How Dynamic are IP Addresses?

Yinglian Xie, Fang Yu, Kannan Achan
Eliot Gillum[+], Moises Goldszmidt, Ted Wobber
Microsoft Research, Silicon Valley
[+]Microsoft Corporation
{yxie,fangyu,kachan,eliotg,moises,wobber}@microsoft.com

ABSTRACT

This paper introduces a novel algorithm, *UDmap*, to identify dynamically assigned IP addresses and analyze their dynamics pattern. UDmap is fully automatic, and relies only on application-level server logs. We applied UDmap to a month-long Hotmail user-login trace and identified a significant number of dynamic IP addresses – more than 102 million. This suggests that the fraction of IP addresses that are dynamic is by no means negligible. Using this information in combination with a three-month Hotmail email server log, we were able to establish that 95.6% of mail servers setup on the dynamic IP addresses in our trace sent out solely spam emails. Moreover, these mail servers sent out a large amount of spam – amounting to 42.2% of all spam emails received by Hotmail. These results highlight the importance of being able to accurately identify dynamic IP addresses for spam filtering. We expect similar benefits to arise for phishing site identification and botnet detection. To our knowledge, this is the first successful attempt to automatically identify and understand IP address dynamics.

Categories and Subject Descriptors

C.2.3 [**Computer Communication Networks**]: Network Operations—*network management*; C.2.0 [**Computer Communication Networks**]: General—*security and protection*

General Terms

Algorithms, Measurement, Security

Keywords

DHCP, dynamic IP addresses, IP volatility, entropy, spam detection

1. INTRODUCTION

Many existing tasks such as malicious host identification, network forensic analysis, and other blacklisting based approaches often require tracking hosts identities. Techniques that use host IP addresses to represent host identities are commonly used (e.g., [13, 26, 32]). These techniques are based on the premise that a vast majority of IP addresses in the Internet are static, and that the fraction of dynamic addresses is small. Unfortunately, the validity or the degree to which this important assumption holds has not been studied in existing literature.

In this paper, we aim to quantify the above assumption, and in the process answer the following questions. Is the set of dynamic IP addresses really a small fraction of the set of all IP addresses in the Internet? How can we automatically identify a dynamic IP address, and meanwhile estimate the frequency at which it is used to represent different hosts?

The answers to these questions clearly have wide applicability. For example, existing blacklist-based approaches for detecting malicious hosts (e.g., botnet members, virus spreaders) should not include individual dynamic IP addresses straightforwardly in their filters, as the identities of such hosts change frequently. Similarly, Web crawlers should pay special attention to IP addresses that exhibit very dynamic behavior, as the records they point to typically expire quickly.

Another application, which we use as a case study in this paper, is spam filtering. Previous studies suggest that spammers frequently leverage compromised zombie hosts as mail servers for sending spam [8, 23], and that many zombie hosts are home computers with serious security vulnerabilities [19]. Therefore, a mail server set up at a dial-up or wireless connection is far more suspicious than one set up with a statically configured IP address. In other words, whether a mail server is mapped to a dynamic IP address or not can turn out to be a useful feature to add to existing spam filtering systems.

Throughout this paper, we use the term *IP dynamics* to refer to the dynamic behavior, over time and in aggregate, of the mapping between IP addresses and host computers. Collecting information about IP dynamics is a challenging task for several reasons. First, such information is essentially very fine grain – even for IP addresses within the same administrative domain and sharing the same routing prefix, IP dynamics can be very different. For example, it is not unusual for the static IP address of a Web or mail server to be adjacent to a wireless DHCP IP range. Second, ISPs and system administrators often consider the configuration of IP address ranges to be confidential and proprietary. Such information can potentially be used to infer the size of customer population and operation status. Finally, the Internet is composed of a large number of independent domains, each with its own policies for IP assignment. Thus *manually* collecting and maintaining a list of dynamic IP addresses requires an enormous effort, especially given the fact that the Internet evolves rapidly.

An important goal of this paper is to develop an *automatic* method for obtaining *fine-grained, up-to-date* dynamics properties for IP addresses. We introduce a metric, *IP volatility*, that expresses the

SIGCOMM'07, August 27–31, 2007, Kyoto, Japan.

rate at which a given IP address is assigned to different hosts. Estimates of IP volatility can help distinguish whether IP addresses are statically assigned, or belong to a block [1] of dynamically configured DHCP [6] addresses such as those commonly used for dial-up, DSL, or wireless access. As we will demonstrate, such fine-grained dynamics information can suggest possible host properties behind the IP address – whether the host is an end user computer, a proxy, or a kiosk-like shared computer.

We propose *UDmap*, a fully automatic method for identifying dynamic IP addresses and estimating IP volatility. The dynamic IP addresses we find are a subset of DHCP addresses that exclude statically configured addresses, such as those based on host-MAC address mapping. UDmap utilizes two types of information. First, we require a log that contains information that roughly tracks host identities at specific IP addresses. In this paper, we use a one-month trace of Hotmail user-login sessions for this purpose. Second, we require access to IP address aggregation information such as BGP routing table entries and CIDR IP prefixes. Overall, our algorithm has the following desirable properties:

- *It is generally applicable.* UDmap can be applied not only to Hotmail user logs, but also to other form of logs, such as Web server or search engine logs with user/cookie information.

- *It runs autonomously.* Each domain or server can independently process the collected data, with no need to share information across domains. Further, UDmap does not require changes to client software.

- *It offers fine-grained, up-to-date IP dynamics information.* UDmap identifies dynamic IP addresses in terms of IP blocks, often smaller than IP prefixes, and thus more precise. As it is fully automated, it can be constantly applied to recent logs to obtain up-to-date information.

We also present a detailed study of IP dynamics at a large scale, and apply our technique to spam filtering using a three-month long Hotmail email server log. Our key findings include:

(1) *There are a large number of dynamic IP addresses that have not not been identified by previous work.* Using the one-month Hotmail user-login trace, UDmap identified over 102 million dynamic IP addresses across 5891 ASes. A large fraction of the identified dynamic IP addresses are DSL hosts, with the top ASes from major ISPs such as SBC and Verizon. Over 50 million of the identified dynamic IPs do not show up in existing dynamic IP lists [7].

(2) *IP volatility varies widely, with IP-to-host bindings changing from several hours to several days.* Over 30% of the dynamic IP addresses we identified had average IP volatility of between 1 and 3 days. As might be expected, IP volatility is correlated to network access method. Our findings suggest that IP addresses configured for dial-up access are more dynamic than those for DSL links, while IP addresses in cable modem networks are least dynamic.

(3) *Spam filtering can benefit from using IP dynamics data.* To our knowledge, we are the first to provide a systematic study on the correlation between the portion of dynamic IP addresses and the degree of spamming activities. By examining Hotmail email server logs, we show that 95.6% of the sending servers from dynamic IP ranges sent *only* spam emails. The total volume of spam from dynamic addresses we detected is significant: it constitutes 42.2% of all spam sent to Hotmail during our trace period.

We acknowledge that, despite the large size, our Hotmail login dataset is still far from providing a complete view of the global IP address space. The purpose of this paper is not to identify all dynamic IP addresses in the Internet. Rather, the goal is to expose IP dynamics as an important feature to consider for various network applications, and more importantly, to offer a practical solution for obtaining and understanding fine-grained IP dynamics information.

[1] We use the term *block* to represent a group of continuous IP addresses, typically of finer granularity than an IP prefix.

2. RELATED WORK

We review related work in identifying dynamic IP addresses in Section 2.1. As we propose spam filtering to be a prime application area of UDmap, in Section 2.2, we briefly survey existing approaches to spam detection, particularly those that relate to the theme of our work.

2.1 Dynamic IP Identification

To the best of our knowledge, we are the first to develop a framework to *automatically* detect dynamic IP addresses on a global scale and simultaneously understand the associated IP volatility. In all prior work, enumerations of dynamic IP addresses have been maintained by hand [9].

Some dynamic IP addresses can be deduced by examining the Reverse DNS (rDNS) and Whois databases [30]. A rDNS record maps an IP address into a host name, providing a way to infer its address properties. For example, the rDNS record for 157.57.215.19 corresponds to the DNS name *adsl-dc-305f5.adsl.wanadoo.nl*, indicating that the IP address is used for an Asymmetric Digital Subscriber Line (*adsl*) in the Netherlands (*nl*). Despite the existence of DNS naming conventions and recent proposals on standardizing DNS name assignment schemes [27], not all domains follow the naming rules. In fact, many IP addresses do not have rDNS records: it is reported that only 50 to 60% of IP addresses have associated rDNS records [10].

Certain enterprises maintain Dialup User Lists (DULs) of suspected dynamic IP addresses, largely to support efforts to aid in spam filtering [29]. Dynablock provides the most well known and widely used DUL [7]. It not only contains dialup IPs, but also other dynamic IPs such as DSL and cable user IP ranges. As of January 2007, the list contains over 192 million dynamic IP addresses. Manually maintaining such a large list requires enormous effort and resources. Moreover, updating dynamic IP addresses relies on the reporting of system administrators. With Internet topology and IP address assignments changing rapidly, Dynablock can be expected to contain increasingly obsolete information and miss newly configured dynamic IPs. In Section 5.3, we show that our automatic method identifies over 50 million dynamic IP addresses that are not covered by Dynablock.

While there are no existing approaches that automatically identify dynamic IP addresses, there has been significant amount of prior work on finding the topological and geographical properties associated with an IP address. Krishnamurthy et al. [15] have proposed to cluster Web clients that are topologically close together using BGP routing table prefix information. Padmanabhan et al. [20] have proposed several methods to obtain geographic locations of IP prefixes. Freedman et al. [10] further extended [20] to provide even more fine grained geographic location information. Recently, Casado and Freedman [3] proposed to identify NAT and proxies by passively collecting client information using active Web content such as Javascript. Our technique is complementary to these efforts by focusing on the dynamic nature of IP addresses, and it does not require actively probing client machines.

2.2 Email Spam Filtering

Spam is an ever growing problem in the Internet. Recently, it has been reported that over 91% of all email generated is spam [21].

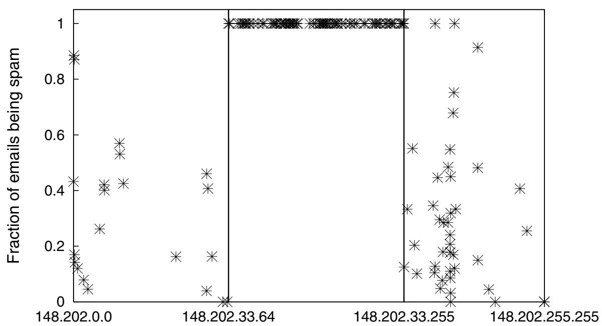


Figure 1: Spam ratio of mail servers in 148.202/16

Despite significant advances in anti-spam techniques (e.g., [11, 16, 18, 31]), spam fighting remains an arms race. Spammers increasingly use sophisticated techniques, such as arranging many tiny images to resemble message content or using animated GIF attachments, to bypass content-based spam detection systems [21]. Moreover, content-based systems are easy targets for spammers who can manipulate content at will until it gets by the filter.

Network-based spam filtering approaches that do not rely on message content have started to receive increased attention. DNS Black Lists (DNSBLs) have been used to record the IP addresses of spamming mail servers captured either through mail server logs or Honeypot projects [1]. In 2004, Jung and Sit [13] showed that 80% of spam sources they identified eventually appeared in one or more DNSBLs within two months. Recent study [23] has shown that spammers are getting more stealthy. Spammers often harvest a large number of zombie hosts to send spam, both to increase throughput and to defeat blacklist-based countermeasures. Some spammers even hijack IP prefixes for spamming [23]. As a result, a decreasing fraction of spamming hosts are listed in DNSBLs. Ramachandran et al. [22] recently showed that only 6% of the botnet IPs they queried were actually blacklisted.

Detecting correlation among email sources or content offers new possibilities for identifying spammers that control large botnets. Li and Hsieh [17] studied the behavior of spammers by clustering, using criteria such as the presence of similar URLs in messages sent out by mail servers. In a similar vein, Ramachandran et al. [24] correlated queries to DNSBL and botnet membership to identify zombie spammers. These approaches are grounded on the implicit assumption that IP addresses are generally static and that the fraction of dynamic IPs tends to be negligible. Under this assumption, recording the IP address of a spamming host in a blacklist is meaningful, as it can help filter out further spam from this host. However, as we show in this paper, this assumption is not valid and the number of dynamic IP addresses is very large. Obtaining the list of *active* dynamic IP addresses and understanding their properties is critical for network-based spam filtering approaches.

3. A MOTIVATING EXAMPLE

In this section, we present a case study that emphasizes the need for IP dynamics information to aid in spam detection. As we will discuss, the knowledge of dynamic IP address ranges can effectively help identify spamming hosts, especially for IP addresses outside the United States, where little information is available from existing data sources.

For our case study, we analyze the IP address block 148.202/16. This is a large block of 65,536 IP addresses owned by Universidad de Guadalajara in Mexico. The main reason for choosing this particular block is the amount of interesting activity happening behind it. 136 mail servers, all in this IP range, were used to send email to Hotmail accounts during the period from June 2006 until early September 2006. It is common for universities to configure mail and other computing servers using static IP addresses, while assigning dynamic IP address blocks for other uses (e.g., for wireless access). However, of the 136 mail servers we detected in this IP range, 75 were *solely* used to send spam, while the rest sent a mix of spam and legitimate email. This is further illustrated in Figure 1: notice that email servers in the address range 148.202.33.64 and 148.202.33.255 sent 100% spam.

As a first step, we searched for records pertaining to 148.202/16 using the Dynablock database and rDNS lookups. Surprisingly, none of the IP address in this range is listed in Dynablock, and a majority (93 out of 136) of these email server addresses don't even have an rDNS record.

Of the 33 IP addresses with rDNS records, only 3 can be verified as possibly legitimate, by virtue of the fact that the keyword `mail` was present in their host names. The remaining 30 IPs could not be classified due to the lack of any meaningful information in their rDNS records. For example, one such IP resolved to *foreigner.class.udg.mx*. From the name alone, we cannot infer either the type of IP address or whether this is a legitimate email server.

Blacklist-based spam filtering techniques are also not effective in the 148.202/16 address range. We screened all 30 popular spam server blacklists [1] for the presence of the offending 136 mail server IP addresses. Unfortunately, we were able to identify only 8 IP addresses from the blacklists. However, as we can see from Figure 1, the number of spamming mail server IPs is far more than 8. We can imagine two possible reasons for the absence of these spamming mail servers in the blacklists. First, they might have been sending a very low volume of spam, possibly below the threshold required to qualify for the blacklist. Second, they might have used dynamic IP addresses, meaning their IP addresses change from time to time, making it hard to set up a history.

Applying UDmap to this range, however, we identified 7045 IP addresses as dynamic. In particular, the range from 148.202.33.64 to 148.202.33.255 was identified as dynamic, where 73 IPs in this range were used to set up mail servers. Since legitimate mail servers must both send and receive emails, they are often configured to use static IP addresses to facilitate establishment of inbound connections. Thus, mail servers set up using dynamic IP addresses are more likely to be spam mail servers, directly controlled by spammers or leveraged as zombie hosts. Indeed, for the 73 mail servers set up with dynamic IP addresses, all of their traffic to Hotmail was classified as spam by the existing Hotmail spam filter.

The above discussion illustrates how the knowledge of IP dynamics can be a helpful feature for spam detection, particularly when existing network-based approaches fail.

4. THE UDMAP ALGORITHM

In this section, we present our methodology for automatically identifying dynamic IP addresses and computing IP volatility. We make the observation that dynamic IP addresses manifest in blocks [2], and therefore we explore *aggregated IP usage patterns* at the address block level. The IP addresses we seek to identify are those actively in use, so we name our algorithm *UDmap* in that it generates a *usage-based* dynamic IP address map.

UDmap takes as input a dataset that contains IP addresses and some form of persistent identification that can aid tracking of host

[2]It is common for system administrators to assign a range of IP addresses for the DHCP pool rather than creating a discrete list of individual IPs.

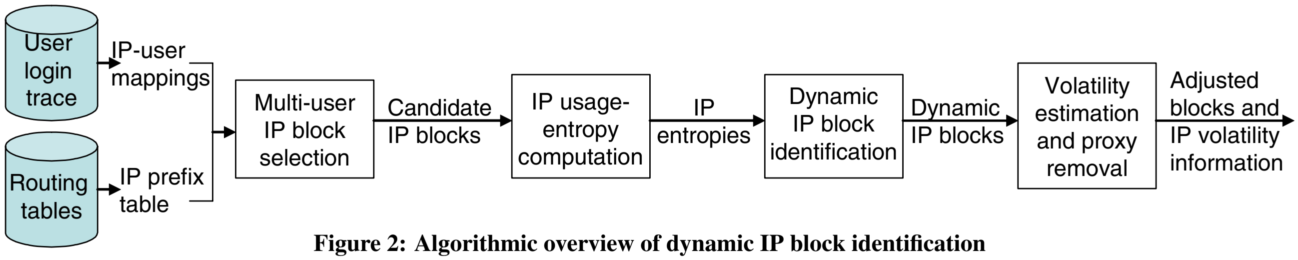


Figure 2: Algorithmic overview of dynamic IP block identification

identities, e.g., user IDs, cookies. Such datasets are readily available in many application logs, including but not limited to search engine and Web server traces. The availability of more accurate host identity information (e.g., OS IDs, device fingerprints [14], or MAC addresses) is not required, but may offer scope for more precise results.

In our study, we adopt a month-long MSN Hotmail user-login trace gathered during August, 2006 as our input data. Each entry in the trace contains an anonymized user ID, the IP address that was used to access Hotmail, and other aggregated information about all the login events corresponding to this user-IP pair in the month. The aggregated information includes the first and the last time-stamps of the login events over the month, and the minimum and the maximum IDs of the OSes used [3].

The output of UDmap includes (1) a list of IP address blocks identified as dynamic IP blocks, and (2) for each returned IP address, its estimated volatility in terms of the rate at which it is assigned to different hosts. In the rest of this section, we first explain the intuitions behind our approach (Section 4.1) and then present the UDmap methodology in detail (Section 4.2 to 4.5).

4.1 Methodology Overview

Establishing IP dynamics with only user-IP mapping information is a challenging task, because it is unrealistic to assume a one-to-one mapping between users and hosts. For example, a user can connect to Hotmail from both a home computer and a office computer. Further, a home laptop could be shared by family members, each having a different Hotmail user ID.

We now make several key observations that collectively make the identification of dynamic IP addresses possible. Although a user can use multiple hosts, these hosts are usually *not* located together in the same network, or configured to use the same network-access method (e.g., a laptop using a wireless network and a office desktop connecting through the Ethernet). Therefore it is very rare for a user to be associated with several to tens of static IP addresses, all from a very specific IP block. It is even rarer to observe a large number of users, with each having used multiple static IP addresses.

To the contrary, it is very common to observe users that are associated with multiple IP addresses from a dynamic IP address range. Dynamic IP addresses are usually allocated from a continuous address range, reachable by the same routing table prefix entries. Further, a user who appears at a given dynamic IP address is likely to use other IP addresses from the same range. UDmap thus explores the aggregated user-IP mappings to identify dynamic IP address ranges. By focusing on address activities at the granularity of IP blocks, it can make estimates about the behavior of addresses that appear infrequently or are absent from the traces.

Figure 2 presents a high level overview of the four major steps involved in identifying dynamic IP address blocks. First, UDmap selects (multi-user) IP blocks as candidate dynamic ones. Second, for each IP address in every candidate block, UDmap computes a score, defined as *usage-entropy*, to discriminate between a dynamic IP and a static IP shared by multiple users. In the third step, UDmap uses signal smoothing techniques to identify dynamic IP blocks by grouping addresses with high usage-entropies. Finally, UDmap estimates IP volatility, and based on it, further filters out proxy cluster IP addresses. The final output is a list of adjusted IP blocks and the associated address volatility. We present each of these steps in detail next.

[3]The trace collection process encodes each distinct type and version of operation system into a unique OS ID.

4.2 Multi-User IP Block Selection

The first step of UDmap is to identify candidate dynamic IP address blocks. Intuitively, if more than one Hotmail user is observed to use the same IP address, it is likely that this IP has been assigned to more than one host and hence is a candidate dynamic IP address. However, counting the number of users for each individual IP in a straightforward way is not robust due to two reasons: (1) it is likely that not all the addresses in a block will appear in the input dataset; (2) a small number of individual IPs in a dynamic IP block may still appear static by having a single user (e.g., a dynamic IP assigned to a home router that rarely reboots). Hence UDmap looks for multi-user *IP blocks*. In particular it selects a set of m *continuous* IP addresses IP_1 to IP_m as a candidate block $B(\text{IP}_1, \text{IP}_m)$ if the block has the following properties:

1. IPs in a block must belong to the same AS and also map to the same prefix entry in a BGP routing table.
2. Each block meets a minimum size requirement by having at least k IP addresses, i.e., $m \geq k$.
3. Both the beginning address (IP_1) and the ending addresses (IP_m) must be present in the input trace. Further, the block should not have significant *gaps*, where we define a *gap* as a region in the address space with g or more continuous IPs that were either not observed in our data, or used by at most a single Hotmail user.

By property (1), we ensure that IP addresses within a same block are under a single domain and topologically close. Properties (2) and (3) ensure that we observe a significant fraction of the multi-user IP addresses within the block. Notice that by returning IP blocks, IP addresses that were not present in the input data can be included in the output.

We used the BGP routing table collected on August 1, 2006 by Routeviews [25] to extract IP prefix entries. The parameters k and g have potential impact on both the coverage and the returned IP block sizes. Intuitively, a small k is likely to have a large coverage by returning even small dynamic IP ranges, while a large k is more restrictive in considering only large, actively used address blocks. A small g tends to break a large address range into small pieces, while a large g is more likely to return large blocks but may potentially result in more false positives (i.e., static IP addresses mistakenly identified as dynamic ones). For better coverage and

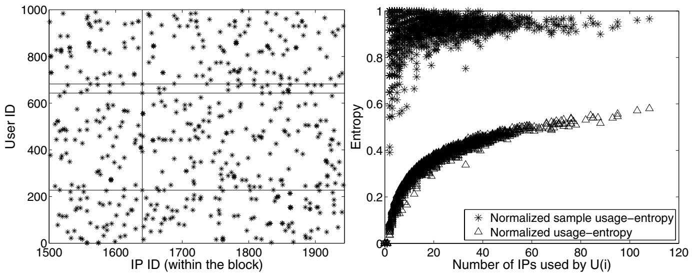


Figure 3: (a) Section of a user-IP matrix, (b) Normalized usage-entropy vs. normalized sample usage-entropy for the IPs in (a)

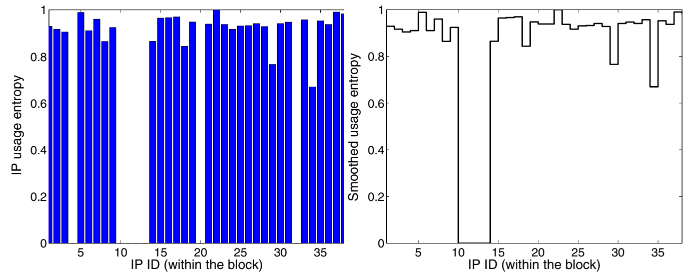


Figure 4: (a) Signal pulses for sample usage-entropy of IP addresses, (b) Smoothed signal after median filter

fewer false positives, we set both parameters to 8, which is often the minimum unit for assigning IP address ranges. We discuss the resulting coverage and block sizes further in Section 5.2 and 5.3.

4.3 IP Usage-Entropy Computation

After UDmap obtains a list of multi-user IP blocks as candidates, it needs to further distinguish between a *dynamic* IP address that had been assigned to multiple hosts (thus multiple users) and a *static* IP address linked to a single host but shared by multiple users. Users of dynamic IP addresses can be expected to log in using other IP addresses in the same block. Hence, over a period of time, a dynamic IP will not only be used by multiple users, but these users also "hop around" by using other IPs in the same block (we discuss other similar cases, such as proxies and NATs, in Section 4.5). From a practical viewpoint, dynamic IPs are often assigned through random selection from a pool of IP addresses [5], and when users "hop around", the probability of them using an IP in the pool can be expected to be roughly uniform

The IP usage-entropy computation is performed on a block-by-block basis. Let U denote the set of all users and $|U|$ the total number of users in the trace. For every multi-user IP block $B(\text{IP}_1, \text{IP}_m)$ with m IPs, we can construct a binary user-IP matrix $A \in \{0,1\}^{|U|\times m}$, where we set $A(i,j)$ to 1 if and only if user i has logged into Hotmail from IP address IP_j. Figure 3(a) shows a section of a user-IP matrix pertaining to a multi-user IP block with 2432 IP addresses.

Given the set of all users $U(j)$ who used a particular IP_j, we would like to know the probability that these users used other IP addresses in $B(\text{IP}_1, \text{IP}_m)$. To quantify the skewness of the aforementioned probability distribution, we introduce a metric, called *IP usage entropy* $H(j)$. If we form a sub-matrix $A_j^{|U(j)|\times m}$ of A that contains only the rows corresponding to users in $U(j)$ (illustrated in Figure 3(a), where UDmap selects only the rows pertaining to the highlighted IP), $H(j)$ can be computed as:

$$H(j) = -\sum_{k=1}^{m}(\frac{a_k}{z_j}log_2(\frac{a_k}{z_j}))$$

where a_k is the k-th column sum of A_j and z_j is the sum of all the entries in A_j.

Since the block size m may vary across different multi-user blocks, we define two normalized versions of the usage entropy, called *normalized usage-entropy* $H_B(j)$ and *normalized sample usage-entropy* $H_U(j)$, computed as follows:

$$H_B(j) = H(j)/log_2 m \quad (1)$$
$$H_U(j) = H(j)/log_2(|C(j)|) \quad (2)$$

Here, $H_B(j)$ quantifies whether the probability of users $U(j)$ (the set of users that used IP_j) using other IPs in the block is uniformly distributed, while $H_U(j)$ quantifies the probability skewness only across the set of IP addresses (denoted as $C(j)$) that were *actually* used by $U(j)$. In the ideal case, where IP addresses are selected randomly from the entire block, we can expect the normalized usage-entropy $H_B(j)$ of most of the IP addresses in the block to be close to 1 (over time). However, realistic traces are only of limited duration. Hence the actual observed set of IP addresses used by $U(j)$, during the trace collection period, may only be a fraction of all the IP addresses in the block, especially when the block size is large. As illustrated by Figure 3(b), due to the large block size ($m = 2432$), normalized usage-entropies $H_B(j)$ tend to be relatively small, and in this case reduce to a function of the total number of addresses $|C(j)|$ used by $U(j)$. With limited data, the normalized sample usage-entropy $H_U(j)$ is an approximation to the ideal $H_B(j)$ as $H_U(j)$ better estimates the degree of uniformity in address selection among the set of users $U(j)$. For our one-month trace, UDmap adopts $H_U(j)$ in computing IP usage-entropies. With enough observation from longer-term data, we expect $C(j) \to m$ for dynamic IP blocks, and hence $H_U(j) \to H_B(j)$.

4.4 Dynamic IP Block Identification

After UDmap computes the IP usage-entropies, one might conclude that those IPs with usage-entropies close to 1 are dynamic IP addresses. However, we emphasize that dynamic IP addresses manifest as blocks. Therefore, for each multi-user IP block, we proceed to identify *sub-blocks* of IP addresses within each multi-user IP block such that the usage-entropies of a majority of addresses in a sub-block are above a pre-specified threshold H_e.

To achieve this fine-grained segmentation, UDmap regards usage-entropy as a discrete signal $s(i)$ in the address space, where $s(i)$ can be either $H_B(i)$ or $H_U(i)$. Figure 4(a) illustrates this representation by plotting the normalized sample usage-entropies $H_U(i)$ as signal pulses. Note the time axis of the discrete signal is the same as that of the IP address space. UDmap then employs signal smoothing techniques to filter the noise that appears as small "dips" along the signal. This signal noise exists due to the fact that the corresponding IP addresses were either not used by any user, or have small usage-entropies due to insufficient usage. We use the well known median filter method for suppressing isolated out-of-range noise [4]. This method replaces every signal value with the median of its neighbors. Specifically, for each variable IP_i, the smoothed signal value $s'(i)$ is computed as:

$$s'(i) = \text{median}(\{s(\llcorner i - w/2 \lrcorner, \ldots, s(\llcorner i + w/2 \lrcorner)\})$$

where w is a parameter of the median filter that determines the neighborhood size. Since our goal of signal smoothing is to adjust the signal "dips" due to insufficient usage of a few individual IPs, UDmap applies the median filter to only those IP addresses with entropies lower than the predefined threshold H_e. Additionally, we do not apply median filtering if a signal value does not have enough neighbors (boundary conditions). In our current process, we set H_e to 0.5. As illustrated in Figure 4(a), the normalized sample usage-entropies are well separated in most cases, and thus not sensitive to H_e. We set w to 5, so that the signal smoothing process can smooth over up to 2 consecutive dips.

After applying the median filter, the identification of dynamic IP blocks is straightforward: UDmap sequentially segments the multi-user blocks into smaller segments by discarding the remaining "dips" after signal smoothing. As illustrated in Figure 4(b), the signal smoothing process "paves over" the sporadic dips in the original signal, but preserves large "valleys". In this example, UDmap will return two dynamic IP blocks.

4.5 Volatility Estimation and Proxy Removal

The final step in classifying dynamic IP address blocks is to estimate IP volatility. This step is critical, as it estimates the frequency at which host identity changes with respect to an IP address. UDmap considers two metrics for every identified dynamic IP address: (1) the number of distinct Hotmail users that have used this address in input data, and (2) the average Hotmail inter-user duration, i.e., the time interval between two different users, consecutive in time, using the same IP. Recall our input data contains timing information pertaining to the first and last time a user connected to Hotmail on a per user-IP pair basis. UDmap leverages these two features to estimate the inter-user duration.

Another important purpose of IP volatility estimation is to remove a class of potential false positive addresses. Using just the previous three steps, we expect UDmap to generate the following two classes of false positives. The first class comprises groups of proxies that employ load balancing to designate users to different servers. The second case includes Internet cafés, teaching clusters, and library machines, where a user physically logs in to any one of a group of equivalent machines.

Both cases correspond to a cluster of machines that are configured with a range of continuous static IP addresses, where a user may use any one of the machines. The difference between these two cases is that, for the first case, multiple users can *concurrently* access Hotmail through a single proxy, while in the second case, requests from different users appear *sequentially* as users can not simultaneously log on to the same machine.

The activity patterns of these two types of static server-clusters are very similar to dynamic IP blocks: they both manifest as blocks, with multiple users being associated with different IP addresses. Therefore, without additional attention, UDmap could potentially misclassify them as dynamic IPs. Note that NAT boxes with single static IP addresses do not manifest as blocks and therefore will not be misclassified.

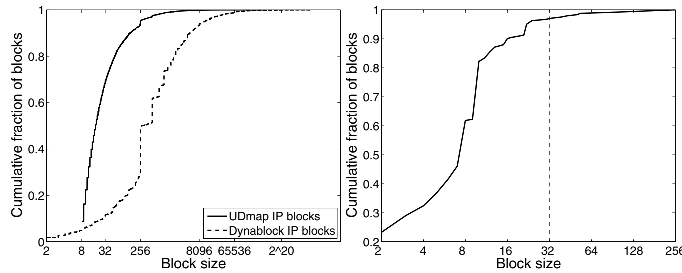
Using IP volatility estimation, UDmap can easily filter the first class of false positives by removing IPs with a large number of concurrent accesses. More specifically, UDmap discards consecutive IP addresses that are each associated with a large number of users ($\geq$ 1000) and also exhibit very short inter-user duration ($\leq$ 5 minutes). We determine the parameters by examining the user population of proxy IPs (identified through rDNS lookup with the keyword `proxy`): they corresponds to 5% false negative rate of known proxies. UDmap currently does not remove the second class of false positives. We will further discuss this topic as future work in Section 8.

5. UDMAP IPS AND VALIDATION

In this section, we present and validate the set of dynamic IP addresses output by running UDmap over our trace. For brevity, we refer to these IPs as *UDmap IPs*. We acknowledge that, given the limited duration of data collected from a single vantage point, UDmap might not be able to identify those dynamic IP addresses that were used infrequently in our data. With sufficient input data, we expect the UDmap coverage to increase over time.

	# IPs	# ASes	# Blocks
UDmap IPs	102,941,051	5,891	958,822
Proxy IPs	2,522	95	242

Table 1: IP blocks identified by UDmap



(a) UDmap IP blocks (b) Proxy IP blocks

Figure 5: Cumulative distribution of IP block sizes

5.1 Input Dataset

Our input dataset contains more than 250 million unique users and over 155 million IP addresses, spanning across 20,167 Autonomous Systems (ASes). Thus it covers a significant, actively used portion of the Internet. Furthermore, Hotmail is widely used by home users, where network connections are typically configured to use dynamic IP addresses. We assume, therefore, that our trace contains a larger fraction of dynamic IP addresses than that would be expected from either random sampling or information collected within an enterprise-network environment. Thus we believe our dataset is sufficient for a study aimed at understanding the broad scope and usage patterns of dynamic IP addresses.

5.2 UDmap IP Blocks

Out of the approximately 155 million IP addresses in input data, around 117 million were used by multiple users, based on which UDmap identified around 2 million multi-user IP blocks with a total of 169 million IPs. As shown in Table 1, using the 2 million multi-user IP blocks as candidates, UDmap returned over 102 million dynamic IP addresses and 2522 large-scale proxy IP addresses. Out of these 102 million dynamic IPs, about 95 million were in our input data. Thus more than half (61.4%) of the IP addresses observed in the trace are dynamic. Around 6.7% of the 102 million dynamic IP addresses did not appear in the trace, but were included because they were located within the address blocks returned by UDmap.

The high percentage of dynamic IP addresses in our input data suggests that dynamic IPs are a significant fraction of the address space. This implies that applications cannot readily assume that IP addresses are synonymous with host identities.

Figure 5(a) and (b) show the cumulative fraction of the UDmap IP block sizes. We observe a few instances containing very large blocks. The rest of the cases, specifically 95% of all blocks, have fewer than 256 hosts. We also plot in Figure 5(a) the CDF of the dynamic IP block sizes reported by Dynablock [7]. Despite the similarity of the two curve shapes, Dynablock IP block sizes tend to be larger, with only 50% of the blocks having fewer than 256 IP addresses.

	# blocks	% UDmap IP	% Dynablock IP
1. Identical $A_i = B_j$	220	0.11%	0.06%
2. Subset $A_i \subset B_j$	399,207	47.93%	79.71%
3. Superset $A_i \supset B_j$	452	1.60%	0.25%
4. New A_i	558,667	48.06%	0.00%
5. Missed B_j	23212	0.00%	15.30%
6. A_i, B_j partially overlap	1735	2.30%	4.69%

Table 2: Comparison of UDmap and Dynablock IP blocks.

The reason we see smaller UDmap block sizes is the sporadic usage of IPs within a large range. The infrequent usage of certain IPs forces the multi-user block selection process to split the large ranges into smaller ones. In particular, over 95% of the multi-user blocks have fewer than 256 IP addresses. A longer-term trace can be expected to contain more usage of dynamic IP addresses over a larger space and hence larger blocks.

Finally, Figure 5(b) shows the block size CDF for the identified proxy IP addresses. Most of the proxy blocks are small, with 95% of blocks having fewer than 32 hosts. Knowledge of proxy clusters can be very helpful, as proxies often need to be treated differently than normal hosts in various applications. For example, applications that rate limit host connections might prefer to choose a higher threshold for connections coming from proxies.

5.3 Validation

It is difficult to verify whether UDmap IPs are indeed dynamic ones, mainly because ISPs and system administrators consider detailed IP address properties as sensitive, proprietary information and hence do not publish or share with others. As discussed in Section 2.1, to date, the best information about dynamic IP addresses comes from two major sources: reverse DNS (rDNS) lookups and Dynablock database [7]. Both of these sources require dedicated, manual maintenance and update. Even so, they are far from being comprehensive to provide a complete list of dynamic IP addresses.

In the lack of better data sources for verifying dynamic IP addresses on a global scale, we use combined information from both rDNS and Dynablock for validation.

First, we compare UDmap IPs with the IP addresses maintained by Dynablock (referred to as *Dynablock IP*). We consider six cases when comparing the list of UDmap IP blocks $\{A_1, A_2, A_3, \ldots\}$ with the list of Dynablock IP blocks $\{B_1, B_2, B_3, \ldots\}$. Table 2 shows, for each case, the number of blocks and the corresponding percentages of IP addresses.

Case 1 (identical): The block returned by UDmap has the exact same address boundaries as a block from Dynablock. A small fraction (0.11%) of UDmap IPs fall into this case.

Case 2 (subset): The identified UDmap block is a subset of addresses from a Dynablock block, and 47.93% of UDmap IPs fall into this category. The main reason that UDmap failed to find the rest of dynamic IP addresses is their insufficient usage in our data. We find 47.6% of the missed IPs did not appear in the trace, and the rest 52.4% appeared but were used infrequently, with the average number of users per IP being 1.72.

Case 3 (superset): The UDmap IP block is larger than the corresponding Dynablock IP block. Only 1.60% of UDmap IPs fall into this category. Many UDmap IP blocks in this category are significantly larger than the corresponding Dynablock IP blocks. We suspect that these IPs beyond the Dynablock IP ranges are also dynamic ones, but not reported to Dynablock. Later in the section, we verify these IP addresses using rDNS lookups.

Case 4 (new): These are UDmap IP blocks not listed in Dynablock. These blocks consist of a large fraction of UDmap IPs (48.06%) and we also verify them through rDNS lookups.

Type	Keyword	Percentage	Total
Dynamic	dialup, modem	0.74%	34.53%
	dsl	18.75%	
	ppp	3.97%	
	cable, hsb	2.48%	
	dyn	5.14%	
	wireless	0.06%	
	pool	1.41%	
	dhcp	0.36%	
	access	1.61%	
Possibly dynamic	not found	21.21%	21.21%
Static	mail	0.0001%	1.63%
	www, web	0.28%	
	static	1.35%	
Rest	IP address	21.54%	43.53%
	unknown	21.99%	

Table 3: Random sampling based rDNS lookup results

Case 5 (missed): UDmap failed to identify any dynamic IP address from an entire Dynablock block. Only 5.78% of such missed IPs appeared in our data, with an average number of users per IP being 0.58. Hence these addresses are also used infrequently.

Case 6 (partially overlap): UDmap IP blocks and Dynablock IP blocks *partially* overlap with each other. This excludes Case 1-3. Only 2.3% of UDmap IPs belong to this case.

After comparing with the Dynablock IP list, we can verify 49.81% of the UDmap IP addresses.

For the remaining 50.19% of the UDmap IPs that are not seen by Dynablock, we verify them through rDNS lookups. Due to the large number of addresses and thus the lookup queries involved, we use two methods to sample the identified IP addresses: *random sampling* and *block-based sampling*, and we perform rDNS lookups on only the sampled addresses. The random sampling method randomly picks 1% of the remaining UDmap IP addresses that are not in Dynablock. The block-based sampling assumes that IP addresses within a same block should be of the same type. So this method picks one IP address from each UDmap block only. Based on the returned host names, we can then infer whether the looked up IP is a dynamic address by checking if the host name contains conventional keywords used for dynamic IP addresses, such as `dial-up`, `dsl`, etc [27].

Table 3 presents the rDNS lookup results using random sampling. The block-based sampling method returned similar results, and thus we do not present them due to space constraints. In total, 34.53% rDNS records contain keywords that suggest the corresponding IP addresses as dynamic. Among those, DSL constitutes a large portion, suggesting that a significant fraction of users access Hotmail through home computers via DSL links.

There are 21.21% lookups returning no rDNS records. These might also correspond to dynamic IP addresses because a static host is more likely to have been configured with a host name for it to be reachable. We do find a small fraction (1.63%) of the rDNS records contain keywords (i.e., `mail`, `server`, `www`, `web`, `static`) that suggest them as static IP addresses. For the remaining 43.53% rDNS records, we cannot infer any network properties based on their returned names. Around half of these rDNS records contain the IP addresses they are pointing to. For example: 190.50.156.163 is associated to *190-50-156-163.speedy.com.ar*.

Due to the incomplete information from both Dynablock and rDNS, we were not able to verify all UDmap IP addresses. In fact, the lack of sufficient existing information about IP dynamics further confirms the importance of an automatic method for inferring

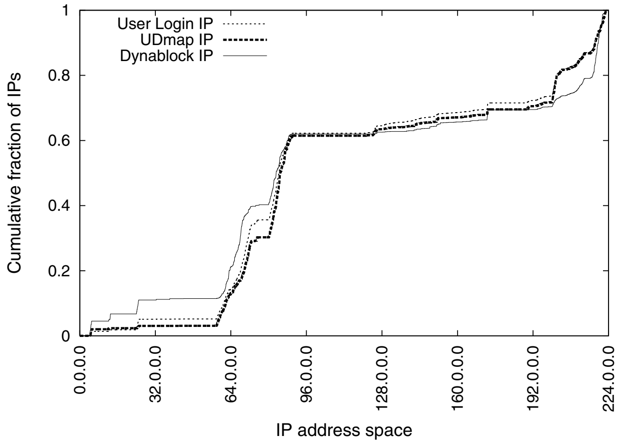


Figure 6: Cumulative distribution of IPs in address order

such properties. We emphasize that UDmap not only outputs the dynamic IP lists, but also returns the fine-grained IP volatility information – the rate at which an IP is assigned to different hosts. Applications can leverage such information to determine the corresponding host properties based on their specific application context.

6. UNDERSTANDING IP DYNAMICS

In this section, we present the detailed study of IP dynamics based on the identified 102 million UDmap IP addresses. Understanding IP dynamics has huge implications to applications that use IP addresses to represent hosts. Broadly, our study seeks to answer the following two sets of questions:

- How are dynamic IP addresses distributed across the Internet, and in particular, what address portions do they originate from and what are the top domains that have the most number of dynamic IPs?
- How *volatile* are dynamic IP addresses, and in particular, how often does the host identity change on average? What types of IP addresses are more volatile than others? Finally, how consistent is IP volatility within address blocks?

6.1 Address Distributions in the Internet

Figure 6 plots the distribution of UDmap IP addresses across the IP address space. As a comparison, we also plot the distributions of the Hotmail user-login IPs and Dynablock IPs. For all three categories, the majority of IP addresses originate from two relative small regions of the address space (58.255-88.255 and 195.128-222.255), suggesting their distributions across the IP space are far from uniform.

Overall, UDmap IPs distribute evenly across the IP space used by Hotmail users. The only notable exception appears within the small address range 72.164-75.0, where UDmap did not classify these addresses as dynamic. Whois database [30] query results indicate this region is used by Qwest (72.164/15) and Comcast (73.0/8 and 74.16/10), both of which are large ISPs in the U.S. Based on sampled rDNS lookups, certain IP addresses from Qwest have the keyword `static` in their resolved names, suggesting the ones not picked by UDmap might correspond to static IPs. In Section 6.2.3, we also present results indicating that IP addresses under Comcast are indeed not very dynamic. There are about 10% of Dynablock IPs within the address range of 4.8-58.255. Only a small fraction of these dynamic IPs were observed in our input data and hence appeared as UDmap IP addresses.

Domain	.net	.com	.edu	.arpa	.org	rest
% UDmap IP	77.35	21.20	1.14	0.13	0.12	0.06
% IP in log	70.74	26.00	2.54	0.29	0.25	0.18

Table 4: Top domains of the IP addresses

AS #	# IP ($\times 10^6$)	AS Name	Country
7132	5.378	SBC Internet Services	USA
3320	4.809	Deutsche Telecom AG	Germany
3215	4.679	France Telecom	France
4134	4.538	Chinanet-Backbone	China
19262	4.081	Verizon Internet Services	USA
3352	3.435	Telefonica-Data-Espana	Spain
209	2.431	Qwest	USA
3356	2.098	Level3 Communications.	USA
2856	1.942	BTnet UK Reg. Network	UK
8151	1.913	Uninet S.A. de. C.V.	Mexico

Table 5: Number of UDmap IPs in the top 10 ASes

In an attempt to study the domains and ASes that have the largest number of UDmap IPs, we extracted top-level domain information from the rDNS lookup results that we obtained during the validation process (see Section 5.3) [4]. As shown in Table 4, among the successfully resolved names, 77.35% are from the `.net` domain, suggesting that these IPs are owned by various ISPs. This is not surprising, given that ISPs typically offer network access to customers using dynamically assigned IP addresses through DHCP. We also notice a significant portion of the IP addresses from the `.com` domain (21.20%). Many of these `.com` host names contain keywords such as `tel` or `net` in their resolved names (e.g., `idcnet.com`, `inter-tel.com`). We manually visited several such Web sites, and confirmed that they are also consumer network ISPs. For example, IP addresses with host names ending in `idcnet.com` are owned by a wireless network provider [12]. Other than the `.net` and the `.com` domains, the percentage of UDmap IPs from other domains is very small. In particular, only 1.14% of the resolved hosts are from the `.edu` domain. For reference, we also report results pertaining to IPs in the input log, shown in the second row of Table 4. The percentage of IPs in the `.net` domain drops from 77.35% (UDmap IPs) to 70.34% (all IPs), while all other categories increase. This suggests that IPs in the `.net` domain are more likely to be dynamic, while IPs in other domains have a higher chance of being static.

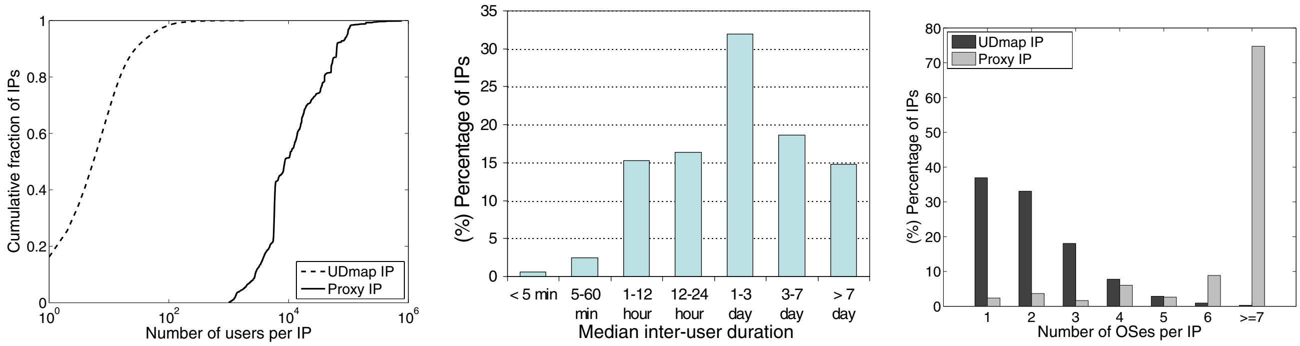
Table 5 lists the top ASes with the most number of UDmap IPs. Interestingly, we find all of the ASes correspond to large ISPs that directly offer Internet access to consumers. Out of the top 10 ASes, four are from the United States, with SBC Internet Services being the top AS with over 5 million of UDmap IPs.

Both Table 4 and Table 5 suggest that a large fraction of UDmap IP addresses are from consumer networks connecting to the Internet using DSL or dial-up links. These IP addresses are thus more likely used by home computers or small enterprise hosts.

6.2 IP Volatility Analysis

In this section, we study the volatility of UDmap IPs. We focus on the following two metrics: (1) the number of users that have used each IP in our data, (2) the median inter-user duration (we use median as opposed to mean to eliminate outliers). We begin by presenting the volatility of all UDmap IPs. We then examine the degree of similarities between IPs in a same block based on IP volatility. Finally, we use a simple, yet illustrative case study to show the impact of network access type on IP volatility.

[4] We excluded the country code before we extracted the top-level domains from host names.



(a) Number of users (b) Estimated inter-user duration (c) Estimated number of OSes

Figure 7: UDmap IP statistics computed with three different metrics on per-IP basis

6.2.1 *Volatility Per IP Address*

Figure 7(a) shows the cumulative fraction of UDmap IPs that were used by varying numbers of users. The majority of UDmap IPs were used by several to tens of users over the 31 day period. Although most of the UDmap IPs had host identity changes, they are not highly volatile. As expected, proxy IPs appear to be extremely volatile, with each having a large number of users.

Figure 7(b) shows the histogram of the average inter-user durations estimated using the procedure described in Section 4.5. We observe that the time between two consecutive users using a UDmap IP is in the order of tens of hours to several days. Over 30% of IP addresses have inter-user durations ranging between 1 and 3 days. We also notice a small set of IP addresses that were highly dynamic with inter-user durations below 5 minutes. Manual investigation of a few such hosts indicates these are likely to be highly dynamic dialup hosts, and we are investigating this further.

Recall that our input trace also contains information regarding the operating system used. Based on this information we can obtain a lower-bound on the number of actual OSes (two OSes are different if they are of different type *or* version) that have been associated with each IP. According to the histogram in Figure 7(c), most of the UDmap IPs have one or two OSes. This characteristics is strikingly different for proxy IPs, where it is very common for 7 or more different OSes to be associated with an IP address. This shows that IP volatility can help us remove proxy IPs and hence reduces false positives of the UDmap algorithm.

6.2.2 *Volatility Similarity within Blocks*

As dynamic IPs are assigned from a pool of addresses, we proceed to examine whether the addresses from the same IP block have similar volatility properties. We introduce a metric, called *dispersion factor*, to quantify the homogeneity of IP volatility across all the addresses returned in a UDmap IP block. Given a set of values $\mathbb{F} = \{v_1, v_2, \ldots, v_m\}$, the dispersion factor R is defined as

$$R = \frac{\text{90th-percentile}(\mathbb{F}) - \text{median}(\mathbb{F})}{\text{median}(\mathbb{F})}$$

The dispersion factor measures the degree of data dispersion by computing the normalized difference between the 90th-percentile value and the median (we use 90th-percentile instead of maximum to exclude outliers). A large dispersion factor suggests the 90th-percentile value significantly varies from the median and hence a large variation across the data.

We again consider the two properties reflecting IP volatility: the number of users per IP and the average inter-user duration. Figure 8(a) shows the distributions of the dispersion factors for these two properties across all the UDmap IP blocks. Overall, dispersion factors pertaining to the number of users per IP are smaller than those of inter-user durations. For the former, 73% of the blocks have dispersion factors smaller than 1, while for the latter, 33% of blocks have dispersion factors smaller than 1. This suggests that the number of users per IP tends to distribute relative evenly inside a block, while the user-switch time has a much larger variation across IPs even within the same address range.

Intuitively, one might expect small blocks to have smaller dispersion factors. We classify the UDmap IP blocks into three categories based on their sizes: small (fewer than 32 IPs), medium (32-256 IPs), and large (more than 256 IPs). Figure 8(b) and (c) show the breakdown of the dispersion factors for these three categories of blocks. For both figures, the X-axis corresponds to the dispersion factor, and the Y-axis represents the fraction of the blocks. Indeed, large blocks tend to be more diversified, particularly for the inter-user duration metric. Homogeneous blocks with dispersion factors smaller than 0.1 are almost exclusively small blocks.

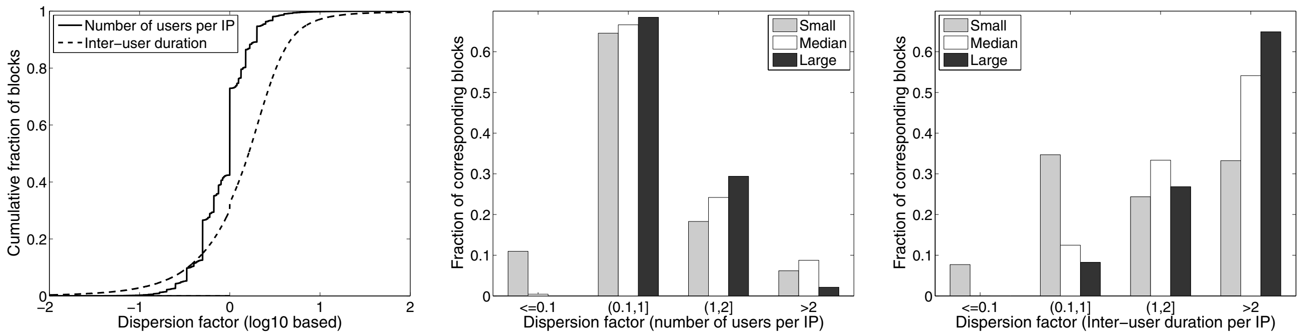
Our volatility analysis suggests that IPs within a block are approximately used by equal number of users. The average inter-user duration varies within blocks, and small blocks tend to be more homogeneous in term of IP volatility.

6.2.3 *IP Volatility and Network Access Type*

In Section 6.2.1, we showed that certain UDmap IP addresses are more dynamic than others. It is often hypothesized that dial-up IP addresses are more dynamic, since every dial-up might return a new address. Similarly, anecdotal evidence suggest cable modem hosts do not change IP addresses frequently. In this section, we present a case study to characterize the inter-user durations with respect to various network access types.

We selected thee known IP blocks that are representative of various network access types: Bell Canada dial-up (206.172.80.0/24), SBC DSL (209.30.56.0/22), and Comcast cable (24.10.128.0/16). UDmap successfully identified the majority of the addresses in the trace for Bell Canada and SBC DSL. However when it came to Comcast cable, UDmap picked 1076 IPs out of the 19512 present in the input trace, perhaps due to the fact that IP addresses from Comcast are generally less dynamic [2].

Figure 9 plots the inter-user duration associated with all the IP addresses that pertain to the three blocks (instead of only those identified by UDmap). If an IP was used by only a single user during the entire month, we set its inter-user duration to 31 days. We have the following observations: (1) Bell Canada dial-up block is much more dynamic than the other two blocks; the majority of the observed inter-user durations are in the order of hours. (2) SBC DSL block also displays dynamic behavior, with inter-user duration



(a) CDF of R across blocks (b) R for number of users (c) R for inter-user duration

Figure 8: Distribution of dispersion factors across UDmap IP blocks

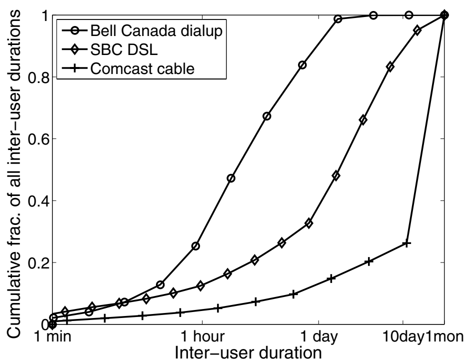


Figure 9: Distribution of inter-user durations

being 1 to 3 days. (3) In contrast, the Comcast IP block is relatively static; over 70% observed IPs did not change user within the entire month.

The distinct IP volatility of these three different blocks suggests it might be possible to classify the type of network access links based on IP volatility. It is an interesting area of research to further understand the correlations between IP volatility and network access type.

7. IP DYNAMICS AND SPAM DETECTION

The motivating example in Section 3 illustrates how knowledge of IP dynamics might help detect spamming email servers from a specific university network. In this section, we systematically investigate the general applicability of using dynamic IP address information for spam detection. We use a three-month long email server log from Hotmail to facilitate our study.

7.1 Data Description

Our Hotmail email server log was collected between June and early September of 2006 (3 months). It contains a record of all incoming SMTP connections aggregated on a daily basis (one entry per sending IP per day). Each entry includes a coarse-grained timestamp, the IP address of the remote email server, and the number of email messages received. In addition, Hotmail applies content-based and history-based spam filtering schemes on received email messages and records the number of spam emails detected by the filter. The spam filter is configured to detect spam with low false positive rates, but there still might be spam emails that slip through the radar. For these false negatives, if a user reports them as spam, Hotmail logs them in a user feedback database.

7.2 Incoming Email Server IP Addresses

Using both Dynablock and UDmap IPs, we classify the remote email server IPs into two categories: (1) *identified dynamic* if it belongs to either Dynablock IPs or UDmap IPs, and (2) *likely static* otherwise. As we will show later in Section 7.3, most of the legitimate email servers are indeed *likely static* servers. Figure 10 plots their IP address distributions in the address space. Despite the difference in their observed dynamics, the two categories of addresses come from roughly the same two regions of address space. This suggests these regions of addresses are used more actively than others in general. Therefore, address space location alone, cannot effectively discriminate a legitimate server from a spam server.

Existing spam filtering techniques use IP address history as a filtering criterium [28]. Recent work [23] showed that most zombie hosts send spam only once. Since hosts using dynamic IP addresses are attractive targets for attackers, the volatility of IP addresses that send email should be a useful metric. Figure 11(a) shows the frequency in terms of the number of days these different categories of IPs appeared in the log. The majority of the *identified dynamic* IP based email servers have very short histories: 55.1% of them appeared only once in the three-month period; only 1% of them appeared more than ten times. As a comparison, 22% of the IPs classified as *likely static* (those not listed by UDmap or in Dynablock) appeared in the log for more than ten days. For those IPs that sent emails only once, there was no history to help determine the likelihood of being a spammer. Even for those reoccurring dynamic IP addresses, history is not helpful, exactly because the host identities might have already changed. In this case, the knowledge of whether a host is set up with a dynamic IP is helpful in determining whether spam filters can leverage its sending history.

7.3 Spam from Dynamic IP Addresses

Although most of the *identified dynamic* email servers sent emails to Hotmail only once during the course of three month, the aggregated volume of spam from these servers is large. Table 6 shows that about 92% of the emails from UDmap IPs and Dynablock IPs are spam, accounting for up to 50.7% of the total spam captured by Hotmail and 49.2% of the user reported spam. We observe that although Dynablock IP list contains more addresses than UDmap IPs, there are fewer Dynablock IPs *actually* used to set up mail servers. Consequently, the total spam volume from Dynablock IPs is also lower (15.8 billion as opposed to 24.1 billion from UDmap IPs). This echoes the importance of an automatic method for keeping track of most up-to-date, popularly used dynamic IPs.

Given the overall high percentage of spam from dynamic IP addresses, a question we ask is whether spam originates from just a few hosts. Figure 11(b) shows that a large fraction of mail servers

	# of IPs	# of IPs used by mail servers	% of emails classified as spam	% of all Hotmail classified spam	% of user-reported spam
UDmap IP	102,941,051	24,115,951	92.4%	42.2%	40.3%
Dynablock IP	193,808,955	15,773,646	92.3%	30.4%	29.3%
UDmap IP $\bigcup$ Dynablock IP	242,248,012	27,163,219	92.2%	50.7%	49.2%

Table 6: Spam sent from UDmap IPs and Dynablock IPs

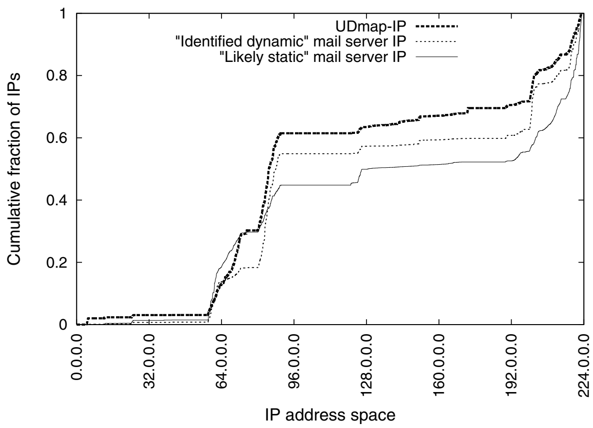
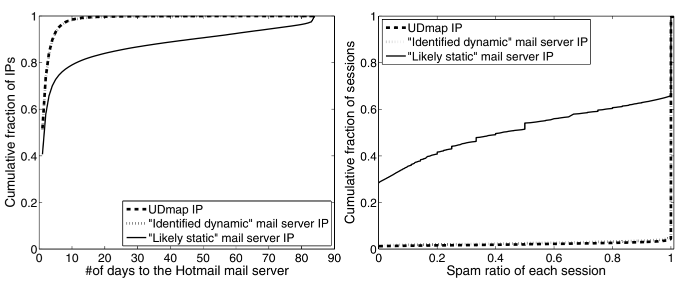


Figure 10: Distribution of email server IPs



(a) (b)

Figure 11: (a) Number of days an IP was used as a mail server to send emails, (b) Spam ratio per session. In both figures, the UDmap IP curve overlaps with the "identified dynamic" mail server IP curve.

with UDmap or Dynablock IPs sent spam emails *only*. The X-axis corresponds to the *spam ratio*, computed as the percentage of spam over the number of mail messages received from per IP per day (referred to as a *session*). The Y-axis is the cumulative fraction of the sessions. Based on the classification results using the existing Hotmail spam filter, 95.6% of the sessions from UDmap IPs sent spam only (spam ratio = 100%), 97.0% of them sent emails with over 90% spam ratio. The remaining 3% can potentially be legitimate mail servers. We note here, however, the 3% is an upper bound of our spammer detection false positive rate because the existing spam filter might miss spam emails. In contrast, there is a much smaller fraction of sessions from the *likely static* IP addresses with a high spam ratio: 31.4% of the sessions sent only spam, and 62.8% of the sessions had spam ratio lower than 90%. More importantly, using the knowledge of dynamic IP addresses, we can filter 40.3% of the undetected spam emails – those slipped through the existing spam filter, but subsequently reported by users as spam (last column of Table 6). Thus we expect using UDmap IP can further reduce the spam filtering false negatives.

We also studied the ASes that sent the most spam emails to Hotmail and the results are presented in Table 7. Notice that the top spamming ASes are spread out across the globe. This finding differs from the results reported by Ramachandran et al. [23], which showed that about 40% of spam originates from the U.S. A possible explanation is that Hotmail's global user presence attracts a broader range of spamming IP addresses worldwide. The third and fourth columns of the Table 7 present results pertaining spamming behavior of dynamic IPs in these top ASes. In particular, the third column indicates that, for majority of the top ASes, over 50% of their outgoing spam emails originate from dynamic IP ranges. This suggests that spam from dynamic IP addresses is prevalent across large, active consumer ASes. The fourth column delivers an even stronger message: the overwhelmingly high spam ratios from these (dynamic IP based) spam sources is highly indicative that a large fraction of them are compromised zombie hosts exploited by the spammers.

As evidenced by the strong correlation between spammers and the *dynamic* portion of the Internet, the knowledge of dynamic IP addresses and their usage patterns has great potential to help combating spam. We believe systematically investigating how to incorporate the knowledge of IP dynamics into existing spam detection frameworks is a future research direction of critical importance.

8. DISCUSSION AND FUTURE WORK

The results in Section 7 provide evidence that IP dynamics can be a successful weapon in the fight against email spam. Yet there is room for improvement: it could be the case that legitimate mail servers are set up using dynamic IP addresses coming from DSL or cable modem networks. We expect those cases to exhibit distinctive email-sending patterns and are currently looking at several possibilities as ongoing work.

As discussed in Section 4.5, UDmap might misclassify certain teaching clusters (i.e., labs in universities) and library machines as dynamic IPs. However these machines are typically in the `.edu` domain, and based on our verification results, they form a relatively small population (see Table 4). In order to classify these machines correctly, one can provide additional information to UDmap – for example, we can augment our framework to include information such as OS ID and device fingerprinting information [14] to more precisely characterize IPs.

The length of the input trace might also impact the quality of results, and we expect that longer traces will lead to better coverage. A thorough analysis of the relationship between length of trace (duration) and dynamics of IP addresses is an interesting problem and deserves attention.

From a security standpoint, spammers might wish to thwart the effectiveness of UDmap by making static IP appear dynamic, or to evade detection by making dynamic zombie IPs appear as static ones. UDmap is robust to such attacks. One cannot let a static IP address appear to be dynamic without controlling a range of consecutive IPs. On the other hand, it is even harder to make a dynamic IP address appear static because one cannot prevent others from observing the dynamic behavior of its neighboring IPs.

AS #	# of spam emails	% of spam from UDmapIP	Spam ratio of UDmapIP	AS Name	Country
4134	6,349,330,892	52.92%	93.21%	Chinanet-backbone	China
4837	5,259,034,812	42.90%	93.20%	China169-backbone	China
4776	4,422,195,227	26.57%	98.70%	APNIC ASN block	Australia
27699	2,359,727,485	95.61%	91.53%	TELECOM DE SAO PAULO	Brazil
3352	2,336,700,524	84.58%	96.28%	Telefonica-Data-Espana	Spain
5617	2,234,104,550	0.54%	97.15%	TPNET	Poland
19262	2,073,172,523	79.60%	96.19%	Verizon Internet services	USA
3462	1,922,291,974	86.31%	93.22%	HINET	Taiwan
3269	1,802,531,410	88.16%	95.52%	TELECOM ITALIA	Italy
9121	1,760,38,6582	89.96%	97.78%	Turk Telekom	Turkey

Table 7: Top 10 ASes that sent the most spam

9. CONCLUSIONS

We presented UDmap, a simple, yet powerful method to automatically uncover dynamic IP addresses and related IP volatility information. Using Hotmail user-login data, UDmap identified around 102 million dynamic IP addresses spanning across 5891 ASes, indicating that the fraction of dynamic IP addresses in the Internet is significant. Our detailed, large-scale IP dynamics study showed that majority of the identified IP addresses are owned by various consumer network ISPs, and hence are likely used by home user computers or small enterprise hosts. Our findings also indicate that IP volatility exhibits a large variation, ranging from several hours to several days.

We applied IP dynamics information to spam filtering as an example application. Using a three-month long Hotmail email server log, our trace-based study showed that over 95.6% of the mail servers set up using dynamic IP addresses sent out only spam, with the total spam volume being 42.2% of all spam received by Hotmail during the trace period. We view this as a significant and important result with wide implications to the field of spam detection.

10. REFERENCES

[1] Multi-DNSBL Lookup. `http://www.completewhois.com/rbl_lookup.htm`.

[2] Braunson. Guide To Change Your IP Address (Part 2). `http://totaldream.org/index.php?page=articles&view=article&id=101`, 2006.

[3] M. Casado and M. J. Freedman. Peering through the Shroud: The Effect of Edge Opacity on IP-based Client Identification. In *Proc. 4th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[4] K. R. Castleman. Digital Image Processing. *New Jersey: Prentice Hall*, 1996.

[5] Cisco Network Registrar User's Guide. `http://www.cisco.com/en/US/products/sw/netmgtsw/ps1982/products_user_guide_list.html`.

[6] R. Droms. Dynamic Host Configuration Protocol. RFC 2131: `http://www.dhcp.org`, 1997.

[7] Dynablock Dynamic IP list. `http://www.njabl.org`, recently aquired by Spamhaus, `http://www.spamhaus.org/pbl/index.lasso`, 2007.

[8] J. Evers. Most Spam Still Coming From the U.S. `http://news.com/Most+spam+still+coming+from+the+U.S./2100-1029_3-6030758.html`, 2006.

[9] S. Foo, S. C. Hui, S. W. Yip, and Y. He. Approaches for Resolving Dynamic IP Addressing. *Internet Research: Electronic Networking Applications and Policy*, 7(3):208–216, 1997.

[10] M. Freedman, M. Vutukuru, N. Feamster, and H. Balakrishnan. Geographic Locality of IP Prefixes. In *Proc. of the ACM Internet Measurement Conference (IMC)*, 2005.

[11] J. Hovold. Naive Bayes Spam Filtering Using Word Position Attributes. In *Conference on Email and Anti-Spam*, 2005.

[12] IDC Netwurx. `http://www.idcnet.com`, 2006.

[13] J. Jung and E. Sit. An Empirical Study of Spam Traffic and the Use of DNS Black Lists. In *Proc. of the ACM Internet Measurement Conference (IMC)*, 2004.

[14] T. Kohno, A. Broido, and K. Claffy. Remote Physical Device Fingerprinting. In *IEEE Symposium on Security and Privacy*, 2005.

[15] B. Krishnamurthy and J. Wang. On Network-Aware Clustering of Web Clients. In *Proc. of Sigcomm*, 2000.

[16] H. Lee and A. Y. Ng. Spam Deobfuscation Using a Hidden Markov Model. In *Conference on Email and Anti-Spam*, 2005.

[17] F. Li and M. H. Hsieh. An Empirical Study of Clustering Behavior of Spammers and Group-based Anti-Spam Strategies. In *Conference on Email and Anti-Spam*, 2006.

[18] D. Lowd and C. Meek. Good Word Attacks on Statistical Spam Filters. In *Conference on Email and Anti-Spam*, 2005.

[19] D. Majoras, T. B. Leary, P. J. Harbour, and J. Leibowitz. Effectiveness and Enforcement of the CAN-SPAM Act: A Report to Congress. `http://www.ftc.gov/bcp/conline/edcams/spam/reports.htm`, 2005.

[20] V. N. Padmanabhan and L. Subramanian. An Investigation of Geographic Mapping Techniques for Internet Hosts. In *Proc. of Sigcomm*, 2001.

[21] Postini Message Security and Management Update for October Reveals that Spam is Back with a Vengeance. `http://postini.com/news_events/pr/pr110606.php`, 2006.

[22] A. Ramachandran, D. Dagon, and N. Feamster. Can DNSBased Blacklists Keep Up with Bots? In *Conference on Email and Anti-Spam*, 2006.

[23] A. Ramachandran and N. Feamster. Understanding the Network-Level Behavior of Spammers. In *Proc. of Sigcomm*, 2006.

[24] A. Ramachandran, N. Feamster, and D. Dagon. Revealing Botnet Membership Using DNSBL Counter-Intelligence. In *2nd Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI)*, 2006.

[25] Route Views Project. `http://www.routeviews.org`.

[26] V. Sekar, Y. Xie, M. K. Reiter, and H. Zhang. A Multi-Resolution Approach for Worm Detection and Containment. In *DSN*, 2006.

[27] M. Sullivan and L. Munoz. Suggested Generic DNS Naming Schemes for Large Networks and Unassigned Hosts. RFC draft: `http://tools.ietf.org/wg/dnsop/draft-msullivan-dnsop-generic-naming-schemes-00.txt`, 2006.

[28] The Apache SpamAssassin Project. `http://spamassassin.apache.org`.

[29] Trend Micro Inc, Mail Abuse Prevention System. `http://www.trendmicro.com/en/products/global/kelkea.htm`.

[30] Whois.net – Domain Research Tools. `http://www.whois.net`.

[31] M. Xie, H. Yin, and H. Wang. An Effective Defense Against Email Spam Laundering. In *Proc. of the ACM Computer and Communications Security (CCS)*, 2006.

[32] Y. Xie, V. Sekar, D. Maltz, M. Reiter, and H. Zhang. Worm Origin Identification Using Random Moonwalks. In *Proc. of the IEEE Symposium on Security and Privacy*, 2005.