# How to Count Bots in Longitudinal Datasets of IP Addresses

Leon Böck* Dave Levin† Ramakrishna Padmanabhan‡ Christian Doerr§ Max Mühlhäuser*

*Telecooperation Lab
Technische Universität Darmstadt

†University of Maryland

‡CAIDA

§Hasso Plattner Institute
University of Potsdam

***Abstract*—Estimating the size of a botnet is one of the most basic and important queries one can make when trying to understand the impact of a botnet. Surprisingly and unfortunately, this seemingly simple task has confounded many measurement efforts. While it may seem tempting to simply count the number of IP addresses observed to be infected, it is well-known that doing so can lead to drastic overestimates, as ISPs commonly assign new IP addresses to hosts. As a result, estimating the number of infected hosts given longitudinal datasets of IP addresses has remained an open problem.**

**In this paper, we present a new data analysis technique, *CARDCount*, that provides more accurate size estimations by accounting for IP address reassignments. CARDCount can be applied on longer windows of observations than prior approaches (weeks compared to hours), and is the first technique of its kind to provide confidence intervals for its size estimations. We evaluate CARDCount on three real world datasets and show that it performs equally well to existing solutions on synthetic ideal situations, but drastically outperforms all previous work in realistic botnet situations. For the Hajime and Mirai botnets, we estimate that CARDCount, is 51.6% and 69.1% more accurate than the state of the art techniques when estimating the botnet size over a 28-day window.**

## I. Introduction

Empirically measuring botnets is critical to understanding how they operate, the threat they pose, and ultimately how to mitigate them. One of the most basic yet important questions researchers ask about botnets is: how big are they, and how does their size change over time?

Although these are such simple questions, accurately measuring the size of a botnet turns out to be incredibly challenging. Ideally, each bot would have a unique, long-lived identifier that a researcher could ascertain. While a small number of botnets do offer such identifiers (e.g., Hajime bots can be queried for a public key that they generate each time they reboot [10]), for most botnets the only identifiers are the bots' IP addresses.

It is therefore tempting to simply count IP addresses, but unfortunately—as is now well-known in the community—this can yield wildly inaccurate results. For instance, Stone-Gross et al. [26] showed that counting the daily IP addresses overestimated the size of the Torpig botnet by 36.5%. The primary reason for this is that IP addresses are not necessarily long-lived identifiers; ISPs regularly *reassign* IP addresses, sometimes after short intervals (e.g., a couple of hours) [18].

Left with no better alternatives, researchers have continued to use IP addresses to count bots, but limit their analyses to short windows of time: typically 15–60 minutes. While operating over short time-windows helps avoid double-counting bots that obtain new IP addresses, it introduces at least two unfortunate limitations in our understanding of botnets: *First*, counting the number of bots every 15–60 minutes misses out on important diurnal patterns that happen on daily or weekly intervals. For instance, if the bots in one country experience a daily lull at the same time that another country experiences a daily spike, then limiting the analysis to small windows of time might make it appear that the set of bots remains largely unchanged over time. *Second*, it makes it difficult to understand the impact of changes to the botnet itself. Herwig et al. [10] observed that the Hajime botnet rapidly increased in size immediately after rolling out the Chimay Red exploit. However, prior techniques do not permit direct comparisons of bots between windows of time, making it difficult to understand who precisely the new bots are.

In this paper, we introduce a new data analysis approach to count IP addresses that accounts for IP address assignment durations. Our approach builds off of prior work that observed that ISPs tend to follow a predictable reassignment duration [18]; for instance, Telefonica in Spain has been reported to reassign approximately 90% of its IP addresses at precise 24-hour intervals. Our central insight is that we can use per-AS (autonomous system) models of address assignment durations to calculate the probability that two IP addresses measured at different points in time correspond to the same physical bot.

We call our approach *CARDCount* (Considering Address Reassignment Duration when Counting). In contrast to traditional counting approaches which consider only IP addresses, CARDCount also incorporates per-AS distributions of IP address durations. These distributions are readily available, thanks to prior work [18], [19], and continue to be collected with efforts such as RIPE Atlas [4], a set of network probes distributed across thousands of networks around the world. Our primary contribution is in the formulation, evaluation, and application of an analytical framework that shows that combining these datasets, with the datasets that botnet researchers collect,

Network and Distributed System Security (NDSS) Symposium 2023
27 February – 3 March 2023, San Diego, CA, USA
ISBN 1-891562-83-5
https://dx.doi.org/10.14722/ndss.2023.24002
www.ndss-symposium.org

we can more accurately count bots regardless of IP address reassignments.

Accounting for address assignment durations confers many benefits over traditional approaches to botnet size estimation. *First*, CARDCount can be applied to large windows of time (weeks or more), and is not limited to the standard small windows of 15–60 minutes. We show that when estimating the size of real botnets (Hajime and Mirai) over 28-day windows, CARDCount is 69% more accurate than the state of the art tool. *Second*, CARDCount takes advantage of the probability distributions of address assignment durations to compute confidence intervals over the botnet sizes it estimates. These allow researchers to reason with greater statistical certainty, especially over large time windows.

**Contributions** We make the following contributions:

- We introduce CARDCount, a data analysis method that accurately estimates (with confidence intervals) a botnet's size by accounting for IP address assignment durations (§III).
- We compare CARDCount to state of the art bot counting approaches on a ground-truth dataset, showing that CARDCount is more resilient to confounding factors such as churn or incomplete data (§IV).
- We show that CARDCount outperforms existing approaches when applied to real botnet measurements of Hajime and Mirai (§V).
- We make our code and per-AS distribution data publicly available at https://github.com/CardCount.

## II. Background and Related Work

We consider the following basic setting that is typical in work that measures botnets. We assume that a measurement infrastructure can periodically ascertain a set $S_T$ of bots' IP addresses over some window of time $T$ (e.g., the amount of time it takes to scan the botnet). Often, these windows of time are small (on the order of a couple hours), but often repeated to obtain longitudinal datasets spanning weeks, months, or even years. The precise methods of measurement vary, for instance, actively scanning a botnet's command and control infrastructure [10] or passively capturing attack traffic [7], [3].

While there are many reasons for studying botnets, we focus in this paper on arguably the most common one: determining how many hosts comprise the botnet. There are two predominant ways by which prior work has attempted to answer this question by counting IP addresses. We review these techniques, and then we review the papers that apply them.

### A. IP-based Size Estimation Techniques

*a) Binned IP Counting (BinCount):* The most common technique for botnet size estimation is to simply count the number of unique IP addresses seen within a small bin of time. We will refer to this technique as $\text{BinCount}_T$ where $T$ represents the size of the time bin:

$$\text{BinCount}_T = |S_T|$$

There are two confounding factors that make BinCount impractical or, at the very least, difficult to interpret. The first is *IP address reassignments*: ISPs typically assign dynamic IP addresses to their customers, and through DHCP can periodically reassign them a new IP address. If a given bot's IP address changes within the observation window $T$, then that one bot can be double-counted, leading BinCount to over-estimate. To minimize the chances of double-counting, many researchers prefer to use the smallest time windows possible, but this exacerbates the next concern.

The second confounding factor is *diurnal patterns*: it has been shown that botnets (including IoT botnets) exhibit diurnal patterns, increasing or decreasing in size when users are at home or at work. For globe-spanning botnets, this means that there is no single window of time $T$ during which the number of bots is maximized across all countries. Thus, while BinCount may accurately capture the number of bots for short windows of time, it does not help in answering how many hosts are infected worldwide over the course of a day or longer.

*b) Max Simultaneously Active IPs (MaxCount):* To further reduce the impacts of address reassignment, another popular approach is to count the number of IP addresses that are *simultaneously* active at any point in time $p$. More precisely, this means that the IP addresses were observed to be sending messages before and after $p$. To compute this, an IP address is assumed to be continuously active as long as it sends a message every $\tau$ seconds. The size of the botnet is then estimated by taking the maximum number of simultaneously active IP addresses. We refer to this method as MaxCount: More precisely, suppose that $S_T^p$ represents the set of IP addresses simultaneously active at time $p$, then:

$$\text{MaxCount} = \max_p |S_T^p|$$

Yan et al. [29] improved upon MaxCount by observing that IP churn commonly occurs within an Autonomous System (AS). They proposed computing MaxCount on a per-AS basis, and then aggregating the sum over all of the ASes, even if the maxima per AS are at different times $p$. We refer to this as $\text{MaxCount}_{AS}$. More precisely, if $\text{AS}_i(S_T^p)$ denotes the subset of IP addresses from $S_T^p$ that are in $\text{AS}_i$, then:

$$\text{MaxCount}_{AS} = \sum_i \max_p |\text{AS}_i(S_T^p)|$$

MaxCount (in both variants) has two main benefits over BinCount with small bin sizes. First, it does not rely on any pre-specified bin size and identifies the maximum number of hosts on a continuous scale. Second, by looking at each AS independently it is less affected by diurnal patterns occurring at different times throughout the world.

Despite these benefits, MaxCount still has several drawbacks. Most importantly, it only counts IP addresses that are active simultaneously, and thus, as we will demonstrate, if the measurement infrastructure is unable to achieve nearly perfect coverage of an AS, then it suffers from *under-estimating* the size of a botnet.

As we will demonstrate, our technique, CARDCount, addresses these shortcomings.

TABLE I: Summary of counting technique usage to estimate the size of botnets.

| Technique | Papers | Botnet(s) |
|---|---|---|
| BinCount$_{\text{20m}}$ | [10] | Hajime |
| BinCount$_{\text{1h}}$ | [15], [3], [26] | Sality, Mirai, Torpig |
| BinCount$_{\text{2h}}$ | [13] | Storm |
| BinCount$_{\text{24h}}$ | [12], [13], [11], [23], [15], [1], [26] | Storm, Sality, ZeroAccess, Torpig, Multiple P2P |
| BinCount$_{\text{14d}}$ | [17] | Multiple |
| BinCount$_{\text{1mo}}$ | [11] | Storm |
| BinCount$_{max}$ | [20], [25], [16], [26] | Multiple IRC, Walowdac, Sality, ZeroAccess, Torpig |
| MaxCount | [20], [11], [9] | Multiple IRC, Storm, Sality, ZeroAccess |
| MaxCount$_{AS}$ | [29] | Sality, ZeroAccess |

## B. Counting Botnet Infections

We have identified 14 papers dating back to 2004 that have provided estimates for various botnets' sizes, summarized in Table I. We observe that BinCount is the most prevalent counting approach, with 11 papers using it exclusively and two additional papers reporting both MaxCount and BinCount results. The most common bin size is 24h with seven out of 13 papers using it. This is somewhat surprising, as IP address reassignment often occurs more quickly than 24h [18], [19]. Four papers use a MaxCount variant; of these, only Yan et al. [29] use MaxCount$_{AS}$, even though it provides the most accurate lower bound estimates. In addition to the IP address counting listed in Table I, some papers also measure botnet ID counts [25], [3], [10], indicating that even when a botnet has countable IDs, it is still valuable to researchers to be able to accurately count IP addresses.

Three papers discuss the challenges of estimating a botnet's size. Rajab et al. [20] investigate how different botnet counting approaches lead to vastly different size estimations. As part of their study, they report on both the total aggregate (BinCount) and simultaneously active IP addresses (MaxCount). They suggest combining multiple independent measurements. Kanich et al. [13] investigate measurement inaccuracies caused by IP address reassignments, other researchers, and differences in the measurement approach. While they highlight and investigate the problem of IP address assignment durations, they do not propose a solution to the problem. Finally, Stone-Gross et al. [26] showed that counting the daily IP addresses of the Torpig botnet overestimated the size by 36.5%. They further observed that this overestimation varied by AS, and posited that a per-AS analysis technique might help achieve greater accuracy; our paper, at last, develops this idea.

# III. CARDCount Technique

In this section, we describe our CARDCount methodology for inferring how many hosts correspond to a set of IP addresses over some period of time. We also show how to compute confidence intervals over these estimates; to the best of our knowledge, this is the first bot-counting technique to do so. We begin by setting up the problem that CARDCount seeks to solve.

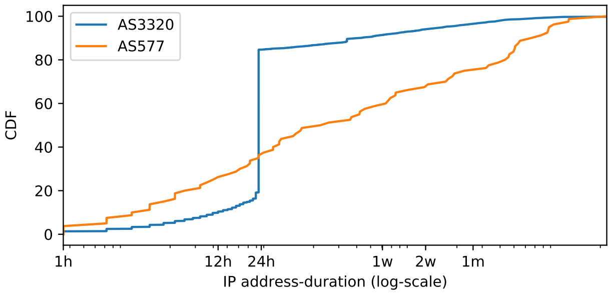


Fig. 1: Example IP address duration distributions

## A. Problem Setup

Consider the following common scenario: a researcher has measured a botnet over some duration of time $T$ and has obtained $A$ distinct IP addresses by crawling or interacting with the set of active bots during that time. The goal is to determine the number of actual infected *hosts* $H$ those IP addresses correspond to.

As noted in Section II, the number of IP addresses ($A$) is typically significantly larger than the number of hosts ($H$) over a long window of time $T$. Prior work has observed that this is predominantly due to the fact that any given host can be assigned multiple IP addresses over time [20], [13].

The key insight behind CARDCount is to infer how each individual AS reassigns its IP addresses, and to use data about address assignment durations to estimate the number of hosts. As a strawman example: if every host on the Internet were assigned a new IP address every $r$ seconds, then over a period of time $T$, each host would have $T/r$ IP addresses. Thus, $A$ addresses over time $T$ would correspond to $Ar/T$ hosts.

This is the main thrust of CARDCount, but of course this toy example fails to account for two critical realities: *First*, not every host on the Internet is reassigned its IP address on some consistent, global schedule; rather, different ASes assign IP addresses for different durations. We account for this by using complementary datasets about AS' address assignment durations. *Second*, not all hosts obtain their IP addresses at the same time; hosts come online at different times, restart due to failures, and so on. As a result, even if a host obtained a new IP address every 24 hours, they might obtain a new IP address seconds into our window of observation $T$. We address this with a mathematical model that accounts for how much hosts' IP addresses overlap with our observation window.

In the remainder of this section, we discuss how CARDCount addresses these practical concerns.

## B. IP Address Durations

The intuition behind CARDCount is that if we can understand the rates at which an individual ASes reassigns its IP addresses, then we can estimate how many hosts a given number of IP addresses correspond to. For instance, if an AS reassigned IP addresses every 24 hours, and if we observed 7 addresses from that AS over the course of a week, then we could estimate that those addresses correspond to a single host.

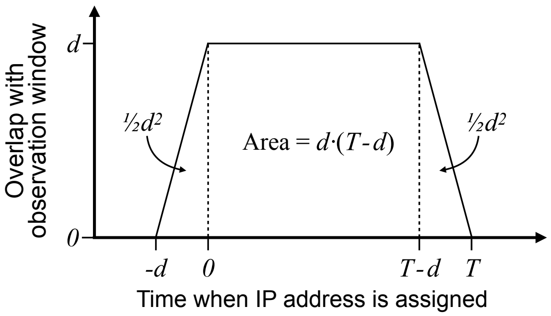

Overlap with observation window
d
0
½d²
Area = d·(T-d)
½d²
-d
0
T-d
T
Time when IP address is assigned

Fig. 2: Computing how much a given IP address assignment of duration $d$ overlaps with an observation window of length $T$. (Note the $x$- and $y$-axes have different scales.) Annotated values denote the areas of the given regions; the area of the entire shape is $d(T-d)+d^2 = d \cdot T$.

In order to perform this analysis, CARDCount requires *address assignment duration distributions* of each AS for each IP address under consideration. Although ISPs do not typically make this information publicly available, Padmanabhan et al. and Griffioen et al. [18], [8] have both independently shown that address assignment durations can be measured directly. For instance, Padmanabhan et al. used data from RIPE Atlas probes: these are small measurement vantage points deployed across thousands of networks around the world that regularly perform basic measurement tasks (e.g., pinging root DNS servers, performing traceroutes to target servers, etc.). RIPE data is available to researchers, and Padmanabhan et al. showed how to infer when a RIPE Atlas probe's IP address has changed, allowing them to measure assignment durations.

A core finding of these prior efforts is that, while IP address assignment durations differ across ASes, durations *within* an AS tend to exhibit repeated patterns. Addresses within a given subnet exhibit even stronger similarity [8]. Figure 1 depicts two example IP address assignment distributions for the ASes 3320 (DTAG: Deutsche Telekom in Germany) and 577 (BELL Canada) taken from the RIPE Atlas probe dataset [19] from 2018. Two-thirds of DTAG's IP addresses are assigned for 24h. In contrast, BELL Canada has no discernible reassignment period, with assignment durations assigned almost uniformly at random throughout this observation window.

CARDCount takes each AS's address assignment duration distribution as input and computes their mean and variance. (We detail how CARDCount applies these next.)

As a practical matter, when applying these datasets to CARDCount, it is ideal to use distribution data that was collected at approximately the same time as the botnet IP address data, but there is some flexibility here. Prior work showed that while address assignment policies can change, the changes are infrequent [18], [19]. We use the existing datasets for address assignment durations, as they were collected contemporaneously with the botnet data we analyze.

### C. Accounting for Overlap with the Observation Window

As mentioned in §III-A, there is no global address reassignment schedule that all ASes follow; addresses can be reassigned at any time, on any day. Thus, it is possible that any host that is observed during an experiment's observation window was assigned its first observed IP address *before* the experiment began. Similarly, a host may obtain a new IP address shortly before the observation window ends.

In trying to convert the number of IP addresses $A$ into an expected number of hosts, it is important to account for the fact that some addresses may have only overlapped for short periods of time with the window of observation. CARDCount accounts for this overlap by computing $O(d,T)$: the expected amount of overlap that an IP address with duration $d$ would have with an observation window of duration $T$.

Figure 2 shows how we compute how much an IP address with duration $d$ overlaps with an observation window of length $T$.[1] If the observation window starts at time 0, then an IP address will overlap with the window only if its assignment started between time $-d$ and $T$. Assuming that address reassignments can occur at any time chosen uniformly at random, the probability of an overlapping IP address starting at any given time in the range $[-d,T]$ is equal to $\frac{1}{d+T}$. The average amount of overlap is equal to the area of the trapezoid depicted in Figure 2, which is equal to $d \cdot T$. Thus, the expected amount of overlap, $O(d,T)$, given average duration $d$ and observation window size $T$, is equal to the average overlap times the probability:

$$O(d,T) \;=\; (d \cdot T) \cdot \frac{1}{d+T} \;=\; \frac{d \cdot T}{d+T} \qquad (1)$$

When $d$ is much smaller than $T$, the overlap approaches $d$. To see this, suppose that $d = \epsilon T$ for some very small $\epsilon > 0$ (e.g., $d$ is on the order of minutes and $T$ is on the order of months). Then we get $O(d,T) = \frac{d \cdot T}{(1+\epsilon)T} = \frac{d}{1+\epsilon} \approx d$. However, we caution against simply using $d$ instead of $O(d,T)$, as even small values of $\epsilon$ can aggregate when considering the sizes of large botnets.

Equation 1 gives us the expected overlap for a given IP address duration $d$. It will also be useful to compute the *average* overlap across a distribution of address durations $D$. We denote this as $\bar{O}(D,T)$ and compute it as follows, assuming that we have taken $n$ discrete samples $\{d_1,\ldots,d_n\}$ from $D$:

$$\bar{O}(D,T) \;=\; \frac{1}{n}\sum_{i=1}^{n} O(d_i,T) \;=\; \frac{1}{n}\sum_{i=1}^{n} \frac{d_i \cdot T}{d_i + T} \qquad (2)$$

Because the samples are selected independently, this in essence gives us a weighted sum corresponding to the expected overlap for any IP address whose duration is drawn from the distribution $D$.

### D. Estimating Botnet Sizes

We are now able to present precisely how CARDCount estimates the number of hosts in a botnet upon observing $A$ addresses over an observation window of duration $T$.

[1] Note that Figure 2 assumes that $d \leq T$. When $d > T$, a similar geometric argument can be made, and it turns out that the resulting expected overlap is precisely the same as Equation 1.

CARDCount analyzes each AS differently. It breaks up the number of addresses $A$ into the number of addresses across each AS: if AS $k$ has $A_k$ addresses then $A = \sum_k A_k$. Then, it estimates the number of hosts on a per-AS basis.

The expected number of IP addresses that will be assigned per host from AS $k$ with duration distribution $D_k$, number of samples $n_k$, over time window $T$ is:

$$T/\bar{O}(D_k, T) \;=\; 1 \bigg/ \frac{1}{n_k} \sum_{i=1}^{n_k} \frac{d_i}{d_i + T} \qquad (3)$$

Thus, the expected number of hosts for AS $k$ with $A_k$ addresses is simply $A_k$ divided by the expected number of IP addresses per host:

$$\begin{aligned} \text{CARDCount}(A_k, D_k, T) &= A_k \bigg/ \frac{T}{\bar{O}(D_k, T)} \\ &= A_k \cdot \frac{1}{n_k} \sum_{i=1}^{n_k} \frac{d_i}{d_i + T} \qquad (4) \end{aligned}$$

Across the entire dataset, we estimate the total number of hosts by summing Equation 4 across all of the ASes: $\sum_k \text{CARDCount}(A_k, D_k, T)$.

### E. Computing Confidence Intervals

CARDCount relies on the mean observed durations in order to make its estimates. While the law of large numbers dictates that with *sufficiently large samples*, i.e., number of assigned IP addresses, the mean of the samples will reflect the mean of the distribution, the mean may still deviate depending on the standard deviation of $D$ and the sample size.

In order to compute how large the possible error of CARDCount could be in practice, we compute the confidence intervals as follows. While the input distribution $D_k$ for a given AS $k$ is usually not normally distributed, the central limit theorem states that the means of any distribution are normal for large sample sizes $A_k$. Therefore, we can estimate the 95th percentiles for a given AS $k$ using the central limit theorem:

$$\text{CARDCount}(A_k, D_k, T) \;\pm\; 1.96 \cdot \frac{\sigma_k}{\sqrt{A_k}}$$

where $\sigma_k$ is the mean sample standard deviation of the distribution $D_k$. To compute the aggregate confidence intervals, we sum across all of the ASes:

$$\sum_k \text{CARDCount}(A_k, D_k, T) \;\pm\; \sum_k 1.96 \cdot \frac{\sigma_k}{\sqrt{A_k}}$$

Based on common recommendations, these confidence intervals can be computed when the number of samples $A_k > 30$. In instances where $A_k \leq 30$, it is still possible to compute confidence intervals, but not with the above formulas. Instead, we repeat drawing $A_k$ samples from the distribution 1000 times and then compute the confidence intervals over those 1000 rounds of sampling.

### F. Assumptions and Threats to Validity

The above mathematical model makes several tacit assumptions related to confounding factors that, if violated, could affect the accuracy of CARDCount. As most of these confounding factors apply to all counting approaches, we discuss and empirically evaluate them—based on real world data and simulations—in Section IV.

## IV. Evaluation using ground truth

Our overarching goal with the CARDCount approach is to improve the accuracy with which botnet sizes are estimated, even in the presence of confounding factors such as short IP address durations, bot churn, etc. In this section, we begin by describing different factors that can affect the accuracy of botnet size estimation techniques. Next, we describe the RIPE Atlas dataset. We use this dataset for two purposes: to obtain IP address assignment duration distributions for ISPs and also as a source of ground truth against which we can evaluate the accuracy of different techniques. We then proceed to evaluate the accuracy of CARDCount for estimating botnet sizes under various settings and in the presence of confounding factors, and compare against state of the art approaches.

### A. Confounding factors for botnet size estimation techniques

**A1. Short IP address durations** Short IP address durations would lead prior techniques such as BinCount to overestimate the number of bots. This factor is the main motivator for alternate approaches such as CARDCount and MaxCount, that mitigate overestimation caused by IP reassignments.

**A2. Bot churn** Botnet size estimation techniques usually do not account for bot churn; specifically, they assume that all bots are online through the entire measurement period $T$. This factor can lead CARDCount and MaxCount to underestimate the number of bots. As an extreme example, consider a time window $T$ of 7 days and an average duration $d$ of 1 day, and suppose we observed $A = 7$ total IP addresses in that time. CARDCount would estimate that this constituted a single host. But with churn, it is possible that each of the 7 addresses was a single bot entering the system and then leaving before obtaining a new IP address. In a sense, what CARDCount computes is the weighted lifetime of active bots: although in this example there were 7 bots, each only lived for 1/7 of the duration window, resulting in a weighted average of 1 bot. MaxCount would also underestimate the number of bots in this case if their IP addresses are not observed simultaneously. Conversely, BinCount would not be affected by this confounding factor, since each observed IP address would be interpreted as a bot. Similar to MaxCount, CARDCount can limit the effect of bot churn by decreasing the size of the observation window. In our simple example, a time window of $T = 1$ would allow CARDCount to accurately estimate the botnet's size for every single day.

**A3. Capturing partial bot activity** Although botnet measurement efforts have become impressive over the years, measurements remain imperfect, and may not track all bots' IP addresses all the time. Recall that MaxCount tries to account for this by assuming that if an address was seen at measurement intervals $i - 1$ and $i + 1$ then it was probably also active at interval $i$. For CARDCount, it will lead to an underestimate only if we miss one of a bot's IP addresses entirely; so long as we see any IP address at least once, CARDCount properly accounts for it.

**A4. Accuracy of the address duration distributions** CARDCount makes use of additional datasets to ascertain the address assignment duration distributions, $D_k$. This introduces another potential source of error: how does CARDCount fare when the empirically measured distributions are not accurate? Note that BinCount and MaxCount are not affected by this confounding factor, since they do not model address assignment distributions. If the true mean of $D_k$ is larger than what is measured, then CARDCount will *underestimate* the botnet's size (because it will think that addresses are changing more frequently than they are). However, we show empirically that CARDCount is surprisingly robust to even large errors in the distributions.

**A5. Shared IP addresses** The scarcity of IPv4 addresses has led to a large scale adoption of Network Address Translation (NAT) in both home and carrier networks. This may lead to multiple bots sharing a single public IP address. Consequently, all IP based measurements, including CARDCount, will underestimate the actual number of bots. Furthermore, absent bot-unique identifiers, there exist no approaches to count the number of bots behind NAT. While this topic should clearly be addressed, its complexity [21] prevented us from including it in this paper. In §VI we discuss this topic in greater detail, and how similar concepts used for CARDCount could be applied to shared IP addresses.

In the rest of this section, we evaluate the extent to which each of these potential confounding factors actually impacts CARDCount's results in practice. In Section VI, we also discuss other potential limitations to CARDCount.

## B. The RIPE Atlas dataset

The RIPE Atlas[2] dataset provides reliable ground truth about the status of hosts and their IP address assignments. This dataset allows us to achieve two goals: it provides us with a source of empirical IP address distributions from ASes around the world and also enables us to measure the accuracy of CARDCount, BinCount, and MaxCount in a controlled setting (i.e., where the number of "bots" is known). We considered using other datasets as a source of ground truth; however, while some botnets implement botnet specific identifiers [10], [6], [2] or can be fingerprinted [8], these identifiers are volatile and cannot provide reliable ground truth.

*RIPE Atlas overview:* In 2018 The RIPE NCC's Atlas project deployed more than 10,000 hardware devices (called "probes") in volunteers' networks around the globe. All RIPE Atlas probes conduct periodic measurements, called "built-in" measurements, such as pings, traceroutes, and DNS measurements. The RIPE NCC makes efforts to deploy Atlas probes across diverse ASes in countries around the world and represents ASes in Europe and North America particularly well. Though its coverage in other parts of the world is lower, the number of probes has almost doubled since 2018. Moreover, the Atlas project actively aims to diversify its coverage, having at least one probe in $86.2\%$ of all countries in 2022. Furthermore, it remains one of the largest publicly available datasets of IP address durations, and as a result, has been used to shed considerable light on dynamic address assignment patterns and practices employed by various ASes [18], [19].

*Obtaining IP address durations and host-counts:* We obtained our IP address durations dataset—which serves as the input for CARDCount's address assignment duration distributions—from RIPE Atlas. We used the methodology described by Padmanabhan et. al [19] to obtain this dataset and we summarize it below. Every RIPE Atlas probe has a globally unique identifier ("probe-ID"). This identifier persists across reboots and is included in each measurement reported by the probe. For obtaining IP address durations, we harness the "IP echo" built-in measurement. RIPE Atlas probes are configured to automatically run "IP echo" measurements every hour towards a measurement server operated by RIPE. In each IP echo measurement, a RIPE Atlas probe executes an HTTP GET request to the RIPE-controlled measurement server, which in turn echoes back the IP address of the client as seen by the server. The response contains the publicly visible IP address of the client in the "X-Client-IP" field in the response header. The probe then reports this response, which contains its IP address and its unique probe-ID (and other fields such as the measurement-time), to a RIPE-controlled server that collects and processes measurements. By stitching together the IP addresses seen over time for each probe-ID, we are able to observe when clients' addresses change, and thereby arrive at the duration that each address is assigned.

The address available in the "IP echo" measurement is typically that of a Customer Premises Equipment (CPE) device (like a home router), especially in the residential ASNs that our study focuses on. Sometimes, the ASN in which the probe is housed may be using CG-NATs (Carrier-Grade NATs), in which case the address in the IP echo measurement is one of the addresses from the CG-NAT's address pool [21]. Prior work [18], [19] has shown that the prevalence of CG-NATs in the RIPE Atlas dataset is low, since the vast majority of ASes with more than 20 RIPE Atlas probes (we focus on these ASes in this work) are major residential ISPs in Europe and North America that are not using CG-NATs [21].

Since our datasets of the Hajime and Mirai botnet used in Section V are both from 2018, we obtained IPv4 address durations from RIPE Atlas for the same year. We follow the recommendations from Padmanabhan et al. [19] to filter probes deployed in atypical scenarios that can lead to the inference of false assignment changes. Specifically, we filter out probes that were observed for very short durations, probes that are multihomed (since inferred addresses changes on such probes could be spurious), probes that are not in residences (using user-provided and RIPE-provided tags), and probes deployed behind atypical NATs [19].

We obtained *probe-counts* from the Atlas dataset by counting the number of unique probe-IDs within the measurement duration $T$. Since every probe-ID is unique and persistent, the probe-count we obtain in this manner accurately represents the number of Atlas probes active during that time. Furthermore,

[2] https://atlas.ripe.net/

TABLE II: Size estimation for RIPE Atlas probes in 2018

| $T =$ | 1 day (std) | 7 days (std) | 28 days (std) |
|---|---|---|---|
| CARDCount | 0.98 (0.0056) | 0.96 (0.0090) | 0.91 (0.0102) |
| MaxCount$_{AS}$ | 1.00 (0.0011) | 0.99 (0.0024) | 0.97 (0.0039) |
| MaxCount | 0.99 (0.0017) | 0.98 (0.0035) | 0.95 (0.0057) |
| BinCount$_T$ | 1.14 (0.0107) | 2.06 (0.0417) | 5.12 (0.1165) |

to obtain reliable distributions $D$ we limited ourselves to ASes with probe-counts of at least 20 probes.

*Key advantages of the Atlas dataset:* The RIPE Atlas dataset has several characteristics that make it ideal for the evaluation of CARDCount. In contrast to botnet-ID-based datasets, RIPE Atlas probes have a unique, persistent identifier. Thus, we can infer probe counts accurately, and we use these probe counts as a source of ground truth against which we compare the size estimates based on IP addresses. Moreover, the individual probes actively contact the servers at regular intervals, whenever they are turned on and connected to the Internet, providing a reliable source of address durations.

### *C. Evaluation in different scenarios*

Next, we evaluate CARDCount, BinCount, and MaxCount against the ground truth Atlas dataset in the presence (and absence) of confounding factors and evaluate their accuracy. We define accuracy as the *estimate* of botnet size divided by the *actual* number of bots.

*1) Minimal confounding factors:* Our first experiment covers the size estimation of the RIPE ATLAS probe dataset without any modifications. The population of RIPE Atlas probes experiences relatively little churn. This behavior is very atypical of botnets [8], [10]. Nevertheless, measuring the accuracy of the approaches in this setting allows us to obtain a baseline for further experiments that simulate confounding factors on top of the RIPE Atlas dataset.

We compare three different measurement windows aligned to common human patterns of one day, one week, and one month ($T$ = 1d, 7d, 28d). We chose these values as they are most likely to capture human influenced diurnal and other churn patterns, i.e., if a machine is used once a day, a week or a month, which should be captured by the chosen intervals. Table II shows the accuracy and standard deviation of the different approaches.

Of these approaches MaxCount$_{AS}$ provides the best approximation of the total population in all three observation windows, followed by MaxCount, CARDCount, and BinCount in that order. This high accuracy is a result of the dataset collection process. As Atlas probes are stand-alone devices intended to provide continuous measurements, they typically remain active for long periods of time and are rarely turned off. Consequently, even probes that churn are typically active simultaneously at least once within the analyzed window sizes. Hence, both variants of MaxCount perform well in these conditions. However, as we will show later, MaxCount strongly drops in performance in the presence of more erratic churn behavior, which is common in botnet measurement [7].

CARDCount performs slightly worse than MaxCount in this scenario. We investigated the cause and found that CARDCount is more affected by the type of churn present in the RIPE Atlas dataset. While probes are active simultaneously at least once, churn causes the probes to generate fewer IP addresses than they normally would. As discussed in (A2) CARDCount tends to estimate the weighted lifetime of active bots if they are not active throughout the entire window $T$. This can be seen in Figure 3, which visualizes the results of Table II. It clearly shows that CARDCount closely matches the weighted average active population of the RIPE Atlas dataset. Moreover, we can see that the ground truth value lies within the confidence intervals of CARDCount.

Lastly, BinCount's accuracy is the worst, with more than $12\%$ overestimation, for even small observation windows of $T = 24h$. This is related to many ASes reassigning IP addresses within $24h$.

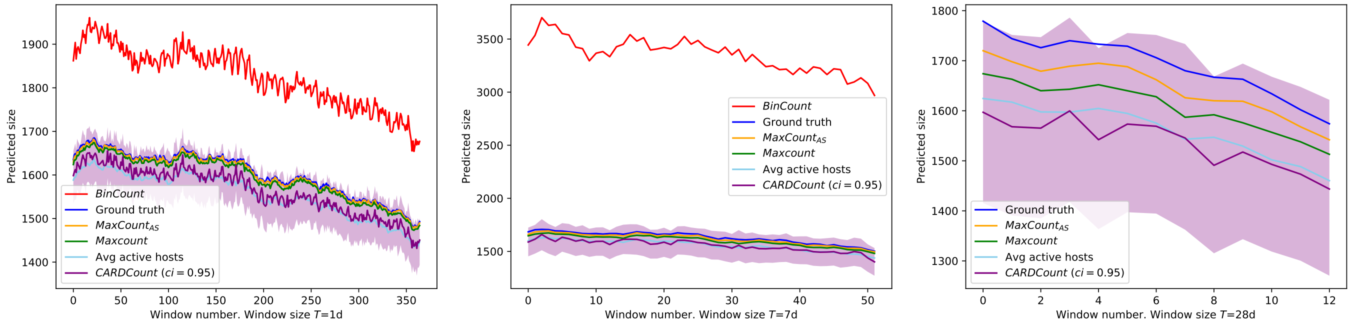
We also observe that *for all counting approaches the size estimation worsens for longer observation windows $T$*. For MaxCount this can be explained by a decreasing likelihood that all devices that were active over a month are all active at the same time. CARDCount's case is similar: bot churn increases over longer periods and some bots consequently do not generate IP addresses throughout the entire period (A1). For BinCount, the reassignment of IP addresses leads to an ever increasing count even if the number of hosts remains constant (or as in our case, when it decreases over time).

*2) Effect of short address durations:* We next examine how the address duration distributions within individual ASes affects the accuracy of various approaches. Intuitively, we would expect ASes with short assignment durations to adversely affect BinCount and for longer address durations to not have as much of an effect. Figure 4 shows how each approach's accuracy is affected by the mean IP address assignment duration per AS and confirms this intuition. BinCount's accuracy is worse when IP address assignment durations are short. CARDCount is also affected as shorter address duration in combination with host churn leads to larger underestimations (A1). In contrast MaxCount$_{AS}$ is not affected by the mean address duration.

*3) Partial Monitoring Coverage:* With this experiment we want to simulate the confounding factors of (A2) bot churn, and (A3) capturing partial bot activity. The most important differences between the RIPE Atlas dataset and real world botnet measurements are bot churn and discovery delays. Griffioen et al. [7] reported that most Mirai bots have a lifetime of a few hours, highlighting extreme amounts of churn. Similarly, measuring large botnets with telescopes or active crawling may easily delay the discovery of a newly infected bot by a few hours. Therefore, the amount of churn, specifically short term bot churn, in real world botnets is much higher than in the RIPE Atlas dataset. To simulate these high amounts of churn in combination with capturing only partial bot activity we did the following.

We chunked the RIPE Atlas dataset into one hour blocks and for each hour chose with a probability of $80\%, 50\%$ or $30\%$ if a given host was captured by our measurement during that hour. These values were chosen to represent a range from mild to heavier amounts of bot churn.

While a rudimentary simulation, it represents most common cases of bot churn, i.e,. devices being inactive for short to long durations, and perceived bot churn, i.e,. not capturing



(a) Size estimate with $T = 1d$

(b) Size estimate with $T = 7d$

(c) Size estimate with $T = 28d$

Fig. 3: Size estimation over varying observation windows $T$ with unaltered ATLAS probe data. The shaded area depicts the 95th confidence intervals of CARDCount. We omitted BinCount for $T = 28$d to keep the figure readable.

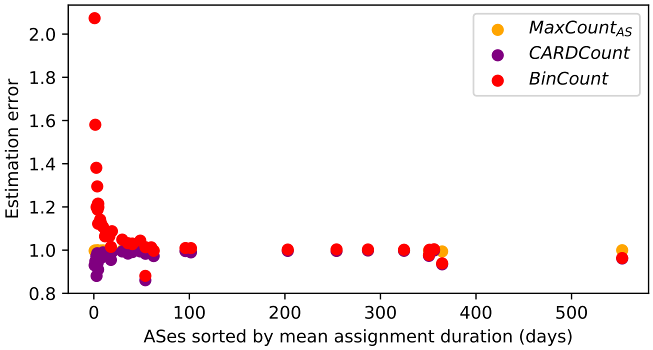


Fig. 4: Estimation error of counting approaches in relation to mean assignment durations. MaxCount is not affected, whereas CARDCount and BinCount are more accurate for higher mean assignment durations.

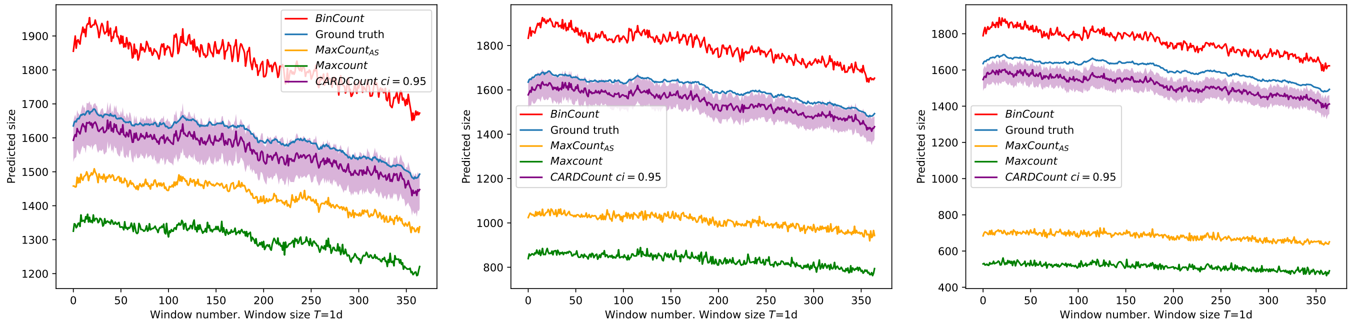
TABLE III: Accuracy with limited coverage per hour.

| | Bots/h | 1 day | 7 days | 30 days |
|---|---|---|---|---|
| CARDCount | 80% | 0.97 | 0.95 | 0.91 |
| | 50% | 0.96 | 0.95 | 0.90 |
| | 30% | 0.95 | 0.93 | 0.89 |
| MaxCount$_{AS}$ | 80% | 0.89 | 0.91 | 0.90 |
| | 50% | 0.63 | 0.67 | 0.67 |
| | 30% | 0.42 | 0.47 | 0.48 |
| MaxCount | 80% | 0.81 | 0.81 | 0.78 |
| | 50% | 0.52 | 0.52 | 0.50 |
| | 30% | 0.32 | 0.32 | 0.32 |
| BinCount | 80% | 1.14 | 2.05 | 5.12 |
| | 50% | 1.12 | 2.03 | 5.06 |
| | 30% | 1.10 | 1.98 | 4.96 |

the activity of a bot in a given time window. Furthermore, for a lesser probability of activity, bots may be inactive for longer periods of time, similar to diurnal patterns. Other factors such as not observing a bot at all, is a problem of the actual measurement, rather than the applied data analysis approach. We also did not simulate IP addresses being reassigned to multiple bots, as we lack a proper model. However, analysing the data available to us, this is not a frequent occurrence.

The results of this experiment are presented in Table III. We can see that both CARDCount and BinCount are almost not affected at all by the changes, even at an hourly coverage as low as $30\%$. This is because both approaches take as input only whether an IP was seen in a given observation window or not. Therefore, even if we only see $30\%$ of the botnet in a given hour, we will likely see the assigned IP address at least once throughout the day. Given that the shortest known reassignment durations are $4$ hours [8] and more commonly $12$ or $24$ hours [18], we have a probability of $75.99\%$ (4h), $98.62\%$ (12h), and $99.98\%$ (24h) to see the bot once while its IP address was assigned. MaxCount and MaxCount$_{AS}$ perform much worse in this scenario. This is to be expected, as even within an AS, only a small fraction of bots will be observed at the same time, limiting the number of bots that can be observed active simultaneously by MaxCount$_{AS}$. Interestingly, MaxCount$_{AS}$ covers a much higher fraction of the total population than the fraction of the population seen per hour, e.g., 89.2% for a coverage of $80\%$ over an observation window of one day. The reason for this is the focus on individual ASes. While it is nearly impossible to observe 89.2% of the total population given the odds of including a host with $80\%$ probability, the probability of seeing more than 80% of smaller ASes in an hour is quite likely. Since MaxCount$_{AS}$ computes the sum of peaks of each individual AS, the peaks can occur at different times throughout the observation window. Therefore, a total sum exceeding the expected value of $80\%$ becomes likely as seen in these experiments.

Figure 5 provides a more in-depth look for $T = 1d$, which is the most common observation window observed in Table I. It once again shows that CARDCount and BinCount are only minimally affected by the reduced coverage. In comparison, both variants of MaxCount strongly underestimate the ground truth population. Interestingly, the two widely applied estimators, BinCount and MaxCount, are also the most inaccurate estimators of population size, in settings with IP address churn, device churn, and limited monitoring coverage.



(a) Botnet coverage 80%/hour 1d (b) Botnet coverage 50%/hour 1d (c) Botnet coverage 30%/hour 1d

Fig. 5: Size estimation over one day windows and varying fractions of the hosts covered by the simulated measurement. CARDCount outperforms the state of the art in all three scenarios of limited coverage.

The high amounts of churn simulated by coverage of $50\%$ and $80\%$ lead to the ground truth being outside the confidence intervals of CARDCount. This is because the confidence intervals address errors of sampling from the distribution $D$, but just like all other approaches cannot account for unknown confounding factors such as bot churn. However, even for the most extreme case of churn CARDCount's accuracy only drops by 3.03%, compared to MaxCount$_{AS}$, which drops in accuracy by $57.27\%$. This highlights that CARDCount retains a high level of accuracy regardless of IP reassignments or bot churn.

*4) Variations in Session Lengths:* Lastly, we analyze how CARDCount is affected when the underlying address duration distributions are inaccurate (A4). Recall that this confounding factor only affects CARDCount.

To recapitulate, the performance of CARDCount relies on knowing the address assignment durations of a given AS. We retrieve this information from the ATLAS probe data and apply it to the ATLAS probe data. However, in practice the *input distribution* $D$ and the (commonly unknown) address assignment distribution of the monitored population, referred to as *actual distribution* $\hat{D}$, could deviate. A deviation in the mean of those two distributions will affect the accuracy of size estimations made by CARDCount. To understand how much discrepancy is acceptable, we artificially increase or decrease the duration of observed IP address assignments. To do this, we added the following modification to each individual session in the input distribution $\{d \pm 2h, d \pm 12h, d + 24h\}, \forall d \in D$, to generate a modified $\hat{D}$. If the modification causes an IP address assignment to have a negative duration, we set it to zero instead. Afterward we recompute CARDCount with $\hat{D}$ as input, but measure the IP addresses of the unmodified RIPE Atlas dataset. It is important to note that $d + 24h$ doubles the IP address duration for many of the ASes in the dataset. Furthermore, we did not evaluate $d - 24h$ as this led to a majority of durations being set to zero.

Figure 6 presents the results of this experiment. The experiments show that smaller differences of $\pm 2h$ in the input and actual distributions do not lead to strong deviations in CARDCount's predictions. An interesting observation is that shortened input distributions lead to greater deviations than longer input distributions. This is because the overlap of $d$ and $t$ grows slower with increasing $d$. Increasing the IP address duration by $24h$ or decreasing them by $12h$ leads to over- and underestimations with the unmodified CARDCount being outside the respective 95th percentiles. Interestingly, for the $24h$ increase the ground truth is within the 95th percentile for windows sizes of 7d and 28d. This is related to CARDCount underestimating the ground truth in the unmodified RIPE Atlas dataset. Consequently, the overestimation introduced by the increased assignment durations cancels out the initial underestimation caused by churn and duplicate assignments.

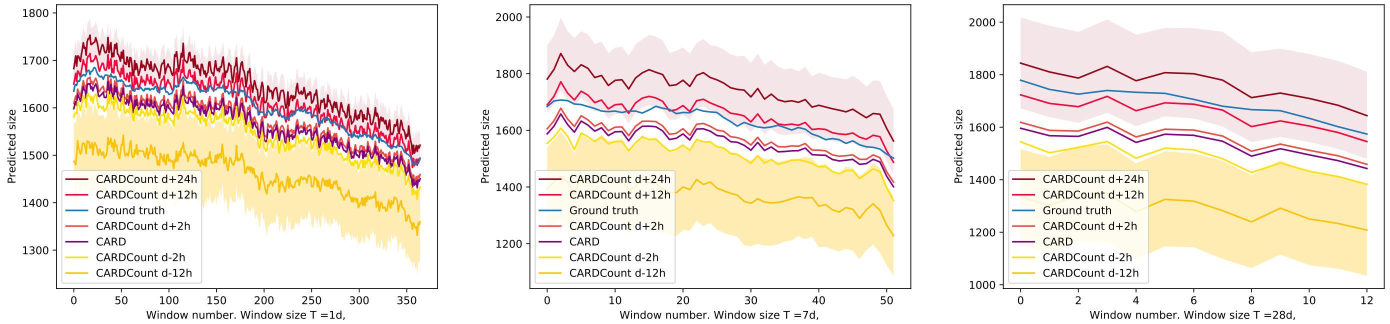
Concluding this experiment, we learned that CARDCount remains highly accurate in the presence of even large deviations between input and actual distributions. Furthermore, the margin of error is larger if the input distribution overestimates the actual mean IP address durations.

### D. Summary

In this section, we compared CARDCount against the state of the art and analyzed the impact of confounding factors on all counting approaches. We found that the accuracy of BinCount—the most commonly used Botnet size estimation technique—is unfortunately the lowest, as it overestimates considerably due to IP address reassignments. MaxCount performs best if the amount of churn is minimal. However, its accuracy dropped tremendously in more realistic scenarios of higher churn and limited capturing of bot activity. CARDCount was accurate throughout all scenarios. Most notably, it performs very well in high churn scenarios. Moreover, almost all confounding factors lead to underestimations by CARDCount. Therefore, CARDCount is very unlikely to overestimate the actual population and can be considered a lower bound. The only reason for CARDCount to overestimate is an overestimation of the mean address duration in $D$. However, even then there is a wide margin of error, specifically if other confounding factors leading to underestimation are present.

## V. Size Estimation of Real-World Botnets

In the previous section, we evaluated CARDCount in a highly stable scenario of long-running measurement endpoints.



(a) $\omega = 1d$

(b) $\omega = 7d$

(c) $\omega = 28d$

Fig. 6: CARDCount mean session length deviations, with differing input ($D$) and actual address assignment distributions ($\hat{D}$).

Although not representative of the Internet as a whole, these long-running nodes enabled us to artificially cut up long running traces, and thereby allowed us to evaluate how the established algorithms and CARDCount perform under different observation periods and how their performance is affected by confounding factors.

In this section, we apply CARDCount against two real world botnets, Hajime and Mirai. We obtained datasets from Herwig et al. [10] and Griffioen et al. [8]; these datasets measured which hosts were infected by the Hajime and Mirai botnets, respectively. Using these datasets, we evaluate and compare the performance of counting algorithms in practice.

### A. Datasets

*Hajime.* The Hajime botnet was assessed by Herwig et al. [10], primarily by crawling the BitTorrent Distributed Hash Table (DHT) to find bots that provide Hajime configuration files or platform-specific malware files for download. Additionally, the researchers passively identified infected hosts, by advertising some botnet-specific files within the network to identify bots trying to download these Hajime files. Within the study we only utilize the active bot identification data. While Hajime bots choose an installation-specific identifier we could not leverage it as ground truth, as it frequently changes upon reboot and is sometimes more volatile than IP addresses.

While the entire dataset for the Hajime botnet spans a period of five months from January until May 2018, there were several updates to the botnet in that time frame. An update to the Hajime botnet causes its bots to restart, leading to changes in the bot's port and ID. Therefore we chose to take a 28-day subset of the data ranging from February 22nd to March 21st, for which there was uninterrupted coverage and no updates to the botnet. This dataset encompasses 2,254,532 IP addresses spread across 2,412 ASes. Within this set, 33,486 IP addresses (1.5%) were observed within 36 ASes in which there are at least 20 hosts in the RIPE Atlas dataset. We focus in the evaluation on these 33k IPs, as we can rely on the RIPE Atlas dataset to obtain $D$ and validate the performance of the algorithms.

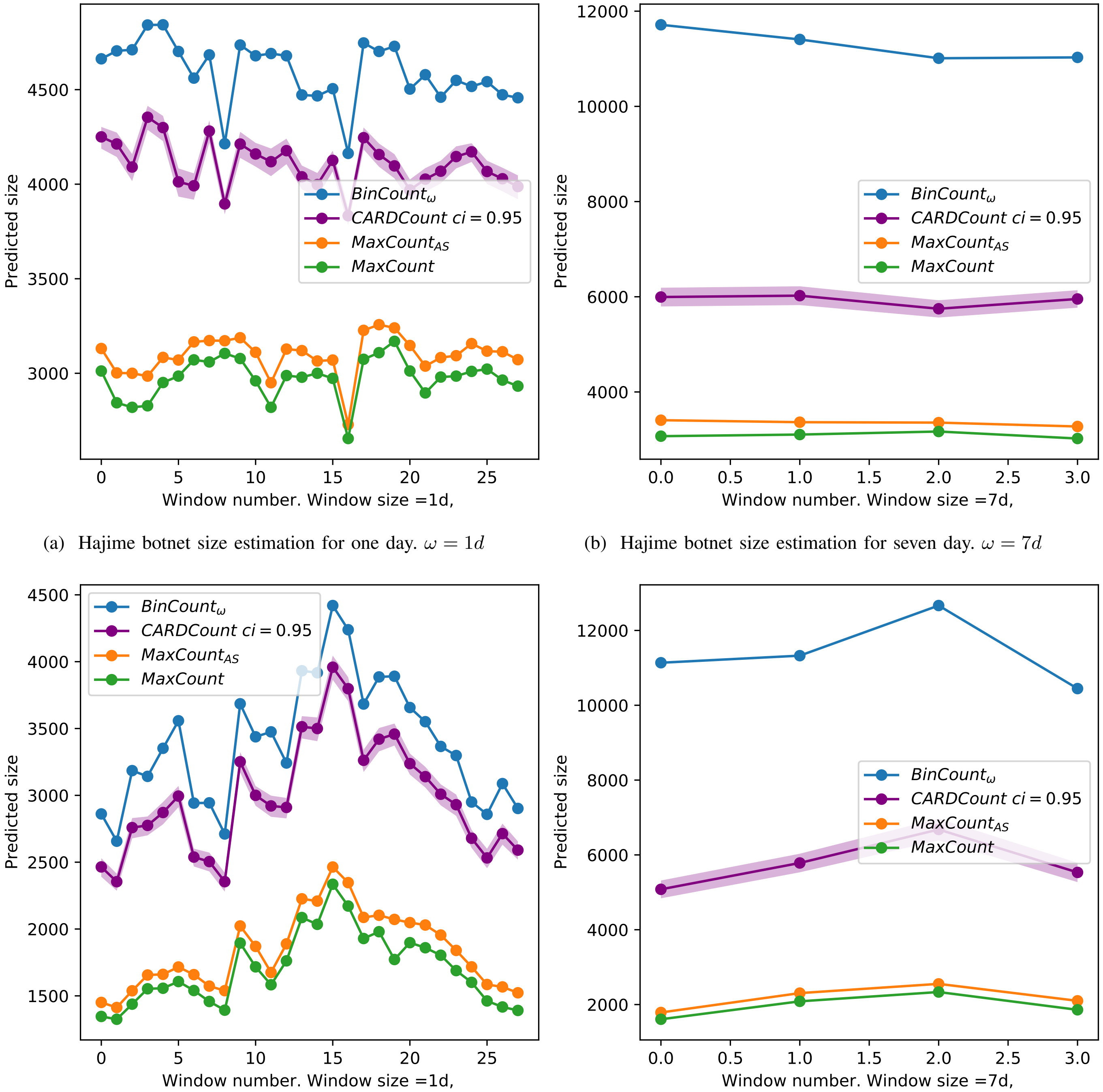
As referenced in the original paper for the dataset, an exhaustive crawl was conducted every 16 minutes. Based on the recommendations of Yan et al. [29], we set the timeout $\tau$ to identify two separate sessions for MaxCount to ten times the crawl interval, i.e., 160 minutes.

*Mirai.* The Mirai dataset was made available by Griffioen and Doerr [8], [7]. As after an infection, Mirai bots immediately start to scan the Internet to identify and infect other vulnerable devices, it is possible to track Mirai-infected hosts by collecting this probing traffic. To do so, these studies utilized a large network telescope of 65,000 IP addresses, tracking various Mirai families based on their attack traffic. To match the time frame of data collection for Hajime and the RIPE Atlas probe dataset, the Mirai dataset is limited to data from February 2018.

The Mirai dataset identifies 1,052,552 infected IP addresses distributed over 6,418 ASes. Also, Mirai contains an infection-unique identifier, which we can use to identify infections across IP addresses. From this set, 40,609 IP addresses (3.9%) were located in 38 ASes which contained at least 20 measurement endpoints in the RIPE Atlas platform.

As the Mirai dataset was collected in a passive measurement, we cannot set the $\tau$ for MaxCount based on the crawl frequency. Griffioen et al. report that if a bot sends 25 packets per second, 95% of all bots should contact their telescope within two hours. Moreover, they report an average time between receiving two packets of 421 seconds. Based on these values, we chose to set $\tau$ to the larger value of two hours to allow a fair comparison of MaxCount, even for slow hosts.

Lastly, as the Mirai dataset is a passive measurement, we observed bot fingerprints that changed IP addresses at an unusually high frequency. Upon further investigation, we could identify them to be part of a Carrier Grade NAT (CG-NAT). As these IP addresses in CG-NATs differ from regular IP address assignment patterns, we filtered them from the dataset. To do this, we identified all Mirai fingerprints that had at least three IP addresses assigned to them for less than 10 minutes. Based on these IP addresses, we derived and filtered all /24 network ranges that contained bots behind a CG-NAT. This reduced the set of IP addresses from 40,609 to 37,389 in the Mirai dataset.



(a) Hajime botnet size estimation for one day. $\omega = 1d$

(b) Hajime botnet size estimation for seven day. $\omega = 7d$

(c) Mirai botnet size estimation for one day. $\omega = 1d$

(d) Mirai botnet size estimation for seven day. $\omega = 7d$

Fig. 7: Size estimations for the Hajime and Mirai botnets.

TABLE IV: Size estimation of Hajime

| | $T$ =1d (avg) | $T$ =7d (avg) | $T$ =28d |
|---|---|---|---|
| $CARDCount$ | 4108 | 5930 | 7269 |
| MaxCount$_{AS}$ | 3096 | 3351 | 3521 |
| MaxCount | 2974 | 3092 | 3169 |
| BinCount$_T$ | 4592 | 11,290 | 33,496 |

TABLE V: Size estimation of Mirai

| | $T$ =1d (avg) | $T$ =7d (avg) | $T$ =28d |
|---|---|---|---|
| $CARDCount$ | 2980 | 5769 | 8654 |
| MaxCount$_{AS}$ | 1837 | 2187 | 2673 |
| MaxCount | 1700 | 1972 | 2335 |
| BinCount$_T$ | 3387 | 11,396 | 37,389 |

### B. Applicability of IP Address Duration Distributions

To understand whether CARDCount can be applied to these botnets, we first ask: Do the bots in these botnets experience similar address assignment durations as the underlying RIPE Atlas dataset we use in CARDCount?

In the Appendix, we provide a thorough comparison of the address assignment distributions of Hajime and Mirai as compared to RIPE. At a high level, our results show that the distributions are close—a mean difference of 8.6h for Hajime and 16h for Mirai—and that these differences are well within CARDCount's resilience to errors in the underlying distribution $D$ that we showed in §IV. Therefore, the size estimations of CARDCount should be highly accurate in comparison to MaxCount and BinCount, which are both heavily affected by confounding factors.

### C. Estimating Hajime

Table IV and Figures 7 (a) and (b) present the results of the size estimation of Hajime for daily, weekly, and monthly observation windows. As a first observation, we can see that the size estimates grow for larger observation windows. This indicates that even if the size estimates for a single day do not exceed 4358 infections for CARDCount and 3316 infections for MaxCount, the total number of infections over a week or month are larger than on any single day. This observation matches the reports of Stone-Gross et al. [26] for the Torpig botnet. This is a clear sign of the presence of churn expected in the Hajime botnet, as infections do not persist, and new machines are actively infected at all times. Therefore, infections on different days are likely to be different devices leading to greater estimates for longer observation windows. Even though the population seems to churn frequently, there are only moderate daily fluctuations for CARDCount (11.9%) and MaxCount (7.9%). Moreover, we can not observe any repeating patterns, e.g., weekly repetitions, for the daily size estimates. Such repeating patterns have been previously reported for Windows-based botnets [9].

The second observation is the considerable difference between the measurement approaches. For BinCount we can clearly observe a stark increase in overestimation with increasing observation window sizes. Similarly, we can observe that the presence of churn causes MaxCount to provide a drastically smaller estimate than CARDCount. Considering the small difference in IP durations (c.f. §V-B) and CARDCount's resilience to even large amounts of bot churn, it represents the most accurate estimate available. Based on this, we provide rough estimates for the actual error of MaxCount$_{AS}$ and BinCount by comparing them to the mean value of CARDCount.

We compute the estimation error as the percent difference between two estimates, $E_1$ and $E_2$:

$$\frac{|E_1 - E_2|}{\frac{1}{2}(E_1 + E_2)} * 100$$

For MaxCount$_{AS}$ this indicates underestimations of the bot population by 24.6%, 43.5%, and 51.6% for observation windows of 1, 7, and 28 days respectively. For BinCount, we assess overestimates by 11.8%, 90.4% and 360.8%. While these numbers may deviate by a few percentage points, the results clearly resemble the results of our experiments conducted on the RIPE Atlas probe dataset. Therefore, CARDCount provides a crucial improvement in the accuracy of estimating botnet sizes.

### D. Estimating Mirai

Figures 7 (c) and (d) and Table V present the size estimations of the Mirai botnet using CARDCount and MaxCount. Again, we can see that the size estimates grow for larger observation windows. This indicates that even if the size estimates for a single day do not exceed 4024 infections for CARDCount and 2451 infections for MaxCount, the total number of infections over a week or month is larger than on any single day. This is to be expected as Mirai infections do not persist and actively infect new machines at all times. Therefore, infections on different days are likely to be different devices leading to greater estimates for longer observation windows. Another observation similar to Hajime is that there are large daily fluctuations for CARDCount (36.8%) and MaxCount (41.1%). In contrast, the weekly fluctuation is only 18.0% for CARDCount and 27.5% for MaxCount. Once again, we can not observe any repeating patterns, e.g., weekly repetitions, for the daily size estimates. Lastly, the larger deviations across time windows can be explained by the finding of Griffioen et al. [7], that some Mirai variants are very unstable and frequently crash. This leads to very short infection durations for these variants.

Following the same arguments stated in the section on Hajime, the data available to us shows that CARDCount should provide the most accurate estimation for the size of Mirai. Based on this, comparing MaxCount$_{AS}$ and CARDCount indicates an underestimation of the bot population by a percent difference of 38.4%, 62.1%, and 69.1% for observation windows of $T =$ 1d, 7d, 28d, respectively. While these results are even worse for MaxCount$_{AS}$ than for Hajime, this can be explained by the greater volatility and churn of Mirai [7]. BinCount once again overestimates at similar ratios to RIPE Atlas and Hajime with 13.8%, 97.5%, and 332.0%. These results once again show that CARDCount provides a significant improvement in accurately estimating a botnet's size.

## VI. Discussion

In the previous sections, we introduced CARDCount and evaluated it together with the state of the art counting mechanisms on three real world datasets. The goal of this section is to discuss the lessons we learned and how to apply them to future botnet measurements. We also discuss the potential limitations of CARDCount.

### A. *Importance of Estimating Botnet Size: Accurate size estimations enable adequate response.*

The size of a botnet oftentimes influences the attention and resources defenders allocate to defend against it. Knowing how many devices are infected, how many users' credentials were stolen, or estimating the capacity of a Distributed Denial of Service (DDoS) attack enables defenders to make better decisions. Furthermore, precisely knowing the size of a botnet, and being able to measure if it is growing or shrinking is essential to judge the (cost-)effectiveness of countermeasures. But as we show in this paper, the state of the art is only accurate for time windows of a few hours. As reported previously by Stone Gross et al. [26] and shown in §V, even 24 hours are insufficient to capture the full size of a botnet. To make things worse, the error of both MaxCount and BinCount is volatile and dependent on the IP reassignment frequency (BinCount) and amount of churn (MaxCount) in the botnet. Therefore, estimates are incomparable between botnets. This is exacerbated by the fact that botnets often exhibit biases towards which ASes they infect, for instance, if a vulnerable device is rolled out en masse by an ISP or service provider, or especially popular in a certain region [10]. Consequently, if the population of the two botnets differs significantly in their churn and IP reassignment characteristics, one may appear much larger than the other, even if it is not. CARDCount addresses these issues by providing an accurate and less volatile size estimate for botnets, which enables defenders to make better decisions in prioritizing and addressing botnet threats.

### B. *Best Practices for Botnet Size Estimations: Size estimations are always context sensitive.*

Our evaluations have shown that CARDCount outperforms the state of the art counting approaches in the presence of IP address reassignment and bot churn, which are present in all known botnet measurements. Nevertheless, in the unlikely case that there is little to no bot churn, CARDCount performs slightly worse than MaxCount$_{AS}$. Therefore, we suggest applying the following guidelines to future botnet size estimation:

- Consider the characteristics of the measured botnet, e.g., if there are large amounts of churn, CARDCount outperforms MaxCount$_{AS}$, whereas MaxCount$_{AS}$ performs better in low churn scenarios. If in doubt, reporting all three counting mechanisms, i.e., BinCount, MaxCount$_{AS}$, and CARDCount, will help comparison between botnets and reports.
- Consider the availability and applicability of the input distribution $D$. We recommend comparing the distribution $D$ to the IP address durations observed in the botnet measurement $\hat{D}$. A detailed example of this comparison is included in Appendix A.
- If $D$ is unavailable or inapplicable for an AS, we recommend falling back to MaxCount$_{AS}$, as it is more accurate that BinCount in most cases.

### C. *Other Applications*

Within this paper, we only applied CARDCount in the context of botnet size estimation. However, the concepts of CARDCount are generally applicable to any scenario involving the counting of hosts on the Internet based on IP addresses. As such, it could also be used to count the active hosts in a peer-to-peer network, count malware infections, or active users in a webservice if no alternative means (such as logins) are available.

### D. *Limitations*

In Section IV, we discussed several confounding factors for estimating botnet sizes, and we showed empirically that they do not significantly affect CARDCount's accuracy in practice. We close this section by discussing two other potential limitations: one specific to CARDCount (the availability of address assignment distributions) and another that affects *all* IP-based techniques (shared IP addresses).

*1) Availability of Input Distributions:* Given accurate IP address assignment distributions, CARDCount is the least volatile and most accurate approach to estimate the size of botnets in practice. Unfortunately, these IP address assignment durations are not yet available for all ASes. At the time of our study, the RIPE Atlas dataset provided sufficient data for 39 ASes, covering 1.49% and 3.9% of all IPs in Hajime and Mirai. This small coverage was a result of the uneven deployment of RIPE Atlas in its initial stages, with countries in North America and Europe hosting the majority of probes. However, in recent years, RIPE Atlas has actively been increasing the number of probes. Just as importantly, it has been diversifying probe locations by distributing new probes in under-represented countries and ASes. At the time of writing in 2022, the number of probes has nearly doubled since the time of the botnet measurements [4]. These probes now cover 4.941% of all ASes and 86.224% of all countries. Specifically, we observe that RIPE has these many probes in Hajime's 10 most-infected countries [10]: Brazil (91), Iran (103), Mexico (29), China (73), India (139), South Korea (27), US (1637), Turkey (53), Russia (604), Indonesia (65). This shows broad coverage even in bot-heavy locations.

Moreover, to date, there has been little incentive to collect and share IP address assignment information. Fortunately, there are other potential sources for $D_k$ beyond RIPE Atlas. For example, UDmap [28] used Hotmail user login traces to study dynamic addressing properties, and Casado et al. used HTTP cookies available from CDN datasets [5]. Ideally, ISPs would share their reassignment policies. While the incentives for ISPs to do so have been lacking in the past, our paper provides a tangible benefit to the ISPs themselves: more accurate estimates of botnet size counts can lead to better mitigation strategies for the ISP. So while we agree that $D_k$ coverage is not perfect, it is growing, and publication might in and of itself help improve it. As part of this work, we make per AS assignment distributions available for public download as well as source code to generate these from the Atlas data for future use at https://github.com/CardCount.

If the necessary distributions remain unavailable for some ASes, we recommend using MaxCount$_{AS}$ as a fallback. We assume, that $D_k$ will eventually be available for big ASes, limiting the lower accuracy of MaxCount$_{AS}$ to smaller ASes with likely lower infection numbers.

*2) Shared IP Addresses (NAT and VPN):* As we mentioned in §IV, CARDCount, like all other IP based counting methods cannot count bots sharing IP addresses. Two common forms of address sharing are NAT and VPNs. When two hosts share an IP address, any IP-based technique risks underestimating the true botnet size. However, while IP address reassignments require us to *filter* noise from the collected dataset, there is currently no generic way to identify shared IP addresses in the first place. An exception to this is the availability of additional identifiers, that if simultaneously active at the same IP provide insight into IP sharing. In practice, such identifiers cannot be assumed to be available.

Another important factor is that shared IP addresses may coincide with IP address reassignments. Consider that three bots share a public IP address. Even if we can identify the three bots, each IP reassignment would cause us to count three additional bots with each new IP address. Therefore, any solution should be paired with an approach like CARDCount.

What complicates the confounding factor of shared IP addresses is the variety of implementations. NAT in customer premises enables sharing of the public IP address among multiple devices in the home. In the case of IP reassignment, these bots will always be grouped behind the same reassigned public IP. Apart from customer NAT, some Internet Service Providers (ISPs) have started to deploy NAT at a network-wide level—so-called Carrier Grade NAT (CG-NAT)—grouping several customers behind a single public IP address. In a CG-NAT IP reassignments can vary from changing with subsequent connection attempts, being stable for individual conversations, or change similar to regular IP address reassignments [21]. Furthermore, bots that are grouped at one point, may be reassigned to two or more different IP addresses. Yet another cause of IP address sharing is VPNs, in which multiple clients make use of the same VPN proxy server (and thus appear to have that VPN server's public IP address). To the best of our knowledge, little is known about this type of shared IP addresses, and how they may change over time. These complexities and differences require a deep understanding on the configuration and extent of CG-NAT in practice, to properly address the confounding factor of shared IP addresses.

The Hajime and Mirai datasets shed light on the extent to which IP address sharing takes place. Specifically, we look at the number of *concurrent infections* in which a single IP address reported two or more bot IDs in an overlapping fashion (i.e., a given IP address identified as bot ID $A$, then ID $B$, then $A$ again). These concurrent infections indicate IP address reuse by two different bots (due to NAT, CG-NAT, or VPN), because both Hajime and Mirai choose bot IDs randomly, and are very unlikely to ever choose the same ID twice.

In the Hajime dataset, we found that 214,686 IP addresses out of the total of 2,254,532 (9.5%) exhibited concurrent infections. Address sharing was not limited to a small number of networks; we found that 1,626 out of 2,412 ASes (67.4%) had at least one concurrent infection. Similarly, Griffioen et al. [8] reported that, for Mirai, 9,370 out of 12,112 ASes (77.4%) had concurrent infections.

These numbers highlight the importance of addressing this topic in future work. Specifically, two things should be addressed: i) means to count bots sharing an IP address without additional identifiers, and ii) models of IP assignment and sharing in CG-NATs, similar to models of how they reassign IP addresses. Provided with these means, we are hopeful that the general approach of CARDCount—reverse-engineering the address assignment policy to infer host count—could apply in CG-NATs, and be combined with CARDCount to provide a holistic approach to accurate botnet size estimation.

## VII. Conclusion and Future Work

Within this paper, we studied and addressed the inaccuracies in IP based botnet size estimation caused by IP address reassignments.

We introduce CARDCount and show that knowledge about the IP address assignment durations and policies of ASes can be leveraged to provide more accurate size estimations in the presence of IP address churn. This greatly improves our abilities to accurately estimate the size of botnets using IP addresses, specifically in the presence of heavy churn and over longer periods of data collection. Interestingly, we found that the two most common estimators, BinCount and MaxCount, are also the most inaccurate estimators of population size, in settings with IP address churn, device churn, and limited monitoring coverage.

While CARDCount provides large improvements in accurately estimating a botnet's size, there are two open problems that should be addressed in future work. To increase the applicability of CARDCount in more ASes, additional means to obtain IP address assignment durations of ASes should be investigated. Furthermore, to obtain a complete picture, we need to investigate generic approaches to identify if, and how many bots share a NAT gateway.

## Acknowledgments

This research work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE. This work was also supported in part by NSF grants CNS-1901325 and CNS-1943240.

## References

[1] D. Andriesse, C. Rossow, and H. Bos, "Reliable Recon in Adversarial Peer-to-Peer Botnets," in *ACM Internet Measurement Conference (IMC)*, 2015.

[2] D. Andriesse, C. Rossow, B. Stone-Gross, D. Plohmann, and H. Bos, "Highly resilient peer-to-peer botnets are here: An analysis of Gameover Zeus," in *International Conference on Malicious and Unwanted Software (MALWARE)*, 2013.

[3] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the Mirai Botnet," in *USENIX Security Symposium*, 2017.

[4] R. Atlas. (2022) Network coverage. [Online]. Available: https://atlas.ripe.net/results/maps/network-coverage/

[5] M. Casado and M. J. Freedman, "Peering Through the Shroud: The Effect of Edge Opacity on IP-Based Client Identification," in *Symposium on Networked Systems Design and Implementation (NSDI)*, 2007.

[6] N. Falliere, “Sality: Story of a peer-to-peer viral network,” *Rapport technique, Symantec Corporation*, vol. 32, 2011.

[7] H. Griffioen and C. Doerr, “Examining Mirai’s Battle over the Internet of Things,” in *ACM Conference on Computer and Communications Security (CCS)*, 2020.

[8] ——, “Quantifying Autonomous System IP Churn using Attack Traffic of Botnets,” in *International Conference on Availability, Reliability, and Security (ARES)*, 2020.

[9] S. Haas, S. Karuppayah, S. Manickam, M. Mühlhäuser, and M. Fischer, “On the Resilience of P2P-based Botnet Graphs,” in *IEEE Conference on Communications and Network Security (CNS)*, 2016.

[10] S. Herwig, K. Harvey, G. Hughey, R. Roberts, and D. Levin, “Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet,” in *Network and Distributed System Security Symposium (NDSS)*, 2019.

[11] T. Holz, M. Steiner, F. Dahl, E. W. Biersack, and F. C. Freiling, “Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm,” in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[12] B. B. Kang, E. Chan-Tin, C. P. Lee, J. Tyra, H. J. Kang, C. Nunnery, Z. Wadler, G. Sinclair, N. Hopper, D. Dagon, and Y. Kim, “Towards Complete Node Enumeration in a Peer-to-Peer Botnet,” in *ACM Symposium on Information, Computer and Communications Security (ASIACCS)*, 2009.

[13] C. Kanich, K. Levchenko, B. Enright, G. M. Voelker, and S. Savage, “The Heisenbot Uncertainty Problem: Challenges in Separating Bots from Chaff,” in *USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET)*, 2008.

[14] S. Karuppayah, *Advanced Monitoring in P2P Botnets - A Dual Perspective*. Springer, 2018.

[15] S. Karuppayah, L. Böck, T. Grube, S. Manickam, M. Mühlhäuser, and M. Fischer, “SensorBuster: On Identifying Sensor Nodes in P2P Botnets,” in *International Conference on Availability, Reliability, and Security (ARES)*, 2017.

[16] S. Karuppayah, E. Vasilomanolakis, S. Haas, M. Mühlhäuser, and M. Fischer, “BoobyTrap: On Autonomously Detecting and Characterizing Crawlers in P2P Botnets,” in *IEEE International Conference on Communications (ICC)*, 2016.

[17] L. McLaughlin, “Bot Software Spreads, Causes New Worries,” *IEEE Distributed Systems Online*, vol. 5, no. 6, 2004.

[18] R. Padmanabhan, A. Dhamdhere, E. Aben, kc claffy, and N. Spring, “Reasons Dynamic Addresses Change,” in *ACM Internet Measurement Conference (IMC)*, 2016.

[19] R. Padmanabhan, J. P. Rula, P. Richter, S. D. Strowes, and A. Dainotti, “DynamIPs: Analyzing address assignment practices in IPv4 and IPv6,” in *ACM Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, 2020.

[20] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, “My Botnet Is Bigger Than Yours (Maybe, Better Than Yours): Why Size Estimates Remain Challenging,” in *Workshop on Hot Topics in Understanding Botnets*, 2007.

[21] P. Richter, F. Wohlfart, N. Vallina-Rodriguez, M. Allman, R. Bush, A. Feldmann, C. Kreibich, N. Weaver, and V. Paxson, “A Multi-Perspective Analysis of Carrier-Grade NAT Deployment,” in *ACM Internet Measurement Conference (IMC)*, 2016.

[22] D. S. Roselli, J. R. Lorch, and T. E. Anderson, “A Comparison of File System Workloads,” in *USENIX Annual Technical Conference*, 2000.

[23] C. Rossow, D. Andriesse, T. Werner, B. Stone-Gross, D. Plohmann, C. J. Dietrich, and H. Bos, “SoK: P2PWNED - Modeling and Evaluating the Resilience of Peer-to-Peer Botnets,” in *IEEE Symposium on Security and Privacy*, 2013.

[24] S. Saroiu, P. K. Gummadi, and S. D. Gribble, “Measuring and Analyzing the Characteristics of Napster and Gnutella Hosts,” *Multimedia Systems*, vol. 9, no. 2, pp. 170–184, 2003.

[25] B. Stock, J. Göbel, M. Engelberth, F. C. Freiling, and T. Holz, “Walowdac - Analysis of a Peer-to-Peer Botnet,” in *European Conference on Computer Network Defense, (EC2ND)*, 2009.

[26] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. A. Kemmerer, C. Kruegel, and G. Vigna, “Your Botnet Is My Botnet: Analysis of a Botnet Takeover,” in *ACM Conference on Computer and Communications Security (CCS)*, 2009.

[27] D. Stutzbach and R. Rejaie, “Understanding Churn in Peer-to-Peer Networks,” in *ACM Internet Measurement Conference (IMC)*, 2006.

[28] Y. Xie, F. Yu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber, “How Dynamic are IP Addresses?” in *ACM SIGCOMM*, 2007.

[29] J. Yan, L. Ying, Y. Yang, P. Su, and D. Feng, “Long Term Tracking and Characterization of P2P Botnet,” in *International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, 2014.

## Appendix

In this appendix, we describe how we compare the input distributions $D$ taken from RIPE Atlas, against the actual distribution $\hat{D}$ in the Hajime and Mirai botnets. As the actual distribution $\hat{D}$ is not readily available, we need a reliable way to measure IP address durations in the botnet datasets. Simply taking the duration for which we observed an IP address in the botnet would underestimate the actual IP duration, as i) the IP may be assigned before or longer than the bot is infected, and ii) our measurements may not immediately discover or track the bot from its infection to inactivity. Therefore, we need a more accurate way to measure the actual IP duration for bots.

For Hajime and Mirai we do this by leveraging the IDs and fingerprints of the botnets. While in each case IDs and fingerprints change upon reboot, they provide a unique identifier for continuous periods of bot activity. This allows us to observe multiple IP address changes for a constant ID or fingerprint. If we see at least three IP addresses $IP_A$, $IP_B$, and $IP_C$, we can compute the minimum IP duration $min(IP_B)$ as the time between the first and last contact with $IP_B$. Similarly, we can estimate the maximum IP duration $max(IP_B)$ as the duration between the last contact with $IP_A$ and the first contact with $IP_C$. This allows us to estimate $\hat{D}$ as a set of IP assignment durations $\hat{d} = (min(IP_B), max(IP_B)) \in \hat{D}$. In some cases, the min and max duration deviated by multiple days, indicating measurement artifacts (caused by, e.g., the loss of Internet connectivity of the infected device). Therefore, we filtered all IP durations where min and max deviated by more than $12h$.

Another issue that needs to be addressed before we can compare the two distributions is the bias toward short durations. This bias is caused by two factors. First, infected devices are less likely to be active for as long as RIPE Atlas probes. Therefore, it is unlikely to see very long IP durations. Second, the limited measurement period of $28d$ allows us to observe at most one IP duration of $28d$ for a bot, whereas we could see over two million IP durations of $1s$. This problem is exacerbated since we can only measure the durations of bots with at least three IP addresses.

To address this bias, we apply the *create-based* method proposed by Roselli et al. [22]. This method was previously applied by Stutzbach et al. and Sariou et al. [27], [24] in the context of measuring the lifetimes of hosts in peer-to-peer filesharing, and Karuppayah [14] for measuring the lifetimes of peer-to-peer bots. The create-based method proposes to split the total period $\tau$ into two equal-sized parts. By considering only IP assignment durations that start in the first half and durations of at most $\frac{\tau}{2}$, all IP durations between $[1, \frac{\tau}{2}]$ have equal probability of being observed.

We use the create-based method to compare $D$ and $\hat{D}$ for a measurement period of $\tau = 28d$, and consider all ASes

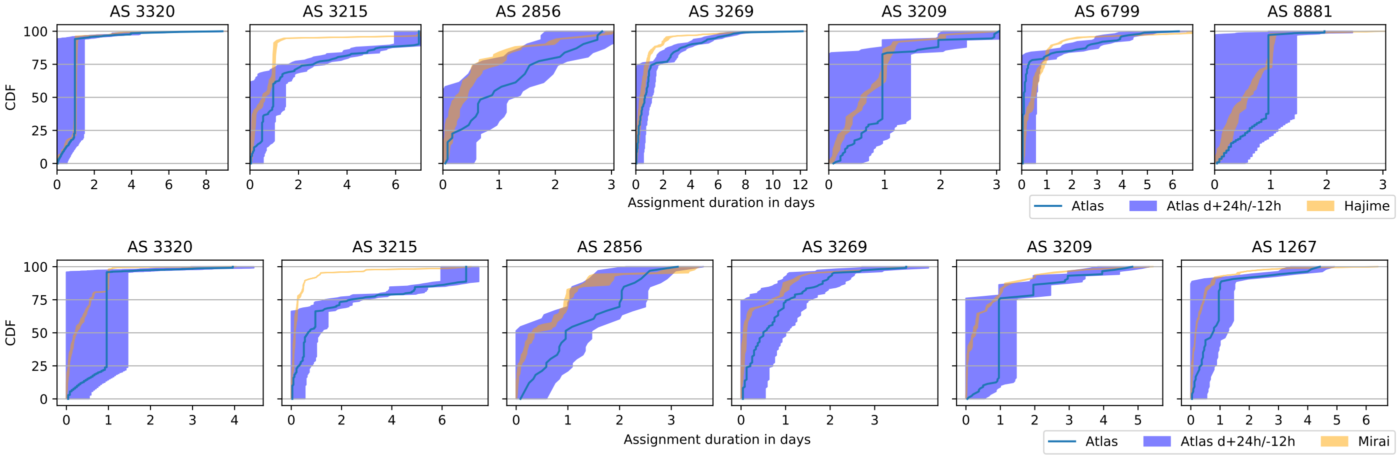


Fig. 8: CDFs comparing IP address assignment durations of RIPE Atlas against available IP durations for Hajime and Mirai.

with at least 30 IP address durations. As the hosts in a botnet are more likely to churn, and we need three consecutive IP addresses, we cannot observe as many long durations even if we apply the create-based method. Therefore, we compare $D$ and $\hat{D}$ by limiting the maximum length to the longest duration observed in $\hat{D}$. This provides us with seven ASes for Hajime and six ASes for Mirai, where we can compare the ground truth and actual distributions $D$ and $\hat{D}$. Figure 8 plots the CDF for all these ASes. For Hajime and Mirai, we plot both min and max durations. Furthermore, the figure includes the actual distribution of the RIPE Atlas distribution, and error margins of $d + 24h$ and $d - 12h$, which we evaluated in §IV. As shown in Section IV, deviations within this range have a limited impact on the accuracy of CARDCount. For Hajime, at least $75\%$ of the actual durations $\hat{D}$ fall within these ranges, with a slight tendency towards shorter durations. Furthermore, in three out of seven cases, the IP assignment durations of Hajime are fully encompassed within the error margin of the RIPE Atlas distribution. For Mirai, we can similarly observe that for five out of six ASes, at least $90\%$ of IP assignment durations are encompassed within the error margin of the RIPE Atlas distributions.

For both botnets, the greatest deviations can be observed for AS 3215. The cause of these deviations is that both botnet measurements failed to capture a large volume of $7d$ durations, which is a common reassignment frequency in AS 3215. The CDF for Hajime exhibits a mode at $7d$, showing that the botnet measurements are also able to capture *some* of these longer duration instances in AS 3215, but are likely underestimating other instances. We thus see that (a) shorter bot lifetimes and (b) the requirement that three consecutive IP address durations need to be observed result in botnet measurements underestimating durations. For Mirai, we can also observe a much larger fraction of short durations in the ASes 3320, 3215, 3269, 3209, and 1267. These are highly likely related to the presence of a CG-NAT, which we could not identify in our sanitization step. This interpretation is supported by the reports of Griffioen et al. [7]. They found that Mirai infections have different lifetimes depending on the AS of the infected bot. They specifically mention that bots in the ASes 3320 and 3269 had among the shortest lifetimes.

All that said, the actual impact on CARDCount's estimates depends on the difference in the mean IP address duration of $D$ and $\hat{D}$. We computed the mean differences between Hajime and RIPE Atlas based on the available data and found a mean difference of $8.6h$ and $16h$ for Mirai and RIPE Atlas. As discussed before, the higher deviation for Mirai is likely caused by our inability to identify and filter all hosts in a CG-NAT, and the unusually short bot lifetimes in some of the ASes. Given these deviations, and based on our experiments in §IV, CARDCount might slightly overestimate the actual bot population. However, this overestimation is counteracted by the underestimation caused by the presence of churn.

Overall, a large similarity in nearly all ASes, for which we could provide an accurate comparison of IP address assignment durations, strongly indicates that CARDCount provides an accurate estimate of the bot population in Hajime and Mirai.