NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INCENTIVIZING AND EVALUATING INTERNET-WIDE NETWORK MEASUREMENTS

by

Gokay Huz

March 2014

Thesis Advisor:	Robert Beverly
Second Reader:	kc claffy

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
2–4–2014	Master's Thesis	2012-04-02—2014-03-28

4. TITLE AND SUBTITLE

INCENTIVIZING AND EVALUATING INTERNET-WIDE NETWORK MEASUREMENTS

5a. CONTRACT NUMBER

5b. GRANT NUMBER
CNS-1111445

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Gokay Huz

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Science Foundation
4201 Wilson Blvd, Arlington, VA 22230

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.IRB Protocol Number: XXXX

14. ABSTRACT

The Internet's size is a primary challenge to researchers attempting to capture its properties. Inferences are therefore often based on available measurements, which may be biased due to the measurement process. We seek to understand the dependence of sampling methodology on two network measurement projects. We examine the potential of Mechanical Turk (MTurk) to guide the selection of samples by country and reward. As a proof-of-concept, we design an IPv6 adoption experiment disguised as a human intelligence task. Using 75 dollars, we obtain an IPv6 adoption estimate that differed by less than 3 percent of public estimates. From this initial success and analysis of the price sensitivity, we attempt a crowd-sourced approach to obtain representative measurements of Internet source address validation. However, this second experiment violated MTurk's terms of service. We therefore perform a per-country sampling analysis of nine years of existing source validation data from the Spoofer project. We conclude that conventional sampling methods do not properly characterize the data, primarily due to the changing nature of the underlying population during the collection period.

15. SUBJECT TERMS

Incentivized Network Measurements, Amazon Mechanical Turk, Spoofer, IP Spoofing, Sampling

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(include area code)*
Unclassified	Unclassified	Unclassified	UU	89	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

INCENTIVIZING AND EVALUATING INTERNET-WIDE NETWORK MEASUREMENTS

Gokay Huz
Lieutenant, Turkish Coast Guard
B.S., United States Coast Guard Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2014

Author: Gokay Huz

Approved by: Robert Beverly
Thesis Advisor

kc claffy
Second Reader, University of California, San Diego
Cooperative Association for Internet Data Analysis

Peter Denning
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Internet's size is a primary challenge to researchers attempting to capture its properties. Inferences are therefore often based on available measurements, which may be biased due to the measurement process. We seek to understand the dependence of sampling methodology on two network measurement projects. We examine the potential of Mechanical Turk (MTurk) to guide the selection of samples by country and reward. As a proof-of-concept, we design an IPv6 adoption experiment disguised as a human intelligence task. Using 75 dollars, we obtain an IPv6 adoption estimate that differed by less than 3 percent of public estimates. From this initial success and analysis of the price sensitivity, we attempt a crowd-sourced approach to obtain representative measurements of Internet source address validation. However, this second experiment violated MTurk's terms of service. We therefore perform a per-country sampling analysis of nine years of existing source validation data from the Spoofer project. We conclude that conventional sampling methods do not properly characterize the data, primarily due to the changing nature of the underlying population during the collection period.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

API	Application Programming Interface
AS	Autonomous System
BCP	Best Current Practices
CLT	Command Line Tools
DDoS	Distributed Denial-of-Service
HIT	Human Intelligence Task
HITs	Human Intelligence Tasks
ICMP	Internet Control Message Protocol
IP	Internet Protocol
ISP	Internet Service Provider
NAT	Network Address Translation
OS	Operating System
SCR	Spoofing Capability Rate
TLD	Top Level Domain
UDP	User Datagram Protocol
URL	Uniform Resource Locator
USA	United States of America

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

I would like to express my gratitude to my advisor, Professor Robert Beverly, for the tremendous support and guidance he has provided me with. His passion for technology and research was very inspiring.

And special thanks to kc claffy at University of California, San Diego, who supported the work despite her very tight schedule. I really appreciate her efforts to proofread my draft at the last minute and provide very valuable feedback. It is also her financial support that funded the Mechanical Turk experiments on Amazon.com and made the experiments possible.

A special thanks to my wife. Words cannot express the patience she showed towards me. She put up with my grumpy days and sleepless nights, and was always supportive of me when I needed her the most. Thank you for taking care of us.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

The Internet is large and complex with thousands of service providers and over a billion devices. The number of domain names alone is over one billion,[1] with the number of active Internet users estimated to be over 2.5 billion.[2]

Characterizing properties of the Internet is often difficult due to many reasons, including architectural limitations, information hiding, and available measurement vantage points. Yet, measuring the Internet is an important research area with implications on network architecture, critical infrastructure protection, security, economics, and policy [1].

This thesis examines an emerging theme in large-scale Internet measurement: crowd-sourced inferences. This introduction provides an overview of the problem, our approach, and a summary of contributions.

1.1 Active and Passive Internet Measurement Efforts

There are many efforts, both in research and industry, designed to actively measure the Internet. Examples of some of these efforts' goals include: evaluate end-host or residential network performance, infer the topology of the numerous autonomous systems and networks, assess security and network policies of different networks, and help inform the future design of networks [2]. For example the Ark platform by CAIDA (Cooperative Association for Internet Data Analysis) provides data that can be used to create annotated internet maps [3], the Dimes platform of Tel Aviv University aims to study the structure and topology of the Internet [4], the Grenouille platform by Grenouille Association aims to measure broadband speed as observed by home users [5], and Google's Measurement Lab provides tools and an open platform for researchers [6]. Other efforts seek to characterize different types of protocol or standards adoption, for instance Zander *et al.* [7] and Dhamdhere's analysis [8] of IPv6 adoption.

The Internet is a dynamic network, in that the number of active hosts and topologies of numerous networks connected to it are constantly changing. Further, the protocols, standards, and security policies of Internet hosts, and the networks to which they attach, change. The IPv4

[1] Internet Systems Consortium, "ISC Domain Survey", https://www.isc.org/services/survey/

[2] International Telecommunication Union, "Statistics", http://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx

address space allows for over 4 billion hosts, therefore it is not feasible to obtain measurements to and from all of the hosts (e.g., full-mesh measurements). In addition, many hosts and networks restrict sending, receiving, or replying to measurement probes, complicating the ability to accurately characterize the Internet. However, the accuracy and validity of Internet-wide measurement tasks increase with the number of data points collected. This basic tension poses a challenge to researchers trying to characterize properties of an Internet that is continually evolving. It is cost and resource prohibitive to build and maintain an infrastructure to constantly monitor Internet traffic, policies, relationships, networks, or hosts. As a result, current research is primarily based on *sampling* properties of the Internet, rather than complete measurements.

Sampling is an important part of all research and measurement efforts, including those in domains outside of network research. This thesis aims to explore various sampling methods that can be employed by Internet measurement researchers, pros and cons of each method, and the suitability of each for particular measurement efforts. We further evaluate the marginal value of individual Internet measurements from an existing measurement project, subject to existing data and results.

1.2 Problem Statement

Passive network measurements analyze properties of the network opportunistically, often by collecting network traffic without injecting any network probes. In contrast, active Internet measurement methods actively send probes to hosts on the Internet to elicit a particular response in order to measure a particular property of interest. Without efficient sampling methods, active probes will inject more artificial traffic than is required to characterize the network property under investigation. The volume of traffic generated by these active measurement systems can be considerable. We investigate sample sizes and stopping points for two systems in this thesis, with the goal of collecting results without wasting available resources (such as bandwidth and processing power required for analysis).

In Chapter 2.6.2, we first examine the potential for using Amazon.com's mechanical Turk, a crowd-sourcing platform, to measure IPv6 adoption. Because of the large and diverse set of users that are part of the platform, we seek to understand the extent to which we can leverage the platform to mitigate sampling bias. We apply various statistical analyses to understand the accuracy of the results we obtain along several dimensions, including per-country inferences. A fundamental finding of this work is that the crowd-sourcing platform itself introduces significant biases into our measurement inferences.

We then analyze data from the Spoofer project, a second existing system that contains a large span of longitudinal data in Chapter 4. We apply some of our developed measurement analysis and methodology design insights to better understand the spoofer project [9,10]. Source address spoofing allows hosts to send forged IP packets. This gives attackers a big advantage for their Distributed Denial-of-Service (DDoS) attacks, makes attack attribution operationally so expensive as to be infeasible for most practical cases, and makes it very hard to block the packets according to their source address. The spoofer project is a long-running effort to characterize the efficacy of source address authenticity mechanisms on the Internet. We aim to take a closer look at the problem and analyze the existing stream of measurements that are collected by the Spoofer project. A fundamental finding of our analysis of the spoofer data is that its inferences are limited by too few sample points.

1.3 Research Questions

In this thesis, we undertake three primary research questions:

- How much of a population needs to be sampled to make statistically significant inferences about properties of the population?
- Is it possible to view individual measurement data points in context, rather than in isolation, and quantify their values, taking into account previous measurements? For example, if the existing measurements allow us to make a conclusion about a small part of the network with a high confidence, can we stop allocating measurement efforts to gather more data points from that section and divert our resources to other "less-measured" parts of the network, where new measurements would have a higher marginal utility?
- How can Internet measurements be "crowd-sourced [11]" using existing tools? Specifically, how can we utilize Amazon.com's mechanical Turk platform [12] to support active Internet measurements? What techniques can be safely used without violating its terms of service?

1.4 Summary of Major Contributions

- To the best of our knowledge, our work was the first attempt to try and use mechanical Turk for internet-wide network measurements by disguising a network measurement task as a Human Intelligence Task and getting people to contribute to our test results without even realizing it.

- Using Amazon.com's mechanical Turk platform, we collected user data about IPv6 adoption rates in the USA and India. Our results for the adoption rate in the USA differed by less than 0.03% of the data published by Google and Cisco, and by less than 2% of the data published by Akamai. For India, our results differed by less than 0.50% of the data published by Google and Cisco and by less than 2% of the data published by Akamai. This initial experiment costed us less than $75. We concluded that mechanical Turk platform offers a diverse user population that is representative of the general Internet users, and compared to other options to collect data from remote locations of the world, the cost is within reasonable limits.
- We conducted another experiment on mechanical Turk to assess the price sensitivity of workers to the monetary awards offered by different HITs. As expected, as the monetary award increases, Human Intelligence Task (HIT)s get completed at increased rates and in shorter time periods. The price also affects the quality of work performed by the workers. When the monetary award is set for less than $0.10, people pay less attention to the driections and make more mistakes, resulting in useless data.
- Mechanical Turk platform claims to have users from over 190 countries. Although this might be true (we did not try to challenge this claim), workers from the USA and India make up a very large majority of the userbase. This might pose a problem for researchers that require a more geographically diverse user population.
- We also tested mechanical Turk to crowdsource more measurements for the Spoofer project, only to discover that Amazon.com does not allow requesters to require workers to download and run executable programs, due to user security and privacy reasons.
- We performed an extensive analysis of the data collected by the Spoofer project, calculating the SCR for individual countries and trying to estimate population parameters using over 9 years of data spanning from 2005 to 2014. Although the historical charts that showed SCRs for individual countries and Top Level Domain (TLD)s were interesting, they do not help to estimate the population parameters for the entire Internet population. This is mainly due to the fact that the underlying population parameter which we are trying to estimate by traditional sampling analysis, is constantly changing during the sampling timeframe.
- Using historical charts, we were able to pinpoint countries and TLDs, for which SCRs were increasing or decreasing. More detailed discussions are presented in section 5.1

1.5 Organization

This thesis is organized as follows:

- Chapter 1 presents a brief introduction about the thesis research.
- Chapter 2 discusses prior and related work, and gives an overview of related statistics concepts and theory. It summarizes key concepts and formulae about population mean estimation, gives an overview of the mechanical Turk platform and the typical life-cycle of a HIT. It also introduces the Spoofer program, what it does and how it works.
- Chapter 3 discusses two experiments that we conducted on the mechanical Turk web site. It explains out methodology, analysis methods and our findings. There are several subsections related to Amazon.com's mechanical Turk, advantages and disadvantages of using the site as part of a systematic approach and the pitfalls that researchers should be aware of then conducting research on it.
- Chapter 4 gives a more detailed explanation of how the Spoofer program works and our analysis of its existing results. We present the changes of SCR for individual countries over years and do a cross-validation of the results from the project.
- Chapter 4 summarizes the results and findings and explains how our findings can be used for providing alternate means for targeted Internet measurements and possible reducing costs.
- Chapter 5 concludes and suggests possible future areas of exploration.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background and Related Work

This background chapter reviews relevant statistical measures that will later be investigated via crowd-sourcing. We will discuss the use and motivation of sampling in section 2.2, how to estimate population parameters in section 2.3, Amazon.com's mechanical platform in section 2.4, and finally, the Spoofer project in section 2.5.

2.1 Enumeration

Before we can start our data collection efforts, we need to enumerate the members of the general population. After enumerating all members, we can either do a census, where we collect population parameters from every single member of the population, or we can take a sample, where we only gather data from a smaller subset of the general population.

There are efforts that try to enumerate and scan all the hosts on the Internet. While traditional tools like `nmap` require more than 2 months to scan the entire Internet in about, new tools, such as `zmap` are emerging that can do the same job in about one hour [13], However, these are the required times for `nmap` and **zmap** to do a simple scan, where they only test if there is a live host at a specific address that answers to Internet Control Message Protocol (ICMP) requests. When the data collection requires more complex tasks, as does the Spoofer [9] and many other measurements, sampling naturally emerges as the only viable method for data collection.

2.2 Sampling and Motivation

Measurements of any particular property in a target population can either be a census where every single observation unit is individually observed and the result recorded, or a sampling where only a selected subset of the target population is observed and recorded. When the population size is big enough, measuring the entire population quickly becomes cost-prohibitive or impossible due to budget or time constraints. For small populations and simple measurements, a census might be preferable to obtain the maximum precision.

For example, a study might try to estimate the number of young people in the United States, aged 16-18 that smoke and their gender distribution. In this example, the population of the survey would be all the people that meet the requirements (aged 16-18, males and females, living in the U.S.). We might not be able to enumerate all the members of this population, and

even if we did, we might not be able to collect information from all of them. We would not be reach all the members, and some of them that we reach will choose to not participate in the survey.

Because certain sub-groups are expected to have similar percentages, dividing the target population into smaller groups allows us higher precision. For example, we can sample males and females separately and then combine the weighted results. (Section 2.2.1 discusses sampling methods that we can use).

Similarly, in the field of active network measurements, sub-groups of the target population will have similar characteristics. For example, if we are trying to measure the performance of peer-to-peer networks across different countries and we are aware of countries that require ISPs to enforce network neutrality, we might evaluate these countries separately from ones that do not. These similarities could be at the country level, Autonomous System (AS) level, or a network level.

When taking measurements from a local network with limited number of hosts and where it is possible to take measurements from any point in the network at any rate, it is simple to produce the exact picture of the network, no matter whatever data you are collecting. However, once the researcher goes beyond the limited local network, a methodology that depends on explicit enumeration is less feasible. Also, in a local network, we might have administrative freedom on what we are doing on the network. If we are taking measurements from networks that are not under our management authority, we are limited in the tools that we can use, and also how much of these tools we can use. Sending too much unsolicited packets to a remote network might trigger an alarm and block our IP address from access to the network, which would not be a problem on a local network.

Lastly, measuring every node or every link in a network does not scale well, especially when the network in research is the Internet with billions of active nodes. Another factor that complicates enumeration of all hosts is the use of dynamic addressing on the edge of the network. While the core of the Internet is using mainly static addresses, most of the end users use dynamic allocation to assign IP addresses to edge hosts. Therefore, when measuring any non-trivial property of the entire Internet with billions of active hosts, sampling is the only feasible method to come up with time-sensitive measurements.

There are a total of 2^{32} or more than 4 billion possible hosts on the Internet. The number of

ASes is over 400,000, the number of domain-names is over 250 million[3]. Because of the vast number of connected computers, measurements and data collection about the Internet inevitably requires sampling. It is not too time-consuming to collect data from every single member of the target population. And in a constantly evolving environment like the Internet, it is not possible to extend the duration of the measurements over a long time span as the duration itself will invalidate the collected data points. That is why researchers often use various sampling methods to come up with snapshots of the Internet.

Sampling allows Internet measurement researchers to estimate, for example, population mean by inspecting only a sample, with far fewer measurement points. The sample properties are then used to estimate the property of the entire population. We use "unbiased estimators" to make inferences about the general population. The idea behind this reasoning is that a perfectly random sample would be representative of the entire population. This does not mean that all the samples have the exact properties, or that they reflect the overall population accurately. The process involves a trade-off; compromising accuracy in exchange for convenience and cost. By adjusting the sample size, we can get estimates of population parameters with any desired confidence level.

2.2.1 Sampling Methods [14]

There are numerous strategies for picking candidates for inclusion into the sample, but at a higher level, they fall into two basic categories:

1. Probability Sampling: also called *random sampling*. In random sampling, each individual in the population has an equal chance of being selected. However, true random sampling is often impractical, as samples may be drawn with *almost* random processes, that suit the practical limitations of the selection process.
2. Non-Probability Sampling: generally referred as *quota sampling*, where the individuals are included in the sample to reflect particular properties of the population.

If large sample sizes can be cheaply and efficiently obtained, probability sampling is preferred. However, quota sampling usually returns more accurate results for smaller samples. If the sample sizes are large enough, by the central limit theorem, the expected distribution of the random samples will approach that of quota sampling.

[3]Verisign Inc, "The Domain Name Industry Brief", http://www.verisigninc.com/assets/infographic-dnib-Q32013.pdf

Simple Random Sampling (SRS) with/without Replacement

Simple random sampling *with replacement* is the only sampling method that ensures that every individual has the same and equal chance of being selected. Practically though, it is very hard, if not impossible, to: (1) enumerate all members of a large population, (2) select individuals randomly to form the sample (3) survey the sample, factoring in the non-response rates.

Stratified Sampling

Simple random sampling may result in the over representation of some populations in the final estimations, while some other populations are not considered at all. For example, if one picks a simple random sample of 100 adults out of a 1000 person target population, a possibility exists, a very small one indeed, that all the people in the sample are males. Although the probability of such an extreme case is very small, many other random sample selections will have males and females unproportionally distributed in them.

For many measurements, this is an undesired occurrence and any manual modification to the 'random' sample frame will distort the results.

To overcome the possibility of some population groups not being represented in the sample, some sample designs employ *stratified random sampling*. In this method, the overall population are split into exclusive *strata*, such that every member of the overall population is a member of one and only one *stratum*. This also prevents individuals from being included in two or more different strata. After the target population is split into the pre-defined strata, each individual stratum is sampled for the target parameters. Then the results from each strata are merged to estimate the population parameters.

Stratified sampling allows a researcher to reduce the required sample size when the target population is more homogeneous than the general population.

Cluster Sampling

Also known as *multi-stage cluster sampling*, cluster sampling divides the population into exclusive "clusters." Each individual cluster is then randomly sampled. For example, if measuring IPv6 adoption rates around the world, the first division would be to split the world into countries (or regions), and then collect samples and measurements from each country randomly. The country results then can be *weighted* and aggregated to estimate the worldwide adoption rates. One thing that is worth noting is that; in cluster sampling, individuals in the population do not have an equal chance of being selected, as their chances are dependent on the cluster into which

they fall.

Quota Sampling
In quota sampling, the researcher first picks some basic characteristic and then tries to match the sample to the population in regards to the chosen characteristics. For example, for measuring IPv6 adoption rates, we can find the distribution of internet hosts across countries. Then for each country, we calculate the target sample size and then collect measurements that fit the requirement. For example, if the total number of IP hosts is 4 billion and we know that 10% of the hosts are located in China; for a target sample size of 5000, we collect 500 measurements from China.

2.3 Estimating Population Parameters

As mentioned in the previous subsection, not all samples will have the same parameters (mean, proportion, variance) as the population. However, if we keep taking random samples from the population and plot their parameters, they will tend to cluster around the population mean. When plotted on a histogram, the sample means will have a symmetrical bell curve, centered on the population mean. This is why the sample mean, proportions and variance are regarded as unbiased estimators of the population properties

We can use *inferential statistics* to make estimates about population parameters. Assuming the sample size is large, relative to the population size, we can base our population estimates on the sample parameters. In the following chapters, we discuss how to calculate the accuracy of this estimate and the confidence levels of our estimate using standard error of the mean.

2.3.1 Notation Used

The following subsection lists the notation used for the formulae that are introduced. [15]

n	Population or sample size
x	A single data point in the population or the sample
μ	Population mean
σ	Population standard deviation
σ^2	Population variance
$\bar{x}$	Sample mean
s	Sample standard deviation
s^2	Sample variance
$S.E.$	Standard Error

For a population, the mean is calculated as:

$$\bar{\mu} = \frac{1}{n} * \sum_{i=1}^{n} x_i \tag{2.1}$$

The population variance is:

$$\sigma^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2}{n-1} \tag{2.2}$$

The standard deviation of the population, σ, is equal to the square root of population variance.

For a sample, the mean is:

$$\bar{x} = \frac{1}{n} * \sum_{i=1}^{n} x_i \tag{2.3}$$

The variance of the sample is calculated as:

$$s^2 = \frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ... + (x_n - \bar{x})^2}{n-1} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1} \tag{2.4}$$

The standard error of the sample mean is:

$$S.E.(\bar{x}) = \frac{s}{\sqrt{n}} \tag{2.5}$$

Estimating proportions is a simpler case of mean estimation and all the above formulas apply, and they take a simpler form. The variance of the population can be estimated using the following formula:

$$V[\hat{p}] = \left(1 - \frac{n}{N}\right) \frac{p(1-p)}{n} \tag{2.6}$$

The standard error is:

$$\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \tag{2.7}$$

where $\hat{p}$ is the proportion of the sample that has the measured property.

In equation 2.6, the first term is called the *finite population correction*, and can be dropped when the population size is much bigger compared to the sample size. For large samples, the size of the sample determines the precision of the estimator. So, for very large samples, equation 2.6 reduces to:

$$V[\hat{p}] = \frac{\hat{p}(1-\hat{p})}{n-1} \tag{2.8}$$

Note that, in practice, the population mean and standard deviation are never known and these are the values that are estimated, using sampling.

After calculating the mean, standard deviation and standard error of a sample, we can estimate the population parameters as follows:

- There is a 66 % chance that the actual population mean will be within ± 1 S.E. of the sample mean
- There is a 95% chance that the actual population mean will be within ± 2 S.E. of the sample mean
- There is a 99 % chance that the actual population mean will be within ± 3 S.E. of the sample mean

This has two important consequences. First that lower standard error of the mean results in estimates with higher accuracies. According to formula (2.5), standard error of the mean is dependent on the standard deviation of the sample mean and the sample size. And the only way of reducing the standard error of the mean is to increase the sample size, or the *sampling frequency*. Sampling frequency is the size of the sample size, divided by the population size. Because we cannot draw a sample that is larger than the population size, the sampling frequency is always smaller than 1. For populations that have a high degree of variation, the standard error will also be higher and we need more samples from these populations. For homogeneous populations, standard deviation will also be small and we do not need large sample sizes.

Another important consequence is that as the *confidence level* of our estimates increases, so does *the margin of error*. In the above statements, the probabilities (i.e. 66%, 95% and 99%) denote the confidence levels and the ranges (i.e. ± 1 S.E, ± 2 S.E, and $\pm$ 3 S.E) denote the margin or error. Large samples allow higher confidence levels at lower margins or error.

2.3.2 Determining Sample Size

When sampling, it is important to pick a sample size that has a high probability of capturing the properties of the population. Generally, bigger sample sizes result in higher accuracies when estimating population parameters. However, bigger sample sizes result in higher cost of gathering data points, and increasing the sample size on large samples result in exponentially diminishing returns. However, even with relatively large samples, there is no guarantee that the sample will perfectly estimate population parameters.

When designing the sampling strategy, we need to first define accuracy requirements of the measurement. This is why we need to define the desired margin of error for the measurements and the target confidence level. Higher confidence levels will result in larger sample sizes. For proportions, we can calculate the required sample size as follows [16]:

$$S = z_\alpha^2 * \frac{\hat{p}(1-\hat{p})}{\delta^2} \tag{2.9}$$

where z is the ordinate on the normal curve corresponding to α

$\hat{p}$is the estimated proportion (if we do not have any prior estimates for $\hat{p}$, we take the proportion to be 0.50)

$\delta$is the specified precision of the estimate

Table 2.1 shows values of z_α for common confidence intervals:

Confidence Interval	z_α
68.27%	-1
95.45%	-2
99.73%	-3

Table 2.1: z and z_α Values for Common Confidence Intervals

For example, if we are trying to estimate the adoption rates of IPv6 and our *initial guess* of the adoption rate is 5% ($\hat{p} = 0.05$), and we want to be accurate within 0.05 ($\alpha = 0.05$) at 95% confidence interval ($z_\alpha = -2$), the required sample size would be:

$$S = (-2)^2 * \frac{0.05 * 0.95}{0.05^2} = 76 \tag{2.10}$$

Sampling Error(%)	Estimated proportion ($\hat{p}$)				
	0.5	0.4	0.3	0.2	0.1
1	10,000	9,600	8,400	6,400	3,600
2	2,500	2,400	2,100	1,600	900
3	1,100	1,067	933	711	400
4	625	600	525	400	225
5	400	384	336	256	144
10	100	96	84	64	36

Table 2.2: Sample Sizes Required to Achieve 95% Confidence Level for Given Precision and Proportion Values

Table 2.2 gives us the required sample sizes to achieve 95% confidence level for estimates of varying degrees of precision, and with varying intra-group proportion parameters. If there are no a priori estimates about the target proportion, we have to use the first column of the table. If we have any prior knowledge or estimate about the proportion and it is lower than 50%, the required sample sizes would be much lower for any given confidence level and precision.

2.4 Amazon.com's Mechanical Turk

Amazon's mechanical Turk (mturk) [12] is an open marketplace and a crowd-sourcing platform where *workers* from around the world meet with *requesters* who publish *Human Intelligence Tasks (HITs)* for the *worker*s to complete. The *HIT*s are small micro-tasks that take anywhere from a few seconds to a couple hours to complete. In return for their work, the *workers* are compensated by the *requesters*. The service was first introduced by amazon.com in 2005 and is open to both workers and requesters from around the world. As of February 2014, there are over 150,000 available HITs on the MTurk web site. Amazon advertises the site to have 500,000 registered workers from over 190 countries worldwide [17]. As discussed in section 2.4.2, the distribution of user countries is very skewed and a vast majority of users are from U.S. or India.

MTurk is used by researchers for conducting user studies [18, 19], behavioral research [20], and other experiments [21, 22]. Kittur *et al.* suggest that "micro-task markets have great potential for rapidly collecting user measurements at low costs" [18], and Buhrmester *et al.* conclude that "MTurk participants [are] more demographically diverse than standard Internet samples and significantly more diverse than typical American college samples" [22].

There are numerous prior efforts that study the quality of work performed by workers on the MTurk platform [22] and the platform's user demographics [23]. These studies praise the de-

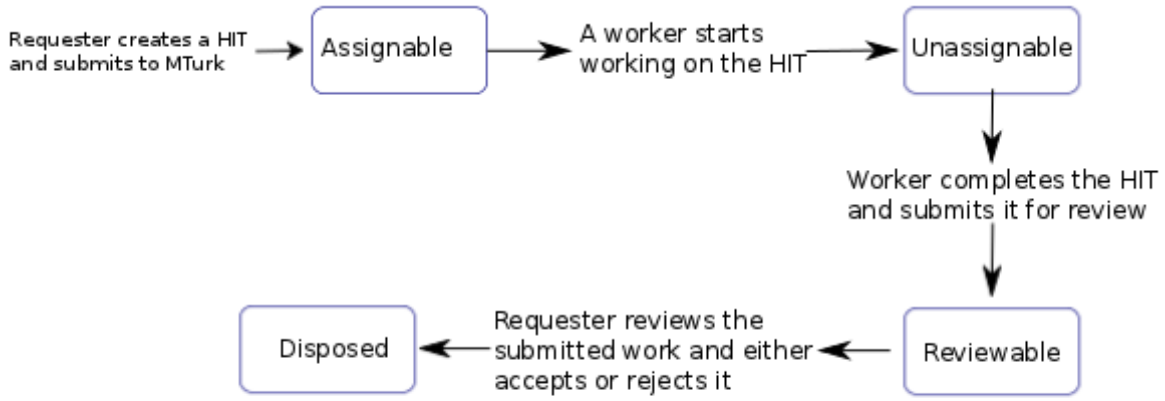


Figure 2.1: Lifecycle of a HIT on the MTurk Platform

mographical diversity of the mechanical Turk workers and suggest the platform as a promising mechanism for online surveys and user research, while a few of them also recognize the fact that most of the workers are compensated below the U.S. minimum salary wage and criticizing the platform as being a digital sweatshop [24].

2.4.1 Life Cycle of a HIT

On the mechanical Turk platform, a HIT is a small task that is to be completed by a single worker. A HIT normally has a single assignment, but a HIT can also have multiple assignments. Multiple assignments of the same HIT is useful when we have one task and we would like the same task to be completed by multiple users. This provides us with multiple answers to the same question from several distinct workers. This can be used for quality control, or cross-verification of answers.

There are five basic states of a HIT [25].

1. Assignable: When a *requester* creates and uploads a HIT to MTurk, the HIT becomes *assignable*, meaning that a qualified worker can accept it and start working on it. HITs can have required worker qualifications to prevent some workers from working on the HITs. For example, a requester can require an approval rating of 96% before the worker can accept the HIT. There are many predefined qualifications on the mechanical Turk platform and requesters can also define and create their own qualifications. When creating the HIT,

the requester also assigns a compensation amount for the HIT. The minimum award is $0.01 (US Dollars). The requester can create multiple "assignments" (or instances) for a HIT, which makes the HIT available to be completed by multiple distinct workers.

2. Unassignable: When a worker accepts a HIT and begins work, the HIT becomes unassignable, so that other workers cannot work on the same HIT. This is similar to checking out a file on a a Concurrent Versions System (CVS). When a worker accepts a HIT, other users cannot accept the HIT or start working on it. Should the worker "return" the HIT, it becomes assignable again. (similar to checking in a file on a CVS) If the worker completes and submits the HIT, it becomes reviewable.
3. Reviewable: After a worker completes a HIT and submits it, the HIT becomes *reviewable*, meaning that the requester can review the results of the work and either approve or reject the HIT.
4. Reviewing (optional): The reviewer can either review all submitted results manually or can set to automatically accept all results. When the requester reviews the completed HIT, the HIT moves on to the next stage. The HIT can be reviewed using the command-line tools, the website or the provided APIs.
5. Disposed: After his/her review, a requester can either accept the result or reject it. If the result is rejected, the worker does not get compensated. Any HIT that the requester rejects gets returned to the pool of available HITs, so other workers can work on the HIT. Note that the requester, when creating the HIT, determines how long the HIT will be active on the site, after when the HITs expire and are no longer available on the web site. If the requester rejects a HIT that has been completed by a worker, and the HIT's time frame has expired, the rejected HITs will not be available for other workers to complete.

   Another useful feature of the mechanical Turk platform is that all workers have an approval rating. This is the ratio of the approved HITs that the worker has completed, divided by the number of all the HITs the worker has completed. This is a feedback and rating system for the workers. This is very similar to feedback rating users have on online marketplaces like eBay.com.

Amazon.com provides requesters a web-based user interface, Command Line Tools (CLT) and Application Programming Interface (API)s for automating the entire process using many high-level programming languages (including JAVA, Python, Perl, Ruby, PHP). Requesters can perform all of the tasks using the CLT and the API, while the web-based user interface provides limited functionality for administering basic, simple HITs.

Especially when conducting research, a requester can have a single HIT, available to more than one worker (usually, to hundreds or thousands of users). For example, this allows multiple distinct workers to complete the same survey, allowing data collection from multiple users.

If a requester does not require unique workers to complete the task, s/he can create multiple HITs, with one assignment each. This will allow all the tasks to be completed by the same user.

The requester can specify requirements that workers need to pass or qualifications they must posses before they are offered a particular HIT. These requirements allow HITs to be targeted to workers in specific countries only, to workers who have a specified approval rating, or only to MTurk *masters*, or "elite group of workers who have demonstrated accuracy on specific types of HITs on the Mechanical Turk marketplace. Masters achieve a Masters distinction by consistently completing HITs of a certain type with a high degree of accuracy across a variety of requesters [26]." The exact requirements of a Masters distinction are not stated by amazon.com and the distinction is given by the amazon.com staff without a screening process, details of which are not publicized. Most of these advanced features, like creating, assigning and revoking worker qualifications to HITs, are available only to users who create their HITs using CLTs or the APIs.

2.4.2 User Demographics of MTurkers

Amazon.com does not publish any public information about their workers, but there are studies that analyze user demographics. Ross, et al [23]. for example, use sample data spanning 20 months and surveying more than 3800 users to report on the nationality, gender, age, and household income of "MTurkers." And, as the studies span over multiple years, it is easy to observe the trends in the shifts of user population. The authors conclude that, in Nov 2009, the users from U.S.A. and India make up the majority of users (56% and 36%, respectively. The ratio of workers from the USA are in a constant decline for the last 18 months, while the ratio of users from India have been rising. The paper analyzes many aspects of user demographics in detail (like age, gender and nationality of users, reported annual income, etc.). One of the reasons of limited participation from other countries is due to distribution of financial awards. Due to tax reporting requirements, only the workers in the U.S.A. and India can receive financial compensation by check. Users from other countries can still complete HITs and earn money, but they can use their money only towards purchases from Amazon.com. Amazon does not mail checks or wire funds to banks in other countries.

2.4.3 Using Mechanical Turk for Research

Researchers actively use Mechanical Turk. Mason and Suri [20] discuss many aspects of conducting behavioral research on the Mechanical Turk platform, including a detailed methodology, quality assurance, security, and ethics and privacy of research. They praise the MTurk platform for providing access to a "massive subject pool available 365 day a year," the diversity it offers in terms of worker age, gender distribution, annual income, etc.. compared to the subject pools available at U.S. universities, and greatly decreased cost of surveys and experiments that are completed in much shorter time frames.

Buhrmester et al. [22] study the quality of survey data available from Mechanical Turk and investigate how compensation affects data quality. They conclude that "the quality of data provided by MTurk met or exceeded the psychometric standards associated with published research."

Oh and Wang evaluate usability of Mechanical Turk as a platform for conducting music perception experiments [21]. After citing limitations of the platform for music perception experiments, they also praise the diversity of user demographics of the platform and suggest that new advances are "heightening the potential of MTurk to serve the scientific communities at large."

Christin et al. [19] demonstrate how they were able to recruit more than 950 MTurk workers to download an executable file, and run the file on the users' local computers with administrative priviledge for one hour. They analyze and report on the relationship between financial reward and increased user participation. The study reveals how users completely ignore traditional security advice and policy in exchange for small financial incentives, as low as $0.01.

Although social scientists have embraced the Mechanical Turk platform for research, there are no experiments, to the best of our knowledge, that utilize MTurk for network measurement, e.g., measuring properties of user's networks or their Internet service provider. In section 2.6.2, we explore one method of using MTurk platform for active, targeted Internet measurements.

2.5 The Spoofer Project

Internet's design stems from the original DARPA project where a small number of connected hosts were known and trusted. The original Internet designers did not anticipate the popularity of the Internet, while potential security vulnerabilities were a secondary design consideration. One property that was not inherently built into the design was source address validity. The

current architecture allows a host to fabricate IP packets with source addresses that are different than its own IP address. Also known as IP "Spoofing," this insecurity allows attacks from spoofed sources, where it is difficult to track an attack back to the true source. Recently, many DDoS attacks have employed IP spoofing. For example, in April 2013, more than 15 US banks and financial institutions were under attack for weeks at a time, which made their web servers unreachable for hours.[4] In March 2013, another attack was launched against Spamhaus, an organization dedicated to tracking and fighting spam on the Internet, that used up to 300Gbps of the organization's bandwidth.[5] Among other tactics, both of those attacks used packets with spoofed IP addresses and they were both hard to evade. Because addresses are forged, it is difficult to attribute the attack to any particular person or organization and often the attacks are distributed, meaning that many hosts from all around the world are involved in the attack.

Spoofed IP addresses are commonly used by attackers in DDoS attacks so that the attacker cannot be identified and also so that the attacked system cannot filter out the packets based on source IP addresses. Some ISPs use ingress filtering [27], which blocks IP packets with spoofed addresses and does not allow spoofed IP packets to get out to the Internet while other Internet Service Providers (ISPs) do not enforce filtering.

The currently ongoing Spoofer project [9] that started in 2005 aims to measure how common it is for ISPs to allow IP packets with spoofed source addresses. Since 2005, it has collected over 19,000 data points from over 15,000 unique IP addresses [28]. Participation in the measurements is completely voluntary, and any user around the world can download the executable file from the project's web site and run it The results are reported to a central server that keeps a record of the results. In order to improve the accuracy of the measurement results, the program requires esers from different networks to run the program on their computers.

The Spoofer program attempts to test whether IP packets with spoofed source IP addresses can be received by the monitoring servers aroung the world or whether they are filtered by any of the intermediate routers. Normally, the packets a client sends have the client's IP address in the source IP field of the packets. If the client's Internet Service Provider (ISP) enforces ingress filtering as defined in Best Current Practices (BCP) 38 [27], no client would be able to send any spoofed packets as the ingress router at the ISP would drop these spoofed packets.

[4]InformationWeek, "Banks Hit Downtime Milestone In DDoS Attacks" http://www.informationweek.com/security/attacks/banks-hit-downtime-milestone-in-ddos-att/240152267, accessed 12 February 2014

[5]Quentin Jenkins, "Answers about recent DDoS attack on Spamhaus", http://www.spamhaus.org/news/article/695/

Some ISPs filter at a prefix granularity, thereby preventing arbitrary spoofing, but permitting clients to spoof IP addresses that are in the same prefix as itself (adjacent addresses). However, there are many ISPs that do perform no source address validation and allow IP packets with arbitrary spoofed source addresses. A client in this case would be able to spoof any of the approximately 2^{32} IP addresses in the IP address space.

The Spoofer project aims to permit users to understand the extent to which their network permits spoofing. A by-product of the measurements is an aggregate *estimate* of the globally spoofable IP address space, networks, and ASs. Further, by analyzing the results of tests run by users from around the world, the project aims to estimate the spoofing capability rate, i.e., the percentage of IP addresses that have the capability to send spoofed IP packets and whose packets are received by hosts across the Internet. Technically, all hosts can send spoofed packets. For most of the clients, their ISP blocks the packets, so the packets never arrive to their final destination. We define Spoofing Capability Rate, SPR, as the fraction of IP addresses whose spoofed packets successfully travese the Internet and arrive to their destinations. The spoofer project classifies the clients' IP addresses as capable of spoofing any arbitrary IP address, capable of spoofing neighboring or adjacent IP addresses only, or not able to spoof any IP address. The program also tests the client's ability to spoof private addresses as defined in RFC1918 [29] and non-routable sources. It also infers the presence of a Network Address Translation (NAT) device between the client and the server. Presence of a NAT device between the client and the test servers prevents useful spoofing measurements. In this case, the source IP address field of packets will be overriden by the NAT device and the the measurement will not be useful.

2.6 How the Spoofer Program Works

Users download the program from its web site and run the executable on their computers. The program requires administrative or root access on the computer as it uses raw ethernet packets or raw sockets instead of using the system networking stack to send spoofed IP packets. The program initially sends non-spoofed UDP packets to ensure that non-spoofed packets sent by the host reach the server. The program then starts sending User Datagram Protocol (UDP) packets with spoofed IP sources.

2.6.1 Adjacent Spoofs

Initially, the program starts with adjacent IP addresses (Real IP address $\pm 2^i$, for $i \in \{0..13\}$). So, for example, if the IP address and subnet mask of the host is 192.0.2.100/24, the program

Sequence #	IP Address
1	192.168.2.101
2	192.168.2.98
3	192.168.2.104
4	192.168.2.108
5	192.168.2.84
6	192.168.2.132
7	192.168.2.36
8	192.168.1.228
9	192.168.1.100
...	...

Table 2.3: Probable Source Address Sequence for a Host with a Real IP Address of 192.168.2.100

might try a sequence as shown in table 2.3 as the source IP address of the packets. The reasoning behind this increasing deviance from the real IP address is to test at what point the ISP blocks and drops the spoofed packets. For example, for the given host with an IP address of 192.168.2.100 and a subnet mask of /24, the ingress filtering described in BCP38 would still allow the host to spoof addresses in the given prefix. So, the host would be able to use any IP address in the [192.168.2.1 - 192.168.2.254] range.

Compared to arbitrary spoofs, adjacent spoofs do not pose an as big threat to Internet security, as even if the specific host that sent the packet can not be traced, it is still possible to track the packets to the source network. An administrator whose network is being attacked with spoofed IP packets can easily block the attacking network.

2.6.2 Arbitrary Spoofs

If BCP38 filtering is not employed by any of the ISP's between two internet hosts, both hosts can use any IP address as the source address on their outgoing packets. In the absence of other filtering rules, a client can also send packets with private IP addresses as the source IP. Private IP addresses are defined in RFC 1918 [29] and these packets should never be routed in the public Internet. These packets, however, are often filtered in the network core because it is safe to do so. [10]

When run, the spoofer client program sends a probe to the main server and receives a list of IP (source, destination) pairs. It then sends spoofed packets to these destinations. If the servers can receive any of the spoofed packets, the client is classified as it can spoof IP packets.

CHAPTER 3:
Using Mechanical Turk to Incentivize Network Measurements

In section 2.4, we introduced several Mechanical Turk (MTurk) use cases. To the best of our knowledge, our work is the first to examine using MTurk for performing Internet measurements. Most prior work that examines the utility of the MTurk platform for conducting research praise it for the diversity of the user base it offers. In this chapter, we discuss the suitability of using Amazon's Mechanical Turk to crowd-source Internet measurements.

As part of this research, we conducted two small experiments on the MTurk platform. The first experiment was designed to understand the feasibility of designing a HIT that performs a useful Internet measurement task. As a proof-of-concept, we designed a HIT to measure the adoption rate of IPv6 and to analyze the diversity that the platform offers as it relates to Internet measurements. We compared the results from this experiment with other publicly available IPv6 data to infer the representativeness of our MTurk-based inferences. This comparison allows us to draw more general conclusions about the population of MTurk users and suitability of MTurk to Internet measurement.

Based on the results of the first MTurk experiment, we ran a second experiment designed to analyze the relationship between HIT compensation, HIT completion time, and on the quality of work performed by MTurk workers from two sample countries.

3.1 Designing Internet Measurement HITs

As will be discussed in section 3.4, it is not possible to directly ask the mechanical Turk users to run a particular test or network measurement utility, e.g., netalyzer[6] or the spoofer client. In an effort to ensure the safety of its users, amazon.com does not allow any HIT that requires the workers to run an executable file on their computers. If MTurk did permit executables, researchers could leverage the platform to distribute a program that leverages the full capabilities of a client's computer. For instance, an executable could send raw packets and conduct a variety of low-level measurements from many vantage points. However, this would expose the mechanical Turk users to increased security vulnerabilities. Therefore, Amazon.com's terms of service does not allow HITs involving executable files.

[6] http://netalyzr.icsi.berkeley.edu

Due to these restrictions, we seek to design an experiment that *appears to be a human centric task, but is, in fact, performing a useful Internet measurement in the background*, without any explicit user interaction.

For this first experiment, we uploaded a HIT to mechanical Turk (MTurk) website. The creation of the HIT required us to create a mechanical Turk requester account, upload money to compensate the works, create the HTML files on our server that displayed the HIT to users and then configuring the parameters of the HIT. The HITs are highly-configurable and using text config files, the requester can assign various parameters to the HIT, like the duration of the HIT, worker requirements for the HIT, compensation amount for workers, locale requirements for workers, etc.. Then we had to upload this configuration file to mechanical Turk using the CLT that Amazon.com provides. As soon as the HIT is uploaded, it is availeble for workers to accept an complete. The goal of the HIT was to measure whether or not the client was IPv6 capable, although the HIT makes no reference to this as its true intent. Instead, the HIT appears to be very simple, basic survey that requires users to count balls. Thus, the human task is simply counting the number of different balls of two different colors. Our HIT displayed a random number of blue and red balls (number ranging from 1 to 4) and required the users to enter the number of red and blue balls. By using a random number of balls, we can verify whether the user completed the actual human task, even if the ball counting is incidental to the IPv6 measurement. Crucial to our experiment was that we hosted the red and blue ball images on a web server we maintain, and the blue balls can only be retrieved using HTTP over IPv6. Thus, our HIT allowed us to infer whether the client was IPv6 capable during the course of the worker completing the HIT. Figure 3.1 shows the IPv6 red versus blue ball HIT as it appears to a worker. Note that, in this example, the system chose to display three red balls, and two blue balls. However, since the client had no IPv6 connectivity, the blue balls did not appear. The user was required to enter the number of red and blue balls she saw using the two drop-down boxes.

We hosted the survey page on an external, dual-stacked server (http://www.cmand.org) that had both an IPv4 and an IPv6 address. Because the HTML page containing the survey was hosted on our server, every time a user displayed the survey, their browser had to access the images hosted on our server as part of the HIT completion process.

The red balls were hosted on the same server. The URL for the blue dot image blue dot used a separate host (http://turk.cmand.org/). This hostname had only a AAAA record in the DNS, en-

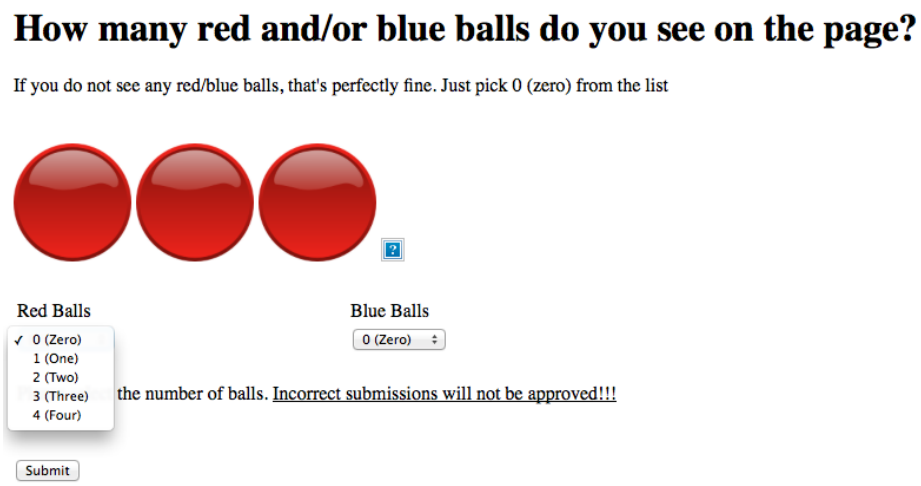


Figure 3.1: Screenshot of our HIT on MTurk

suring that users without IPv6 connectivity had no means of accessing the image file. Therefore, these users without an IPv6 address were able to only see a broken link (or nothing, depending on the behavior of the particular web browser used by the client) instead of the blue balls.

The image files for red balls were simple bitmap files. The images for blue balls were generated on-the-spot using a PHP script. For blue balls, the IP address of the client was embedded within the Uniform Resource Locator (URL) delivered to the user's web browser when she accessed the HIT. The blue ball image file URL was of the form:

```
http://turk.cmand.org/turk/img.php?A.B.C.D
```

where the second part of the URI represented the user's IPv4 address. If the user had IPv6 support, their browser would also fetch the blue ball image(s). If not, the browser would display a broken link since there is no `A` record in the DNS for turk.cmand.org. This allowed us to match the IPv4 requests with IPv6 requests for a given HIT. A sample request for a blue image file was logged on the server as follows:

```
2002:e60:239f::e60:239f - - [11/Mar/2014:01:17:36 -0400]
"GET /turk/img.php?14.96.35.159 HTTP/1.1" 200 37977
"http://www.cmand.org/turk/?assignmentId=2OQ5COWOLGRZ5GYJKV366NUZ12Y7YF
&hitId=2Z3KH1Q6SVQ8JGUEV4XLGOBVDDWL2L&workerId=A1JVUD5XUB9H48
```

Test ID	Completed HITs	Award Offered	Completion Time	Completion Rate
1	200	$0.26	10 hours	20 HIT/hour
2	142	$0.11	72 hours	2 HIT/hour

Table 3.1: HIT Properties and Their Effect on the Completion Times

```
&turkSubmitTo=https%3A%2F%2Fwww.mturk.com"
"Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/31.0.1650.57 Safari/537.36 OPR/18.0.1284.63"
```

Each user was awarded $0.11 to $0.25 for completing the HIT. For the first batch of 200 HITs, we awarded all users $0.26. All of the HITs were completed in less than 10 hours. Next we uploaded the same HIT, reducing the award amount to $0.11 and increasing the number of available HITs to 300. After three days, we had 142 of the 300 available HITs completed before we expired the HIT prematurely after three days as we had gathered enough data points to calculate completion rate over time and enough data points to start our preliminary analysis. The only reason we reduced the award was because there was a demand by the users to complete the simple $0.26-task, and we tried to maximize the number of measurements for the experiment. Table 3.1 shows the HIT running and completion times. We did not allow any user who has participated in the first batch of HITs to complete the HIT again. From this initial experiment, we observe a strong relationship between the HIT reward and completion time. We investigate this relationship in more detail in section 3.2.

Next, we analyzed the web logs of the web server for the duration of the experiment. The web server logged all the requests for resources, including the blue and red balls. Because all the resources used for the experiment were hosted under a specific directory on the web server, it was easy to categorize requests that were a result of our experiment.

From the log files, we extracted all HTTP requests that were due to our our survey. Due to the way HTTP works, a single test requires multiple HTTP requests from each user. For example, the web server logs a separate HTTP request for the `.html` file, and separate HTTP requests for each image file.

It is worth noting here that before MTurk users complete a HIT, they preview the HIT. The worker can then decide to complete the HIT, or they can return it (choosing not to perform the work). For our experiment, MTurk users that only previewed the HIT, without completing it, also generated HTTP web requests to our server and these requests are also included in the

Country	Number of IPv4 Requests
United States	322
India	148
Great Britain	13
Japan	7
Canada	7
Ireland	3
Others	28
Total	530

Table 3.2: Distribution of IPv4 HIT Requests by Source Country

results. Of the 3339 requests, 1485 of them were due to workers previewing the HIT.

A single user that completed the HIT is expected to generate about 10 HTTP requests (about three requests for the image files, and two requests for the image-generating `.php` files. Users generate, on average, five requests for previewing the HIT and another five requests while completing the HIT). The total number of requests to the web server is within the expected range.

Based on all of the HTTP requests, we extracted the source IPv4 and IPv6 addresses. We obtain a total of 530 unique IPv4 and 38 unique IPv6 addresses. We map the IPv4 and IPv6 addresses to countries using the maxMind.com's database [30], enabling us to determine from which countries the HTTP requests were originating.

3.1.1 Experiment 1 Results

Table 3.2 shows the country distribution of the IPv4 requests, grouped by country. Not broken down individually are 28 IPv4 requests from the following countries:

1. New Zealand, Malaysia, Hong Kong, Egypt, Germany (2 each)
2. Turkey, Singapore, Russia, Romania, Pakistan, Philippines, Netherlands, Indonesia, Hungary, Guatemala, Finland, Denmark, Switzerland, Brazil, Bangladesh, Australia, Argentina, United Arab Emirates (1 each)
3. Unknown countries[7] (2 total)

During the course of our experiment, we received a HTTP GET requests for the HTML file from 38 unique IP addresses using IPv6 source addresses. Of these 38 IPv6 addresses, nine

[7] maxMind.com's database is not complete, and due to the ever-changing nature of Internet Protocol (IP) address allocations and mappings, it is not always possible to map an IP address to a country.

ISP Name	Number of Requests
Comcast	9
AT&T	4
Verizon	2
Time Warner	1
Virginia Polytechnic Inst	1
CENIC[9]	1
Total	**18**

Table 3.3: Source ISPs of the IPv6 Requests

were using teredo tunneling (IPv6 addresses allocated from 2001:0::/32) and 11 were using 6to4 tunneling (addresses that start with 2002::/16). Tunneling technologies were designed to be just transition technologies. A user that uses IPv6 tunneling is not connected to an ISP using an IPv6 address. Instead, it establishes a tunnel with another tunnel provider, sends all the packets to this provider using IPv4 packets, with IPv6 packets as the payload of the IPv4 packets. The receiving host then decapsulates the IPv6 packets and sends the newly created IPv6 packets to the destination host. As our focus is on measuring the penetration of native IPv6, we do not consider these tunnel-using hosts as IPv6-enabled.

We used maxmind.com's database [30] to map the IPv6 addresses to countries and the ARIN whois service[8] to map the addresses to their ISPs. All of the remaining 18 IPv6 addresses geolocated to the USA. The ISPs that these IP addresses belong to are shown in Table 3.3 and the distribution of tunneling technologies are shown in table 3.4.

Tunneling Method	Number of Requests
6to4 Tunneling	11
Teredo Tunneling	9
Total	**20**

Table 3.4: Distribution of Tunneling Technologies

Finally, we attempt to compare our results from using MTurk to infer IPv6 adoption to other, publicly available data sources. Numerous entities measure the IPv6 adoption rates among Internet users. Table 3.5 shows the IPv6 adoption rates, as measured by Google, Akamai, and Cisco, respectively.

[8]https://www.arin.net

[9]Corporation for Education Network Initiatives in California. This request was used by us for testing the HIT and is removed from analysis.

We used these numbers from Google [31], Akamai [32], and Cisco [33] as the ground truth to compare our results against. Our results coincided closely with Google and Cisco's public measurements.

	Google [31]	Akamai [32]	Cisco [33]	MTurk Experiment
USA	5.27%	3.2%	5.25%	5.28%
India	0.16%	0.05%	0.14%	0.00%
Total	2.72%	1.50%	2.72%	3.21%

Table 3.5: IPv6 Adoption Rate Ground Truth vs. MTurk Experiment Inference

As can be seen in table 3.2, we received HTTP requests from 322 unique IPv4 addresses from users in the USA, and 148 from users in India. The number of requests from other countries were too few to make any statistically significant conclusions about other countries and are not included in the table. Because of the few data points we had, it would be incorrect to conclude, for example, that our measured IPv6 adoption rate for Great Britain was 0%. That is why we present the IPv6 adoption rates for two countries only in table 3.5

Through our experiment, we demontrated that the mechanical Turk platform can be utilized for crowd-sourcing internet measurements, as long as the researcher is aware of its limitations and its offerings. The results are self-selected and are not random. However, the user base of mechanical Turk workers are diverse and they present a representative sample of the overall Internet population.

3.2 Measuring our HIT Price Sensitivity

The second experiment that we conducted on mechanical Turk was designed to analyze the effects of the compensation amount on the completion time of HITs and on the quality of work performed by the workers. For this task, we used the same HIT that we had previously designed, except that on the HIT page, there was an explicit note saying that incorrect submissions will not be approved (therefore, it was explicitly stated that the workers who submitted incorrect answers would not receive any compensation).

To assess the effect of the compensation amount on the completion rates of HITs, we used the previous HIT as a template and uploaded more HITs on the mechanical Turk web site. Initially we priced the HIT at $0.05 for each successful submission and uploaded 100 HITs. Once the HITs were completed, we uploaded 100 more, increasing the compensation to $0.10. After all 100 HITs were completed, we uploaded 2 more sets with compensation amounts of

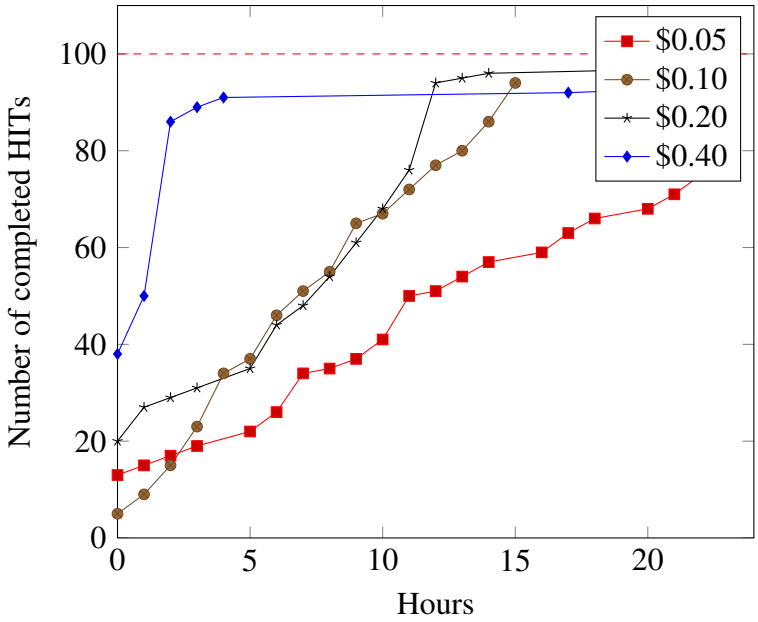


Figure 3.2: Effects of the Compensation Amount on HIT Completion Rate for Workers in India

$0.20 and $0.40. We conducted two concurrent versions of these experiments, one for users in USA only and one for users in India only. Because the workers without an IPv6 address were not able to see the blue balls on the HIT page, we ignored the user submitted values for blue balls and checked only the number of red balls for correctness. Any user who submitted an incorrect number of red balls was rejected, and no compensation was given for these incorrect submissions. In the end, we had about 800 data points (minus HITS that we rejected, discussed later in the section).

As discussed in section 2.4.1, each user was allowed to complete only one assignment in each batch of 100 assignments. However, we did not restrict users who participated in one batch from participating in other batches. So, a single user could potentially complete any one or more of the different priced HITs, i.e., they could complete the $0.10 HIT and the $0.40 HIT.

3.2.1 Experiment 2 Results

When searching for a HIT to complete, workers can sort and filter the available HITs based on the price. Table 3.6, figure 3.2 and figure 3.3 summarize our price sensitivity results.

Figures 3.3 and 3.2 show the effect of the compensation amount on HIT completion rates. The x-axis shows the number of hours that pass after the HIT is uploaded and is available to be completed by MTurk users. The y-axis shows the number of HITs that are completed. As

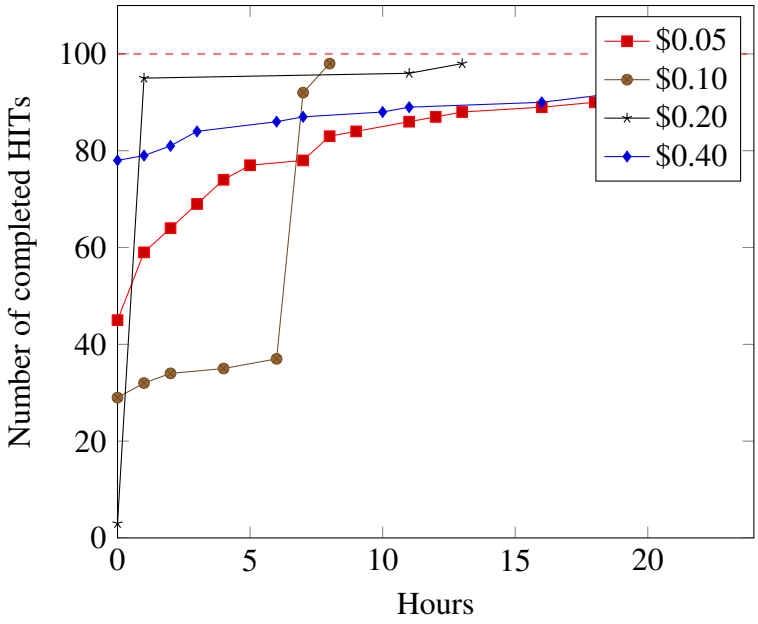


Figure 3.3: Effects of the Compensation Amount on HIT Completion Rate for Workers in the USA

expected, as we increased the compensation amount, HITs are completed more quickly.

Of note is that the $0.40 HIT is completed much more quickly than other HITs, and that there are some irregularities in the plots. For example, for the HIT that was targeted to workers in the USA only, the HIT that was paying $0.20 was completed much faster than the HIT that was paying $0.40, and the HIT that was paying $0.05 had about 80% of the HITs completed within 10 hours of being submitted to the webpage, which is faster than the HIT that payed $0.10 and had about 30% of the HITs completed in the initial 7 hours. We postulate that these differences are attributable to the time of the day that the HIT was submitted, and to the day of the week when it was available, though it is hard to isolate the effects of time without more controlled experiments.

On the mechanical Turk platform, we let the tests run for 24 hours, after which they expired and were not available for submission. For some of the HITs, not all of the 100 tests were completed. This is because of the design of the mechanical Turk site. When a worker accepts a HIT, he or she has a set amount of time to complete the HIT (for our tests, the time limit was set to its default value of 1 hour). If the worker does not complete the HIT in the allowed time, it is returned back to the system and becomes available for other workers to complete. Also, when the requester reviews the completed HITs and rejects a submission, it should again be available

Country	$0.05		$0.10		$0.20		$0.40	
	Total	Correct	Total	Correct	Total	Correct	Total	Correct
USA	94	92	100	100	100	100	100	100
India	99	95	100	93	100	97	93	88

Table 3.6: Compensation Amount vs. Number of Completed and Approved HITs

for others to complete. If the HIT's allowed time expires during these two time frames, not all tests may be completed. Therefore, for some tests, we obtain fewer than 100 data points.

3.2.2 Quality of Work Completed by Workers

For our IPv6 measurement inference task, we are not interested in whatever data the user submits (we are only interested in the HTTP request; the submitted number of balls is incidental to our true task). However, still we compare the number of displayed balls with the number of balls identified by the worker as part of the HIT. We identified few instances where a worker submitted the incorrect number of balls. Even for such a simple HIT, there were few users that did not click the correct radio button. This could be a result of user error, language barrier or ignorance. These users, as per our policy, were not compensated. Table 3.6 shows the number of completed HITs and the number of HITs that were approved (where the worker selected the correct number of red balls). It is worth noting that the workers in the USA generally produced more correct answers than the workers in India.

Therefore, we recommend that the requester should plan ahead and have some built-in screening questions to detect and prevent this form of automated behavior. It is easy to add a screening question; for example, asking for today's date somewhere in the middle of a survey-type HIT would eliminate most of these auto-clickers.

3.3 Issues Related to the Mechanical Turk Platform

The scope of our research was understanding the feasibility of using the mechanical Turk platform for general active Internet measurements. While there are some tasks for which the mTurk platform is very useful, researchers should also consider its limitations, and be familiar with all aspects of mTurk before utilizing it for active network measurements. Using an online platform like mechanical Turk with strong per-country population bias, where users have to actively go through the registration and user verification phases of amazon.com, inherently introduces different biases (such as self-selection bias) to the sampling process.

The intent of our mechanical Turk experiment was to understand whether sampling biases

prevalent in current active Internet measurement systems could be mitigated. However, introducing a new platform, namely the Mechanical Turk, brings new biases into the equation and these are discussed in the following subsections. Researchers should be aware of these restrictions before designing any experiments and analyzing their results.

3.3.1 Malicious Requesters / Malicious Users

As is the case with any other market, there are people that attempt to manipulate the system and gain an unfair advantage. On the mTurk platform, the requester has the upper hand, as the requester approves completed HITs before the worker can receive compensation. There is a grace period, where the requester can reject the work performed and the worker does not get any compensation. Although it might seem to be unfair for the worker, it is advantageous to the requester as it allows the requester to monitor the quality of the work performed.

To keep low-performing users that consistently produce a low-quality result out of the system, there is an approval rating for all workers. This works like the feedback system on online auction sites. When a requester rejects a HIT completed by a requester, the requester's approval rating goes down. When placing HITs on the system, the requester can require a certain approval rating. The requester can also require a certain minimum number of completed HITs to prevent newly-registered users from completing his/her HITs.

3.3.2 Malicious / Automated Activity on the Mechanical Turk

There are many tasks at which computers excel, and others where computers do not perform as well as humans. Amazon mechanical Turk platform was designed to be used for tasks that require human intelligence. So, if the task does not require any human interaction, it is easier to write a program or a script to complete the task. Also, for tasks that do not have an objective, true answer, such as an on-line survey, there is nothing to prevent the requester from simply clicking through the survey without even reading it. Most of the HITs on the mechanical Turk require human intelligence, as there is no point in paying a remote user for work that could have easily been completed freely by a computer script.

With the limited results that we had, it is not possible to rule out the presence of automated computer scripts that complete HITs on the mechanical Turk platform. However, it is easy to visualize automated human clickers that complete HITs. And our HITs were so easy to complete correctly that incorrect answers to our HITs suggest automated human behavior, where the user simply clicks on the requested links, without even reading the instructions or attempting to

complete the task correctly. These automated users will eventually be weeded out by the worker rating system and will not be able to qualify to complete any HITs because of their own low ratings.

3.3.3 Previewing HITs and Over-Constrained HITs

The mechanical Turk platform allows requesters to host the HIT on amazon.com's servers or on an external server. Hosting on the mechanical servers requires the requester to pick from available templates and is suitable for many tasks. However, hosting the HIT externally, where the requester has more fine-grained control, gives the requester more freedom on the layout and design of the HIT. The HIT is then advertised on the mechanical Turk web site so users can find it, and so that the workers are compensated according to mechanical Turk's rules.

Hosting the HIT externally was necessary for our HIT as we needed to collect the IP addresses of the workers and we needed one of the ball images to be served from a IPv6-only host. Before a worker accepts and starts working on a HIT, he or she previews the HIT. While previewing the HIT, the user's web browser makes an HTTP request to our server. The HTTP requests to our web server is all we needed for our particular measurement; the remainder of the HIT, reporting the number of balls, is incidental. Even if the user does not accept the HIT and does not complete the simple survey, we capture the user's IP address and whether his/her computer was IPv6 enabled or not.

This behavior can be exploited by requesters seeking to perform measurements similar to our own. To exploit this behavior, one can design a HIT that is over constrained. For instance, a HIT that requires the user to be located in, for example, U.S. *and* in India cannot be completed.

The requester can set a high compensation award for the HIT. For example $25.00 is a very high price for a simple HIT, when most HITs are priced for pennies. Many MTurk users sort the HITs by their compensation award and this would guarantee that the HIT receives high visibility. Even before previewing the HIT, the user sees the HIT *and* its requirements. It would not be possible to accept or complete this HIT, as the worker needs to meet *all* requirements of the HIT before s/he can complete it. Still, many users will probably preview the HIT, making the HTTP requests that the researcher wants. This would allow a requester to direct traffic to his/her externally hosted web site for free.

3.3.4 User Demographics and Geopolitical Distribution

Amazon does not disclose any information about the users on the Mechanical Turk Platform. The only information amazon advertises is that there are "more than 500,000 workers from 190 countries [34]." According to a study in 2008 [35], 76% of the users were from the USA and 8% were from India. Another survey by the same author in 2011 claims that 47% of the users were from the USA and 34% were from India [36]. Another paper [23] gives more information about demographics of mTurk users and analyzes shifts in users demographics of "mTurkers".

Mechanical Turk users in the USA can have their earnings wired to their bank accounts. Worker in India can receive checks mailed to their addresses. Workers in other countries can receive only Amazon.com gift certificates for their work. This limits user participation of users from countries other than India and the USA, and explains why the majority of workers are from these two countries.

Despite the fact that the majority of the mTurk users are from the USA and India, there are workers from all around the world that actively complete tasks on a daily basis. Even our HIT, which ran for a few days, was completed by users from 29 distinct countries.

3.3.5 Self-Selection Bias

Any sampling that is performed amongst mechanical Turk users will have a self-selection bias. The distribution of mechanical Turk workers are not uniform around the world, and workers actively need to preview and elect to complete any HITs that they perform, which is again not uniformly distributed. Because of these non-uniformities, it is not possible to speak of a truly random sampling in the mechanical Turk platform. The researchers should always be aware of this self-selection bias when conducting researcher on mechanical Turk platform.

3.3.6 Language Barrier

Although there is no requirement regarding it, the rules of supply and demand and the dynamics of the workplace on mTurk has limited the number of non-English HITs that are available. Of over 300,000 HITs that are available, a search for "Spanish" results in only 4 HITs. A search for "para", a common occurring word in Spanish, results in 12 HITs, all of which are English (para is used as a prefix to paragraph in all HITs).

The prevalent language of the platform necessitates that all workers should have a basic understanding of the English language. This again limits the workforce of the platform to English-

speaking workers.

3.3.7 Geographical Non-Diversity

Amazon offers researchers easy and cheap access to users from around the world. There is no other existing platform that would allow a researcher to acquire samples from around the world, in a timely manner, and for a very low price. However, despite Amazon.com's claim of having workers from 190 countries, it is very hard to get results that are targeted to users in specific countries. Amazon allows the requesters to limit the HIT completion to users in specific countries. The requester can make a HIT available to users in a certain country if he so desires.

We tried to take advantage of this feature (this is specified in the *Locale* property of the HIT) and targeted one of our experiments to users in Japan, Turkey and USA only (with a price-tag of $0.26 for each completion). After two days of running the HIT, we received only 56 results, all from the USA. This clearly hints at the difficulty one might have when targeting users from specific countries. Although there are "occasional" workers from around the world, the majority of the active workers are in India and the USA.

Amazon.com's advertised user diversity somewhat reflects that of the general population in terms of age, income distribution, etc. However this is mostly true for users in the USA. If a researcher is targeting the U.S. (or Indian) population only, MTurk might serve the researcher's needs of respondents or data points. However, partly due to lack of financial incentives for users outside these two countries, we were not able to reach many users outside these two countries. In any study that requires user participation from other countries, it might not be possible to to collect samples of statistical significance.

3.4 Future Work

Amazon.com's Mechanical Turk offers a new platform that has not been exploited for Internet measurements. This project acts as a proof-of-concept of the idea only.

In this experiment, we gathered HTTP requests from 530 unique IP addresses. The limited number of data points limited our analysis to a per-country level. In a future study with more data points, the analysis can be done on a per-AS or per-ISP level.

For this thesis, we used simple "webbugs" to measure IPv6 adoption rate. The same idea can be used on any server that serves HTML pages to gather IPv6 penetration statistics for the website's users. However, there might be tasks other than HTTP requests or simple web surveys that the

Mechanical Turk platform can use to gather Internet measurements. Researchers are limited only by the Amazon Mechanical Turk Participation Agreement [37], its policies [38] and the creativity of the researcher.

Christin, *et al.* [19] have successfully demonstrated another way to utilize MTurk for world-wide participation in a research. They were able to get about 1000 people to download an executable file to their computers, run it for an hour and do a survey about their computer usage. They concluded that users frequently disregard common security best practices and download random, unverified executables from the Internet and run it on their computers with full local administrator access for compensation amounts less than $1.00.

We also experimented in using a similar approach. We prepared a simple HIT, requiring the users to download the Spoofer [9] program from the Internet and run it on their computers. We initially set the compensation amount to $0.26. After a very short time, our HIT was reported to Amazon.com for removal and was subsequently removed from the MTurk site. We received an email about our HIT, citing the MTurk's terms of service [37]. Although the Terms of Service does not mention anything about the nature or the content of the HITs that are allowed on MTurk, Amazon.com's general policies explicitly prohibit any "HITs that require Workers to download software [38]".

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Spoofer Project

In this chapter we describe and present analyses of the data collected by the Spoofer project. The goals of our analysis were as follows:

1. Better understand how the SCR has changed over time for various IP prefixes, autonomous systems, and countries.
2. Ascertain the quality of our inferences by sub-sampling the data, i.e., can we use a portion of the data to make accurate conclusions about the whole. For example, how much data do we need to estimate the SCR within a given confidence interval of 5%?
3. How does the introduction of additional data points affect the overall result? For this, we used the bubble charts in 4.2.1 to plot the changes in the population parameters with the addition of new data.

4.1 Spoofer Data

As detailed in section 2.5, clients voluntarily download and run the spoofer test program on their computers, which must run with administrative or root access. After the spoofer program finishes probing, the results of the tests are stored in a database. We have full access to the complete spoofer results database for our analysis. We analyze spoofer data inclusive of the time period between 12 February 2005 to 26 February 2014. Table 4.1 shows the database tables that were analyzed and the number of records (i.e., database rows) in each table, together with a short description of the table's purpose.

There are 33,683 sessions and clients make multiple attempts to send spoofed packets using numerous source IP addresses to multiple destinations. This testing process produced a total of 806,678 test results, where every attempt by a client to send a spoofed packet to one of the destinations is considered a result. Each result has a disposition, either "spoofable" or not.

Of the 33,683 sessions, we filter out (1) those that fail due to Operating System (OS) restrictions[10] and (2) the ones that were run behind a NAT device. Of the 806,678 data points, 23,068 were filtered out due to (1) and 422,509 were filtered out due to (2).

[10]The client program does not run on machines running Windows 9X

Table Name	# of records	Description
Sessions	33,683	Stores session information about client tests
Spoofs	52,256	Contains information about successful spoof tests
Failed	295,929	Contains information about failed spoofing tests
AdjacentSpoofs	156,990	Contains information about successful adjacent spoofing tests
AdjacentFailed	301,503	Information about failed adjacent spoofing tests
NonSpoofed	45,052	Contains information about destination addresses that were able to receive non-spoofed UDP packets. This information determines the base case
DNS	40433	Contains information about TLD of clients

Table 4.1: List of Tables Used by the Database

4.2 Analysis

After filtering, we are left with 361,101 test results spanning 9 years, from February 2005 to February 2014. We conducted two analyses over the data. The first analysis was aimed to determine the SCRs for each country over time and alowed us to visualize how the rate has changed over the time span of the Spoofer project. The second analysis was cross-validation of the existing data.

We first analyzed the IP addresses of the hosts that we had test data from. If any of our servers was able to receive a spoofed packet from an IP address, we classifed that IP address as it can send spoofed IP packets. If none of the spoofed packets from the IP address were received by our servers, we classified that IP address as it cannot send spoofed IP packets. For some IP addresses, we had mixed results. When an IP client runs the Spoofer program, all the test results from that run are stored in the database with a unique *session ID*. The servers allow a client to report results no more than once in a week. When an IP client runs the program, it sends spoofed packets to multiple servers. In all cases where at least one of the servers was able to receive a spoofed IP packet, we classified that client's IP address as it can spoof packets and then ignored any failed attempts from the same client at the same session. And for this analysis, only IP clients that can spoof arbitrary, or non-neighboring, IP addresses are considered. If there are multiple sessions from any IP address, they are all considered as part of the analysis. So, if the status of an IP changes over time, we would have both results as separate in our results.

Initially, we grouped the results by their countries. For each country, we calculated;

1. the number of successful spoofs,

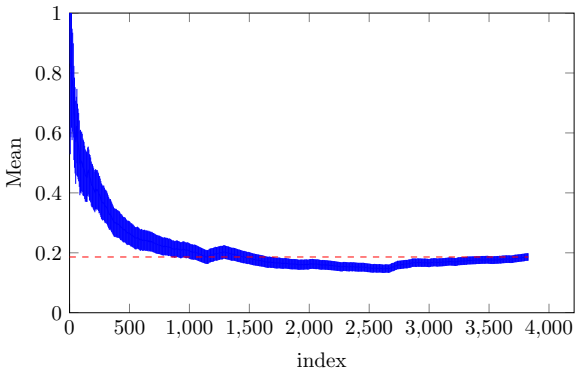


Figure 4.1: Cumulative SCR for the USA Displays an Initial Decrease in the SCR. The Large Number of Data Points Results in a Tight Confidence Interval.

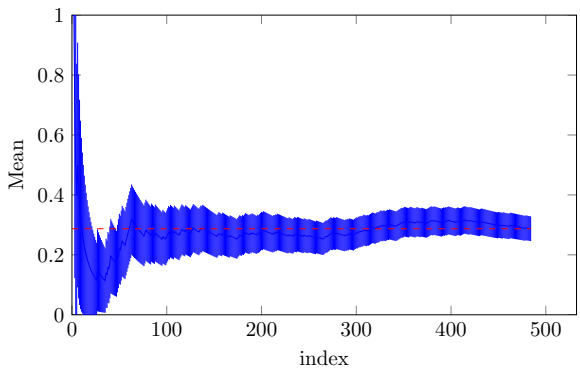


Figure 4.2: Cumulative SCR for India Shows a Similar Initial Decline, and Due to the Fewer Data Points, the Confidence Interval is Larger.

2. the number of failed attempts, and
3. overall SCR

4.2.1 Analysis of Spoofing Capability Rate Over Time for Countries

We plotted the changes over the SCR for each country and TLD. For example, figure 4.1 and figure 4.2 show the cumulative SCR for IP addresses that are located in the USA and India, respectively. These two countries have the highest number of measurements, compared to other countries, so we focus on them initially (Appendix 5.3 examines other countries). In the figure, the x-axis shows the number of data points we have and the y-axis shows the percentage of hosts that can send spoofed IP packets. Because the test data are not uniformly distributed over time, rather than using the time/date of test data, we used the indices of the test data as the label for the x-axis.

The figure shows the cumulative mean of all tests up to a specific point. For example, the first data point shows only the results of the first test. The plot for $x = 1000$ shows the mean of the SCR up to the 1000th data point. The dashed, red line shows the mean for all measurement data. The graphs helps us to visualize the changes over time. The graph also includes the 95% confidence interval values. We used the Equation 2.5 to compute the standard errors for each data point and added error bars to each data point.

Appendix 5.3 contains observed SCRs for the countries that we have at least 50 measurements from, and appendix B contains the same analysis for the current year (from 26 February 2013

to 26 February 2014). We have a few observations about the plots:

1. The country plot for USA in appendix A.1 has the most number of data points. The initial SCR for USA starts very high and declines gradually for the initial 1000 data points. After that point it settles around at 20% and for the majority of the test period, the SCR stays within the 95% confidence interval.
2. Due to the number of test results, USA also has the tightest confidence interval.
3. Although SCRs across countries differ, the SCRs for individual countries are, for most part, stable. There are no noticeable increasing or decreasing trends in individual SCRs.
4. Some countries have long series of spoofed or non-spoofed IP addresses at the beginning of the observation period. This results in some charts starting with a SCR of 100% and it takes a long time for these countries to converge to their mean values.
5. Each country has a distinct shape in their long-term SCR plots, however, the plots are not enough to make any conclusions about the underlying reason of the changes in the SCR.

4.2.2 Cross Validation of Test Data

For the cross-fold validation, we performed an analysis of test data by grouping the test results by the country first and then repeated the same analysis by grouping the test results by the TLD of the IP address. For each country and TLD, we split the data points into two complimentary sets: one set contained all the measurements in the initial 80% of data points, and the other set contained all the remaining 20% of the data points. We used the 80% set as the "training" set, and the 20% set as the "testing" set.

We calculated the SCR for each set and then calculated the difference by subtracting the SCR of the training set from the SCR of the testing set. We then normalized the values by dividing the difference by the SCR of the training set. A positive value implies an uptrend of the SCR in the test set, and a negative value implies a downtrend in the testing set. The x-axis of the charts show this difference in the SCRs between the two sets.

$$Difference = \frac{Rate_{test} - Rate_{train}}{Rate_{train}}$$

where $Rate_{train}$ is the SCR in the training set and $Rate_{test}$ is the SCR in the testing set. A value of 1 indicates that SCR of the testing set is twice that of the training set, and a value of 0 indicates that the SCRs of the two sets are identical.

Then for each country, we inspected each test data in the test set and checked if the ASN of the data was already in the training set. If the ASN did not exist in the training set, we classified the test point as a "new" measurement. The y-axis of the charts show the fraction of new AS numbers in the test set for that specific country. A value of 1 indicates that all the measurements in the test set were "new" ASes, while a value of 0 indicates that all measurements in the test set were from existing ASes.

Figure 4.3 and figure 4.4 show the chart for each country and TLD, respectively, when the data points are analyzed using ASNs for 80/20%, 60/40%, 40/60% and 20/80% splits of the data points.

The size of each bubble on the figures is proportional to the square root of the number of measurements in the dataset. For comparison, the US had a total of 8280 data points, Great Britain (GB) had 898 data points, and Turkey (TR) had 188 data points. Table 4.2 shows the number of data points that were used to build the bubble charts.

4.2.3 Grouping By Prefixes

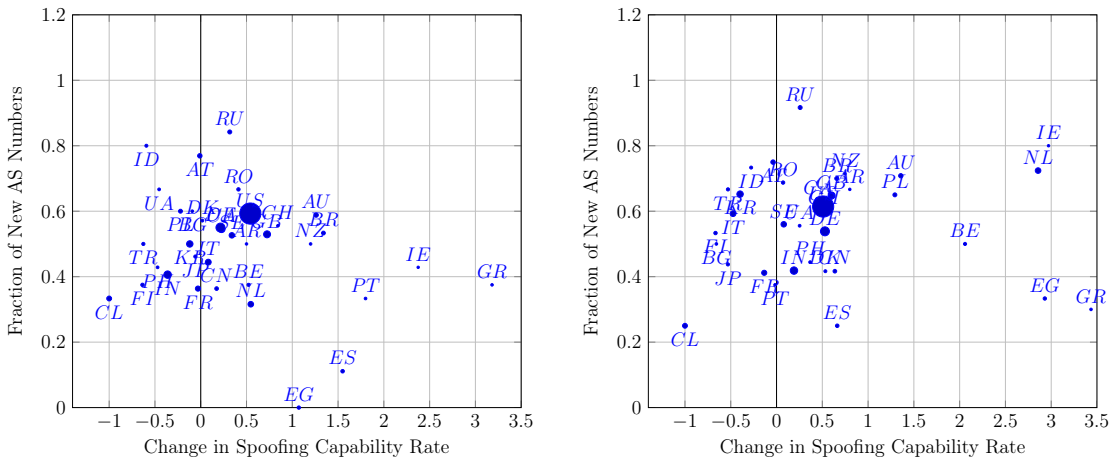
We observed that the spoofing policy in many ASes is not consistent across the network prefixes advertised by that AS. An AS consists of multiple ingress points and each ingress point, or router, might have different configurations. Also the customers of an AS may implement different policy than the AS as a whole. So, the filtering policies of ASes are not consistent across all the prefixes that they advertise.

We then did the same analysis described in section 4.2.1. We again divided our measurements into training and testing sets, and performed the same analysis for each country and for each TLD. But this time, we inspected the advertised prefix to which the IP host belonged. This resulted in a more granular cross-fold validation analysis, as each AS advertises multiple prefixes. For each data point in the testing set, if the new data point's prefix was not in the training set, we classified it as a "new" data point.

Figures 4.4 and 4.6 show the results of our analysis. When categorizing data points as new, according to their prefixes, results in higher y-values for all countries and TLDs, compared to categorization by AS Numbers.

4.2.4 Cross Validation Using Different Fold Ratios

We then performed the earlier analysis, this time using a 60/40%, 40/60% and 20/80% splits of the data. For each split, we used the initial set for training and the remaining set for testing purposes. The charts from each split is given in figures 4.3 to 4.6.



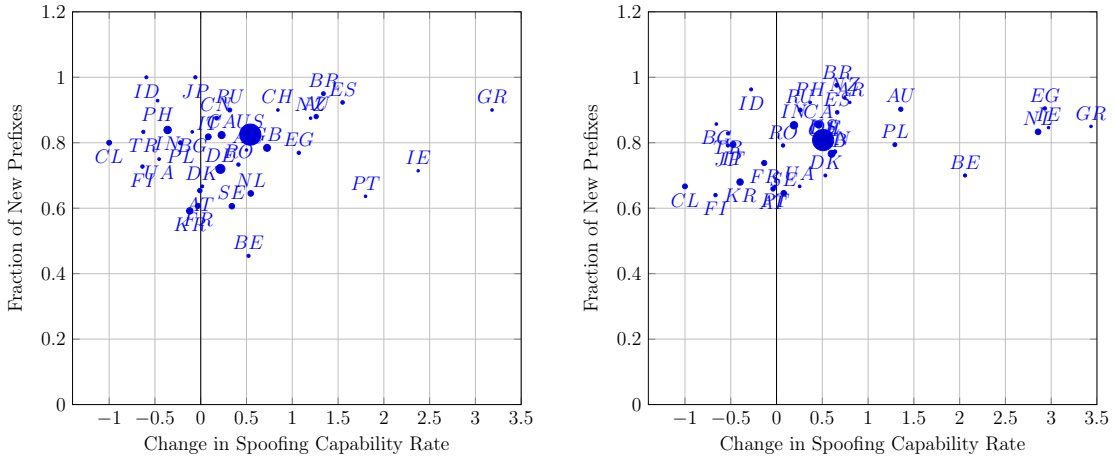
(a) ASN 80/20

(b) ASN 60/40

(c) ASN 40/60

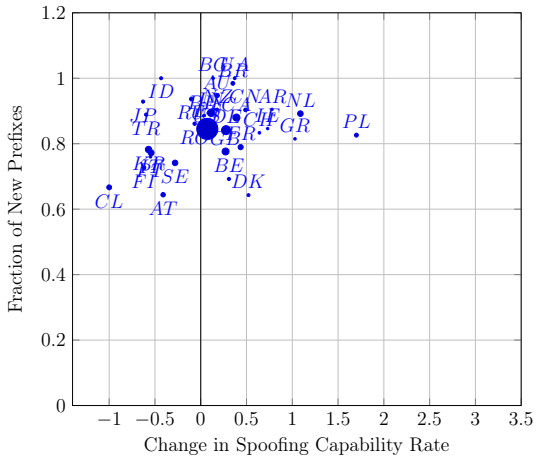
(d) ASN 20/80

Figure 4.3: Bubble charts created using different split points for training and testing data sets. As the training set size decreases, more data points are categorized as "new", resulting in overall lower y-values.

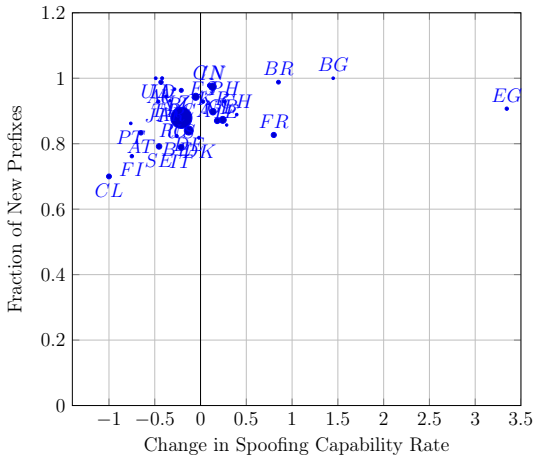


(a) Prefix 80/20

(b) Prefix 60/40

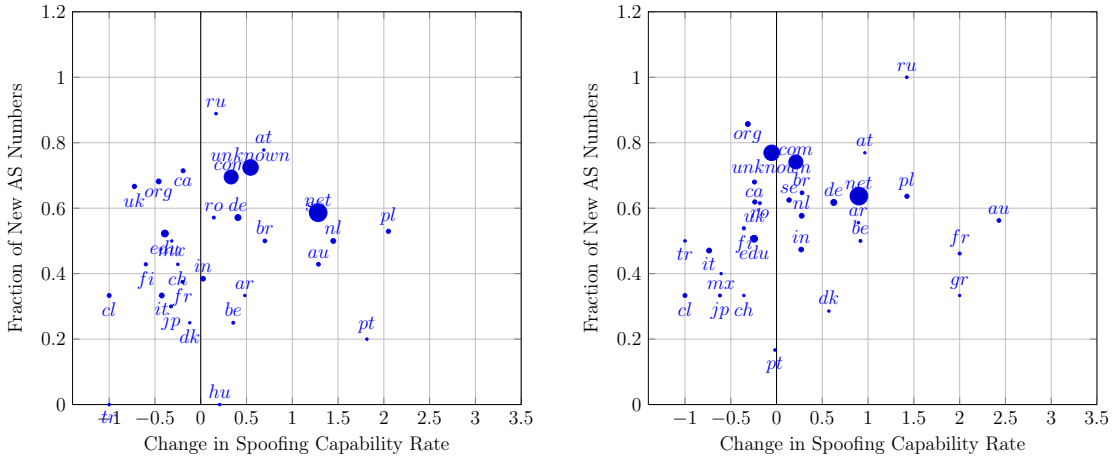


(c) Prefix 40/60



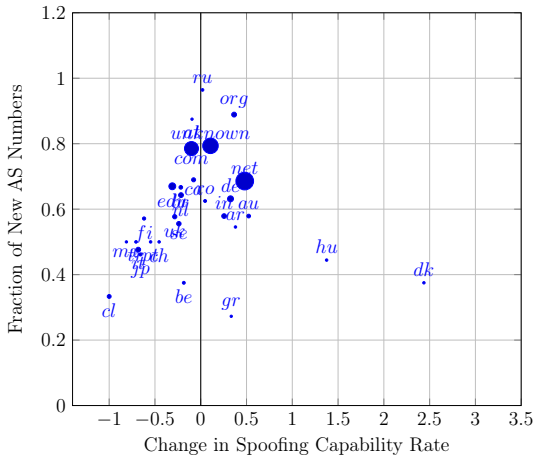
(d) Prefix 20/80

Figure 4.4: Categorizing data points as new, according to the prefixes results in higher y-values for all countries, compared to categorization by AS Numbers. Similar as before, decreasing the size of the training set results in more "new" daa points, noticable by overall higher y-values

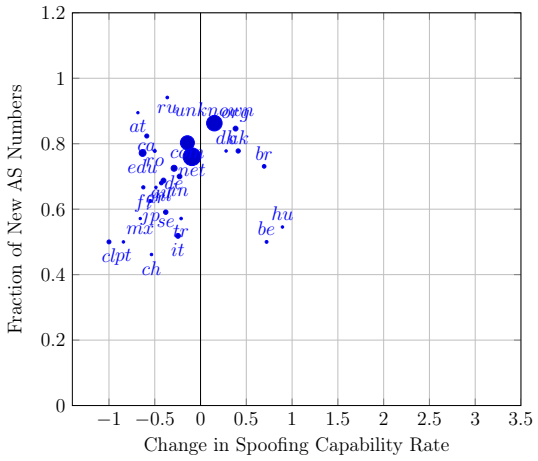


(a) ASN 80/20

(b) ASN 60/40

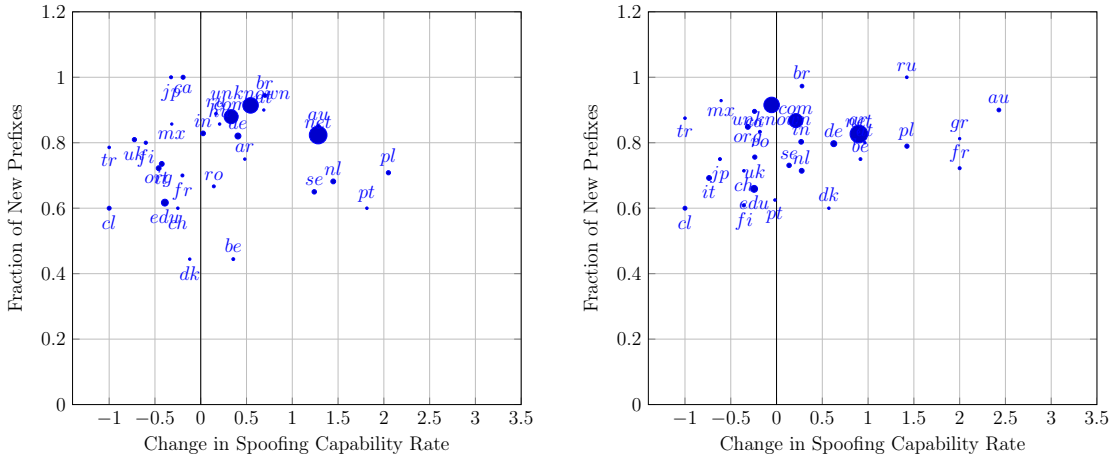


(c) ASN 40/60



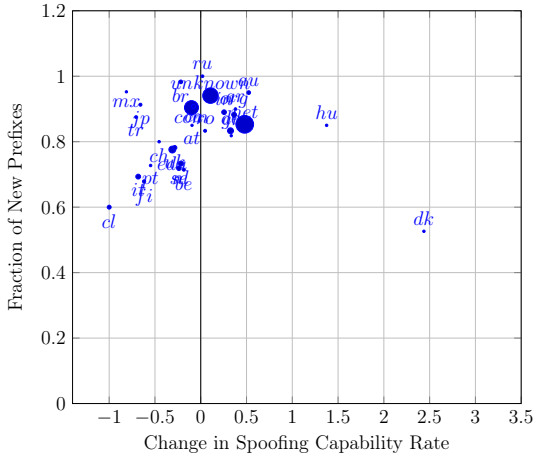
(d) ASN 20/80

Figure 4.5: Bubble charts created using different split points for training and testing data sets. As the training set size decreases, more data points are categorized as "new", resulting in overall lower y-values.

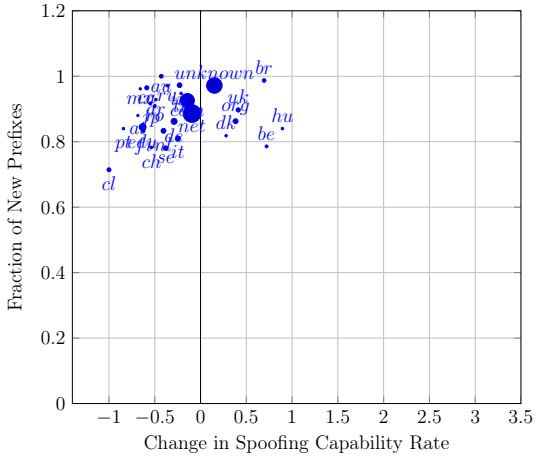


(a) Prefix 80/20

(b) Prefix 60/40



(c) Prefix 40/60



(d) Prefix 20/80

Figure 4.6: Categorizing data points as new, according to the prefixes results in higher y-values for all countries, compared to categorization by AS Numbers. Similar as before, decreasing the size of the training set results in more "new" daa points, noticable by overall higher y-values

4.3 Results

When creating the bubble charts, we were expecting the bubbles to cluster together, and we would make inferences as follows:

1. If the introduction of new ASes or prefixes had caused big changes in the SCRs (bubbles with a high y-value and a high absolute x-value), we could conclude that the samples that we drew and used for training were not representative of the overall population and that more samples had to be taken for conclusive estimates.
2. If the data in the testing set for any country or TLD were mostly new ASes or new prefixes, and there was not a big change in the SCR (bubbles with a small x-value and a large y-value), we could conclude that the samples in the training set were representative of the overall population.
3. For any country or TLD that had a bubble with a high absolute x-value and a small y-value, we could conclude that there has been a shift in policy in filtering policies of that particular country or TLD.
4. For any country or TLD that had a bubble with a low x-value and low y-value, we could conclude that the data points in the test set are just repetitive and are not giving us any new information.
5. After using different points to split the data into training and testing sets, we observed that changing the split point for different bubble charts do not tighten the variability of the results. Using different split points to separate the training data from the testing data, changes the x-values for different countries. However, the changes do not lead to a definite point that tightens the variability for all countries. If any split had resulted in country marks to center around the y-axis, we could have concluded using that specific split point would result in a good estimate of the population parameters. However, the analysis failed to produce such a split point.

Analyzing all the bubble plots, we could not see any correlation between the introduction of new data points and any change in the SCRs. Also, we could not see any plots that exhibited any of the listed behavior and that had enough data points to make any significant conclusions. Most of the data points were clustered in the middle of the graph, with y-values close to 0.5 and x-values in the range $(-1, 1)$. Two main observations about the data are:

1. None of the countries in the bubble charts had large absolute x-values with corresponding high y-values (case 1 above), and there were many countries with small x-values and large

y-values (case 2). This suggests that existing data points are representative of the general Internet.

2. There were no countries that exhibited the behavior described in case 3 and case 4 above, so this suggests that we do not have many repetitive measurements.

When using the 80/20% split, the country with the most number of measurements, U.S., had an increase of 54% in its SCR between the training and the testing set. For the training set, the SCR was 16.79%, and the SCR for the testing set was 25.85%. Plot 4.1 shows the SCR for the USA over years. Figure 4.3 (a) also shows the change in SCR for the USA over time and shows an 50% increase from the training set and the testing set.

This might initially seem inconsistent. However, because the training set contains 80% of the values for figure 4.3 (a), it takes a lot of data points to make the trend noticeable in figure 4.1. At the 80% mark, the SCR is about 17% for USA. The SCR for the testing set is about 26%. But because of the weight of the initial set, the SCR for the entire set (80% and 20% combined) becomes 18.6%.

Some of the countries had major increases in their SCRs. For example, the SCR in Greece more than tripled (increased 3.18 times). However, in the absence of more data, it is not possible to make any conclusions as to why this might have happened. However, the bubble charts show that the fraction of new ASes is relatively low for Greece between the testing and the training sets (it is 0.38). This suggests that the ASes that we obtained measurements from might have shifted their policies and stopped implementing ingress filtering to allow clients to send spoofed packets.

What was interesting to see in figure 4.5 (a) and 4.5 (b) is that there is a big increase in the SCR in `.net` TLD, a smaller increase in `.com` TLD and a small decrease in the `.edu` TLDs.

Deciding on a split point for the test data has a big effect on the outcome of the bubble charts. Picking the testing set to be bigger than the training set increases the overall y-values of the test data, as shrinking the training data makes it more likely for the data points in the training set to be a "new" data point.

At the beginning of the analysis, we were hoping to get a split, where most of the countries or TLDs had x-values close to zero, suggesting that using this split, we could use the sample parameters in the training set to predict the outcome of the testing set. However, none of the many data splits that we used gave us the expected chart. This suggests that we cannot use past

sample parameters to predict the overall population parameter accurately.

Another thing worth mentioning about the bubble plots is that the increases are normalized by dividing the increase by the SCR of the training set. This implies that a large increase in the spoofing difference between the training and data sets does not necessary mean a large increase in absolute terms. For example, in the 80/20% and 60/40% bubble charts, Greece has exhibits an over 300% increase in SCR between the training and the testing sets. However, Greece has an overall SCR of about 7%, and has only 121 test data. Depending on the split of the training and testing sets, where the 8 sppofed tests falls can cause a very large increase in the bubble charts. Other countries with similarly few data points exhibit similar behavior in the bubble charts.

Country	Number of Data Points
US	8280
DE	1497
IN	1020
CA	924
GB	898
KR	763
IT	626
NL	581
SE	558
FR	489
CL	440
AT	367
AU	350
RU	296
PL	287
BR	272
CN	250
ES	234
FI	214
RO	211
EG	207
TR	188
JP	187
BE	184
ID	166
DK	160
UA	142
PH	137
BG	130
PT	129
CH	128
NZ	122
GR	121
AR	114
IE	109

Table 4.2: Number of Data Points that were Used to Create Bubble Charts

CHAPTER 5:
Conclusions and Future Work

In this chapter we summarize our findings and suggest areas for future work.

5.1 Summary

In chapter 2, we outlined various sampling methodologies that are used by researchers when taking measurements from a larger population and how the results of these samples can be used to estimate parameters of the overall population. Sample size plays a large part in the accuracy of the estimated parameters. Normally, larger sample sizes give tighter confidence levels and lower sampling errors. In table 2.2, we also gave the required sample sizes for a given sampling error and desired confidence level.

We alos outlined the ecosystem on Amazon's mechanical Turk platform and described the workflow of HITs. We included information about geographical distribution and user demographics of mechanical Turk users. We concluded that a vast majority of users are from the USA and India. Despite Amazon.com's claims of having users from more than 190 countries, we presented the reasons why the user participation is mostly limited to only two countries and how this complicates efforts to target users in other countries. We also provided guidelines to network measurement researchers wanting to use mechanical Turk.

Next, we examined Amazon's mechanical Turk as a platform for crowd-sourcing active Internet measurements and analyzed how successfully the results match up with real-world data. We concluded that, except for the restrictions discussed earlier, the parameters that we collected from mechanical Turk users differed by less than 3% of the otper publicly available data. resources [31–33].

Initially, we had planned on using mechanical Turk for collecting more data points for the Spofer project. We designed a HIT that would require users to download, install and run the Spoofer program. The program would send data to the server and provide us with data points. However, after submitting the HIT, we were informed that requiring users to download and install executables was against mechanical Turks's terms of service, and our HIT was rejected. This limits the methods that researchers can employ for research on mechanical Turk.

We also analyzed how the amount of compensation on the mechanical Turk platform affects

the rate at which the tasks are completed and how it affects the quality of work completed by the workers. We concluded that as the compensation amount increases, tasks are completed at increased rates, but that we did not observe any correlation between the compensation amount and the work quality. We did observe that workers in the USA completed almost all of the HITs correctly, regardless of the compensation amount. The work quality for users in India was comparatively lower.

Also in chapter 2, we discussed how IP source address spoofing works and how it is hurting the overall Internet population. We also gave a few examples of how IP spoofing is being used by attackers for DDoS attacks, and how ISPs can prevent spoofed packets by installing ingress filtering at the ingress points to the Internet.

In chapter 4, we presented an analysis of Spoofer data by the prefix, autonomous system, and country-level granularity. We plotted the SCRs and the 95% confidence intervals over time for different countries, and analyzed the plots. The plots highlighted the changes in the SCR over time. We also did an analysis of the Spoofer data for the current year for selected countries with the most data points.

Next, we proposed a method for using existing measurements to predict the outcome of future measurements and the actual parameters of the population. Using bubble charts, we analyzed how data from new prefixes, autonomous systems or countries affect the estimated population parameters, and how we can use these bubble charts to classify reasons for change as policy change, or simply changes due to new measurements.

After our analysis, we failed to see any correlation between the introduction of new data points and any change in the SCRs. This can partly be attributed to temporal changes in the underlying system and the very long sampling process.

We concluded that for any project that aims to monitor changes in population parameters over time (like Spoofer), a constant influx of measurements from different networks is needed, and we can not estimate population parameters using traditional sampling methods.

5.2 Future Work

- For the Spoofer project, we need a constant influx of measurements in order to correctly estimate the SCR of the Internet. Other methods to solicit user participation for the Spoofer project would help to measure the Internet's spoofable address space.

- The bubble charts and n-fold analysis can be used for similar analysis of other, larger datasets. It would be useful to use bubble charts for analysis of other data sets and perform a quantitative comparison and analysis of different splits of data. However, these analyses were beyond the scope of this thesis work.
- In chapter 4, we plotted the SCRs for various countries over time However, the data points are not uniformly distributed over the time span of the project. So, instead of using time values for the x-axis, we used the index of our chronologically sorted data points as values for the x-axis. Additional insights could be derived from plots with the time-values of data points on the x-axis, instead of indices.
- Last, developing new, creative use cases for the mechanical Turk platform to conduct network measurements is an active area of current research.

5.3 Conclusion

In addition to its commercial users, Amazon's mechanical Turk offers a new venue for crowd-sourcing user studies and surveys. Especially for research in the social sciences and psychological surveys, it offers very cheap data collection, with more diverse user demographics than the average campus environment. We have demonstrated that Amazon's mechanical Turk can also have limited use for performing active Internet measurements. However, non-deterministic and unpredictable user participation limits its use for particular research projects. User acceptance of HITs is completely voluntary. Although researchers can solicit higher participation by increasing the compensation amount, there is no guarantee that the researcher will collect any given number of data points in a set time, or from a set distribution of user locations. Also, for studies or research that needs large user participation, the costs of the research can be prohibitive for small-budgeted projects as the cost of data collection increase linearly with the number of data points collected.

A large limiting factor, as it relates to Internet measurement, is the geographical distribution of mechanical Turk users. If the researcher needs user participation from users outside USA or India, he or she might not be able to solicit enough data collection in a deterministic time. This would create bottlenecks in the data collection stage if the researcher relies exclusively on mechanical Turk for data collection.

From the Spoofer data, we conclude that measurements over time are affected by changes in the measured system, and it is not easy to use traditional sampling strategies on a ever-changing system. Data collection for the Spoofer project is completely self-selected and requires active

voluntary participation of the user to download and run the test program. This results in a very slow sampling process. In the meantime, the Internet is constantly evolving, and the underlying parameter that we are trying to estimate is changing. This change results in unpredictable results with the sampling process and the traditional methods for estimating population parameters do not apply verbatim.

For any dynamic system in general, it is usually not possible to "freeze" the system to take a snapshot. Therefore, it is important to take all measurements within the shortest possible time span. For the Spoofer project specifically, we cannot conclude at any time that we have enough data points and can stop gathering new data points.

In chapter 5, we also tried to use existing data to predict the values of future measurements. However, this is usually not possible for a dynamic system such as the Internet. Using data that is many years old to forecast future data did not lead to any conclusive results.

APPENDIX A:
Spoofing Capability Rate Changes for Countries over Time

The plots below show how the SCRs have changed over time for various countries. The values on the x-axis show the number of measurements from that specific country, and the y-axis shows what percentage of clients in that particular could send IP packets with arbitrary source IP addresses. The plots also show the 95% confidence interval for the measurements. The shaded areas show the mean value $\pm 2 * standarderror$.

Note that clients that were able to spoof only neighboring addresses are not included in these plots. The values on the y-axis show the mean of all measurements upto and including the value on the x-axis. We have included only the countries from which we had over 50 measurements.

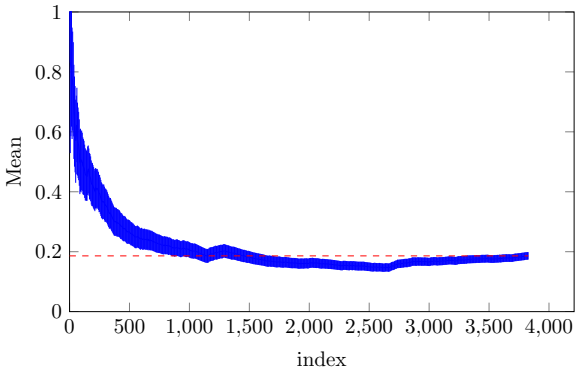


Figure A.1: Cumulative SCR for the USA

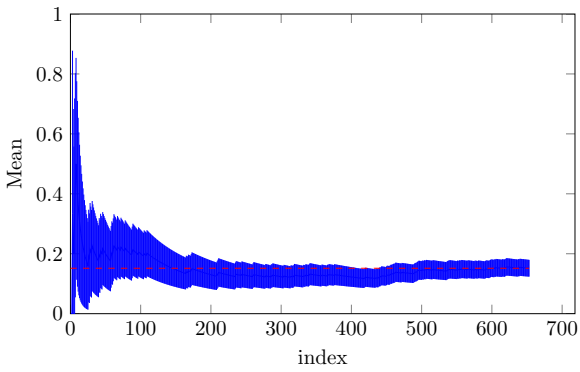


Figure A.2: Cumulative SCR for Germany

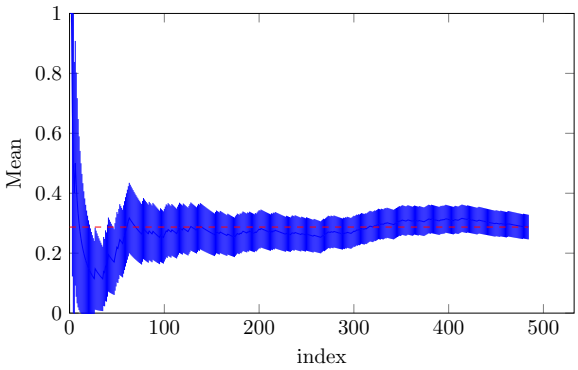


Figure A.3: Cumulative SCR for India

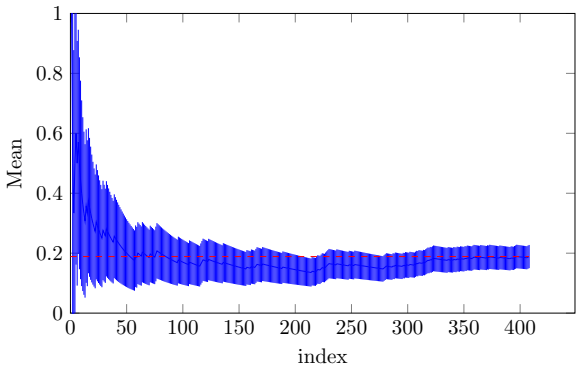


Figure A.4: Cumulative SCR for Canada

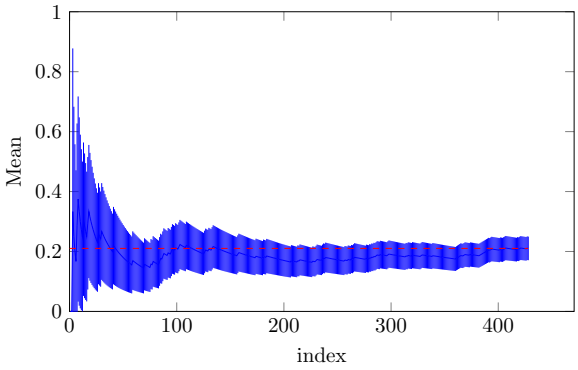


Figure A.5: Cumulative SCR for Great Britain

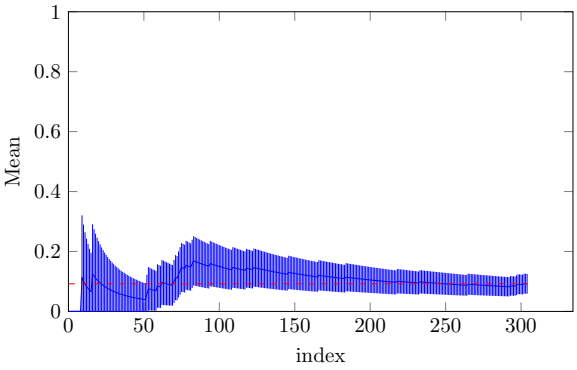


Figure A.6: Cumulative SCR for South Korea

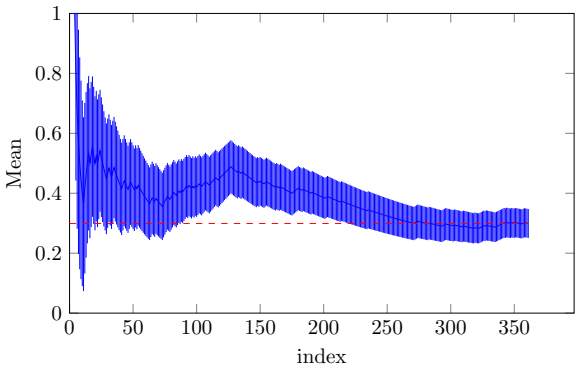


Figure A.7: Cumulative SCR for Italy

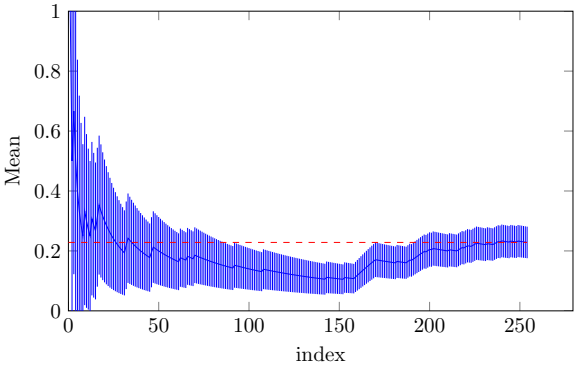


Figure A.8: Cumulative SCR for The Netherlands

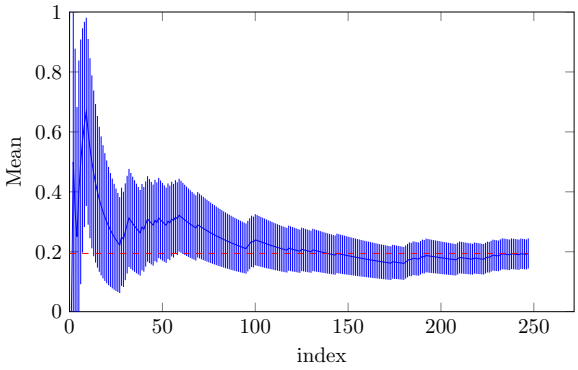


Figure A.9: Cumulative SCR for Sweden

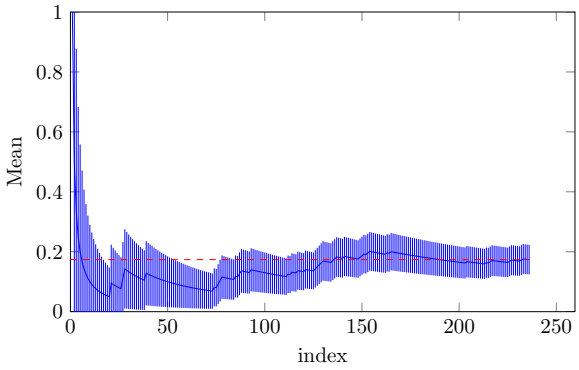


Figure A.10: Cumulative SCR for France

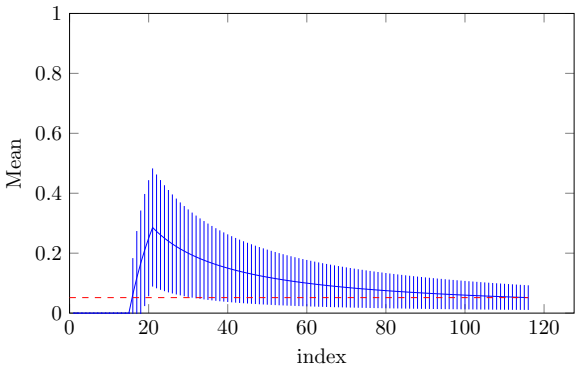


Figure A.11: Cumulative SCR for Chile

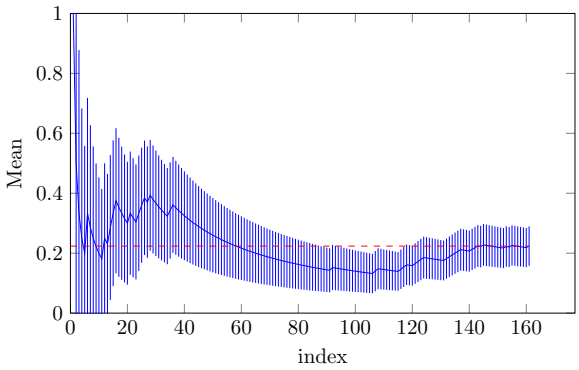


Figure A.12: Cumulative SCR for Australia

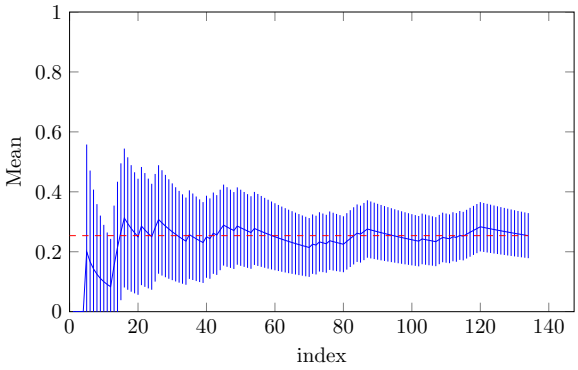


Figure A.13: Cumulative SCR for Russia

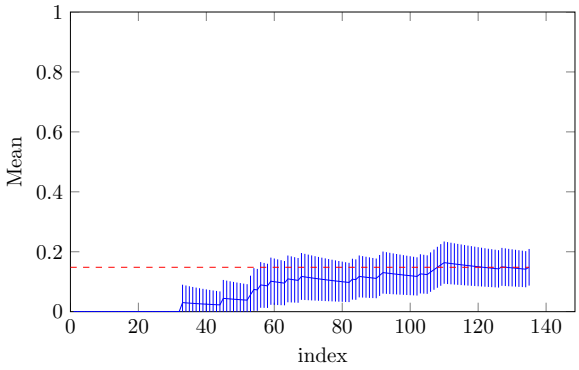


Figure A.14: Cumulative SCR for Poland

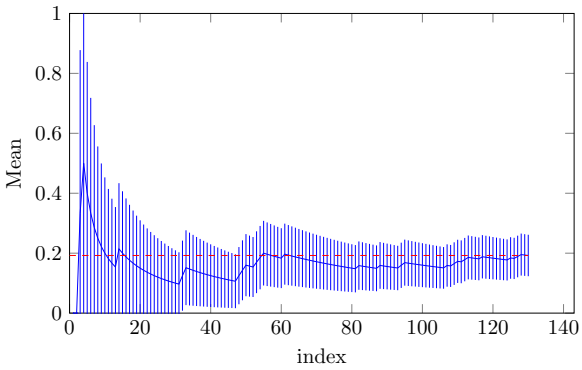


Figure A.15: Cumulative SCR for Brazil

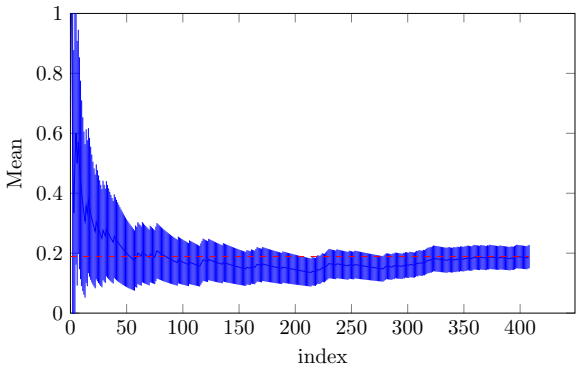


Figure A.16: Cumulative SCR for Canada

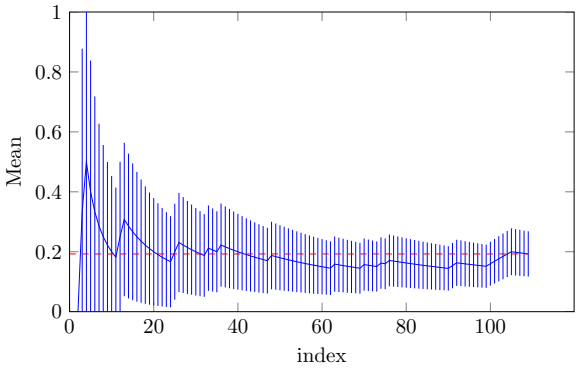


Figure A.17: Cumulative SCR for Spain

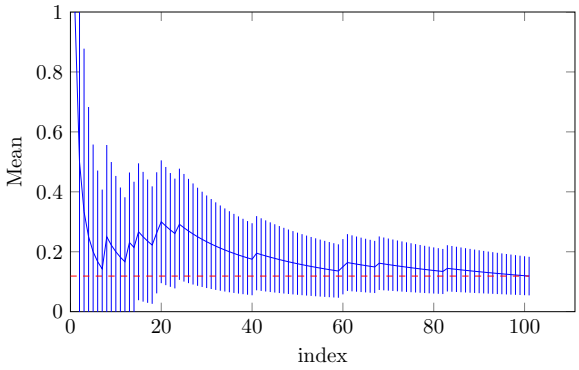


Figure A.18: Cumulative SCR for Finland

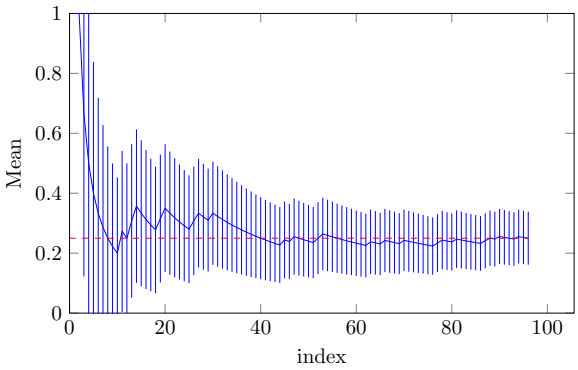


Figure A.19: Cumulative SCR for Romania

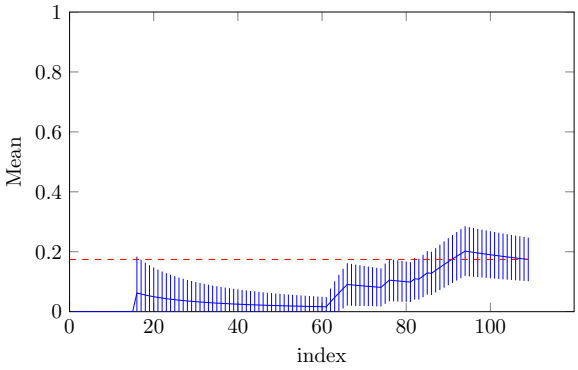


Figure A.20: Cumulative SCR for Egypt

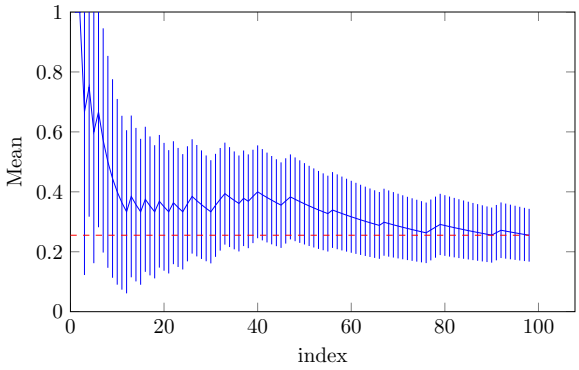


Figure A.21: Cumulative SCR for Turkey

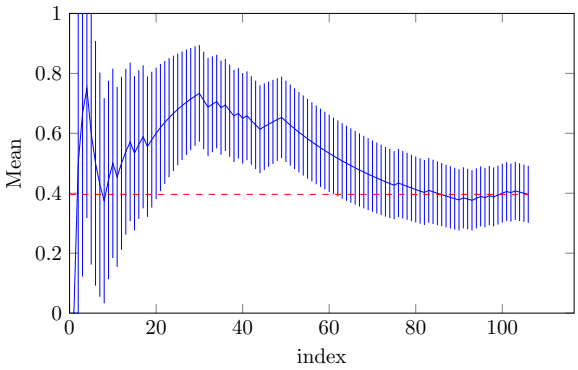


Figure A.22: Cumulative SCR for Japan

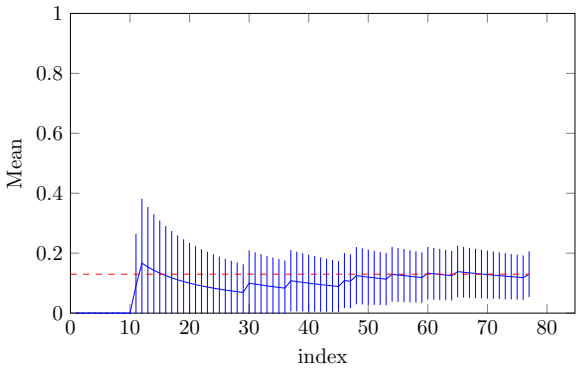


Figure A.23: Cumulative SCR for Denmark

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B:
Spoofing Capability Rate Changes for the Current Year

The graphs below show how the SCRs have changed over the last year. These graphs contain data that span from 26 February 2013 to 26 February 2014. We have included only the countries that have more than 60 data points.

The values on the x-axis show the number of measurements from that specific country, and the y-axis shows what percentage of clients in that particular could send IP packets with arbitrary source IP addresses.

Note that clients that were able to spoof only neighboring addresses are not included in these plots. The values on the y-axis show the mean of all measurements upto and including the value on the x-axis. We have included only the countries from which we had over 50 measurements.

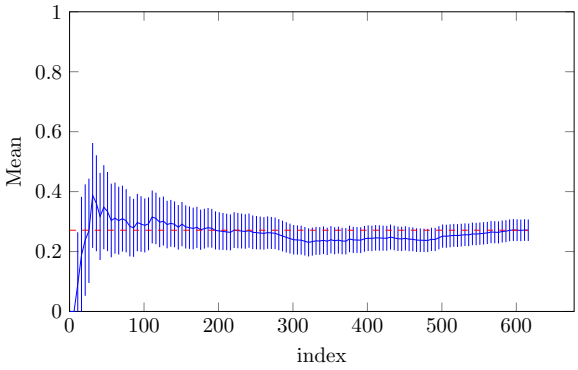


Figure B.1: SCR for the USA

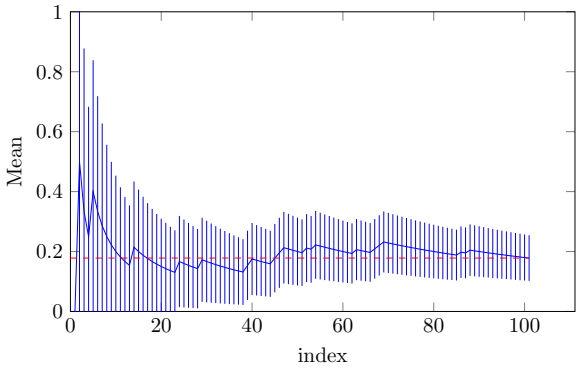


Figure B.2: SCR for Germany

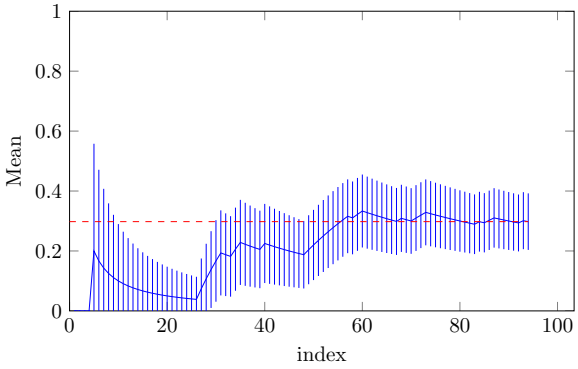


Figure B.3: SCR for Great Britain

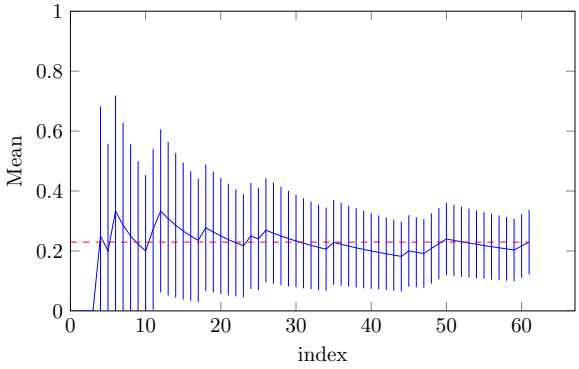


Figure B.4: SCR for Canada

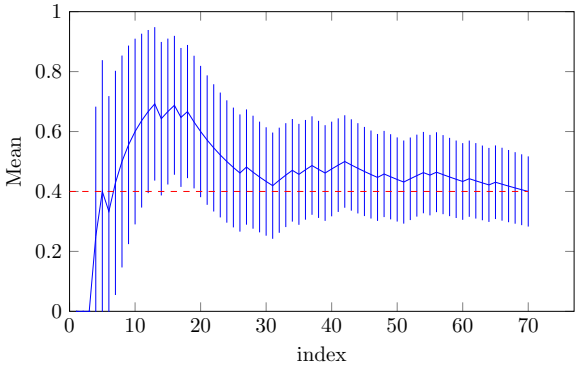


Figure B.5: SCR for The Netherlands

REFERENCES

[1] National Research Council Committee on Research Horizons in Networking, *Looking Over the Fence at Networks: A Neighbor's View of Networking Research*. National Academies Press, 2001.

[2] k. claffy, M. Fomenkov, E. Katz-Bassett, R. Beverly, B. Cox, and M. Luckie, "The Workshop on Active Internet Measurements (AIMS) Report," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 39, no. 5, pp. 32–36, Oct 2009.

[3] k. claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov, "Internet Mapping: from Art to Science," in *IEEE DHS Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*, Watham, MA, Mar 2009, pp. 205–211.

[4] Y. Shavitt and E. Shir, "Dimes: Let the internet measure itself," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 71–74, Oct. 2005. [Online]. Available: http://doi.acm.org/10.1145/1096536.1096546.

[5] Grenouille.com, January 2014. [Online]. Available: http://www.grenouille.com/.

[6] Google.com, "What is Measurement Lab?" January 2014. [Online]. Available: http://www.measurementlab.net/about.

[7] 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*. ACM, 2012, pp. 87–100.

[8] A. Dhamdhere, M. Luckie, B. Huffaker, A. Elmokashfi, E. Aben *et al.*, "Measuring the deployment of ipv6: topology, routing and performance," in *Proceedings of the 2012 ACM conference on Internet measurement conference*. ACM, 2012, pp. 537–550.

[9] R. Beverly and S. Bauer, "The spoofer project: Inferring the extent of source address filtering on the internet," in *Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet (SRUTI) Workshop*, Jul. 2005, pp. 53–59.

[10] R. Beverly, A. Berger, Y. Hyun, and k claffy, "Understanding the efficacy of deployed internet source address validation filtering," in *Proceedings of the Ninth ACM SIGCOMM/USENIX Internet Measurement Conference (IMC)*, Nov. 2009.

[11] Wikipedia.org, "Crowdsourcing - Wikipedia, the free encyclopedia," January 2014. [Online]. Available: http://en.wikipedia.org/wiki/Crowd-sourcing.

[12] amazon.com, "Amazon Mechanical Turk - Welcome," January 2014. [Online]. Available: https://www.mturk.com/mturk/welcome.

[13] 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*, Aug. 2013.

[14] S. L. Lohr, *Sampling: Design and Analysis*. Pacific Grove, California: Brooks/Cole Publishing Company, 1999.

[15] P. S. Alan Buckingham, *The Survey Methods Workbook*. Cambridge, UK: Polity Press Ltd, 2004.

[16] NIST/SEMATECH, "E-handbook of Statistical Methods," January 2014. [Online]. Available: http://www.itl.nist.gov/div898/handbook/.

[17] Amazon.com, "AWS Developer Forums: MTurk CENCUS," January 2014. [Online]. Available: https://forums.aws.amazon.com/thread.jspa?threadID=58891.

[18] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proceedings of the SIGCHI conference on human factors in computing systems*. ACM, 2008, pp. 453–456.

[19] N. Christin, S. Egelman, T. Vidas, and J. Grossklags, "It's all about the benjamins: An empirical study on incentivizing users to ignore security advice," in *Financial Cryptography and Data Security*. Springer, 2012, pp. 16–30.

[20] W. Mason and S. Suri, "Conducting behavioral research on amazon's mechanical turk," *Behavior research methods*, vol. 44, no. 1, pp. 1–23, 2012.

[21] J. Oh and G. Wang, "Evaluating crowdsourcing through amazon mechanical turk as a technique for conducting music perception experiments," *Proceedings of the 12th International Conference on Music Perception and Cognition*, pp. 1–6, 2012.

[22] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk a new source of inexpensive, yet high-quality, data?" *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011.

[23] J. Ross, L. Irani, M. Silberman, A. Zaldivar, and B. Tomlinson, "Who are the crowdworkers?: shifting demographics in mechanical turk," in *CHI'10 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2010, pp. 2863–2872.

[24] E. Cushing, "Amazon Mechanical Turk: The Digital Sweatshop - Science Technology," January 2014. [Online]. Available: http://www.utne.com/science-and-technology/amazon-mechanical-turk-zm0z13jfzlin.aspx.

[25] Amazon.com, "Overview: Lifecycle of a HIT - The Mechanical Turk Blog," January 2014. [Online]. Available: http://mechanicalturk.typepad.com/blog/2011/04/overview-lifecycle-of-a-hit-.html.

[26] Amazon.com, "Amazon Mechanical Turk," January 2014. [Online]. Available: https://www.mturk.com/mturk/help?helpPage=worker.

[27] "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing," May 2000, bCP38.

[28] CMAND, "Spoofer Project: State of IP Spoofing," January 2014. [Online]. Available: http://spoofer.cmand.org/summary.php.

[29] P. Srisuresh and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, Feb. 1996.

[30] "MaxMind - IP Geolocation and Online Fraud Prevention," March 2014. [Online]. Available: http://www.maxmind.com/en/home.

[31] "IPv6 - Google," December 2013. [Online]. Available: http://www.google.com/intl/en/ipv6/statistics.html.

[32] "World IPv6 Launch Anniversary: Measuring Adoption One Year Later - The Akamai Blog," December 2013. [Online]. Available: https://blogs.akamai.com/2013/06/world-ipv6-launch-anniversary-measuring-adoption-one-year-later.html.

[33] "Cisco IPv6 Lab: IPv6 Deployment," December 2013. [Online]. Available: http://6lab.cisco.com/stats/.

[34] "Service Summary | Tour | Requester | Amazon Mechanical Turk," December 2013. [Online]. Available: https://requester.mturk.com/tour.

[35] “Mechanical Turk: The Demographcis,” December 2013. [Online]. Available: http://www.behind-the-enemy-lines.com/2008/03/mechanical-turk-demographics.html.

[36] “The New Demographics of Mechanical Turk,” December 2013. [Online]. Available: http://www.behind-the-enemy-lines.com/2010/03/new-demographics-of-mechanical-turk.html.

[37] “Amazon Mechanical Turk Participation Agreement,” December 2013. [Online]. Available: https://www.mturk.com/mturk/conditionsofuse.

[38] “Amazon mechanical turk,” December 2013. [Online]. Available: https://www.mturk.com/mturk/help?helpPage=policies.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California