How Dynamic is the ISPs Address Space? Towards Internet-Wide DHCP Churn Estimation

Giovane C. M. Moura, Carlos Gañán, Qasim Lone, Payam Poursaied, Hadi Asghari, and Michel van Eeten
Delft University of Technology
Email: g.c.moreiramoura@tudelft.nl

***Abstract*—IP address counts are typically used as a surrogate metric for the number of hosts in a network, as in the case of ISP rankings based on botnet infected addresses. However, due to effects of dynamic IP address allocation, such counts tend to overestimate the number of hosts, sometimes by an order of magnitude. In the literature, the rate at which hosts change IP addresses is referred to as DHCP churn. Churn rates vary significantly within and among ISP networks, and such variation poses a challenge to any research that relies upon IP addresses as a metric. We present the first attempt towards estimating ISP and Internet-wide DHCP churn rates, in order to better understand the relation between IP addresses and hosts, as well as allow us to correct data relying on IP addresses as a surrogate metric. We propose an scalable active measurement methodology and then validate it using ground truth data from a medium-sized ISP. Next, we build a statistical model to estimate DHCP churn rates and validate against the ground truth data of the same ISP, estimating correctly 72.3% of DHCP churn rates. Finally, we apply our measurement methodology to four major ISPs, triangulate the results to another Internet census, and discuss the next steps to more precisely estimate DHCP churn rates.**

I. Introduction

There have been a number of measurement studies into the usage of the IPv4 addressing space, mostly focusing on the degree in which allocated address space is actually used [1], [2], on quantifying statically versus dynamically managed address space [3] and, to a lesser extent, on the duration of use of addresses [4]. Relatively little work has been done on measuring the relationship between addresses and hosts [5]–[8], especially for large-scale, dynamically-managed networks of Internet Services Providers (ISPs).

The differences among networks are substantial, and such a variation within and among ISPs poses a challenge to any research that, in lack of a more precise solution, relies upon IP addresses as a surrogate for the unique identifiers of hosts. This is the case for much research in Internet security. Take as example the ranking and comparison of Autonomous Systems (ASes) by total numbers of infected IPs (part of botnets [9]): it is well-known fact that the number of IP addresses does not reliably correspond to the number of infected hosts [10], due to differences in the rates at which hosts change IP addresses (commonly referred to as DHCP churn rates). By hijacking the Torpig botnet for 10 days, for example, Stone-Gross *et al.* [11] showed how on average, Germany bots had 1.3 IPs/day, while U.S. bots had 0.18, which compromise the reliability of rankings based on such IP counts. After 10 days, the count of IP addresses was overestimating the number of infected hosts by more than one order of magnitude.

ISBN 978-3-901882-68-5 © 2015 IFIP

This paper provides the first attempt to estimate ISP-wide churn rates. This can be used to understand the relation between IP addresses and hosts, and ultimately normalize any metric that relies on counting addresses, such as bot counts in ISP networks, besides providing valuable insight on how ISPs manage and utilize their IP address space.

Currently, there is no authoritative way to estimate churn rates across multiple ISPs. We present a scalable, active measurement methodology based on [4] and employ it to measure the dynamics of all prefixes of several Autonomous Systems (ASes). Even though session durations have been previously measured for random prefixes [4], it is unclear to which degree such active measurement-based methods are capable to capture the dynamics of all addresses allocated to different ISPs or ASes. We assess the precision of our methodology by comparing our measurements against ground truth data from a mid-size ISP ($\sim$ 1 million addresses).

We make the following contributions: (*i*) we present a scalable measurement methodology to measure session times of all active IPs within an AS (Section II), and (*ii*) we assess its precision by comparing to ground-truth data from a medium-sized ISP (Section III); we then (*iii*) develop a statistical model (Section IV) to estimate the number of different users behind IP addresses (DHCP churn rates) over the monitoring period and validate it. Next, (*iv*) we apply the methodology to ASes of four large ISPs and show how their IP usage, visibility, session duration and inactivity varies (Section V). We triangulate these results against the measurements made with the Carnabotnet [12], [13]. Related work is presented in Section VI and conclusions and future work in Section VII.

II. Measurement Methodology

To understand why there is a large DHCP churn variation among and within ISPs, we have first to understand the relation between ISPs and IP addresses. To connect customers, ISPs are *allocated* with network prefixes [14] by their respective Regional Internet Registrar (RIR) [15], which in turn, receive those prefixes from the Internet Assigned Numbers Authority (IANA). ISPs then advertise their prefixes to other ISPs usually employing the Border Gateway Protocol (BGP) [16].

Various technologies can be deployed by ISPs to assign IP addresses to hosts. These include Dynamic Host Configuration Protocol (DHCP) [17] servers, Remote Authentication Dial In User Service (RADIUS) servers [18], IP pools managed by Broadband Remote Access Servers (BRAS) using Point-to-Point Protocol (PPP) [19], among others. Due to space constraints, we do not delve into the specifics of these protocols, and instead look at IP assignment in generic terms.

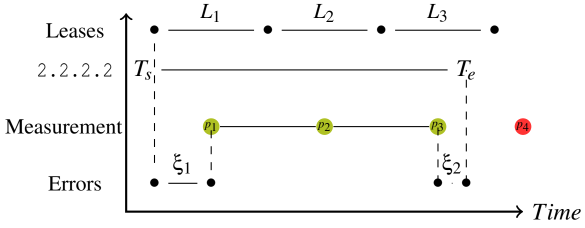

Fig. 1. Measuring Session Durations

ISPs, in turn, have freedom to decide the management policies of their pool of addresses. They may configure their DHCP/RADIUS servers with a large variation in their parameter values, such as the size of the IP address pool and default lease time (the time an address is assigned to a client) [20]. This, in turn, may lead to sub-prefixes exhibiting different usage patterns from one another [7]: business customers are likely to differ from a wireless hot-spots prefixes and home DSL block in terms of session times duration and prefix usage, which are more likely to be managed at /24 [3]. We also expect the usage patterns to vary when comparing different ISPs.

A. IP Address Assignment

ISPs assign either static or dynamic IP addresses to clients for a certain period of time (*lease time*), which can also be extended upon request issued by the client. Figure 1 shows an example in which a device has been assigned with the address `2.2.2.2` at T_s, having L_1 as initial lease with default lease time. This lease was then renewed twice (L_2, L_3). The device disconnects at T_e. The *session duration* of a device in a network is a function of the default lease time (*Lease*, in time units):

$$Session(Lease) = n \times Lease - (Ln_{end} - T_e) \qquad (1)$$

in which n is the number of leases ($n \in \mathbb{N} | n \in [0 \ldots \infty]$) and Ln_{end} is the instant at which the last lease ends. Note that this might differ since DHCP, for example, does not mandate a client to inform a server when it disconnects [17], therefore active leases might be allocated to offline devices. Moreover, choosing optimal default lease time for is far from being an obvious task; short leases times lead to high volume of broadcast traffic, while long lease times can lead to exhaustion on the address pool space [7]. Ultimately, the session duration of a device therefore is not only influenced by the way IP pools and leases are configured, but also by human behavior (e.g., users deciding when to connect or disconnect from the network) as well as external-factors, such as network failures and power outages [21].

B. Method and Metrics

We employ active measurements to estimate online session duration of clients. Figure 1 summarizes the relationship between our probing method. A random device uses the IP `2.2.2.2` for the time interval $T_e - T_s$. To actively estimate this session, we send four periodical probes (p_n). In this example, three probes were successfully replied, indicating the device was active and reachable, while p_4 did not succeed, to which we assume the device disconnected from the network.

We define *measured session duration* of an IP address to a device as the interpolation of the timestamps of continuously acknowledged probes (ACK). For `2.2.2.2` this is the time difference between the timestamps of the ACK messages of p_3 and p_1 (we disregard the time interval between the time the packet is sent and received). The measured session is an approximation of the *actual session duration*, which, for the same figure, is $T_e - T_s$.

Whenever a host disconnects, its former IP address might be reassigned to a different user. As a consequence, an IP address may have multiple users over a measured period of time. In our method, we also calculate the *number of sessions* each IP has been assigned. Finally, based on the same method, we can also calculate for each IP the *time in between sessions*, i.e., how long it takes for an IP address to be re-assigned.

Finally, we compute the *number of distinct sessions* for each probed IP address. As shown in Figure 1, an IP address can be either in the state online or offline. Distinct sessions therefore refer to the number of continuous online sessions an IP has exhibited.

C. Probe Design and Measurement Setup

The two main requirements for our probing design is to be *ISP-independent* and *scalable*. In this sense, we employ active probing as a measurement technique to be ISP-independent and employ `ZMap` [22], a high-performance network scanner to achieve scalability. Additionally, the design should minimize traffic footprint and respect user's privacy, i.e., collect the minimum information necessary about the probed systems. Next we present our choices for the probing design:

Measurement Protocol: Several protocols can be used to probe the state of an IP address (probes p_n in Figure 1). We choose to use ICMP [23] echo request/reply messages (types 8 and 1) over TCP and UDP since ICMP has proved to be less firewalled, generated less abuse messages (and usually considered "benign traffic"), and be more accurate than TCP and UDP [4], [12]. ICMP also generates a smaller traffic footprint, and better respects user's privacy, since no information other than system status is obtained.

Number of Probes: As discussed in [21], "one ping is not enough". Whenever an ICMP packet reaches a router that does not know the MAC address of the destination, the ARP RFC [24] states that the router should drop the packet and then, send a ARP request instead, impacting our results. Therefore, we choose to send two probes per IP per measurement instead. More probes could possibly lead to more accurate results, at expenses of increased traffic footprint.

Measurement Tool: Standard Linux measurement tools, such as `nmap`, `ping`, and `hping3` can be used as probing tool in our design. However, none of these aforementioned tools is designed with scalability as a main requirement. Therefore, we employ `ZMap` [22], an open-source network scanner. Besides being more scalable, `ZMap` outperforms `nmap` in accuracy, since it has a higher connection timeout when waiting for echo reply messages. In our measurements, we could easily probe more than 400K IP addresses per second, using one single computer.

Frequency of Measurements: The frequency of the measurements plays a crucial role in network measurements.

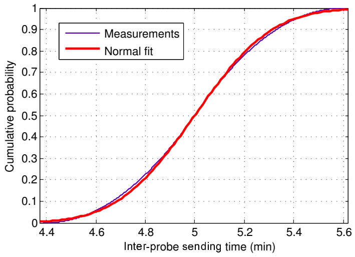

Fig. 2. CDF of the inter-probes sending interval

One common sampling scheme is to send the probe packets separated by a fixed sampling interval. However, using a uniform sampling interval the probes might not capture the true system behavior. Due to DCHP polices and user behavior, there is a possibility that periodic samples may be synchronized with a periodicity in the system under observation. Moreover, commonly used uniform sampling misses high-frequency components and causes aliasing in low-frequency components. Some sampling problems can occur where the samples and system periodicities are not synchronized.

Random sampling is an important step towards more accurate network measurements [25]. It has long been recognized that one way to overcome aliasing in sampling is to sample at random intervals rather than at uniform intervals. Therefore, our approach is based on random additive sampling: samples are separated by independent, randomly generated intervals that have a common statistical distribution $G(t)$. $G(t)$ is defined by ZMap randomization algorithm. ZMap selects addresses using a random permutation of a cyclic multiplicative group of integers modulo a prime and generating a new primitive root (generator) for each scan.

To verify this, we have carried 144 measurements (1 every 5 minutes) over 1 million IP addresses and analyzed $G(t)$, and obtained the timestamps from the outgoing IP packets from the pcap files. Figure 2 shows the empirical cumulative distribution function of $G(t)$ compared to a normal distribution. As expected, ZMap sends the probes randomly according to a normal distribution with mean equal to the sending interval, i.e., $\mathcal{N}(5,0.06)$ in this specific case. Thus, by using ZMap we achieve a Non-Uniform Probabilistic sampling strategy avoiding phase-lock problems while being non-intrusive [25]. We have determined empirically the most suitable interval in between each measurements in Section III-A.Thus, we run the scans every 10 minutes with an average inter-probes sending interval equal to $\overline{G(t)}$.

It is important to emphasize the difference between our work and [4]. Contrarily to theirs, our probe selection method is random and does not have bias to active portions of the address space. Moreover, we probe entire IP address spaces of ISPs, while they use sampling instead (24K /24 prefixes, 9,200 probes/s), at more than 400K probes/s. In addition, their interval between measurements is 11 minutes while we employ 10 minutes.

Measurement Setup: Our probing setup was configured in a Ubuntu 12.04 Server edition, in a Kernel-based Virtual Machine (KVM), with 6 3.3GHz Xeon cores and 8GB of RAM. The measurements were originated from the network of Delft University of Technology (TU Delft, AS 1128), which has SURFNet (AS 1103) as upstream provider. In this setup, the most demanded resource is CPU power – our average network throughput was $\sim$ 25Mpbs on a 1Gbps line. We found that the version of ZMap we used did not guarantee packets transmission[1], and had to run it with only three threads to avoid packets being dropped on our side. We probed and logged the IP address and the timestamp of the corresponding ICMP echo response (SRC_IP, timestamp).

D. Limitations

As any active measurement approach, ours also has its own limitations:

Visible IP Addresses: As discussed in [4], any active probing method can only account for the "visible" part of pool of probed addresses. Many online hosts are expected to be located behind network/application firewalls, network address translators (NAT) which may block all probes destined to a certain network. Moreover, when not behind network firewalls, hosts/customer-premises equipment (CPE) may have their own firewall, and block probes. To cope with that, we validate our method against the ground truth of a mid-size ISP and compare the results for the other larger ASes by using the datasets of the Carna Botnet.

Transient Errors and α threshold: Packet losses due to network failures, limiting-rate network firewalls, intrusion detection and preventions (IDPS) systems, may also lead to incorrect measurements. In Figure 1, if probe p_2 would have been lost, the ACK message related to p_2 would not be received, and therefore there would be two sessions, $p_1 - p_1$ and $p_3 - p_3$, instead of $p_3 - p_1$. To cope with errors incurred by transient failures, we introduce a tolerance threshold α. This threshold defines how much longer (in seconds) the algorithm should wait before considering a host offline whenever a probe is not acknowledged. By definition, the algorithm waits for the fixed period of measurements $(1/f)$. We added α seconds to this period in order to cope with such errors.

Sampling Errors: Our measurements are subject to random sampling errors. Uncertainties associated with the divergence due to sampling errors are generally small compared to the average measured magnitude. For example, a measurement may start after a session has been initiated on the DHCP server, and therefore, not measure it $(p_1 - T_s)$ in Fig. 1. Similarly, it may miss the ending of a session, which leads to other errors – $(T_e - p_3)$ and $(p_4 - \xi_2)$. To mitigate such errors, we weight each timestamp with a uniform distribution of mean $1/f$ as in [26].

III. Validation

Any active measurement method requires its precision to be verified. In our case, it requires us to rely on sources that have ground truth data on the session durations. We collaborated with Shatel [27], a mid-size ISP with approximately one million IP addresses. Shatel is the largest privately held broadband service provider in Iran providing a range of services, mostly based on DSL technologies. We carried out the measurements and provided the ISP staff with the results from ZMap; they then compared this against their customer IP log files, and

[1] See https://github.com/ZMap/ZMap/issues/136

Shatel DHCP logs - Sessions Duration (h)

	m-0	m-600	%-m0	%-m600
5min	29,560,569.95	33,071,437.91	58.58%	65.54%
10min	29,248,506.23	31,594,275.51	57.97%	62.61%
20min	28,630,164.97	28,630,164.97	56.74%	56.74%
30min	28,233,964.92	28,233,964.92	55.95%	62.95%

Shatel DHCP - Sessions

	m-0	m-600	%-m0	%-m600
5min	26,536,848	41,899,400	226.62%	357.29%
10min	15,192,248	8,143,872	129,55%	69.44%
20min	9,324,855	9,324,855	79.51%	79.51%
30min	7,179,625	7,179,62	61.22%	61.22%

TABLE I. RESULTS OF INTERVAL IN BETWEEN MEASUREMENTS

provided us aggregated information on the results. For privacy reasons, we were not given access to the session logs directly, and data processing was performed at their servers.

A. Probing Interval

To determine the probing interval, i.e., the time in between two consecutive measurements (p_1 and p_2 in Fig. 1), we probed the IPs announced by Shatel (obtained from BGP feeds from RIPE Routeviews [28], 1,081,344 unique IPv4 addresses), twice every IP every 5 minutes, for one week (May 22–28, 2014), using the methodology described in Section II.

We then generated a file that reconstructs the DHCP sessions of the IP addresses (m-0) and another file (m-600), by adding a offset α of 600s (threshold parameter Section II-D), to cope with possible transient/network errors. Then, we vary the probing rate and observe how the accuracy of the results changes. To that end, we process `ZMap` output files for 10, 20, and 30 minutes probing interval.

Table I shows the results for each probing interval with the DHCP log files. In this table, m-0 refers to either sum of session's duration or sum of sessions that the m-0 leases file has with the ground truth (DHCP logs), while % refers to the ratio between it and the ground truth. These summations, in fact, express how much our measurement was capable to correctly measure the connection status of the entire AS.

As can be seen, short probing intervals (i.e., < 5m) lead to underestimate the number of session (error $\sim$35%, plus overestimating the total number of sessions in more than 200%), while larger intervals ($>$10m) increase only marginally the precision in terms of session hours. Thus, there are trade-offs among the measurement accuracy, the probe rate, and the overhead on the network. Increasing the probe rate beyond 10 minutes might lead to the situation that the probes themselves skew the results.We therefore choose 10 minutes intervals and use it in the remainder of this paper.

B. Visibility and Usage of Addresses and Prefixes

After determining the probing interval, we employ a measurement dataset that we have generated for 2 continuous weeks (March 22nd – April 5th, 2014). That lead to a file having a total 14,533,525 measured sessions (m-0), and m-600 with 8,215,301 measured sessions (m-600).

Before evaluating the precision of our method, we first need to determine whether our ICMP-based method is able to obtain response from a significant part of addresses allocated.

Shatel DHCP Session Logs

	# IP addresses
Measurements	714,139
DHCP Session Logs	752,098
Measurements ∩ DHCP Logs	709,586 (94.95%)
Only DHCP Session Logs	42,510
Only Measurement	4,551

TABLE II. VALIDATION DATASETS

We have sent 4,641,128,448 probes (two probes per IP, per measurement), to which 356,805,959 IPs (non-unique, total) responded. In average, 166,266 of the $\sim$1M IP addresses responded per measurement, which shows that Shatel, at any given time, has in use 15% of its pool – which was confirmed by Shatel, providing an insight on how they (re)use its pool of IP addresses.

Table II shows the number of unique IP addresses observed on the measurements and on the ground truth. As can be seen, our method was capable to obtain response from 94.95% of the addresses employed by Shatel (the ratio of intersecting IPs between measurement and DHCP logs) during the measured period. The remainder IPs in the DHCP log files of Shatel (42,510) did not respond either because of firewalls or because our probes might have missed those IPs due to sampling rate. Interestingly, there were 4,551 IP addresses only found in our measurements: those are assigned to devices such as routers and servers that do not have their IP addresses recorded in the ground truth. Part of these addresses were allocated to business customers, which, in turn, maintain their own independent DHCP servers, therefore not included in the ground truth.

C. Session Duration Distribution

We compare the measured DHCP sessions to the ones in the ground truth of the log files, for both $\alpha = 0$ and $\alpha = 600$ (m-0 and m-600). Table III summarizes the results. In the first line of the table ($\sum$ **all**), we show the sum of the duration of all sessions for the measurements and ground truth. Then, in the second line $\sum \cap_{IPs}$, we show what portion of these measured hours overlap in time with the measured hours from the DHCP session log files – that is, that captured correctly online time intervals of the intersecting IP addresses. This is shown in the "Ratio" row, which is obtained by dividing $\sum \cap_{IPs}$ of the measurements by the $\sum \cap_{IPs}$ of the ground truth (DHCP). As can be seen, for both measurement files (m-0 and m-600), our method was able to account for $\sim$ 65% of the online time of all the observed IP addresses. It is important to highlight the meaning of these findings: by only sending frequent ICMP messages, we were able to infer correctly 65% of all of the sessions' duration by a 1M IP addresses ISP (Shatel). Due to our sampling rate, we technically miss all sessions that start and end in between two consecutive time intervals (10 minutes). Increasing the frequency could possibly lead to better results, however at the price of increasing traffic footprint.

Another finding is that the threshold parameter α only slightly improves the accuracy of the session durations. To understand why, we further analyze the number of sessions by comparing m-0 to m-600. We can see that m-600 in fact reduced the total number of measured sessions, by merging two distinct sessions. This, in turn, has led to 37,378.42 more hours being correctly estimated, which is a small fraction of

Shatel DHCP logs - Sessions Duration (h)

	m-0	m-600	DHCP
$\sum$ all	59,676,305.56	60,733,610.81	96,899,976.59
$\sum \cap_{IPs}$	59,336,733.11	59,374,111.53	90,874,619.90
Ratio	65.29%	65.33%	
RMSE	0.29	0.28	

DHCP - Sessions

	m-0	m-600	DHCP
$\sum$ all	14,533,525	8,215,301	19,877,570
$\sum \cap_{IPs}$	14,432,133	8,182,572	18,498,448
Ratio	78.01%	44.23%	
RMSE	0.42	0.50	

TABLE III. VALIDATION RESULTS

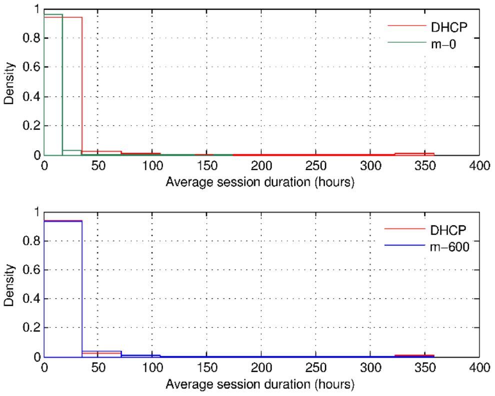

Fig. 3. Normalized histogram of average session duration per IP for Shatel

the total hours ($\sum$ all) with a slightly smaller root-mean-square error (RMSE).

However, when comparing the average session duration per IP m-600 outperforms m-0 in estimating the average session time on IP addresses. Fig. 3 shows the normalized histogram of these results: m-600 follows closer the shape of the ground truth (Spearman's $R^2 = 0.69$), and is capable to estimate average sessions from IPs having long average duration sessions. m-0, on the other hand, is sensitive to any packet loss, and estimates a larger number of very short average sessions ($R^2 = 0.50$). Comparing Fig.3(b) to Fig.3(a), we can see that m-0 performs poorly in estimating the correct number of sessions with duration inferior to 50 hours, which is explained by the fact that those IPs did not respond to the probes while they were actually online. The reasons for that are hard to pinpoint, but include either real-time firewall/IDPS, probe loss, transient errors, greylisting, as discussed in Section II-D.

IV. Estimating DHCP Churn

Our measurements were capable to account for 78% and 44% of the observed sessions for our ground truth datasets (Table III, row Ratio). If each new session would be associated with a new user, then the number of sessions of an IP address would yield to the number of distinct users that an IP has been assigned to. However, a user might be re-assigned to the same IP multiple times over the measured period. To cope with that, one could employ device fingerprinting [29], but this approach requires a large number of packets to be sent in order to measure clock skews, which is hard to scale when probing entire ISPs' address pools, not to mention the privacy implications.

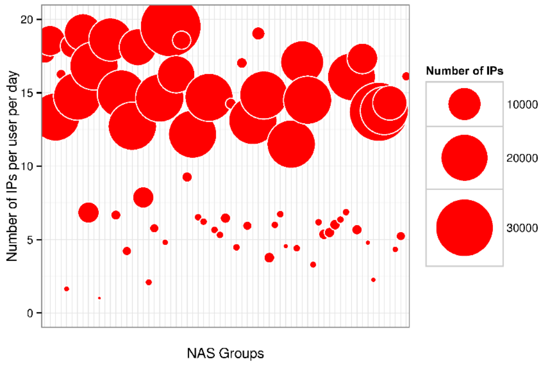

Fig. 4. NAS group size vs churn rate

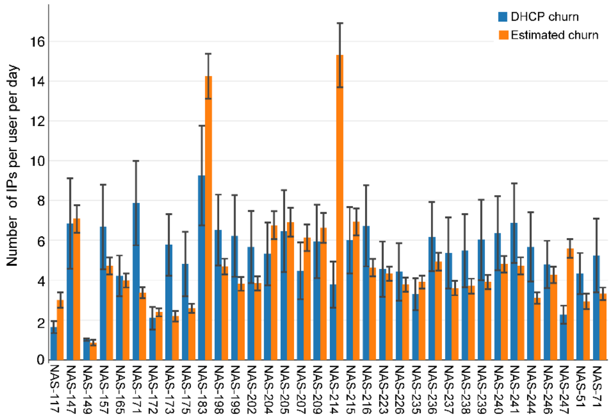

Fig. 5. Number of IPs per user per day

We envision an approach to statistically estimate the churn rates of IP address, using a counting Poisson process. Our churn estimator is based on the number of sessions of an IP address to approximate the number of users behind it. We start by determining, for each IP, the number of sessions as described in Section II-B.

ISPs typically configure Network Access Server (NAS) with pools of addresses, which are then assigned to users. Since the logs are stored by the ISP for each individual NAS,instead of analyzing the churn of IPs altogether, we divide the pool of visible addresses into sub-groups, which correspond to each NAS server as configured at Shatel. Shatel uses 36 different NAS groups with different number of IP addresses assigned ranging from groups with hundred IPs to groups with more than 30,000 different IP addresses. As can be seen in Fig. 4, larger NAS groups have larger churn rate than groups with less number of IP addresses.

For any session, as the churn rate is verified to follow a Poisson distribution [30], then from the properties of the distribution, the number of IPs per user can be estimated based on its rate. Consider a Poisson process $\{\mathcal{A}(t)\}_{t\geq 0}$ that counts the number of active IPs in intervals of $[0,t]$. Assuming that all users were online at some point of time $t' < t$, we can use the intensity process λ of the Poisson process $\{\mathcal{A}(t)\}_{t\geq 0}$ as an estimator churn rate.

Figure 5 shows the mean number of IPs per user per day

AS	CC	ISP	BGP-2014	Visible (total)	Visible (mean)	σ	Carna(Total)	BGP-2012
7018	US	AT&T	73,820,672	3,836,880 (5.19%)	2,195,068 (2.97%)	56,437.0	3,355,650 (5.73%)	58,492,421
2856	UK	British Telecom	11,352,576	2,673,034 (23.54%)	563,635 (4.96%)	15,439.3	2,337,454 (15.23%)	15,344,640
3320	DE	Deutsche Telekom	34,404,352	17,450,601 (50.72%)	4,705,551 (13.67%)	176,638.3	18,514,491 (53.47%)	34,621,952
3215	FR	Orange	15,273,728	2,680,682 (17.55%)	408,537 (2.67%)	11,374.9	3,654,191 (24.75%)	14,762,496
Total:			**134,851,328**	**26,641,197 (19.75%)**	**7,872,791 (5.83%)**	–	**27,861,786 (22.61%)**	**123,221,509**

TABLE IV. EVALUATED ISPs – MARCH 13TH–26, 2014

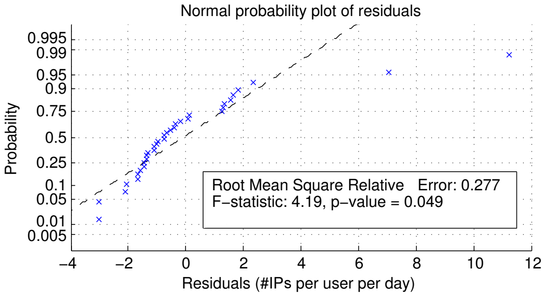

Fig. 6. Error estimation

for those NAS groups that are visible in Shatel. We observe that around 2% of the groups have at most one IP per user per day in average; while the mean number of IPs per user per day is around 5. It is also worth noting more than 60% of the groups have 10 or more IPs per user per day. By measuring the leverage and the Cook's distance we can detect two outliers (NAS-183 and NAS-214). Removing these outliers, the root mean square relative error is equal to 0.27, which confirms the notable accuracy of our estimation. Figure 6 shows the normal probability plot (NPP) of the error of the churn estimation. Despite a short curvature in the NPP, the probability plot seems reasonably straight, meaning an accurate fit to normally distributed residuals. The F-statistic of the linear fit versus the constant model is 4.19, with a p-value of 0.049. Hence the model is significant at the 5% significance level.

Despite the notable accuracy of the results (72.2%), the churn estimator is constrained by the sampling rate. Thus, if an IP change occurs in an interval below 10 min, the estimator will not be able to capture with probability $p = \lambda \cdot 10e^{-\lambda 10}$ where λ represents the number of IP changes per minute [30]. Note that in non-disruptive networks, it is reasonable to assume that $\lambda \in (0, 0.05]$ (i.e, one IP per user per day), therefore the probability of not capturing an IP change is below 0.3. It is worth noting that our estimator relies on the a priori knowledge of how Shatel manages their IPv4 pool of addresses. Without this information, we should perform an additional step to cluster the pool of addresses according to the duration and afterwards apply the estimator.

V. ANALYZING LARGER ISPS

In this section, we assess the scalability and performance of our method: we apply it to four major ASes from different countries – three major European ISPs (British Telecom/UK, Deutsche Telecom/DE, Orange/FR) and one American (AT&T). We chose these ASes because they are large, provide heterogeneous connectivity services (retail, business, wireless, CDN, etc.) and cover a large base of customers. Using the same setup described in Section II-C, we probed these ISPs for 17 continuous days (March 13th – March 29th, 2014), which amount to 134 million IPv4 addresses, as can be seen in Table IV (column BGP-2014).

Before starting these large-scale measurements, however, we met with TU Delft's Security Incident Response team and coordinated how the measurement would be carried out and how the requests would be handled. We ran a web server in the same measurement VM with a web page describing the project goal, our credentials, and how users could opt-out of our measurements, to which we promptly conformed.

In total, we have received a total of 35 e-mails requesting IP addresses to be removed, which we did immediately. All requests were from system administrators in small-businesses and few tech savvy home users. In only one instance one user wrongly thought we were carrying out a denial-of-service attack (DoS) on his server, which was not the case and we have confirmed him once he shared with us his intrusion detection system (IDS) log files, which showed we sent only 2 ICMP echo-requests per IP per measurement. In general the users were understanding and supportive; they only requested few of their IPs to be removed from our measurements.

A. Addresses Visibility/Usage

First, we are interested in evaluating how much of each ISP pool of addresses our method is able to capture – being the results the lower bound of the usage of addresses (Section II-D). Table IV summarizes the results. For the four ISPs, we can see that on average only a small percentage of the pool addresses for these ISPs is in use and visible to ICMP (column Visible (total) < 15%). However, as users connect/disconnect and more IPs get assigned by the ISPs, we see that the total number of visible IPs increases over time (Deutsche Telekom having more than 50% of its pool in cumulative use). This has to do also with diurnal patterns that can be observed on the Internet [31], as can been in Figure 7 that shows the time series of the active IP addresses per ISP.

In the lack ground truth data, we estimate the precision of our results by comparing it against the datasets produced by the infamous Carnanet botnet [12], in which the anonymous authors allegedly hijacked 420,000 home users CPEs (e.g., DSL/cable modems) to carry out Internet census for 8 months in 2012, which seems to be authentic [13]. By comparing our measurement against Carna botnet ICMP datasets, we can see that our method yields to similar visibility rates (Table IV, column Carna (Total)). By using a single source IP address, our method provides very similar results to probing using at least 420k source IP addresses, even when we measure for a shorter measurement window and when the size of the ASes has changed over time (column BGP-2012 shows the number of IPv4 for these ASes in May 2012).

Low visibility/usage for the pool of IP addresses may be due to the presence of network firewalls at prefixes. To rule it

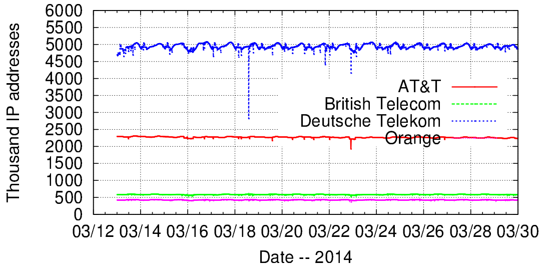

Fig. 7. Time Series of Online IPs per ISP

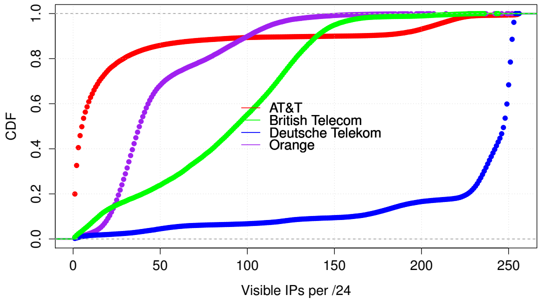

Fig. 8. CDF of Visible IPs per /24 prefix

out, we also measure the visibility at /24 prefixes level – i.e., we consider a /24 in use if it has at least 1 active IP at any moment of the measurement window. Figure 8 shows the CDF of each /24 by the number of visible IP. As can be seen, AT&T and Deutsche Telekom exhibit almost opposite behavior: most of the visible /24 prefixes of Deutsche Telekom are heavily used (> 200 IP addresses), while 60% of the /24 prefixes of AT&T have less than 10 active IPs, for the 17 day monitoring period. This suggests that either these prefixes are sub-utilized or that they are assigned to devices which are configured to not respond to ICMP probes. Such sub-utilization of prefixes can be further investigated when assessing fairness in the IPv4 address space distribution [32].

B. Session Duration Distribution

Table V shows the results of session duration for each ISP, for both $\alpha = 0$ and 600 values (m-0 and m-600). Figure 9 shows the empirical cumulative density function of average session duration per IP for each ISP for m-600. On average, a bot would have its IP address renewed every 61, 20, 10, and 14 hours for AT&T, British Telecom, Deutsche Telekom, and Orange, respectively, which wind up inflating at different rates the actual number of compromised computers per ISP. We can also observe that most addresses have an average lease inferior to 50 hours, and that for AT&T, we see spikes around $t = 75, 100, 140$ hours, which indicates that large portion of the /32 IP addresses are managed by DHCP servers that enforce IP address changes after reaching these session durations.

C. Number of Sessions per IP

This metric indicates how many times an IP address is continuously active and visible for the monitoring period. Figure 11 shows the ECDF of the number of sessions per

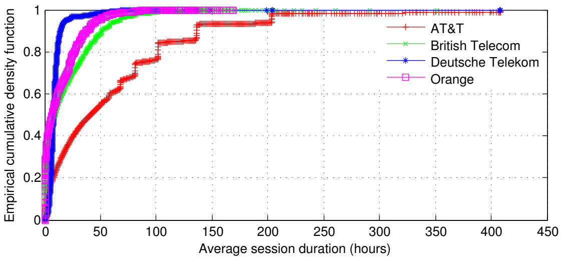

Fig. 9. ECDF: mean session duration/IP (m-600)

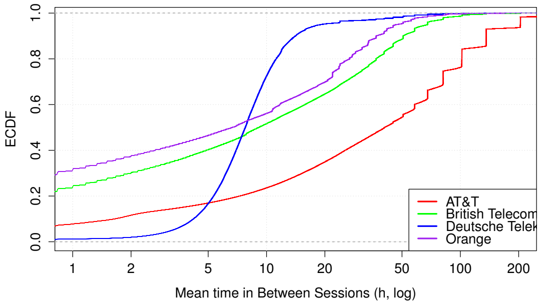

Fig. 10. ECDF: mean time in between session/IP (m-600)

IP address, for each ISP obtained from our measurements, by computing the number of distinct sessions each IP address.

We can see that there is a significant distinction among each ISPs; being Deutsche Telekom the one having most sessions/IP (60% of IPs have more than 10 sessions). This confirms similar findings [11], in which the authors hijacked a botnet for 10 days and show significant discrepancy for bots from different countries (US: 0.18 IPs/day, Germany: 1.3).

D. Discussion: classification of IP addresses

Large ISPs operate large heterogeneous networks, with various offered services – mobile, wireless, home, small and large businesses. These services, in turn, require different addresses management policies. Therefore, churn rates should be better determined accordingly to the type of service the IP address is used for.

Whois and reverse DNS (r-DNS) information can be used to classify IPs according to that. To illustrate that, let us consider Deutsche Telekom. We resolved 13.07M IP addresses (out of 17+ visible). Out of those, 12.9M are associated with retail/business services (domain t-ipconnect.de), while 56K are associated with Akamai content distribution network (CDN) (akamaitechnologies.com), and 37K are associated with infrastructure (DTAG.DE). Similarly to time-series decomposition, the CDF of the number of sessions (Fig. 11) can be decomposed per domain, which is shown Figure 12. As can be seen, most of the churn rates on Deutsche Telekom are explained by the larger retail/business domain.

Therefore, for such large heterogeneous ISPs, we first need to develop a methodology to classify IP addresses accordingly to their usage. Then, we can cluster IPs that have similar usage and properties (session durations, number of sessions), and we feed it to our model to estimate more precise DHCP churn rates per cluster of addresses.

	AT&T		British Telecom		Deutsche Telekom		Orange	
	m-0	m-600	m-0	m-600	m-0	m-600	m-0	m-600
Mean	4.854	61.140	3.280	19.451	3.169	9.900	2.769	14.115
Median	5.354	40.725	3.416	9.247	3.113	7.914	2.716	6.816
Max	68.669	408.212	67.195	408.159	27.679	408.158	50.071	408.018
Min	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Std Dev	2.259	68.634	2.496	24.473	0.905	10.249	2.383	17.967
Coeff Var	0.465	1.123	0.761	1.258	0.286	1.035	0.861	1.273
Trimmed Mean (95%)	4.848	55.723	3.150	17.634	3.157	8.751	2.630	12.769

TABLE V. STATISTICS SUMMARY SESSION DURATION IN HOURS.

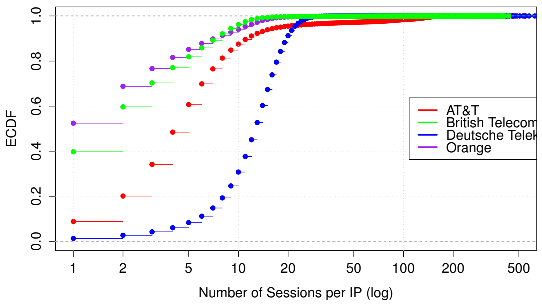

Fig. 11. ECDF: mean number of sessions/IP (m-600)

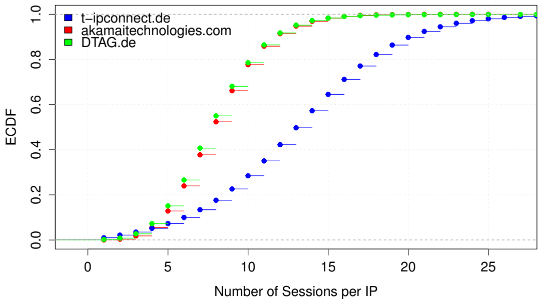

Fig. 12. ECDF: mean session Duration (Deutsche Telekom domains)

VI. Related Work

To the best of our knowledge, this is the first study that employs high-performance large-scale probing with the goal of estimating session durations and dynamics of IP addresses of entire ISPs. Heidemann *et al.* [4] have analyzed the session time of random 24,000 /24 prefixes. We extend their methodology to reconstruct sessions and validate against a mid-size ISP, and then apply it to the entire IP addresses of four major ASes (∼ 280,000 /24). We show how the distribution of sessions varies within and for different ISPs. Another Internet-wide probing was carried out by anonymous authors in [12]. Since its validity questionable [13] (data was obtained by hacking users' CPEs), we only use their datasets to compare the visibility of the four large ISPs we have probed in Section V. Schulman *et al.* [21] have employed ICMP-based measurements to detect network failures incurred by the weather.

DHCP leases have been analyzed in previous works [5]–[8]. However, they have employed DHCP and http server logs. Brik *et al.*, for example, monitored DHCP servers of University of Wisconsin-Madison for 3 weeks, while Khadilkar *et al.* [6], analyze four days of DHCP logs at George Tech. Papapanagiotou *et al.* [7], have monitored two networks for less than 6 weeks, having less than 6,000 active IP addresses. Finally, Xie *et al.* [8] analyzed the http log files for MSN Hotmail, which also included user login information for one month period. Our method differs with these since it enables session duration estimation independent of an ISP and does not require access to log files, making it scalable to the entire Internet.

It is also important to highlight our measurements do not allow to monitor/fingerprint individual users, since the information we collect (IP address, timestamp) is not enough to single out unique users. Active probing has been used to perform device fingerprint. Kohno *et al.* [29] have employed ICMP and TCP-based active measurements to measure clock-skews of devices, which ultimately may allow fingerprint. However, their method requires a vast number of probes per individual IP to be sent, and it is not easily scalable. Eckersley [33], in turn, develop a method to measure the entropy of a users' browser, based on the parameter automatically provided by the user's browser. However, in this case, is a passive measurement approach, in which users must voluntarily access websites that may fingerprint their browsers.

Previous works either used passive data to estimate this churn rate [30] or described complex stochastic models that are not able to capture the whole nature of the dynamic allocation of addresses [34]. However, none of these models was able to establish a methodology valid for the whole Internet. Contrarily, our methodology is scalable and valid for any network.

VII. Conclusions and Future work

This paper provided the first steps towards estimating Internet-wide DHCP churn rates. We proposed an validated a methodology that allows to probe entire ASes. This can be therefore used to produce reliable Internet security metrics to compare ISPs, besides providing insights on how ISPs manage their allocated pool of addresses. We learned that, by employing only ICMP-based measurements, we were able to successfully probe 94% of all online IP addresses of a 1M IP addresses ISP. From these, we are able to capture 65% of their online time. Based on the measurement data, we developed a model based on a counting Poisson process to estimate DHCP churn rates. By comparing our model with the ground truth data, we were able to estimate correctly 72.3% of the DHCP churn rates.

After that, we applied our measurement methodology to four large AS from three European ISPs and one American. We have shown the significant differences among them, with regards to number of sessions, session durations, and number of sessions, and triangulated the results with the Carna botnet Internet census 2012. Moreover, we have shown that to better estimate DHCP churn rates for such large ISPs, we should first classify IP addresses taking into account the connectivity service each IP is employed for (retail, wireless, etc.). By doing that, we will be able to cluster IP addresses according to their usage as well as similar measurement properties, which can be used to feed our DHCP churn estimation model. These are the next steps we are taking in this research.

Acknowledgments: We would like to thank Shatel for the collaboration in this research. Special thanks to Aiko Pras and Ramin Sadre for the valuable comments, as well as to TU Delft's ICT and Abuse staff (Lolke Boonstra, Adrian Cooke, and Paul Keekstra), SUFRNet (Wim Biemolt) for their support. This work was partly funded by the EU Advanced Cyber Defence Centre *(ACDC)* project (#325188), and by the Dutch ISP's Abuse Internet Exchange initiative.

REFERENCES

[1] S. Zander, L. Andrew, G. Armitage, and G. Huston, "Estimating IPv4 address space usage with capture-recapture," in *Local Computer Networks Workshops (LCN Workshops), 2013 IEEE 38th Conference on*, Oct 2013, pp. 1010–1017.

[2] A. Dainotti, K. Benson, A. King, k. claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos, "Estimating Internet Address Space Usage Through Passive Measurements," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 1, pp. 42–49, Dec. 2013.

[3] X. Cai and J. Heidemann, "Understanding block-level address usage in the visible Internet (extended)," USC/Information Sciences Institute, Tech. Rep. ISI-TR-2009-665, June 2010.

[4] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, "Census and Survey of the Visible Internet," in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008.

[5] V. Brik, J. Stroik, and S. Banerjee, "Debugging DHCP Performance," in *Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004.

[6] M. Khadilkar, N. Feamster, M. Sanders, and R. Clark, "Usage-based DHCP Lease Time Optimization," in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 71–76.

[7] I. Papapanagiotou, E. M. Nahum, and V. Pappas, "Configuring DHCP Leases in the Smartphone Era," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 365–370.

[8] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, "How dynamic are IP addresses?" *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 301–312, 2007.

[9] E. Cooke, F. Jahanian, and D. McPherson, "The zombie roundup: understanding, detecting, and disrupting botnets," in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet Workshop*. Berkeley, CA, USA: USENIX Association, 2005, pp. 6–6.

[10] M. A. R. J. Z. Fabian and M. A. Terzis, "My botnet is bigger than yours (maybe, better than yours): why size estimates remain challenging," in *Proceedings of the 1st USENIX Workshop on Hot Topics in Understanding Botnets, Cambridge, USA*, 2007.

[11] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: analysis of a botnet takeover," in *Proceedings of the 16th ACM conference on Computer and communications security*. ACM, 2009.

[12] Carna Botnet, "Internet Census 2012–Port scanning /0 using insecure embedded devices ," 2012. [Online]. Available: http://internetcensus2012.bitbucket.org/paper.html

[13] T. Krenc, O. Hohlfeld, and A. Feldmann, "An Internet Census Taken by an Illegal Botnet: A Qualitative Assessment of Published Measurements," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, 2014.

[14] V. Fuller and T. Li, "RFC 4632: Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan," August 2006.

[15] K. Hubbard, M. Kosters, D. Conrad, D. Karrenberg, and J. Postel, "Internet Registry IP Allocation Guidelines," RFC 2050 (Best Current Practice), Internet Engineering Task Force, Nov. 1996.

[16] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," RFC 4271 (Draft Standard), Internet Engineering Task Force, Jan. 2006, updated by RFC 6286.

[17] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131 (Draft Standard), Internet Engineering Task Force, Mar. 1997, updated by RFCs 3396, 4361, 5494, 6842.

[18] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865 (Draft Standard), Internet Engineering Task Force, Jun. 2000.

[19] W. Simpson, " The Point-to-Point Protocol (PPP)," RFC 1661 (Draft Standard), Internet Engineering Task Force, Jul. 1994.

[20] Cisco, "Configuring the Cisco IOS DHCP Server," 2014. [Online]. Available: http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_dhcp/configuration/12-4t/dhcp-12-4t-book/config-dhcp-server.html

[21] A. Schulman and N. Spring, "Pingin' in the Rain," in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC '11. New York, NY, USA: ACM, 2011.

[22] Z. Durumeric, E. Wustrow, and J. A. Halderman, "Zmap: Fast internet-wide scanning and its security applications," in *Proceedings of the 22nd USENIX Security Symposium*, 2013.

[23] J. Postel, "Internet Control Message Protocol," RFC 792 (INTERNET STANDARD), Internet Engineering Task Force, Sep. 1981.

[24] D. Plummer, "Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware," RFC 826 (INTERNET STANDARD), Internet Engineering Task Force, Nov. 1982.

[25] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection," RFC 5475 (Proposed Standard), Internet Engineering Task Force, March 2009. [Online]. Available: http://www.ietf.org/rfc/rfc5475.txt

[26] S. Zander, L. L. Andrew, G. Armitage, G. Huston, and G. Michaelson, "Mitigating Sampling Error when Measuring Internet Client IPv6 Capabilities," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12, 2012, pp. 87–100.

[27] Shatel, http://en.shatel.ir, 2014.

[28] Reseaux IP Europeens Network Coordination Centre (RIPE NCC), "Routing Information Service (RIS)," 2014. [Online]. Available: http://www.ripe.net/data-tools/stats/ris

[29] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," *Dependable and Secure Computing, IEEE Transactions on*, vol. 2, no. 2, pp. 93–108, 2005.

[30] A. Metwally and M. Paduano, "Estimating the Number of Users Behind Ip Addresses for Combating Abusive Traffic," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11, 2011, pp. 249–257.

[31] L. Quan, J. Heidemann, and Y. Pradkin, "When the Internet Sleeps: Correlating Diurnal Networks with External Factors," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 87–100.

[32] I. Livadariu, A. Elmokashfi, A. Dhamdhere, and k. claffy, "A First Look at IPv4 Transfer Markets," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '13. New York, NY, USA: ACM, 2013, pp. 7–12.

[33] P. Eckersley, "How unique is your web browser?" in *Privacy Enhancing Technologies*. Springer, 2010, pp. 1–18.

[34] A. Wahid, C. Leckie, and C. Zhou, "Estimating the number of hosts corresponding to an address while preserving anonymity," in *Proceedings of the 6th International Conference on Network and System Security*, ser. NSS'12, 2012, pp. 166–179.