Exploring Browsing Habits of Internauts

Exploring Browsing Habits of Internauts: a Measurement Perspective

Danilo Giordano, Stefano Traverso, Marco Mellia,

Politecnico di Torino - first.last@polito.it

ABSTRACT

In this work we investigate the browsing habits of Internauts. We leverage a large dataset collected from more than 25,000 residential users, and characterize their browsing behavior over a period two weeks. We consider the websites they access to, and observe how and when they access them. We aim at verifying if it would be possible to build profiles of users to be used for instance to identify the same user in different time windows.

Results show that the multitude of websites, the heterogeneity of habits, and the limited periodicity make the browsing habits of users unique, dramatically complicating the task of building any reliable profiling. Our results are preliminary, and we encourage the research community to run further experiments in this direction by sharing our dataset.

1. INTRODUCTION

The characterization of users' browsing habits has always attracted the interest of researchers. Since the seminal work of Catledge [1], many studies have been focusing on Web usage, for specific applications [2], or devices [3, 4, 5], or at usage of social networks [6]. Understanding how the content is consumed by users is fundamental to improve service design, to offer novel solutions, and, in general, to augment our understanding of the Internet.

In this work, we focus on the exploration of users' habits when browsing the Web. In particular, we take the point of view of the network, from where DNS requests of users are observable, and from which we can extract the *names* of the *services* they are accessing. To simplify the picture, the service is defined by the "second-level domain" found in DNS traffic, e.g., *google* and *nyt* from *www.google.com* and *www.nyt.com*. We leverage actual anonymized traffic traces collected from operational networks, where 25,000 people access the Web from home. We consider a time window, e.g., one hour, or one day, and we observe how many different websites an Internaut visits, if she keeps discovering new services, or, conversely, if she exhibits repetitive patterns that potentially allows one to build a profile to identify her in a crowd.

We first provide a characterization of the *distinct services* accessed by the entire population. In total, we count more than 400,000 services, with only very few of them being well-known and very popular. Interestingly, the discovery process does not saturate even after two weeks of observation. We next check how the number of *common services* grows over time, i.e., those services that are accessed in multiple periods of time. Intuitively, one would expect that the number of times people access a given service increases for increasing observation period. We instead find out that there is a large number of services that are accessed only once, and by single users. Surprisingly, the growth rates of the two discovery processes (the total number of distinct services, and the total number of common services) is proportional, hinting for a random discovery process.

We then turn our attention on how single users browse the Web, and check if the contacted services can be used to form a fingerprint of each person. We use the Jaccard index to quantify how similar is the browsing performed by i) the same user, and ii) two different users. Intuitively, one would expect that the browsing activity performed by the same user is repetitive over time. Instead, we find limited similarity suggesting a more random exploration of the Web. The affinity among two random users is even more limited, with most of the "common" services being only the most popular ones.

Given this, we run a simple experiment: we test if it could be possible to identify a target user by observing her browsing habits. We extract a user model by monitoring the services she visits during a period of time. Next, we compare the model against 1000 users in a different period of time. The experiment shows that it is possible to identify the target user in less than 50% of cases. This suggests that each individual user has unique interests, but these change over time complicating the task of building a good fingerprint.

While our work is preliminary, we believe it contributes to the understanding the Web, people browsing habits, and the Internet in general. We make the dataset used in this paper available to the research community, and invite the researchers interested in this field

to contact us.

2. RELATED WORK

In our work we consider how people browse the Web by characterizing the services they access to, as exposed by the DNS names of servers, or hostnames. We rely on DNS information to simplify our analysis, and to be able to extract information from passive measurements also in case encryption is in place. Indeed, while HTTPS is becoming more an more popular thus preventing passive extraction of data from traces, no deployment for encryption of DNS traffic is in place yet. Indeed, DNSSEC [7] provides data integrity and authentication, but no confidentiality, i.e., traffic encryption.[1]

Several works tried to profile users based on their browsing behavior. Authors of [10] tackle the user tracking problem. They exploit HTTP, HTTPS and SSH information to profile and re-identify users over time, using the cosine similarity and clustering techniques based on monthly profiles. Their results show that even by using a month-long profile, the false positive rate is very high 68% for HTTP and 21% for SSH. This confirms the difficulty of tracking user based on their traffic.

In the field of identification of DNS patterns, authors in [11] study the feasibility of tracking users looking on their DNS traffic. Using a trace collected in a campus network, they evaluate different classification techniques and their effectiveness. Results show that the tracking precision can reach up to 86% by using the Jaccard index as distance metric in a controlled scenario. However, in real case scenarios, accuracy drops to ≈50%. Our work differs as we present a much more detailed characterization of browsing behavior. Moreover, data used in [11] is of 2010, while the Web as evolved significantly. Finally, authors in [12] propose a high level co-clustering algorithm called *Phantom* to clusterize hostnames of servers based on the clients that access them. Differently to our work, they consider classes of services (e.g., e-commerce, news) instead of single services (e.g., ebay, nyt).

3. DATASET

To build the dataset upon which we base our analysis, we rely on a passive probe running Tstat[2] and installed in the PoP (Points-of-Presence) of the operational network of a national ISP. Users in the PoP connect to the Internet using ADSL modems. The traffic they generate is then routed through high-speed links, where Tstat passively analyzes it in real-time. Tstat monitors all packets and rebuilds TCP and UDP flows. Whenever a flow ends, Tstat dumps a line in a text log containing more than 100 detailed statistics.

For this work, we consider two anomymized 14-day long datasets, *VP1* and *VP2* respectively, collected from two different Vantage Points, located in PoPs in two large cities in the same country. In total, the datasets offer a snapshot of Web browsing activity of more than 25,000 residential customers. Table 1 provides details of the datasets. As it can be seen, more than a billion entries are at our disposal.

Trace	Period	Users	HTTP(S) flows
VP1	5-18 May 2014	12,262	532M
VP2	7-20 April 2014	13,473	614M

Table 1: Details of the datasets considered in this study.

3.1 Data extraction

For our study, we are interested in a specific subset of information collected by Tstat. In particular, we are interested in the data extracted by an Tstat plugin, named DN-Hunter [13]. In more details, when a browser is used to access a website, it has first to perform DNS resolutions of the hostnames associated to the objects found in the page. For each hostname, this returns the IP address(es) of server(s) to contact to fetch the desired object. DN-Hunter parses DNS requests and responses, allowing Tstat to annotate the subsequent TCP flow originated by the same client, and directed to a returned IP address of a server with the original hostname of the service being contacted. Since DNS messages are not encrypted, this allows Tstat to extract the hostname for both HTTP and HTTPS traffic with a very limited complexity, thus offering visibility on Web traffic even in presence of encryption.

Hence, for each TCP flow carrying Web traffic, we extract i) the timestamp at which the flow started, ii) the anonymized IP address of the client, which we employ as user ID hereinafter[3], and iii) the hostname of the server, as recovered by DN-Hunter.

3.2 Service definition

The aim of this study is characterizing the Internet users' online activity, and the information they expose. We are thus interested in which "services" users access to, as exposed by the server hostnames. Observe that the hostname offered by DNS is often redundant or not particularly interesting. For instance, consider *www.google.com* or *www.nyt.com*. Clearly, the so called "second-level domain" is the most significant

[1] There are proposals that address the problem of DNS traffic encryption, the most notable one being DNSCrypt [8]. Their deployment is hampered by the significant infrastructure changes they require[9].

[2] `http://tstat.polito.it/`

[3] The ISP assign static IP addresses to customers' access router, so that the client IP address is a stable and consistent ID.

part, e.g., *google* and *nyt*, which is the one that identifies the service being accessed. Similarly, *www.google.de* or *www1.nyt.com* offer little or no additional information. Hence, we consider only the second level domain to identify the services in the following.

Some considerations hold. First, when a user accesses a website, the browser generates a lot of different requests, one for each object composing the webpage. Most of these objects are being served by servers which are not related with the service the user is interested into, but they are used to support the page delivery. For instance, Content Delivery Networks (CDNs), or advertisement platforms, or video streaming caches are regularly contacted when browsing a webpage. These are *support services* and do not identify *actual services*, i.e., the name of the service the user is interested into. Unfortunately there is no easy way to identify them, so that we cannot actually filter them. Second, the frequency with which these support services are contacted is very high [14]. As we will show in Sec. 5, the most popular services are indeed support services, which however do not characterize the actual browsing habits of a user. In Sec. 5, we provide a characterization of their impact.

If not otherwise stated, the results presented in the remainder of the paper are obtained from the analysis of dataset *VP1*. However, all presented experiments have been conducted on *VP2* too, and we could not observe any significant difference.

4. METHODOLOGY AND METRICS

In the following, we provide a formal description of the methodology upon which we base our analysis. We follow a simple approach based on set theory. Given a flow i, generated by user u_i at time t_i and accessing service s_i, and considering a time interval of duration ΔT, we define the set of services S a user u accessed in the time interval starting from t_0 as

$$S(u, t_0, \Delta T) = \{s_i \mid t_0 \leq t_i < t_0 + \Delta T,\ u_i = u\} \quad (1)$$

Similarly, we can define the set of services accessed by all users during a period of time as

$$S(*, t_0, \Delta T) = \{s_i \mid t_0 \leq t_i < t_0 + \Delta T\} \quad (2)$$

Let $|S|$ be the number of elements in the set. We can define the *similarity* between the two sets by computing the Jaccard index [15] as

$$Sim(S1, S2) = \frac{|S1 \cap S2|}{|S1 \cup S2|} \quad (3)$$

It returns the ratio between the number of common elements over the total number of distinct elements in $S1$ and $S2$. Sim equals 0 if there is no common element. If $S1$ and $S2$ contain the same elements, Sim equals 1.

Note that we do not consider the frequency with which a service is accessed, but only its presence in the set. Indeed the frequency is highly skewed because of the presence of support services. As we show in the following section, these are contacted by users' browsers much more frequently than actual services.

5. AGGREGATE CHARACTERIZATION

In this section we present measurement results that are useful to understand how an aggregate of ≈12,000 Internet users from *VP1* accesses the Web. This helps in characterizing the dataset size and growth in time.

5.1 Service characterization

Starting from t_0 = May 5^{th}, Figure 1 plots the cumulative amount of distinct services we observe over the 14 days of time, i.e., $|S(*, t_0, \Delta T)|$ for increasing ΔT. In total we count about 400,000 distinct services. Interestingly, we observe that, despite we focus on second level domain name only, the number of newly accessed services keeps increasing over time, and does not saturate over the considered period. This reflects the humongous "catalog" of services that people can access on the Internet. The curve exhibits a daily pattern that reflects the typical day/night activity users follow when browsing the Web.

We next analyze the popularity of the services and how it changes when increasing the observation window. Figure 2 reports, for several observation intervals ΔT, the service rank according to their normalized popularity, which we compute as the fraction of users accessing a given service among the population of active users at the considered time interval. Each curve corresponds to a different time scale. In particular, the two bottom curves report the ranks computed on one-hour and four-hour long activity periods at peak time for the first day of our dataset, i.e., for t_0 = May 5^{th} at 8pm, $\Delta T = 1h$ and $\Delta T = 4h$ respectively. For the remaining periods, t_0 = May 5_{th} at 0am, and we consider $\Delta T = 1, 3, 7, 14$ days. Notice the log-log scale. Independently on the time scale, all curves present a Zipf-like distribution, with a few services being very popular, and the vast majority of them being contacted by a small number of users. Zipf's distribution is known for governing many aspects of the Internet [16]. It entails that the popularity of services quickly decreases, so that the majority of them are actually contacted by a handful of people, and only few services can reach large popularity. Indeed, considering the ΔT = 24h, only the top 0.26% services are contacted by more than the 10% of active users, and only ≈36% of services are contacted by more than one user. The shift of the curves toward the right part of the plot reflects the growth in the number of *distinct services* seen in Figure 1. Conversely, the shift toward the top of the plot reflects the growth in the

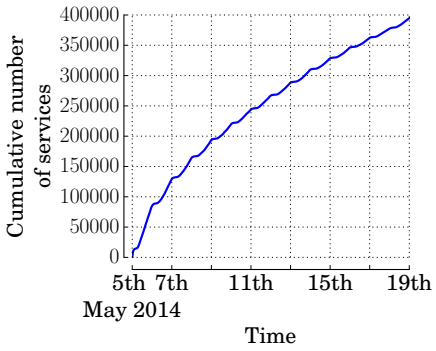


Figure 1: Cumulative number of distinct services over time, *VP1*.

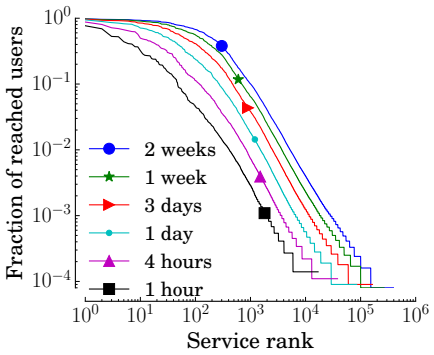


Figure 2: Service rank based on popularity among users, *VP1*.

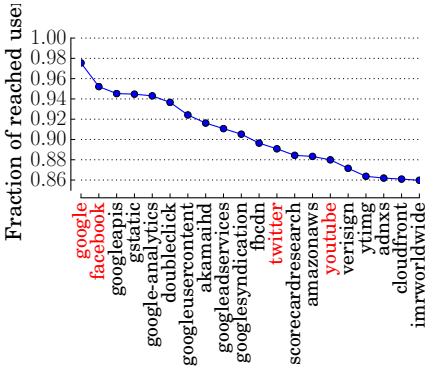


Figure 3: The top 20 most contacted services in dataset *VP1*. Actual and support services are in red and black, respectively.

number of active users. Finally, the shift of the knee toward the upper right part of the plot reflects the increase of *common services*.

Most of the common services are actually support services. To show their impact, we focus on the service popularity rank computed considered $\Delta T = 1$ week, and detail the top 20 most popular services. We report them in Figure 3. Names in red correspond to *actual services*, while names in black are *support services* which the browser contacted, but which the user did not explicitly access. Interestingly, among the 20 top popular services, only four, i.e., *google*, *facebook*, *twitter* and *yahoo*, are those expected to be actually requested by the users. Others are CDNs (*akamai*, *googleusercontent*, *fbcdn* - the Facebook CDN, *ytimg* - the YouTube CDN serving images, *cloudfront* - the Amazon CDN), or advertisement and user-tracking platforms (*google-analytics*, *doubleclick*, *googleadsservice*, *googlesyndication*, *scorecardresearch*, *adnxs*, *imrworldwide*), or cloud service (*googleapis*, *gstatic*, *amazonaws*, *verisign*). Investigating further, the vast majority of the services a user encounters during her online activity are *support services*, and this is nicely reflected in DNS traffic. Those are among the most popular services, as clearly shown by Figure 3. However, those are only a fraction of the services users access in the Internet. For instance, by randomly picking 100 services, and manually checking them, we count approximately 15% were support services, with 85% being actual services.

Next, we conduct a simple experiment to calibrate the parameter ΔT described in Sec. 4. We divide the dataset in smaller portions of duration ΔT. Hence, we obtain time bins aggregating the services contacted by all users, and that we represent as $S(*, t_i, \Delta T)$ for which $i \in 1, ..., \frac{L}{\Delta T}$, where L is the total duration of the dataset. Then, we consider each bin pair $(S(*, t_i, \Delta T), S(*, t_j, \Delta T)), \forall i, j \neq i$. We define the set of services which are present in both bins, i.e., the *common* services, as

$$C(t_i, t_j, \Delta T) = S(*, t_i, \Delta T) \ \cap \ S(*, t_j, \Delta T) \ \forall \ i, j \neq i, \tag{4}$$

and we compute the average number of common services over all possible pairs for a given ΔT as

$$E[|C(\Delta T)|] = \frac{\sum_{i,j \neq i} |C(t_i, t_j, \Delta T)|}{\left(\frac{L}{\Delta T}\right)\left(\frac{L}{\Delta T} - 1\right)} \tag{5}$$

For each bin pair we also compute the union, thus obtaining the set of *distinct* services appearing in the two bins as

$$D(t_i, t_j, \Delta T) = S(*, t_i, \Delta T) \ \cup \ S(*, t_j, \Delta T) \ \forall \ i, j \neq i \tag{6}$$

and we compute the average number of distinct services over the number of pairs for ΔT as

$$E[|D(\Delta T)|] = \frac{\sum_{i,j \neq i} |D(t_i, t_j, \Delta T)|}{\left(\frac{L}{\Delta T}\right)\left(\frac{L}{\Delta T} - 1\right)} \tag{7}$$

We now compare the growth of the number of common services to the growth of the number of distinct services for increasing ΔT. One would expect that the number of common services would grow faster, since, for increasing time, the chance that a service appears in two different snapshots of time is higher. Figure 4 shows the results. It reports $|C(\Delta T)|$ versus $|D(\Delta T)|$ for different values of ΔT. Blue bars report the 20^{th}- and 80^{th}-percentile of the distribution among all pairs. Surprisingly, observe how the growth of number of common and distinct services is linearly proportional, i.e., the number of common services grows with a rate that is proportional to the growth of the number of distinct services. The ratio among them is 1.44. This entails that the common services are approximately 35% of distinct services, independently with respect to the size of the observation window.

In summary, the catalog of services in the Internet is very large. People keep accessing previously unseen services, so that the total number of distinct services

keeps growing over time. Surprisingly, the chance that one visits a service that has been already visited or a new service does not depend on the observation interval. Given this, for the experiments presented in the remainder of the paper we choose ΔT =24 hours.

5.2 Population characterization

We now focus on observing the users' activity. In particular, we are interested in characterizing users based on the amount of services they contact, to check for instance for the presence of heavy-hitter users, and occasional users. For each user, we count the number of visited services, considering one day. Results, not shown here for lack of space, show that in a single day 50% of users contacts less than 100 services, with only 15% of the most (least) active users that contact more (less) than 250 (30) services.

We leverage this distribution to arbitrarily define three classes of users upon which we will base the experiments presented in the following. First, we define as "inactive" the users who contact less than 50 services per day. For instance, for the first day, these represent 36% of users. We next pick 1000 "moderately active" users, set $\mathcal{MA}$, as those users who contact from 650 to 750 services over the whole trace period.[4] Finally, we consider the top 1000 most active users, i.e., those with at least 1200 contacted services, which we call "very active", set $\mathcal{VA}$.

6. CHARACTERIZATION OF USERS' HABITS

In this section we employ the Jaccard index to first observe how repetitive is users' browsing by comparing services contacted during different time periods. Then, in the second part of the section, we compare browsing habits among different users.

6.1 Are user's habits similar over time?

We aim at gauging whether users tend to be repetitive in their browsing activity. To this end, we consider all users in the very active and moderately active classes. For each user u, we build the subsets containing the services corresponding to their activity in different time period t_i of duration ΔT=24 hours. For each user, we obtain 14 independent sets, $S(u, t_i, \Delta T)$, one for each day in the trace. Next we compute the similarity index across all set pairs

$$Sim(u, t_i, t_j) = Sim(S(u, t_i, \Delta T), S(u, t_j, \Delta T)) \; \forall i, j \neq i. \tag{8}$$

To provide an intuitive representation of the outcome of the experiment, Figure 5 reports two examples of users we randomly picked in the very active (top) and moderately active (bottom) classes. For each pair of days (t_i, t_j), it reports $Sim(u, t_i, t_j)$ using a color map; The darker the color, the higher is the similarity. Days start from Monday the 5^{th} of May, from the bottom left of the plot.

In general, we observe that the similarity is fairly low, with most of values that are smaller than 0.5. For the active user, Figure 5(a) the similarity index decreases during the weekends (notice the yellow columns on Sundays). This corresponds to the user not heavily browsing during the weekend, but leaving some device connected to the network, and periodically polling web services, e.g., looking for software updates, or syncing with cloud services. As such, there is a common substrate of support services that are contacted, and that generate a minimum amount of similarity.

Consider the case of the moderately active user, Figure 5(b). Some days present a very high similarity, while others seem lightly similar. The white bar on Monday is due to the fact that the user contacted less than 50 services during that day, thus falling in the "inactive" group for which we do not compute the similarity at all. In general, it is hard to identify a regular pattern. For instance, the activity the user performs on Wednesday, first week, is very dissimilar to any other day in the dataset. Or conversely, Tuesday of the first week, and Wednesday of the second week exhibit a very high similarity. Yet, the same Tuesday results rather different when compared to the following Wednesday.

6.2 Impact of the number of contacted services

Next, we explain how the properties of the considered time periods impact the similarity index $Sim(u, t_i, t_j)$. In particular, we investigate the effect of the number of services found in the two time intervals. To this end, for each user u in the moderately active class $\mathcal{MA}$, and for all pairs (t_i, t_j), we compute the similarity $Sim(u, t_i, t_j)$. Let $m = min\left(|S(u, t_i, \Delta T)|, |S(u, t_j, \Delta T)|\right)$ be the minimum number of services in the subsets of services contacted during the considered time intervals. We plot $Sim(u, t_i, t_j)$ versus m for all possible users u and all possible pairs (t_i, t_j).

The result is depicted as a scatterplot in Figure 6. Curves reports the average (solid red curve) and median (dashed red curve) similarity for samples that fall in bins of size 10, i.e., where of $10k < m < 10(k+1), k = 1, 2, \ldots, 30$. Observe that when the minimum number of services is small ($m < 50$) the similarity is on average very small, and varies widely, reaching 1 for very small values of m. This is due to time period pairs in which the number of contacted services is very low, and their intersection contains very popular services such as, e.g., *google* or *google-analytics*, the majority of which are support services. This further justifies the choice of not computing the Sim index when the minimum number

[4]We chose these thresholds to be just above the median, so that we are relatively confident to focus on a set of consistently active users.

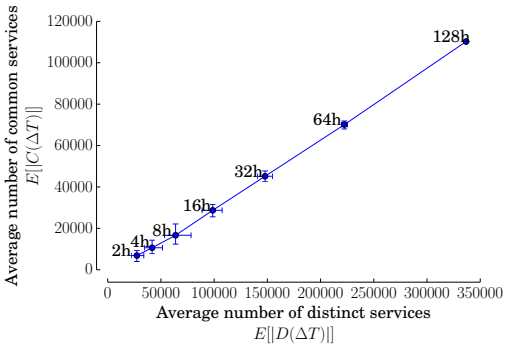
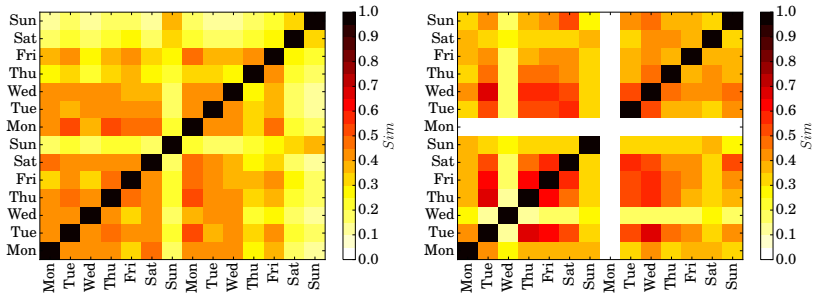


Figure 4: Average number of distinct services, $E[|D(\Delta T)|]$ vs the average number of common services, $E[|C(\Delta T)|]$, for $\Delta T \in [2, 4, 8, 16, 32, 64, 128]$ hours, *VP1*.



(a) Very active user.

(b) Moderately active user.

Figure 5: Heatmaps reporting the matrices of similarity calculated across 24-hour long bins for two example users.

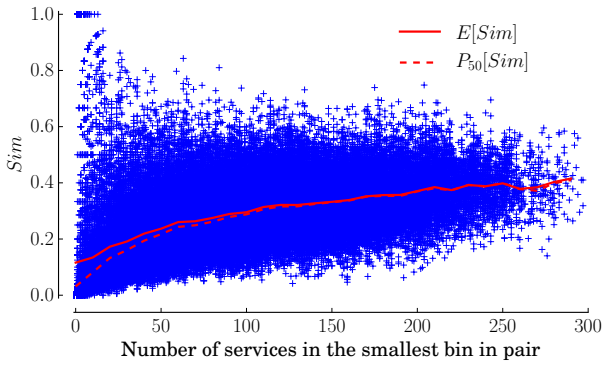


Figure 6: Per-bin-pair similarity index vs the minimum number of services, *VP1*.

of services contacted in one of the two bins is below 50. Conversely, as the number of services grows, the similarity score becomes higher, and at the same time the maximum observed score decreases. Looking at the mean and median values, both stabilizes to a value of ≈ 0.4. Conversely, both are below 0.25 when the minimum number of services is smaller than 50. This is explained by the fact that, in general, sets with a small number of services are compared against sets with large number of services. By definition of the Jaccard index, the denominator (the number of element in the union) has a size which is much larger than the numerator (the number of elements in the intersection).

We complement above observations with another experiment. Again, for each moderately active user $u \in \mathcal{MA}$ we collect the number of contacted services in each time period t_i, and we compute the number of elements it contains, i.e., $n_i = |S(u, t_i, \Delta T)|$. We then consider all pairs (t_i, t_j) $i \neq j$, and consider a discretised grid counting the number of services in buckets of size 10. In other words, we count the number of pairs falling in each bucket. Intuitively, this shows the number of pairs having a given number of services in each. The result is a symmetric matrix, depicted in Figure 7. Darker colors are assigned to buckets containing larger values elements. The plot shows that a considerable fraction of pairs falls in the area where the number of services is smaller than 50, with a very large number of pairs that falls in the $(< 10, < 10)$ bucket (observe the dark block in the origin). These are time periods during which the user is actually inactive, and during which mostly support services are contacted. As explained above, when the number of services is so small, the similarity index computation leads to very variable results, and thus we prefer to not compute it.

Observing Figure 7, we notice that the largest portion of pairs falls in the 70,220 area, i.e., where the number of pairs is large, and thus allows us to obtain a significant similarity measure.

6.3 Self similarity

In this section we analyze how habits of the same users look similar to each other. We take each moderately and very active user separately, and for each of them we compute the $Sim(u, t_i, t_j)$ index across different time periods (t_i, t_j) for which they results active, i.e., $n_i \geq 50$, $n_j \geq 50$. Then, for each class of user, we build the distribution of the similarity values. Results are depicted in Figure 8(a). First, observe that similarity is higher than 0.1, and it saturates at $Sim = 0.7$. This means that, independently of the volume of their activity, the sets of services contained in time bins are significantly different. In other words, users tend to contact same services over time, but the number of new services is however fairly large, meaning that the degree of repetitiveness is low. No significant difference is observed comparing very active and moderately active

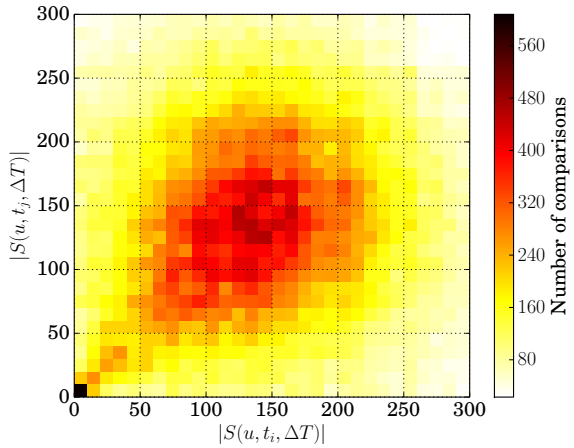
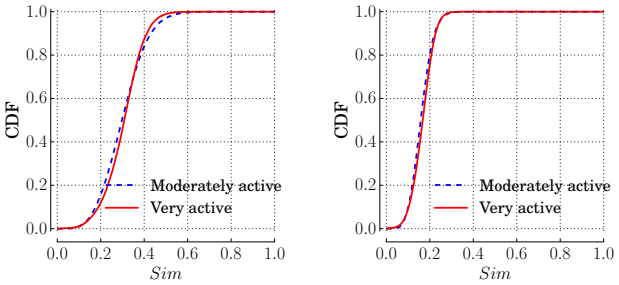


Figure 7: Number of pairs (t_i, t_j) with a given number of elements in each..



(a) Same users. (b) Different users.

Figure 8: Distribution of the similarity index computed across time bins belonging to the users in very active and moderately active classes.

users.

6.4 Similarity across users

In this section we analyze how habits of different users look similar to each other. Again, we consider the services contacted by users in different time periods. We calculate the Sim index across different users, and we build the distributions for the values obtained separately for the very and moderately active users. We report the results in Figure 8(b). As shown, the similarity among different users is very small, and smaller than the one among the same user, cfr. Figure 8(a). Indeed, CDFs now saturate at $Sim = 0.3$, meaning that the activity of different users is in 70% of cases very different. As above, no significant difference is observed when considering very active or moderately active users.

This may induce one to conclude that a user is easily identifiable by the set of services she contacts. We next run a simple test to check weather this intuition is correct or not.

7. BUILDING A SIMPLE USER CLASSIFIER

We run now a simple test to check the capability of correctly identifying a user based on the set of services she contacts. To this end, we focus on the moderately active user class. We use the datasets we obtain from the fist day of *VP1* to build a user signature, and then check if it is possible to recognize her in the subsequent days. In more details, for each user $u \in \mathcal{MA}$, we extract the set of services it contacts on the first day t_0, $S(u, t_0, \Delta T)$. Next, we compute the similarity index against, $S(u_i, t_j, \Delta T), \forall\ u_i \in \mathcal{MA}$ and $j \in \{1, ..., 13\}$, i.e.,

$$Sim(u, u_i) = Sim(S(u, t_0, \Delta T), S(u_i, t_j, \Delta T)). \quad (9)$$

We then rank users for decreasing similarity. Intuitively, we want to check how many times the most similar user turns out to be u in a group of a 1000 users $\{u_i\}$.

In our experiments we observe that the choice of the services upon which we compute the similarity is crucial. For instance, we have seen in Sec. 5 that the top popular services tend to be support services that users connect to, but do not explicitly browse. Since those are extremely popular, the chance they are in the common set is high, but they do not characterize the user behavior. Similar, those services that do not belong to the common service group are by definition accessed once by one user. Those again would not contribute to identify the user behavior in a different time period. Hence, we create three simple policies to filter the services to build the sets used to characterize the user:

•**No Filter**: no filter is applied and all services are considered when building $S(u, t_i, \Delta T)$.

•**Top Filter**: We filter out the most contacted services, i.e., those contacted by more than 50% of users considering the whole dataset.

•**Top+Bottom Filter**: We filter the most contacted services considering the whole dataset, plus those accessed only once in each $S(u, t_j, \Delta T)$.

Figure 9 depicts the probability that the user u is found to be among the X^{th} most similar users to herself in the following days, being X the position in the similarity rank. For instance, the leftmost point reports the probability that the user is found to be the first in the rank, i.e., $argmax_{u_i}(Sim(u, u_i)) = u$.

As shown, when no filtering is applied, in 36% of the cases we can successfully identify a user. Conversely, in 50% of the cases a user has at least other 9 users who exhibit a higher similarity index than herself. The Top and Top+Bottom filters achieve better results, but the chances for a user to be uniquely identified increase only to 44%. All these observations demonstrate that users' browsing habits are only partially repetitive enough to allow one to easily build a model of the user so that later would it be possible to identify it in a population of Internauts. Our results confirms previous finding [11].

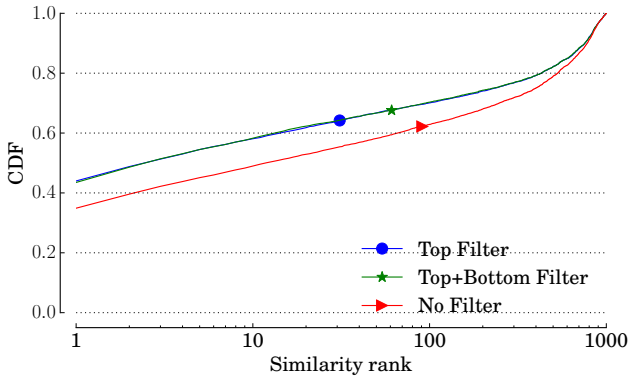


Figure 9: Distributions of the number of times a user is found to be the X^{th} most similar to herself across different observation windows for three different filtering policies.

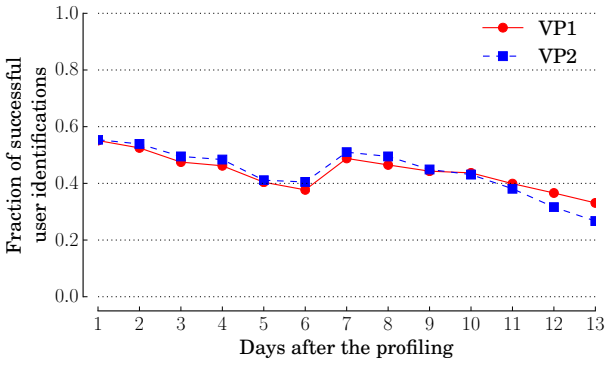


Figure 10: Fraction of times users are successfully identified by their profile at day t_0 over the subsequent days.

7.1 Time stability of the model

We repeat the above experiment considering how similar the results are when comparing sets obtained from far apart days. For a user u, we build a model considering the services she visits during the first day, i.e., $S(u, t_0, \Delta T), t_o = \text{May } 5^{th}$. We then compute the similarity for all days in the following two weeks among all $u_i \in \mathcal{MA}$. For this experiment we employ the Top-Bottom filter. Figure 10 shows the probability to correctly identify a user u in the next days, i.e., when $Sim(u, u, t_j) > Sim(u, u_i, t_j)$. We report results for *VP1* and *VP2* to show how similar are results. Interestingly, we observe that the probability to correctly identify the same user decreases over time. This hints that users tend to visit different services depending on the day of the week, and their online activity slightly differentiate over time. However, observe how after one week the similarity index increases again. This suggests that users tend to exhibit a weekly periodicity when browsing the Web.

8. CONCLUSIONS

In this study we leveraged an actual dataset we obtained from an ISP to analyze how Internauts browse the Web by using simple DNS traffic.

We have seen that in practice the similarity of the browsing activity during two different periods of time is rather limited. This surprisingly entails that users are not repetitive in their browsing. However, the practice of building user profiles based on fingerprints made by services contacted is very unreliable. Indeed, in our experiment we could uniquely identify users based on their activity in only 44% of the cases. Moreover, we observed that the profiling effectiveness vanishes over time.

The topic is worth further investigation. Thus we invite researchers that interested to access our dataset that we made available to the community upon request.

9. REFERENCES

[1] L. D. Catledge and J. E. Pitkow, "Characterizing browsing strategies in the world-wide web," *Computer Networks and ISDN systems*, vol. 27, no. 6, pp. 1065–1073, 1995.
[2] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *ACM IMC*, 2009.
[3] A. Finamore, M. Mellia, M. M. Munafò, R. Torres, and S. G. Rao, "Youtube everywhere: Impact of device and infrastructure synergies on user experience," in *ACM IMC*, 2011.
[4] C. Tossell, P. Kortum, A. Rahmati, C. Shepard, and L. Zhong, "Characterizing web use on smartphones," in *ACM SIGCHI*, 2012.
[5] D. Olmedilla, E. Fras-Martnez, and R. Lara, "Mobile web profiling: A study of off-portal surfing habits of mobile users," in *User Modeling, Adaptation, and Personalization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6075, pp. 339–350.
[6] F. Benevenuto, T. Rodrigues, M. Cha, and V. Almeida, "Characterizing user behavior in online social networks," in *ACM IMC*, 2009.
[7] N. W. Group *et al.*, "Request for comments (rfc) 4033,," *Protocol Modifications for the DNS Security Extensions, Mar*, 2005.
[8] https://www.opendns.com/about/innovations/dnscrypt/.
[9] B. Ager, H. Dreger, and A. Feldmann, "Predicting the dnssec overhead using dns traces," in *IEEE CISS*, 2006.
[10] M. Kumpošt and V. Matyáš, *User Profiling and Re-identification: Case of University-Wide Network Analysis*. Springer, 2009.
[11] D. Herrmann, C. Banse, and H. Federrath, "Behavior-based tracking: Exploiting characteristic patterns in dns traffic," *Computers & Security*, vol. 39, pp. 17–33, 2013.
[12] R. Keralapura, A. Nucci, Z.-L. Zhang, and L. Gao, "Profiling users in a 3g network using hourglass co-clustering," in *ACM MobiCom*, 2010.
[13] I. Bermudez, M. Mellia, M. Munafò, R. Keralapura, and A. Nucci, "DNS to the Rescue: Discerning Content and Services in a Tangled Web," in *ACM IMC*, 2012.
[14] V. Gehlen, A. Finamore, M. Mellia, and M. M. Munafo, *Uncovering the big players of the web*, 2012.
[15] M. Levandowsky and D. Winter, "Distance between sets," *Nature*, vol. 234, no. 5323, pp. 34–35, 1971.

[16] L. A. Adamic and B. A. Huberman, "Zipfs law and the internet," *Glottometrics*, vol. 3, no. 1, pp. 143–150, 2002.