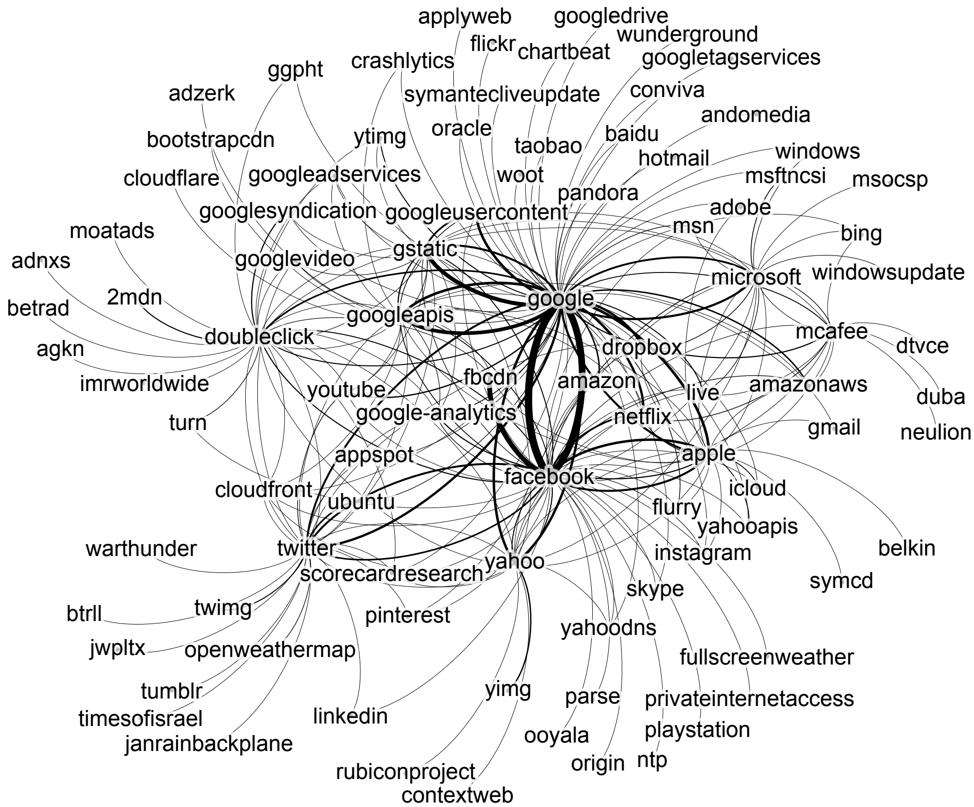

Figure 11: Web browsing behavior in the evening hours.

[12] B. Forrest. Bing and Google Agree: Slow Pages Lose Users. `http://goo.gl/xV1uYl`, Jun. 2009.

[13] S. Goel, J. Hofman, and M. I. Sirer. Who Does What on the Web: Studying Web Browsing Behavior at Scale. In *Weblogs and Social Media (ICWSM)*, Jun. 2012.

[14] S. Guha. Efficiently Mining Frequent Subpaths. In *Australasian Data Mining Conference - Volume 101*, AusDM, Dec. 2009.

[15] J. Hu, H. Zeng, H. Li, C. Niu, and Z. Chen. Demographic Prediction Based on User's Browsing Behavior. In *World Wide Web*, May 2007.

[16] J. Huang and R.W. White. Parallel Browsing Behavior on the Web. In *ACM Conference on Hypertext and Hypermedia*, Jun. 2010.

[17] M. Kakkar and D. Upadhyay. Web Browsing Behaviors Based Age Detection. In *Journal of Soft Computing and Engineering (IJSCE)*, Mar. 2013.

[18] P. Mockapetris. Domain Names - Implementation and Specification. `https://www.ietf.org/rfc/rfc1035.txt`, Nov. 1987.

[19] A.L. Montgomery and C. Faloutsos. Trends and patterns of WWW browsing behavior. `http://goo.gl/mPNRbv`, May 2000.

[20] E. Nygren, R.K. Sitaraman, and J. Sun. The Akamai Network: A Platform for High-performance Internet Applications. *SIGOPS Oper. Syst. Rev.*, 44(3), Aug. 2010.

[21] S. Radhakrishnan, Y. Cheng, J. Chu, A. Jain, and B. Raghavan. TCP Fast Open. In *ACM CoNEXT*, Dec. 2011.

[22] V. Rideout and V. S. Katz. Opportunity for all? Technology and learning in lower-income families. `http://goo.gl/3YfDPu`, Jan. 2016.

[23] S.L. Scott and I. Hann. A Nested Hidden Markov Model for Internet Browsing Behavior. `http://goo.gl/K6joQs`, Dec. 2006.

[24] H. Shang and C.E. Wills. Piggybacking Related Domain Names to Improve DNS Performance. *Computer Network*, 50(11), Aug. 2006.

[25] A.N. Shiryayev. Markov Chains with A Countable Number of Possible States. In *Selected Works of AN Kolmogorov*, 1992.

[26] T. Zhu, R. Greiner, G. Haubl, B. Price, and K. Jewell. Behavior-based Recommender Systems for Web Content. In *Workshop Beyond Personalization*, Jan. 2005.