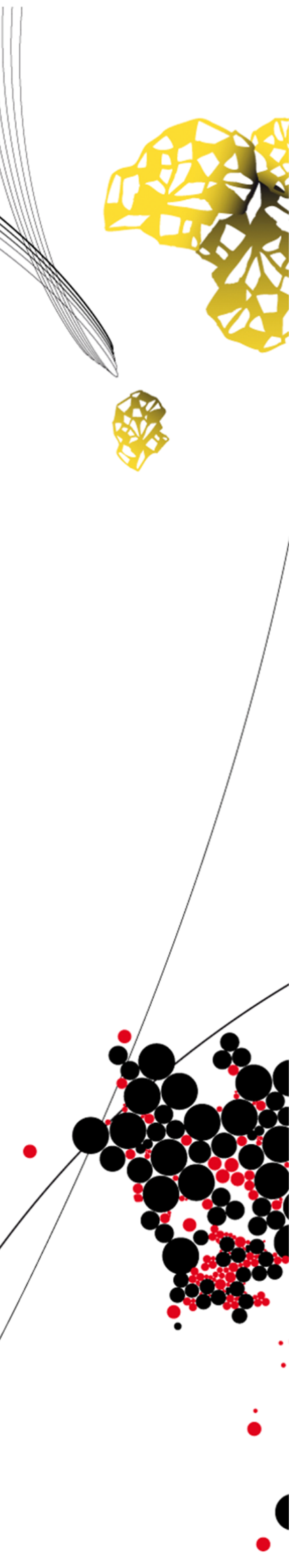


UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

Sustainable Hitlist : Targets for Internet Scans

Danish Suhail Shabeer Ahmed

Master Thesis Report
November 27, 2020

Academic Supervisors:
dr Anna Sperotto
dr Ralph Holz
dr Jasper Gosling

UT Research Chair:
Design and Analysis of Communication Systems (DACS)

UT Address:
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Summary

Internet scanners are a vital tool to study the growth of the Internet and to gain granular insights in network properties such as topology, routing, deployments, and security mechanisms. In the early days of scanning IPv4 address space, a set of IP targets called hitlist was probed. With technological advances resulting in powerful tools, Internet-wide scanning became possible, achieving full scans in as little as 45 minutes. Although this sounds promising, only a few targets respond to a probe request, and hence much overhead traffic is produced. This is amplified by the fact that Internet-wide scans can be carried out with few resources. It is vital to look for an alternative approach that avoids excessive, wasted traffic. It also makes sense to revert the scanning practice of using hitlists in future studies.

This report presents a study on the performance of various statistically generated hitlist that best answers the queries raised by researchers in recent times while conducting Internet-wide scans. A goodness-of-fit test is used to estimate the measure of discrepancy between the sampled hitlist and the Internet. The final results confirm that the stratified-based sampling with a smaller sample size generalizes better than selecting the representatives randomly. The longitudinal study guarantees the stable performance of the stratified hitlist, and a fresh scan is required every 2-3 months to capture the Internet-wide deployments. Once the scanned data is 2-3 months old, only the responsive stable IP hosts feature in the population, and correspondingly the same trait reflects in the hitlist. Dynamic IP allocation and the presence of middleboxes are the main contributing factors that lead to the unavailability of IP hosts.

Contents

List of acronyms

ASN	Autonomous System Number
BGP	Border Gateway Protocol
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
CWMP	CPE WAN Management Protocol
DNS	Domain Name System
ECDF	Empirical Cumulative Distribution Function
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IANA	Internet Assigned Numbers Authority
ICMP	Internet Control Message Protocol
NAT	Network Address Translation
RTT	Round Trip Time
TCP	Transmission Control Protocol
TLS	Transport Layer Security

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

With the evolution of the Internet leading to rapid growth in terms of complexity, size, and importance to the present society, Internet Measurement has become essential to track Internet's continued evolution [1]. Initially, hitlists, a sample set of IP sources, was probed to capture a general overview of the IPv4 address space [2]–[4]. The need to gain a granular understanding of the Internet artifacts and all-the-more technological advancements have motivated researchers to develop powerful tools that execute Internet-wide scans and obtain results almost instantly [5], [6]. With such a promising technology comes the challenge of generating enormous traffic overhead, and this is due to a small amount of response rate. As these scans can be carried out with minimal resources, the overhead created is further magnified, which constitutes an immense Internet background noise. Thus, an alternative approach towards scanning that ensures the mitigation of the excessive traffic generation is highly appreciative. A promising solution is to revert back to the initial scanning practice of using a hitlist for future studies.

The ZMap Project [7] is a parent project which is responsible for a wide range of open-source tools, and these tools are specifically developed to aid experts (security/network) and researchers to perform extensive studies of the hosts and services that constitute the public Internet. ZMap scanner [5] is the very first contribution of this project, and it is an optimized network scanner designed to probe the entire Internet space quickly. General-purpose hardware with few gigabit connections is the minimal requirement for scanning throughout the public IPv4 address space. This tool marked the beginning of developing many other similar open-source tools and libraries for executing large-scale empirical analysis of the Internet end-hosts. With the disposal of these powerful scanners, researchers are enthusiastic to perform regular periodic scans for a longitudinal study to track protocols deployment and trend analysis. Censys.io [8] is a public search engine developed by the same

group of researchers who are responsible for the ZMap scanner. In this project, ZMap is used for scanning and three datasets are made available on a daily basis by probing the entire Internet, including obscure ports. As a result, this project generates 72.2 billion IP-packets every week [9]. On the other hand, many researches initiate their own scans using ZMap, as it is an open-source tool that is designed to offer flexibility such that the scanner could be customized to their project needs [10]. Scans.io [11], a precursor to Censys.io is another public data repository that archives Internet scans. ZoomEye [12] and fofa.so [13] are Chinese based services that are similar to Censys, which scans for services and open ports throughout the Internet and makes data publicly available.

In the event of an Internet-wide scan, a researcher is subjected to the following challenges as mentioned below:

- ZMap gained popularity in the Internet Measurement Community as it prioritized scanning time as its paramount importance [6]. Ever since then, the evolution of the Internet scanners has witnessed that almost every successive release of a scanner manages to outperform its precursor in performances like scanning speed and efficiency by multiplying the number of parallel processing and concurrent probing. Thus, it is evident that scanning properties like time and efficiency have a high correlation to physical resources like bandwidth and Central Processing Unit (CPU). These types of scanning can be resource-draining, resulting in service degradation by flooding the end-hosts' infrastructure [5]. This kind of service degradation threats could be mitigated by strategically probing, and despite such a precautionary measurement, there is still a small probability of such occurrence.

- While performing a longitudinal study in a rapidly growing environment like the Internet, the amount of data generated posts a great challenge to handle and store it. Such big data requires seemingly high computational power to process them and equally requires massive storage space [14].

- The response rate is an important parameter that affects the effectiveness of an Internet-wide scan. The response rate of an Internet-wide scans are generally low. The low IP responsiveness can be due to dynamic addressing or due to the filtering process implemented with middleboxes like firewalls, Network Address Translation (NAT)s, and proxies [4], [15].

Scanning only a sample set of the parent population will serve as a metadata to the complete Internet-wide measurement. This approach will considerably require minimal computational, physical resources, and also minimize the overhead generated due to a low response rate. This observation is a direct manifestation as to

why the Internet Top List is familiar with the research community in recent times. In this thesis, I attempt to identify the best approach to generate a sustainable hitlist that complies with the requirements of the researchers such that it motivates them to utilize our hitlist for their future endeavors.

1.2 Thesis Goal

The goal of this thesis is to develop a tool that generates a hitlist whenever required. This hitlist represents a particular characteristic of all the responsive hosts from the input population in the present real-time scenario. The key traits of the generated hitlist are: to guarantee robustness, remain stable over time, and the methodology used in the hitlist generation needs to transparent, reproducible, and well-documented.

1.3 Research Questions

Based on the problem statement and the goal of this thesis, the following main research question is derived:

How to generate a sustainable hitlist that reputedly represents the general Internet behaviour?

In order to address our main question, we formulate the following sub-questions:

RQ1: What are the current strategies employed in generating a hitlist?

RQ2: What is the best sampling technique used to generate a hitlist? Whether each characteristic of a protocol that needs to be studied demands a different sampling approach?

RQ3: Is the generated hitlist stable and invariant of time?

RQ4: Can Internet centralization aid in influencing the hitlist generation tactics?

1.4 Thesis Contribution

The novelty and contribution of this thesis will be:

C1: Summarize and document the available state-of-the-art hitlist generation techniques.

C2: A command-line tool that generates a hitlist which portrays the characteristics of the responsive parent population.

C3: An analytical methodology that is used to evaluate the hitlist and its results.

1.5 Research Method

The research methodology sketched in this thesis to address each of the research questions are tabulated in Table 1.1. The first sub-question is solved by carrying out a literature survey based on the existing works. The remainder of the sub-questions is addressed by generating a hitlist and evaluating it.

Research Question	***Research Method***
RQ1	Literature Study
RQ2	Design and Evaluation
RQ3	Design and Evaluation
RQ4	Design and Evaluation

Table 1.1: Research methodology to address the defined sub-questions

1.6 Thesis Outline

The remainder of this thesis report is structured as follows: Chapter 2 is dedicated to introduce basic concepts related to this thesis and the following Chapter 3 elaborates the different generation techniques of the hitlists. Chapter 4 documents the methodology and techniques utilized to conduct the experiments. The results are interpreted in Chapter 5 and the last Chapter 6 is dedicated to the conclusion of the report.

Chapter 2

Background

2.1 Motivation

The background section is a refresher that provides a high-level perspective of this research activity and briefs through the basic concepts related to this thesis. This chapter eases the reader to understand and grasp the technical arguments presented in the upcoming sections of this report. Figure 2.1 shows a simplified framework of this thesis.

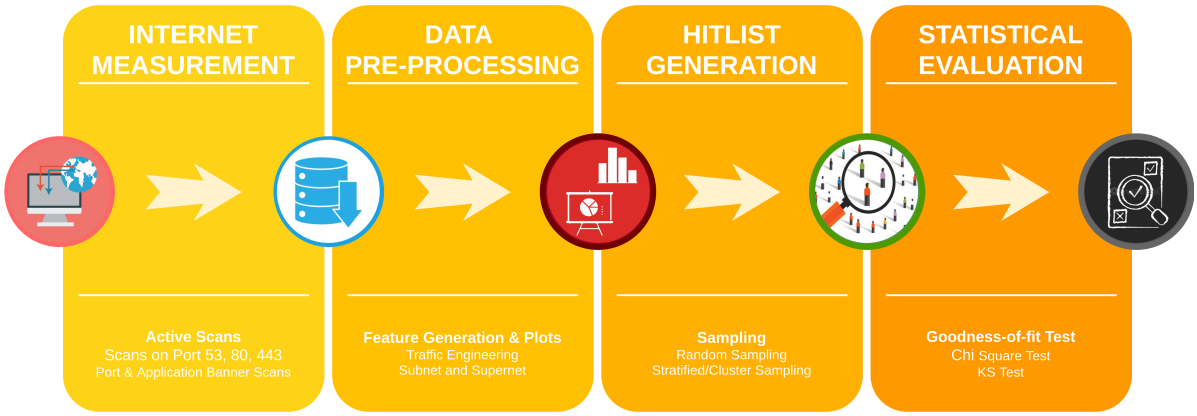


Figure 2.1: High Level Process Flow

2.2 Internet Measurement

Internet Measurement is the process of collecting a broad range of data to study the deployments and characteristics on the Internet. These data provide the impetus to the network architects and experts to ensure operational infrastructures and simultaneously also perform engineering tasks like monitoring, conducting forensic analysis, planning upgradations, and deployments of hosts/services. It is of utmost

importance to assure that the measuring activities do not cause any hindrance to any of the network operations and strictly adhere to all the ethical considerations.

Internet Measurement can be broadly classified into two as follows:

- **Passive Measurement:** This approach simply monitors the target of interest without any discrepancy, as they do not generate any additional traffic. The measurement data is collected just by observing the network activities from a vantage point. This method provides information like operating systems, services, applications, and open ports of the active hosts in the network. It is intrusive to privacy and fails to capture the services that are idle during the period of the scan.

- **Active Measurement:** Active scanning is a measurement technique where the packets are injected into the network and directed to remote targets on the Internet to identify end-hosts with open ports/services on the Internet. It is critical to study deployments, conduct network or application performance tests, and to detect protocol services and its vulnerabilities [16].

2.2.1 Internet-wide Scanners

Internet Scanners are an essential tool to probe and collect data that assists in conducting investigations to understand and categorize the network deployment. It is generally an automated and modularly designed tool, with the fundamental blocks of these tools constituting as follows: *scanner core*, *user interface*, and *output handler*. The scanner core executes the main activities like probing, parsing input and excluding blacklisted targets. The user interface could be a command-line interface or a Graphical User Interface (GUI) that helps to interact with the scanner. The output handler is responsible for directing the scanned result to further processes in the pipeline or store it in the database.

Active scanning is the most popular approach to assess the Internet. Paxson's work [17] was one of the earliest effort to develop a measurement framework that measured the end-to-end behavior over a few numbers of Internet hosts, in order to understand the dynamics of the Internet. His initial efforts strongly influenced later developments to scan the entire Internet. However, the first Internet-wide active probing in modern times (21^{st} Century) was carried out by Heidemann et al. [2]. They performed an Internet Control Message Protocol (ICMP) scan for a period of three months to identify the active IP address throughout the Internet. Durumeric et al. [5] introduced ZMap, an open-source network scanner that is capable of actively probing the entire IPv4 address space within a duration of 45 minutes. It is the first Internet-wide scanner, which led to the explosion of many research groups doing

this now, on a global landscape. Further, Adrian et al. [6] introduced ZGrab, a Go application-layer scanner that complements ZMap to achieve a milestone by reducing the scanning time to 5 minutes. Further, ZMap is extended to support the IPv6 version [10].

The Internet-wide scanners can support both port scanning and an application banner scanning. Port scanning is a technique to probe end-hosts for open ports, whereas application banner scanning is used to obtain application information like application's name and version. A broad range of scanners are available for network measurement today, and the following Internet-wide scanners are well known and popular among researchers and hackers:

- ZMap [5], Masscan [18], Scanrand [19] for port scanning.
- Unicornscan [20] is familiar for application banner scanner and Shodan [21] is a search engine that faciliates with application information.

In spite of these numerous highly efficient Internet-wide active scanners, the concern of low response rate prolongs. Durumeric et al. [5] recommends scanning on default ports of each protocol to increase the response to request ratio. However, the response to an active measurement depends on numerous factors, like dynamic address allocations and the presence of middleboxes [4], [22], [23]. In our work, we propose a hitlist where the number of probing requests is limited and yet revealing the same response to request ratio, which should directly result in minimizing needless traffic overhead.

2.2.2 Target Space

IPv4 address space is really massive with $\sim$4.3 billion addresses, so it is very much essential to determine the target of interest before performing any measurement over the Internet. The process of identifying suitable targets for the study is called

Scanning Scale	***Target Population***
/0 Prefix	$2^{32} \sim$ 4.3 billion
Public IP Space	$\sim$ 3.7 billion
Announced Prefix	$\sim$ 2.8 billion
Sample List	1000 - 1 million

Table 2.1: Estimation on Target Population

Target Selection [24]. In this thesis, we confine our scanning range to only the IPv4 address space, as the infrastructure where our scanner is located does not support IPv6. When considering only IPv4 addresses, the target to scan will generally range as low as few thousands while using a hitlist and the upper bound being $\sim$4.3 billion addresses for the entire IPv4 population. Table 2.1 further elaborates on the target coverage based on the scale at which the scan is performed.

Complete Scan

The address limit of IPv4 is 2^{32}, which is $\sim$4.3 billion addresses. The easiest way to scan the entire Internet is by using the /0 prefix. However, scanning all the combinations of addresses is not an ideal approach because Internet Assigned Numbers Authority (IANA) has explicitly reserved a few of the address blocks for private and special purposes. This privileged address space is almost 13% of the total addresses, and these addresses surely will not respond to any probe request. Table 3.1 tabulates the special address blocks allocated by the global registry bodies. The most convenient and obvious method of scaling down the scanning targets is by only probing the public addresses on the Internet, and this limits the background noise to a certain extent [5], [25]. An Internet-wide scan cannot be prevented completely, but it is advisable only when a precise, accurate, and holistic measurement is required.

Routable Sources

RouteView, an open-project conducted at the University of Oregon, facilitates Internet users with announced Border Gateway Protocol (BGP) prefix information about the inter-domain routing system [26]. Further targets can be scaled down, by probing only the announced IP addresses available in the global BGP routing tables. These addresses would approximately be around 2.8 billion addresses.

Sample List

Sample list[1] is a set of Domain Name System (DNS) domain names or IP addresses/blocks which reputedly represent the global Internet population. There are numerous techniques to create these sample set, and a detailed overview of these lists are presented in Chapter 3. Generally, these lists recommend probing around 1000 - 1 million targets instead of a complete Internet scan.

[1] Also known as Probe list, Seed list

IP Blocks	*Scope*
0.0.0.0/8	Local System
10.0.0.0/8 172.16.0.0/12 192.0.0.0/24 192.168.0.0/16 198.18.0.0/15	Private Addresses
127.0.0.0/8	Loopback
169.254.0.0/16	Link Local Addresses
192.0.2.0/24 198.51.100.0/24 203.0.113.0/24	Documentation Purpose
192.88.99.0/24	IPv6 to IPv4 anycast
224.0.0.0/4	IP Multicast
240.0.0.0/4	Future Use
255.255.255.255/32	Broadcast Address

Table 2.2: Reserved IPv4 Address Blocks

2.3 Data Pre-Processing

Upon receiving the scanned results, it is vital to extract additional information like Autonomous System Number (ASN) and BGP Prefix for every individual IP host as this information will be useful while sampling. A brief overview of the Internet and its functionality will highlight the importance of these attributes.

The Internet can be defined as a group of autonomous systems interconnected with each other. BGP is a commonly used routing protocol that interlinks these autonomous systems. This protocol utilizes information like the ASN and prefix details for the smooth functioning of the Internet by advertising, establishing, and maintaining routes.

Autonomous System Number: A single or a group of networks under the control of an individual authority is called the Autonomous Systems. Each of these systems

is identified with a unique number called the Autonomous System Number. The ASNs are under the control of the IANA and other Regional Registry Bodies.

Prefix: A Prefix is an aggregation or a block of IP addresses. IPv4 addresses are 32-bit numbers represented in dotted decimal notation. The network prefix and the host number is embedded into this 32-bit IPv4 address. The prefix length information differentiates the prefix and host number in the IP address. The prefix length determines the size of the prefix or the number of host addresses in the prefix.

2.3.1 Internet Traffic Engineering

Internet traffic engineering is a practice to mitigate performance issues and optimize the functionality of the IP infrastructures [27]. The main objective is to plan future deployments, traffic flows and utilize network resources efficiently, such that the network is reliable, robust, and meets all the business requirements. One of the common practices in network engineering is subnetting, and this is an approach of breaking down a large network/prefix lengths into smaller contiguous networks or subnets.

Subnetting is quite common due to numerous reasons. For example, an organization allocated with a large address space breaks it down into smaller subnets and assigns every block for a dedicated purpose. Few subnets can further be subdivided for fine-grained control to ensure load balance and optimal routes. These operations are accomplished to achieve higher efficiency, survivability, and modularity on the Internet. Therefore, almost every organization with a registered ASN distributes its allocated space into smaller subnets to meet its operational and contingency requirements. Such activities can influence our measurements and thus needs to be considered while processing the scanned outcome.

2.4 Hitlist Generation

Hitlist generation is the process of shortlisting a subset of representative IP host/prefix, which captures the trends of the entire global deployments. There are numerous approaches to create a hitlist, and a detailed overview is presented in the following Chapter 3. However, probabilistic sampling approaches like random, stratified, and cluster sampling are considered in this research.

2.5 Statistical Evaluation

As the final step of this workflow, the generated hitlist needs to be assessed whether the hitlist represents the parent population or not? When the categorical value has two or more values, the ideal solution to address such a query is by conducting a goodness-of-fit test. We find the relative difference when we observe only the count of a single value.

2.5.1 Goodness-of-fit Test

The goodness-of-fit test is a statistical test that reveals whether a sampled subset matches with the distribution/trend revealed from the parent population. To further simplify, this test determines the measure of discrepancy between the hitlist and the output from the Internet scanner. There are numerous techniques available to conduct a goodness-of-fit test, but only Pearson's chi-squared test and Kolmogorov–Smirnov test are considered in this thesis. It is because these two tests are familiar approaches to estimate the equality of distribution in Internet Measurement studies. It is important to note that we require an evaluation metric that could be utilized to compare the performance of different sample sizes and different categorical values. A sound procedure to conduct a goodness-of-fit test is as follows:

1. Identify the hypothesis question.
2. Sketch an analysis plan.
3. Execute the plan on the sampled subset.
4. Interpret the obtained result.

We considered the following tests and their evaluation metric in this study. Average Normalized Deviation approach and Nominal Association using Phi Coefficient are the two shortlisted metrics that were independent of both the sample size and categorical values[2]. We performed an empirical study between these two tests and chose the Average Normalized Deviation method because its results were easier to interpret and make comparisons. In Sub-section 4.2.5, the reason for selecting this metric is discussed in detail.

Pearson's chi-squared goodness-of-fit

Pearson's chi-squared goodness-of-fit test is a statistical hypothesis testing technique that applies to categorical data to estimate any difference between the ob-

[2]Both these metrics are inspired and modified version of Phi Chi-square Test.

served and the expected event. This test is concerned with the frequency distribution of categorical events.

The most common approach to finding the discrepancy value among categorical values is by using χ^2 value from the chi-square test [28]. The χ^2 value is derived using the count value observed in the population data (O) and expected count from the sample set (E) for each bin [3] in the data:

$$\chi^2 = \sum_{i=1}^{B} \frac{(O_i - E_i)^2}{E_i}, \text{ where B} = \text{No. of Bins} \tag{2.1}$$

The p-value is determined from the χ^2 table and the hypothesis is answered based on the p-value. The hypothesis is rejected when the p-value is less than or equal to the significance level (α). Generally the significance level is considered to be 0.05 (95% confidence interval).

Assumptions: This test is considered only when the following assumptions are satisfied:

1. Observation values need to be independent of each other[4].
2. Data should be a categorical variable.
3. The expected count value should be more than 5.

Kolmogorov–Smirnov test

The Kolmogorov-Smirnov test is a non-parametric test that aids in concluding if a sample derived from the population expresses a similar distribution. The principle idea of the KS test is that when two data are identical, their Empirical Cumulative Distribution Function (ECDF) must be quite similar. Thus, it compares the cumulative distributions of the two data and derives a test statistic value. The test statistic value (D) is the greatest vertical distance between the two Cumulative Distribution Function (CDF) curves. Based on this value and p-value, the results can be obtained. The test statistic is defined by:

$$D = Max|F_0(x) - F_{data}(x)| \tag{2.2}$$

where,

[3] Bins are the possible values that can feature in a categorical variable/interval.

[4] This assumption is almost certainly not the case in Internet Measurement due to the existence of firewall. However, we are motivated to assume the data independent because the firewalls are independently controlled, designed, and managed by every ASNs. [29]

- $F_0(x) =$ the cdf of the population data,
- $F_{data}(x) =$ the sample distribution.

The p-value is determined from the KS table and the hypothesis is rejected if the p-value is less than or equal to the significance level (α).

Nominal Association using Phi Coefficient

Phi coefficient (φ) is another non-parametric statistical test that provides the degree of correlation between two distributions. This test is also called the mean square contingency coefficient. This technique is similar to Pearson's Chi-square test, with the only difference being that its result is independent of sample size. Phi (φ) value can be easily be estimated using the chi-square (χ^2) value as follows:

$$\varphi = \sqrt{\frac{\chi^2}{N}}, \text{ where N = Sample Size} \tag{2.3}$$

While evaluating φ using chi value, it is also important to note that Yate's correction [5] (i.e., expected count should be more than 5) is not applicable. Phi value ranges between the interval (+1, -1), where +1 denotes a positive association and a negative value indicating a negative correlation between the two distributions. Because of its range, interpreting its results are often challenging (Especially while comparing two phi values).

Average Normalized Deviation

The average normalized deviation is another discrepancy metric that is determined using the χ^2 value. This value estimates the measure of discrepancy between the two distribution and remaining independent of both the sample size and the number of values in a categorical variable.

$$\lambda_{avg} = \sqrt{\frac{1}{B}\sum_{i=1}^{B}\frac{(O_i - E_i)^2}{E_i^2}} \tag{2.4}$$

The mathematical difference between the χ^2 value and λ_{avg} estimations are listed as follows:

- The expected count in the denominator of the χ^2 formula is squared to ensure that the discrepancy measure is invariant to different sample sizes.

[5] Yate's correction eliminates the error introduced by approximating the discrete probabilities of frequencies as a continuous distribution

- The modified χ^2 value from the previous step is divided by the number of categorical values (B), and finally taking the square root. These additional mathematical operations provide an average discrepancy value, which is independent of the number of categorical values.

In general, the λ_{avg} value ranges between the interval (0, ∞). A small value indicates that the deviation between the observed and the expected distribution is small. Similarly, a higher value indicates that there is a significant discrepancy. Table 2.3 is a reference list that provides a rough percentage estimate for interpreting the average normalized deviation value.

Percentage estimate (%)	***Average Normalized Deviation (λ_{avg})***
1%	0.0144
2%	0.0295
5%	0.0699
7.5%	0.1057
10%	0.1448

Table 2.3: Percentage Estimate of the Average Normalized Deviation

2.5.2 Relative Difference

Relative difference (C) estimates the difference or change in the count of a particular value over time. For example, this metric estimates the difference between the number of counts for a particular bin value over time.

$C = \frac{x_1 - x_2}{x_1}$, where x_1 = initial value, x_2 = new value

Chapter 3

Related Work

3.1 Motivation

The first step towards addressing our research question is by conducting a literature survey and understand the different strategies embraced in recent times to create a sample list. This chapter provides the initial motivation and learning from previous related work that foster in developing a robust hitlist with minimized limitations.

The following Section 3.2 discusses the different types of sample lists and their generation techniques. Finally, this chapter is concluded with a short discussion in Section 3.3.

3.2 State-of-the-art Sample Lists

Hitlist is a subset of IP addresses/blocks that represents the Internet. This category of lists were used in the early days of the Internet era to capture a generic view of the Internet. Whereas, the top list is a sample of DNS domain names that are visited frequently. These lists are prevalent in recent times and feature more often in various academic research works due to the following reasons:

1. To obtain extended visibility. Complement a large-scale scan by probing a small set of targets multiple times.

2. Minimize the traffic overhead caused due to the low response rate.

3. Reduce the load in data analytics and storage process.

The upcoming two sub-sections briefs through the related work carried out on these two diverse types of sample lists.

3.2.1 Top List

A sample of popular DNS domain names is probed for scientific purposes to conduct a thorough analysis, trace the adoption of a new protocol or security mechanism on the real domains. These list can only be used in domain-based scans. There are numerous top lists available online, for example Alexa [30], Cisco Umbrella [31], Majestic [32], Quantcast [33], Statvoo [34], Chrome UX report [35], and SimilarWeb [36] top list. However, Alexa 1M top list is quite popular as it is used in most of the researches conducted in the Internet Measurement community [37]. The generation method for each of the aforementioned top lists is unique, and thus the top lists are diverse with very minimal overlap. For instance, the Alexa top list represents top domain choices among the netizens. It is achieved by a proprietary methodology that considers the site's estimated workload and visitor engagement monitored by the Alexa browser web plugin. Cisco Umbrella is a list of domains observed by their products like Cisco's OpenDNS service, Phishtank, DNSStream, BGPStream, DNSCrypt, and several other data sources. Majestic 1M top list, on the other hand, generates its list using a custom web crawler and sorts sites based on the number of /24 IPv4 blocks linked to it.

The use of the top list is an effective way to minimize the needless traffic generated while scanning the Internet, but these top lists are unstable with a 50% churn rate per day, and most importantly, the research results could be biased based on the time the scan is conducted [37]. As most of the top lists lack transparency on their generation technique, barely any scientific paper could justify the reason for selecting one of these top list [37]. Pochat et al. [38] identified the possible hidden properties and biases of the four most popular top lists (Alexa, Cisco Umbrella, Majestic, and Quantcast) that could skew the research results and combined these four top lists. Further, they filtered the undesirable domains from the aggregated top list and this aggregated list is called TRANCO. Despite this new top list exhibiting higher stability, the reason for selecting only these four popular top lists is again not justified.

3.2.2 Hitlist

Several scientific studies used different ideas to extrapolate the available data to generate the hitlist, and Table 3.1 tabulates these works. These generation techniques are broadly classified into four different approaches based on the following shortlisting procedure:

Random Selection: In this method, the hitlist is generated by randomly or pseudo-randomly picking the hosts from the probed output. The motivation of this technique

is to obtain a general view of the Internet by scanning on a particular open port or service. The distinctive property of this technique is that there is no sample bias as the probability of selecting a representative host is identical. Alt et al. work [39] developed a tool called degreaser, a fingerprinting tool to detect honeypots remotely. In their work, they pseudo-randomly probe to spot honeypots on the Internet. The scanning process ensured that at least one host in all of the 14.5M routed /24 subnets is present, and their hitlist contained 20 million IP hosts.

Study	***Technique***	***Scan Type***	***Request***	***Start***	***End***	***Interval***	***Size***
Alt et al. 2014 [39]	Random Selection	Active Degreaser	TCP	1st May 2014	31st May 2014	1 month	20 million
Cai & Heidemann 2011 [3]	Prioritize by weight	Active	ICMP	June 2006	February 2010	56 months	24,000 /24 blocks
Fan & Heidemann 2010 [4]	Prioritize by weight	Active	ICMP	March 2006	March 2010	48 months	1.5-13 million
Klick et al. 2016 [9]	Prioritize by weight	Active Censys.io	TCP	September 2015	March 2016	6 months	8-20 million
Heidemann et al. 2008 [2]	Hybrid	Active	ICMP	June 2003	August 2008	62 months	24,000 /24 blocks

Table 3.1: Different Techniques on Hitlist Generation

Prioritizing based on weights: This approach attempts to optimize the hitlist performance by introducing biases and this technique demands numerous Internet-wide scans performed over a period of time. Each host is assigned a value based on their response rate during the observation period. The hosts with a higher value are prioritized and more likely to feature in the hitlist. This technique is usually preferred when the hitlist needs to have a very high response rate. Cai and Heidemann [3] statistically investigated the responsiveness of consistent blocks. Data was collected by generating ICMP request to 1% of the allocated Internet address space throughout the week in a time interval of approximately 11 minutes. A hitlist comprising of 24,000 hosts from /24 blocks is generated. These representatives are selected based on their responsiveness evaluated using the earlier scanned results. They found that almost 40% of the /24 blocks to be allocated dynamically, and one-

fifth of the /24 blocks were underutilized (less than 10%). This research study is a good example of obtaining extended visibility using a hitlist.

Klick et al. [9] presented a topology-aware and IP prefix-based scanning strategy that achieved to develop a stable hitlist, at the cost of losing a small amount of accuracy. They narrowed their focus to four protocols, namely File Transfer Protocol (FTP), Hypertext Transfer Protocol (HTTP), Hypertext Transfer Protocol Secure (HTTPS), and CPE WAN Management Protocol (CWMP). Internet-wide scans were periodically conducted for almost six months from September 2015 to March 2016. They performed a detailed analysis and eliminated the prefixes with hosts that are of least interest. In other words, prioritization was based on the density of the responsive prefixes. By doing so, they reduce 25-90% overhead and miss only 1-10% of the hosts of actual interest. In all scenarios, they ensure that the hitlist retains the 80% of the prefixes with the highest number of active hosts.

Hybrid: This technique is a combination of the earlier two approaches where the hitlist is partially selected randomly, and the remainder of the hitlist is selected based on prioritization. This method has been used to complement a global scan with multiple smaller scans such that it extends visibility. One of the earliest work was of Heidemann et al. [2], they attempted to gain extended visibility of the Internet by performing an extensive scan called a census, and further complementing it by executing multiple scans on a smaller scale called a survey. By doing so, they were able to estimate that there is a growth of 4% per year in IPv4 allocated address from 2004 to 2008. They probed the entire Internet 16 times with ICMP requests and once with Transmission Control Protocol (TCP) requests from June 2003 until May 2007, to capture an exact snapshot of the Internet. Though TCP based scanning is more accurate than ICMP, they persisted with ICMP because they experienced the TCP scans to elicit more abuse complaints. Further, they scanned a subset of IP prefixes from March 2006 until August 2008, to complement their census results. The survey hitlist was a combination of the following: (a) 50% prefixes was a random selection from the latest census data available, (b) 25% prefixes were randomly selected, which was active in one of the previous census results, (c) 25% prefixes selection was based on the priorities/weights.

Fan and Heidemann [4] work was one of the very first research whose primary goal is to create a hitlist. They automated the process of hitlist generation by utilizing the previously scanned census results. They probed the entire IPv4 address space using ICMP requests at regular intervals and filtered the most active addresses. The process ensured that the hitlist representatives were responsive, complete, and stable. By selecting a representative from each /24 block, it achieved completeness. Stability is assured by setting a threshold, and a representative is changed in a /24

block only when the change brings about significant improvement. Based on the past census records, the IP addresses that are most likely to be responsive in the future were selected. The generation technique used a random prediction method for prefixes, which were active in one of the previous scans, and also chose an IP address with the last octet as .1 for prefixes, which was never active throughout their study. They concluded by stating that only one-third of the Internet allows informed selection, and 50-60% of representatives responded three months later, which is probably due to dynamic IP addresses. Despite scanning numerous times throughout the globe, their hitlist remained stable only for a couple of months.

Utilizing Top list: The motivation of this technique is to develop a hitlist that is more stable than its source (top lists like Alexa).This approach is similar to a DNS lookup process, where a hitlist is developed by extracting IP address information from the top list domains. Naab et al. [24] developed a prefix based hitlist by using a DNS top list as it's base. All the domains available in the top lists were mapped to their subsequent IP addresses, and in turn, converted the IP addresses to their respective prefixes. Further, they applied zipf distribution to the domains, assigned weights to each of its IP addresses, and prefix, to rank the prefix based on their cumulative weightage score. Prefixes with minimal change of weights over a period are shortlisted to form the hitlist. This hitlist is more stable than a top list and better classifies the deployments.

3.3 Discussion

From the earlier sections, it is evident that there is a steep development in the evolution of Internet-wide scanners, which attracts research groups to conduct more scientific studies towards the mechanism of the Internet. As a consequence, a considerable amount of traffic overhead is produced due to the low response rate. A suitable approach to minimize this effect of overhead is by leveraging every single measurement data to its maximum potential. The use of a sample list is a customary solution that assists in extracting valuable information from each measurement data to a greater extent and minimize this adverse effect.

The main objectives while preparing a sample list are enumerated as follows:

1. The sampled subset should be able to generalize the complete Internet deployment.

2. It should manage to express the relationship between the probed measurement result and the changes observed in the Internet deployment currently.

A sample subset is qualified to be representative only when both of these principles are accomplished. From the related work section, it is clear that the use of top lists has been prevalent in the Internet Measurement community. However, their generation techniques are proprietary and thus lack transparency. Alexa top lists is highly unstable, as 50% of the sites are replaced by a new one daily [37]. Efforts have been made to develop a top list by aggregating the four most popular top lists (Alexa, Cisco Umbrella, Majestic, and Quantcast) and filter out the undesirable sites to achieve a robust, resilient, and reliable top list. Nonetheless, it is not preferable to utilize these proprietary third-party top lists as input because it introduces dependencies, and it is not possible to guarantee stability always. For example, Alexa changed their domain ranking algorithm in 2018 [37], and such an occurrence will directly influence the behavior of the sample list. These reasons are also applicable when the hitlist is generated by using top lists as its source. Thus, we aim at developing hitlists that do not use DNS resolution, and we do not want to consider rankings by proprietary vendors.

While generating a hitlist based on prioritization, it helps in achieving a high response rate. But the bias introduced in this technique fails to generalize the overall Internet behavior as it fails to estimate the change between the scanned measurement and the present Internet occurrence. A hybrid approach requires probing the complete IPv4 address space multiple times to generate a single hitlist (minimum four to five times). The time interval between these scans is a sensitive choice. When these intervals are short, the weights assigned are ineffective as most of the hosts have similar values, which is equivalent to random sampling. When the time intervals are large, the effect of prioritization nullifies the contribution of random selection and leads to inaccurate generalization.

Based on the previous work, the random selection process is the most suitable and benign approach to accomplish the desired objectives as it generalizes comparatively better than the remaining techniques. A random sample is generally an approximation of the entire population because there is an equal chance of selecting each host. However, a random sample is less effective when the hitlist needs to express a particular characteristic of the IP host. In such a scenario, statistical sampling methods like stratified sampling or cluster-based sampling is highly effective [40]. Thus, our hypothesis in this study is that sampling techniques to generate hitlist vary based on the requirements or the characteristics that need to be studied.

Chapter 4

Methodology

4.1 Motivation

This chapter documents the methods executed to accomplish the research goal of generating a sustainable hitlist. The primary motivation of this chapter is to provide the reader with a transparent hitlist generation technique and a clear idea of how the hitlist has been evaluated. The remainder of this chapter presents with a short summarization of the methodology workflow, followed by illustrating the objectives of the hitlist respectively. Further, the next three sections elaborate on the data collection, pre-processing, and sampling procedure. Finally, the evaluation process on the sampled dataset is defined to conclude on which sampling technique best represents the entire population for each of the individual protocols in the wild.

4.2 Research Roadmap

Figure 4.1 provides an elaborative and self-explanatory flow-chart of this research methodology. This flow-chart can be split into six different steps in the pipeline as follows:

1. Determine the Target of Interest
2. Data Collection
3. Data Pre-processing
4. Sampling Strategies
5. Statistical Evaluation
6. Interpreting Results and making Recommendations

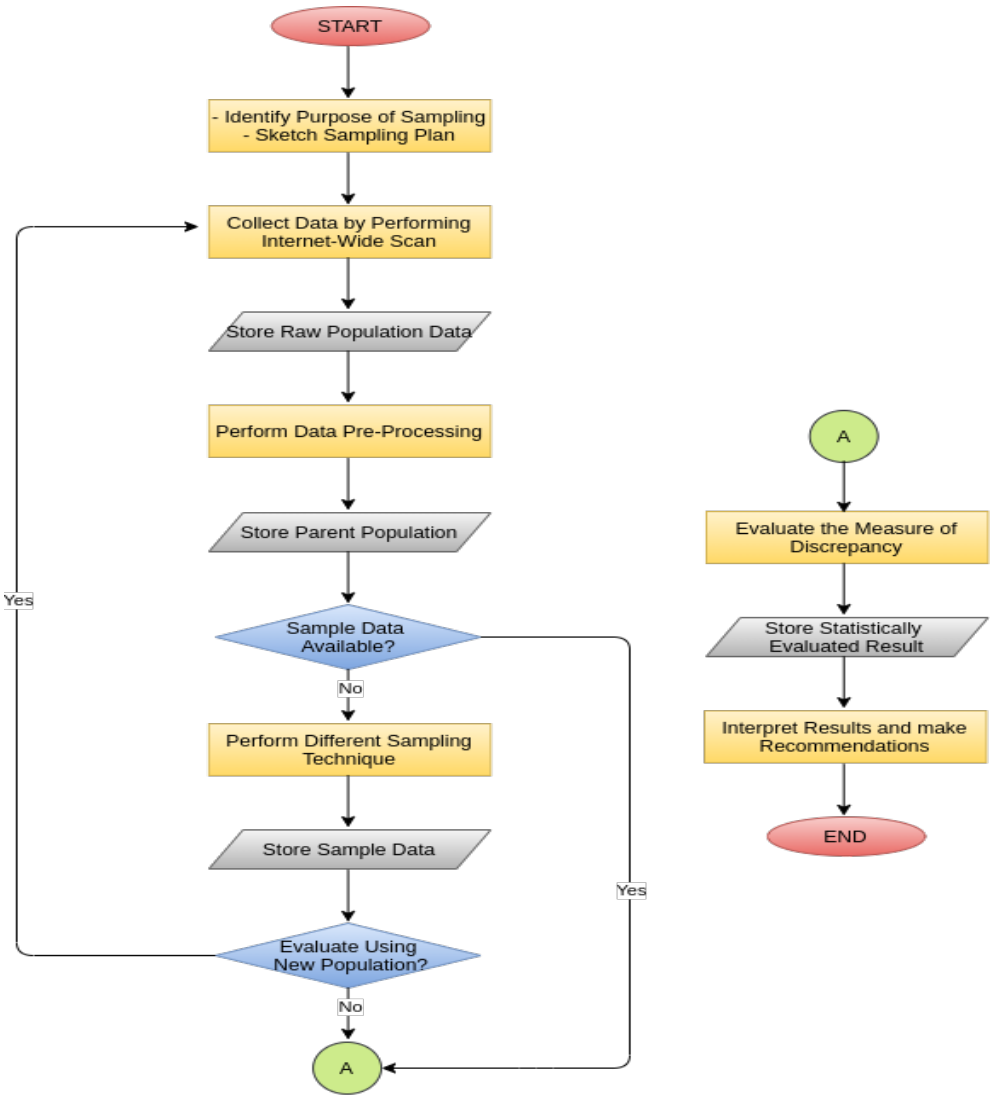


Figure 4.1: Methodology Flow Chart

4.2.1 Determine the Metrics of Interest

To implement a robust sampling process, it is crucial to define a set of objectives as follows:

- The characteristics that the hitlist needs to express,
- The type of information required for sampling, and
- The desired degree of accuracy[1].

[1] In the remainder of the document, accuracy refers to the difference between the average normalized deviation of the hitlist and parent population. Generally, the difference value needs to be zero for 100% accuracy.

Hitlist's Objective: The key takeaway from our literature survey is that random sampling is an ideal approach to prepare a generalized hitlist. Prior to this research, all the other studies dealing with hitlist worked towards the optimization of hitlist's properties like responsiveness and stability. Therefore, this observation of random sampling is presumed to be legit. However, our research is the very first attempt that aims towards generating a hitlist that could replicate the characteristics of all the responsive hosts from the parent population. Thus, our hypothesis is to perform stratified-based sampling using priori information and achieve better accuracy in providing granular-details like the hosts' characteristics.

A short survey has been conducted to identify what are the IP hosts' characteristics that are of interest to the researchers in recent times. Papers of an Internet Measurement conference (i.e., ACM IMC) from 2018-2019 were studied to understand the main characteristics that researchers analyze in their respective work [41]–[50]. Based on the survey conducted, the following aspects of IP hosts are frequently studied:

1. The deployment trends of a particular protocol and its different version supports.
2. The routing characteristics of the end-hosts and their behavior across different prefix lengths.
3. Interdependencies between protocols (i.e., figure out the likelihood of the host that supports a particular protocol also responds to another protocol request.)

Based on the information gathered, the following questions that our hitlist needs to exhibit were framed and enumerated as follows:

Q1: Which sampling technique provides a hitlist that replicates the responsiveness and protocol version trends of the parent population?

Q2: Which sampling approach facilitates a hitlist that captures the parent population's responsiveness and its global distribution of the protocol deployment over the prefix lengths?

Q3: Which sampling approach exactly reproduces the cross-protocol responsiveness?

Priori Information for Sampling: Apart from random sampling, the other two sampling techniques which are considered in this experiment are stratified and cluster-based sampling. These two approaches are executed by distinguishing the entire population based on a particular feature as desired. The features, like Autonomous

System Numbers, allocated BGP prefix, and BGP prefix lengths are utilized to sample and enhance the accuracy of the hitlist.

Error Tolerance: This particular requirement in specific has a very major impact on the recommendations in terms of sample size, and to some extent, the sampling technique used to prepare the hitlist as well. Generally, as the sample size increases, the margin of error reduces. To offer flexibility, we choose the following error margins, and recommendations are made on sampling techniques and size accordingly: 1%, 2%, 5%.

4.2.2 Data Collection

Data collection is the process of gathering relevant information about the end-hosts. This data provides a complete snapshot of the ecosystem and serves as an input to perform the sampling process. This collected data is called as the raw population data. All the scans were probed from a vantage point located in Australia.

Data is gathered by actively scanning to discover a respective open port of interest for all the IPv4 addresses among the announced prefix space. The response to such a probe request facilitates with the information of a particular protocol's deployment, trends, and support. TCP Half-Open scan otherwise, referred to as an SYN scan, is the technique adopted for active scanning. TCP based active scanning has been opted because of its accurate measurement and the flexibility it offers. In this approach, only the first half of a three-way handshake is performed. This type of scanning is most commonly preferred in an Internet-wide scan as it is fast and leaves no record in any of the target's regular system logs [51].

The scope of this thesis is limited to TLS, HTTP, and DNS protocol for brevity and meticulous purposes. However, the combination of these three protocols is considered due to the correlation between them [42] and is also comparatively prone to fewer complaints/abuse emails from providers than other protocols. Zmap scanner introduced by the University of Michigan is utilized to probe the announced prefix addresses. Goscanner introduced in Amann et al. [52], an implementation of Go is used to obtain application-level information[2]. These scanners find themselves located at a research university in Australia. The scans are performed approximately 5-6 times per protocol throughout the thesis, and the relevant scanning information like the epoch is featured in the Annexure B.

[2]We grab application information for TLS protocol only.

4.2.3 Data Pre-processing

Data pre-processing is the step that converts raw data into a useful and coherent information. The raw population data obtained from the scanner as output generally comprises of the following information, namely sync time, Round Trip Time (RTT), and five-tuple (which includes a source IP address/port number, destination IP address/port number, and the protocol). The targets' IP address form the source for data preparation and analysis. A lookup tool extracts relevant information of the IPv4 addresses like it's ASN and BGP prefix. This information is explicitly obtained to assist in the sampling procedure and to achieve better accuracy. The final output after this process is called the parent population. The following processes are necessary to obtain the desired attributes which could assist in the sampling process:

Lookup Tool: `pyasn` is the chosen lookup tool that provides ASN and BGP prefix information for IPv4 addresses. `pyasn`, a Python extension module that provides offline and historical lookups based on the Routing Information Base (RIB) BGP archive. The corresponding RIB at the time of the scan is explicitly downloaded and used to obtain the necessary information.

Data Cleaning: Occasionally few IP address details (less than 6K IP addresses per scan, i.e., $\sim$0.01% of the parent population) are not available in the RIB. Thus, these addresses are discarded as part of the data cleaning process.

Supernetting: Unlike ASN, the network traffic engineering process can result in manipulating the BGP prefix information. `pyasn` provides the advertised BGP prefix information of the IP address. Zhu et al. [53] state that the use of advertised subnet prefix for measurement studies is not recommended and can lead to compromised results. The advertised prefixes are strategically planned and are subjected to modification as per the organizational needs, like setting an upper bound to the routing table, announcing multiple blocks to protect themselves from route hijacking. Thus, in this research, all the neighboring prefixes of a particular autonomous system (AS) are grouped to obtain the allocated prefix information. This allocated prefix holds more generalized information over the advertised prefixes and yields an unbiased and robust result. This process of grouping the sequentially neighboring prefixes is called supernetting.

4.2.4 Sampling Strategies

Sampling is an essential part in this entire pipeline as it is the process of choosing a subset of IP hosts that provisions a particular characteristic of the parent population. Sampling techniques are broadly classified either as probabilistic or non-probabilistic sampling. Probabilistic sampling is well-suited for researches that aim at gaining insights about the parent population. For the same reason, our work focuses on probabilistic sampling techniques.

The sample size is the process of determining the total number of hosts required to represent the parent population. It directly influences the precision and the conclusions drawn from the study. In our study, we consider the following sizes: 1.5k, 10k, 100k, 1M. These sample sizes are explicitly selected as these size ranges are common in recent research works dealing with the top list.

The sole intent of this selection process is to ensure that the generated sample list represents the population dataset. The parent population data is a large file ($\sim$25GB), so the sampling process needs to be tactfully carried out to avoid memory overflow error. All the sampling techniques must select the end-hosts stochastically to ensure that the prescribed hitlist does not result in affecting the service of any specific end-host. A random seed value is set and recorded for every experiment to facilitate reproducibility. Following are the three sampling methods considered in this work to generate a hitlist:

Simple Random Sampling

This sampling approach is a process of randomly or pseudo-randomly selecting 'n' number of hosts from the population data. The selection process is independent and requires only the parent population and sample size as input. The likelihood of selection is uniform among all the hosts and nullifies predictability even while sampling a geometrically distributed or exponentially distributed data [54].

Stratified Random Sampling

In this sampling technique, all the hosts are divided into multiple groups based on a specific attribute. An exact proportion is randomly selected from each of the strata to form a sample set. The core idea behind this sampling approach is to utilize a priori information while sampling, and the chosen feature needs to have a high correlation to the characteristic that needs to be revealed by the sample.

In this study, the features chosen for stratification are protocol version, ASN, prefix-length, and combination of ASN with prefix-length information. This technique is more effective with heterogeneous population data exhibiting linear trends as it

Algorithm 1: Stratified Random Sampling

function StratifiedRandomSample $(df, k, a, seed)$;
Input : Population data df, sample size k, attribute a and random seed $seed$
Output: $sample\_data$
$n \leftarrow$ No. of IP hosts in df;
$df \leftarrow$ sort df based on a in ascending order;
group_dict $\leftarrow$ df.groupby(a);
dict[freq] $\leftarrow$ group_dict.count();
dict[cum] $\leftarrow$ dict[freq].cumulative_sum();
dict[low_limit] $\leftarrow$ dict[cum] - dict[freq];
dict[up_limit] $\leftarrow$ dict[cum] - 1;
dict $\leftarrow$ sort dict based on freq in ascending order;
actual_size $\leftarrow$ 0 ;
while k *is not equal to actual_size* **do**
    portion $\leftarrow$ (dict[freq] / n) * k + 0.5;
    $seed \leftarrow$ Set random seed;
    rand[] $\leftarrow$ random_gen_without_replacing((dict[low_limit], dict[up_limit]), portion);
    actual_size $\leftarrow$ actual_size + portion ;
rand[] $\leftarrow$ sort rand[] values in descending order;
while *rand[] is not empty* **do**
    x $\leftarrow$ rand[].pop();
    $sample\_data \leftarrow sample\_data$ + df.pop(x);

has better precision over random sampling [54]. The implementation of the stratified sampling method is explained by using the pseudocode in Algorithm 1.

Cluster Based Random Sampling

In this work, cluster-based random sampling is similar to stratified random sampling. However, the only difference is that the clusters are randomly selected, and there is no priority given to the highly-dense strata/cluster group. Algorithm 2 explains the implementation of this sampling technique. Attributes like ASN, allocated BGP prefix, prefix-length, and combination of ASN with prefix-length are used to perform cluster-based sampling.

As a special-case of cluster-based sampling, a single IP host is picked from each cluster to form the hitlist. This approach is a motivation from Fan et al. work [4], where they generate a hitlist by stochastically selecting at least one IP representative

Algorithm 2: Cluster Based Random Sampling

function ClusterRandomSample $(df, k, a, seed)$;
Input : Population data df, sample size k, attribute a and random seed $seed$
Output: $sample\_data$
$n \leftarrow$ No. of IP hosts in df;
$df \leftarrow$ sort df based on a in ascending order;
group_dict $\leftarrow$ df.groupby(a);
dict[freq] $\leftarrow$ group_dict.count();
dict[cum] $\leftarrow$ dict[freq].cumulative_sum();
dict[low_limit] $\leftarrow$ dict[cum] - dict[freq];
dict[up_limit] $\leftarrow$ dict[cum] - 1;
$seed \leftarrow$ Set random seed;
dict $\leftarrow$ shuffle dict's row randomly;
actual_size $\leftarrow$ 0 ;
while k *is not equal to actual_size* **do**
    portion $\leftarrow$ (dict[freq] / n) * k + 0.5;
    $seed \leftarrow$ Set random seed;
    rand[] $\leftarrow$ random_gen_without_replacing((dict[low_limit], dict[up_limit]), portion);
    actual_size $\leftarrow$ actual_size + portion ;
rand[] $\leftarrow$ sort rand[] values in descending order;
while *rand[] is not empty* **do**
    x $\leftarrow$ rand[].pop();
    $sample\_data \leftarrow sample\_data$ + df.pop(x);

from each BGP prefix. This sampling technique is commonly used in many works to develop a complete hitlist covering representatives from all the BGP prefixes. We want to study the performance of this technique as it is a widely used method. This special-case sampling process is referred to as per-prefix sampling in the remainder of this document.

4.2.5 Statistical Evaluation

The next step after the generation of the hitlist is to statistically evaluate and estimate whether the sampled hitlist replicates the parent population and its changes in the current Internet deployment. The evaluation and its metric are entirely dependent on the characteristic expected to be revealed from the hitlist. To achieve a generalized result, we conducted 100 sampling iteration. Studies have shown that the average metric value of 100 runs yielded an acceptable statistical precision [55], [56].

Responsiveness: Responsiveness(R) is a necessary property of the hitlist, and this entire study focuses on the characteristics of only the responsive hosts. Responsiveness is important because our goal is to predict the characteristics of the responsive hosts using the hitlist. It is evaluated by finding the number of IP hosts featured in the hitlist that responded to a scan (N_R) and dividing it by the total number of IP hosts in the hitlist (N). This method supports retroactive evaluation and is ideal in performing longitudinal studies [4].

$$R = \frac{N_R}{N}$$

where,
N_R = Total number of responsive IP hosts
N = Total number of IP hosts
R = Responsiveness

Measure of Discrepancy: The measure of discrepancy facilitates in gauging the degree of difference for a categorical variable between two datasets. The average normalized deviation metric is used as the metric to investigate the measure of discrepancy. This metric is preferred as it is invariant of both sample size and number of values in the categorical variable. Our metric must be independent of the earlier mentioned factors as we use the same metric to make comparisons between hitlists of different sizes and categorical values.

$$\lambda_{avg} = \sqrt{\frac{1}{B}\sum_{i=1}^{B}\frac{(O_i - E_i)^2}{E_i^2}}$$

where,
O = Observed count of particular categorical value
E = Expected count of particular categorical value
B = Number of bins (Number of categorical values)

The only notable limitation of this metric is due to the inclusion of Yate's correction. This metric provides a biased result for a small sample size, which contains many categorical values with an expected count less than 5. In our experiment, only sample sizes of 1.5k host experienced such a bias. However, this biasing effect is efficiently minimized by using extreme values while comparing[3]. Pearson's Chi-squared test, KS test, and phi coefficient are the other test that was also studied, and the reason for rejection is as follow was conducted:

1. The chi-squared test is highly sensitive to sample size.
2. KS test is more sensitive in the central region of distribution than the tail. It is possible to overcome this limitation by applying the logarithmic transformation, but this also makes the evaluation process complex [28].
3. Phi-coefficient is independent of sample size but is affected by the number of categorical values. We conducted an empirical evaluation to analyze the trends of the average phi-coefficient (ϕ_{avg}) across all bins and the average normalized deviation (λ_{avg}) metric. Both the values showed similar trends, but the λ_{avg} value was much easier to interpret the marginal differences.

Relative Difference: Relative difference (C) is a metric that compares a particular count between two values. For example, this metric estimates the change in the density of an address block between two scans with a short interval.

$$C = \frac{x_1 - x_2}{x_1}$$

where C = Relative Difference
x_1 = initial value,
x_2 = new value

[3]It is discussed in the upcoming Sub-section 4.2.6.

4.2.6 Interpret Results

The primary focus of the hitlist is to replicate the respective characteristics and the changes incurred in the parent population over the period. Initially, the parent population is evaluated against a new Internet-wide scanned output to find the degree of deviation. This estimated value is called the ground truth, and our hitlist needs to replicate a similar result. The following step is a two-level screening process to select the sampling technique with better accuracy and precision:

1. **Based on average value:** Only those sampling techniques whose estimated result falls within $\pm$5% of the ground truth are shortlisted. This step applies to all the metrics discussed in the previous Sub-section 4.2.5. These selected sampling techniques are further screened. After careful examination of these shortlisted values, it is understandable that this preliminary screening ensures to select only those sampling techniques that are most likely to exhibit better accuracy.

2. **Based on extreme value:** Extreme value (S) is the largest difference between the ground truth and the maximum or minimum value estimated by a hitlist. We estimate the possible range of values that a hitlist can take using the average (μ) and standard deviation (σ). These two values (μ and σ) are obtained while evaluating 100 iterations of sampling. We do not assume any particular distribution while determining this range of values because different sampling techniques exhibit different distribution patterns.

   $$S = max[g - (\mu + \sigma), g - (\mu - \sigma)]$$

   where g = Ground Truth,
   S = Extreme Value
   μ = Average
   σ = Standard Deviation

   Based on this value (S), the top three sampling techniques with the smallest values are selected for each size category. A longitudinal study is performed to analyze the temporal performance of shortlisted sampling techniques. This screening process eliminates the bias introduced due to Yate's correction and also result in shortlist sampling techniques with better precision.

A longitudinal study is carried out to study the stability of the sampling protocol for a period of 90-170 days. Each protocol was probed on their default port approximately five times as a part of data collection. Due to limited time, different

combinations of data were evaluated against each other to study the sampling performance over a different time interval. For example, scan 1 and scan 3 data were used to present the sampling performance on the 60th day, whereas scan 4 and scan 5 data were used to study the result on the 15th day. Finally, plots were drawn to interpret the results and make recommendations on the sampling technique and size based on performance, time, and tolerant margin of error.

Chapter 5

Results

5.1 Motivation

This chapter provides the information inferred upon the examination of the different sampling techniques. Section 5.2 recalls the requirements that the hitlist needs to express and the scanning details. The best sampling approach to generate a hitlist is identified in Section 5.3. In Section 5.4, the stability property of the top three performing sampling techniques is observed. Finally, to further scrutinize the scanning targets of the parent population, two-level sampling is attempted where we considered the impact of Internet Centralization in Section 5.5.

5.2 Hitlist Characteristics and Data Collection

Hitlist Characteristics: Web traffic occupies a large portion ($\sim 70\%$) of the total traffic on the Internet, and further increases due to the booming growth of social networking, cloudification, and video streaming sites [57]. Such a progression on the web directly results in the evolution of web-related technologies, services, protocols, and security. Researchers are motivated to study these changes to enhance their respective features that result in providing a better user experience and a secure environment. Our data comprises various information like ASN, BGP prefix, BGP prefix length, and cipher details, etc. So it is possible to study the performance of a hitlist based on its responsiveness and any protocol feature. Based on our literature survey, as mention in Section 4.2.1, we focus on the attributes of interest to researchers in recent times and frame the following questions that need to be captured in the hitlist:

Q1: Which sampling technique provides a hitlist that replicates the responsiveness and protocol version trends of the parent population?
This question is only applicable to the Transport Layer Security (TLS) protocol

in our thesis because it is the only protocol where studies have shown a keen interest in tracking the growth of protocol version, particularly TLSv1.3, and understanding the applications that support it. From the survey, it is clear that every study has a different requirement with respect to the protocol version, based on their research goal. For instance, a longitudinal study over the TLS protocol versions[1] considers even the error responses (i.e., TLS_NULL) as it adds value to their research. Whereas, TLS_NULL is not required while studying the application usage over different protocol versions. Based on these diverse requirements, this question is subdivided into two: primarily, the hitlist is evaluated by considering all the protocol versions, including TLS_NULL. In the second part, we concentrate only on the valid TLS protocol version without any error.

Q2: Which sampling approach facilitates a hitlist that captures the parent population's responsiveness and its global distribution of the protocol deployment over the prefix lengths?
This question is useful while observing the growth of the protocols across different prefix lengths, and this question has three variants of interest, namely deployment over all the 25 prefix lengths (/8 to /32), routable prefix lengths (/8 to /24), and only /24 prefix length. Most of the study adheres to observe the deployment growth in /24 prefix length because it comprises the most announced BGP Prefix and ASN, as seen in Fig 5.1. Fig 5.1 represents that number of unique BGP Prefixes and ASNs feature across different BGP prefix length.[2].

Q3: Which sampling approach precisely replicates the cross-protocol responsiveness??
This question is relevant to studies that focus on analyzing the responsiveness of the IP hosts and especially the correlation between protocols in terms of responsiveness.

Data Collection: Zmap scanner introduced by the University of Michigan and a modified version of the zgrab scanner [6] located in Australia is used to collect the relevant Internet-wide information on port 443, 80, and 53 (default port of TLS, HTTP, and DNS respectively). Additionally, in the case of TLS, we performed an application-level scan using Goscanner. An application banner scanning is needed because we require application-related information (i.e., protocol version) for TLS

[1] A overview of the TLS versions is available in Appendix A

[2] Fig 5.1 is an observation of an Internet-wide scan on port 443 conducted in July 2020. More granular details about the scan are available in Table B.2

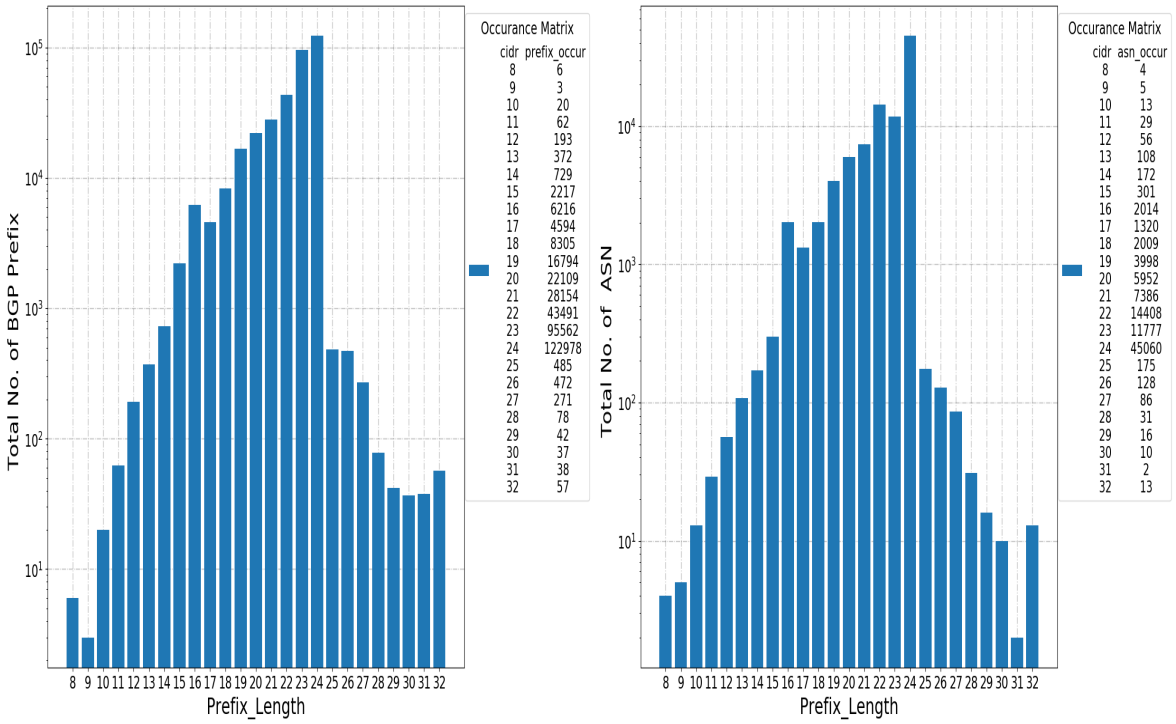


Figure 5.1: Number of announced BGP Prefix and ASN in every Prefix Length

protocol. We collect the version information because many researchers perform longitudinal studies to track the growth of the TLS protocol version. We scanned the entire announced IPv4 address space five times for TLS and DNS protocol. In the case of the HTTP protocol, we probed the Internet six times. Upon receiving an output from the scanner, the raw data is pre-processed to extract valuable information.

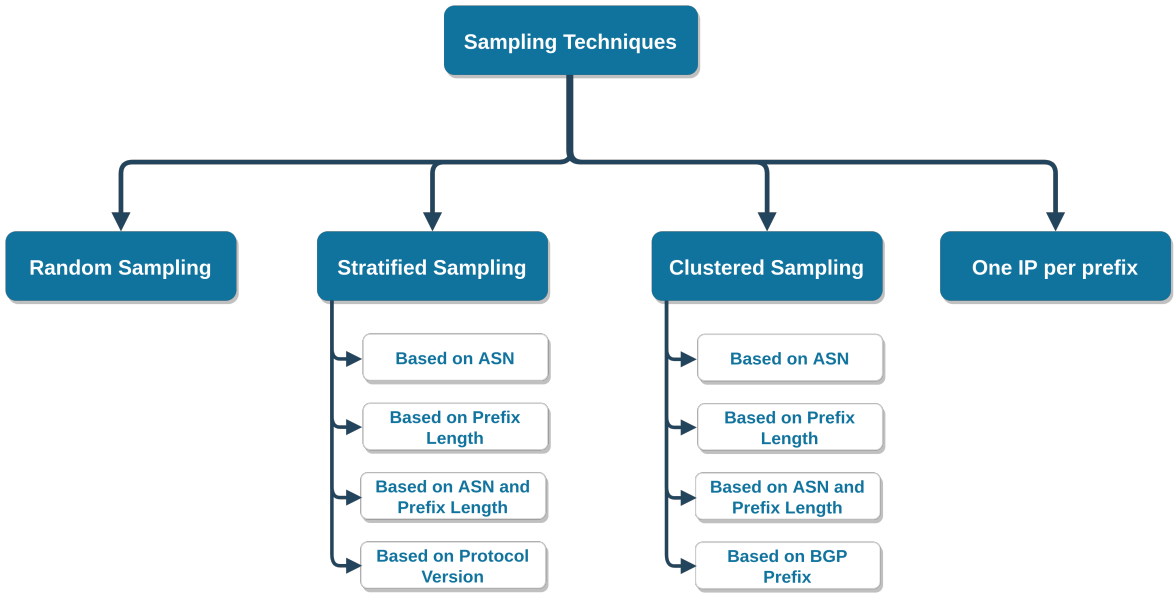


Figure 5.2: Different Sampling Techniques

Subsequently, different sampling techniques are attempted on the pre-processed data and evaluated against its successive scans. For example, a hitlist is generated based on the data collected in April and is evaluated against its succeeding scan from July. The results[3] are visualized using plots for interpretation and carefully analyzed in the upcoming section.

5.3 Best Hitlist Generation Technique

Based on the literature survey, we understand that the random sampling method is the most effective and efficient approach in creating a hitlist. However, our hypothesis is to challenge the random selection process with stratified and cluster-based sampling. All the ten sampling techniques, as mentioned in Figure 5.2, are attempted in the collected data and evaluated.

Our main objective is to develop a hitlist that minimizes the overhead and generalizes the ground truth. To this requirement, it is essential to consider only the protocol features of responsive hosts. Only the responsive representatives reply to our probe request based on which we can classify the hosts using the protocol features. Upon general observation of the results, we find that the hitlist performance is directly proportional to its size. The input based on which stratification is performed decides the accuracy of the respective stratified sampling technique. In other words, the attribute used to split the data into multiple strata needs to exhibit a high degree of correlation to the desired characteristic in the hitlist. On the other hand, cluster sampling is more effective than the rest of the sampling techniques when the population data can be divided into a large number of clusters ($\sim$25) based on a protocol feature. In the remainder of the section, we would study the characteristics of the responsive hosts like protocol version and prefix length in Sub-section 5.3.1 and 5.3.2. In Sub-section 5.3.3, we would observe which sampling technique best replicates the cross-protocol responsiveness.

5.3.1 Protocol Version

We study the protocol version feature only for TLS protocol because it is the only protocol where its versions have been of interest among researchers in recent times. TLS scan performed in April 2020 (scan 1 in Annexure B.2) is utilized as an input to sample and evaluated against the data collected in July 2020 (scan 2 in Annexure B.2). In this sub-section, the average normalized deviation (λ_{avg}) is the only metric

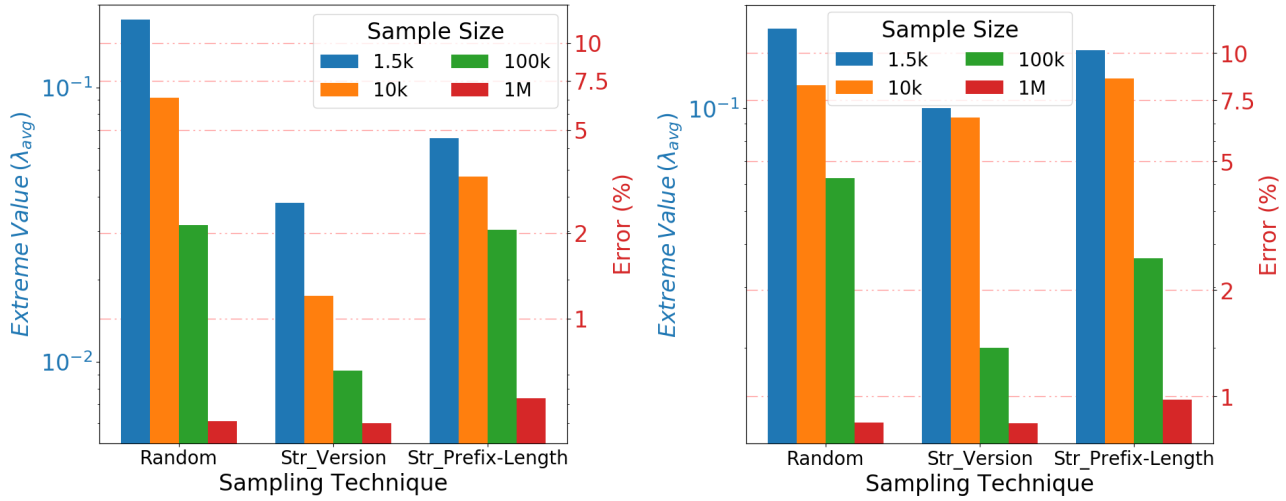
[3]For brevity purposes and to avoid redundant claims/statements, only top-performing sampling techniques results and key observations are discussed in this chapter.

used to determine the measure of discrepancy between the hitlist and parent population. We will look into the performance of different hitlist techniques on both the conditions, with and without TLS_Null.

Primarily, let us focus on the performance of the hitlist that represents all the TLS protocol versions, including TLS_Null. We observe that random sampling, stratified sampling based on the protocol version, and prefix-length are the best performing sampling techniques. Figure 5.3 reflects the performance of these three sampling techniques with respect to extreme value based on λ_{avg} and the error value as their two y-axes, respectively. The error in the y-axis is based on the table 2.3.

Figure 5.3a confirms the fact that with increasing sample size, the hitlist generalizes much better. Also, stratification based sampling performs better than random selection and requires a smaller stratified sample to capture the bigger picture. The use of a specific feature (Eg. protocol version) in distinguishing strata allows stratified sampling to ensure an accurate representation by choosing a smaller sample size than random sampling. Another advantage offered by stratified sampling is that it facilitates researchers to use the same hitlist to examine an individual protocol version separately.

When an error of 1% is acceptable, a hitlist developed using stratification requires only 100K IP hosts to capture the parent population's behavior on the Internet. Stratified sample based on version performs best, but, interestingly, there is a positive correlation between prefix-length and protocol version that leads to a good performance of stratified sampling based on prefix-length. Thus, when a researcher lacks the protocol version information, stratification based on prefix-length can be done to develop the hitlist.



(a) Including All Protocol Version

(b) Error-less Protocol Version

Figure 5.3: Performance of Sampling Techniques based on TLS Protocol Version

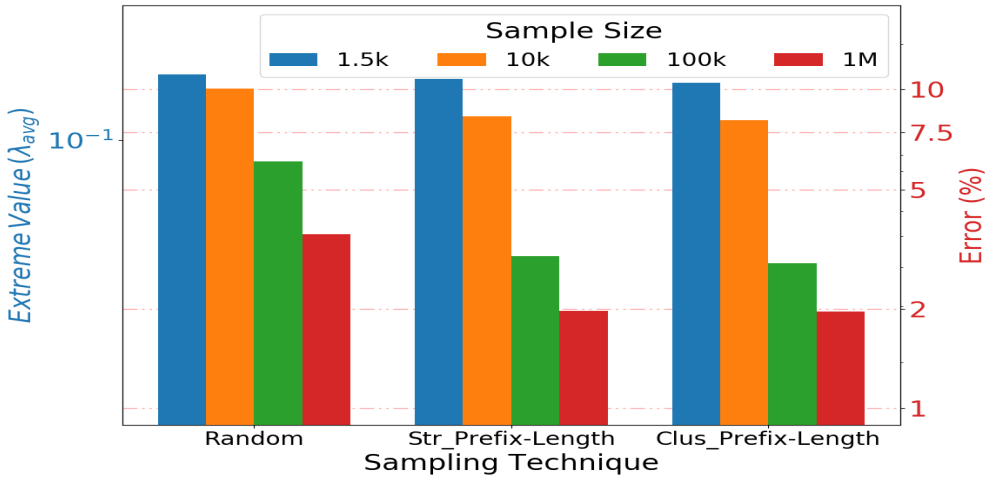
A similar trend among the sampling technique is observed in Figure 5.3b while considering only the errorless TLS versions (excluding TLS_Null). Once the TLS_Null is discarded in our evaluation, the proportion between the remaining protocol version gets unbalanced. 70% of the Internet hosts support TLSv1.2, and the other three protocol versions (TLSv1.0 or TLSv1.1 or TLSv1.3) constitute the remaining 30% of the hosts. This results in extreme values being comparatively higher and thus requires a large sample size to capture the generalized snapshot.

5.3.2 Prefix Length

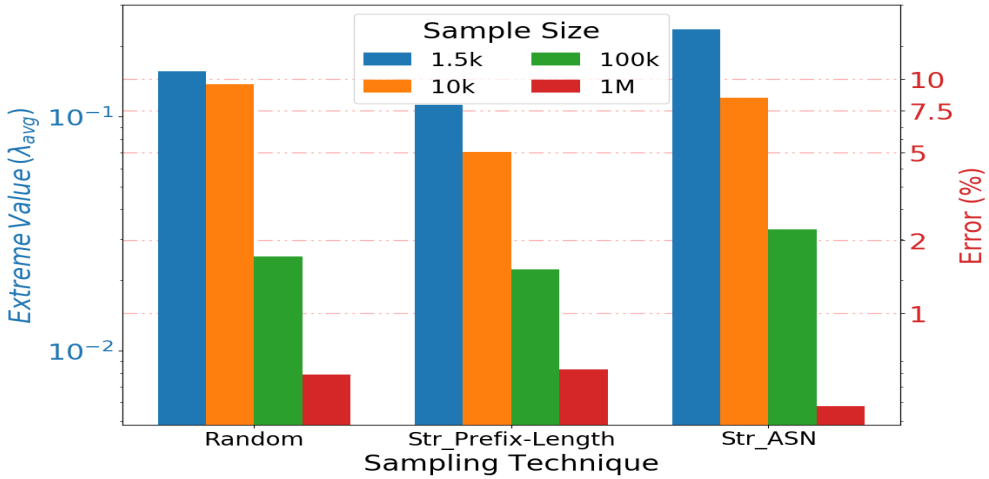
In this section, we will discuss the performance of the different hitlist generation techniques that captures the HTTP[4] deployment across different prefix lengths. The reason behind choosing the HTTP results is because it is pretty much representative, and it is the only protocol where the stratified based on ASN features in the best three techniques while studying the deployment across routable prefix lengths. HTTP scan from July 2020 (scan 1 in Annexure B.1) is the input for sampling and has been validated against the scan conducted in August 2020 (scan 1 in Annexure B.1). Average normalized deviation (λ_{avg}) metric is used to determine the measure of discrepancy when studying the deployment across all the prefix lengths and routable prefix lengths. While examining the deployment across /24 prefix length alone, the relative difference (C) metric is utilized. Figure 5.4a,b reflects the three best performing sampling techniques with respect to extreme value based on λ_{avg} and error value whereas the extreme value is calculated based on relative difference (C) in Figure 5.4c. The error in the y-axis is based on the table 2.3.

While considering the HTTP deployment across all the prefix lengths, i.e., /8 to /32, we observe that random sampling, stratified, and cluster sampling based on prefix-length are the best performing sampling techniques. The random sampling method requires a minimum of one hundred thousand (100K) hosts to select at least one representative from all the 25 prefix lengths. The stratified approach is not able to choose end-host across all the prefix-lengths, even in a bigger sample size of about one million(1M) representatives. A cluster sampling can select at least a single representative from all the clusters, even in a sample size of 10K. Based on Figure 5.4a, the cluster-based sampling method is the most effective and prominent technique available to capture the deployment across all the prefix-lengths. Unlike stratified sampling, the cluster-based sampling technique does not prioritize a stratum with higher density and thus manages to capture hosts from all the prefix lengths and marginally exhibit better performance than the stratified approach. Cluster-based sampling is more effective with population data that shows linear

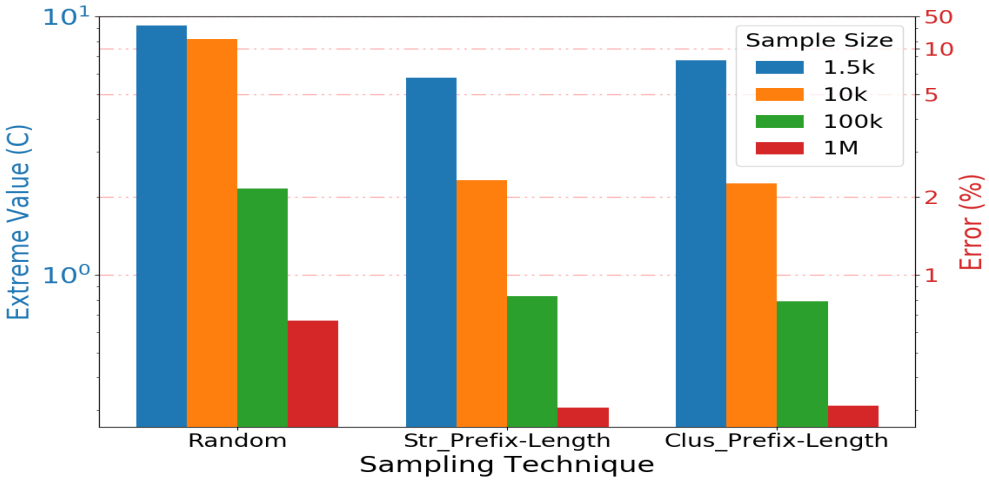
[4]TLS and DNS results are available in Annexure C.



(a) All 25 Prefix Length



(b) Routable Prefix Length



(c) Only /24 Prefix Length

Figure 5.4: Performance of Sampling Techniques based on HTTP Deployment across Prefix Length

trends, and it also has better precision over the other two sampling techniques when the data is heterogeneous and can be split into many clusters [54].

When deployment across only the routable prefix-length (/8 to /24) is considered, a smaller stratified sample (1.5K, 10K, 100K) based on prefix length information portrays the parent population's current status with comparatively lesser error than the remaining two sampling techniques. Stratification based on ASN is highly accurate in a sample size of one million hosts and comprises $\sim$20K unique ASNs. As only a large sample size can provide better accuracy while using stratification based on ASN, we understand that the inclusion of smaller ASNs is very much essential in generalizing the Internet's behavior. A similar observation is witnessed in Section 5.5, where the IP addresses of top 'k' ASNs are skewed at certain prefix instances while attempting the generalize the Internet by only considering the hosts from the top 'k' ASNs that constitute 50% of the overall deployment. Finally, while concentrating on the deployment pattern on /24 prefix-length, both stratified and cluster-based sampling performs better than the simple random sampling technique. However, stratified sampling is marginally more efficient than the cluster-based sampling approach as the number of hosts from /24 is comparatively chosen more (which also means more responsive hosts) in the stratified method than in the cluster-based technique.

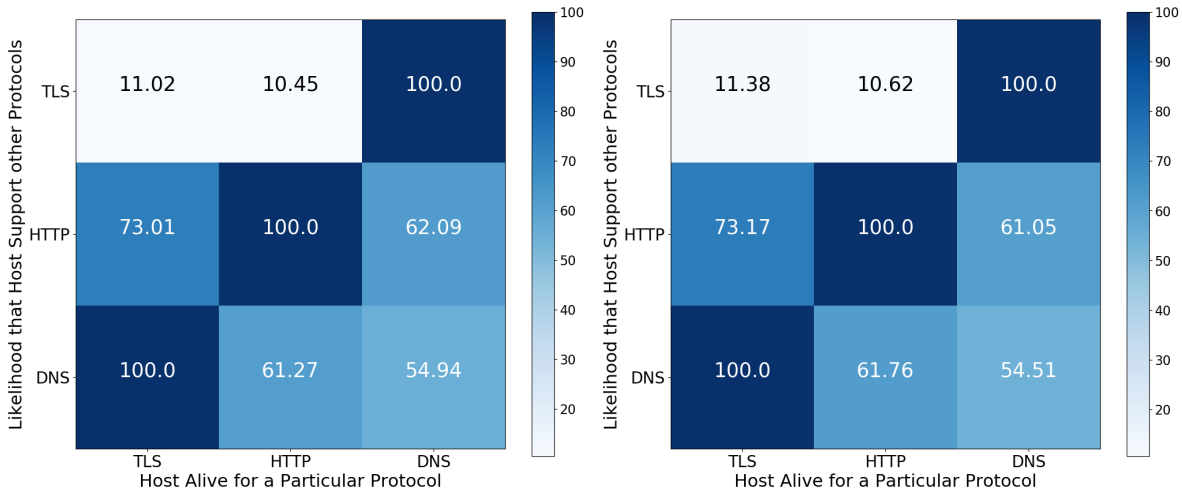
5.3.3 Cross-Protocol Responsiveness

Generally, cross-protocol responsiveness is studied in researches that attempt to understand the responsiveness trait of the end-hosts and to propose a measurement framework. Determining the responsive nature of the hosts is a complex task and depends on numerous factors like:

1. Probe type, target's environment (networking policies enforced using firewalls).
2. Temporal churn in IP addresses.
3. Responsiveness correlation (The likelihood of a host responding to a particular protocol also responsive to another protocol).

We consider the scanning results that were conducted in the first half of September 2020 for all three protocols. Figure 5.5a represents the overall fraction of hosts that support a particular protocol is also likely to respond to another protocol. The following formulae estimate the fraction of the host that is responsive to a particular protocol also responds to different protocols:

$$R_{xy} = \frac{N_{xy}}{N_x} * 100$$



(a) Results based on Parent Population *(b)* Results based on Random Sampling

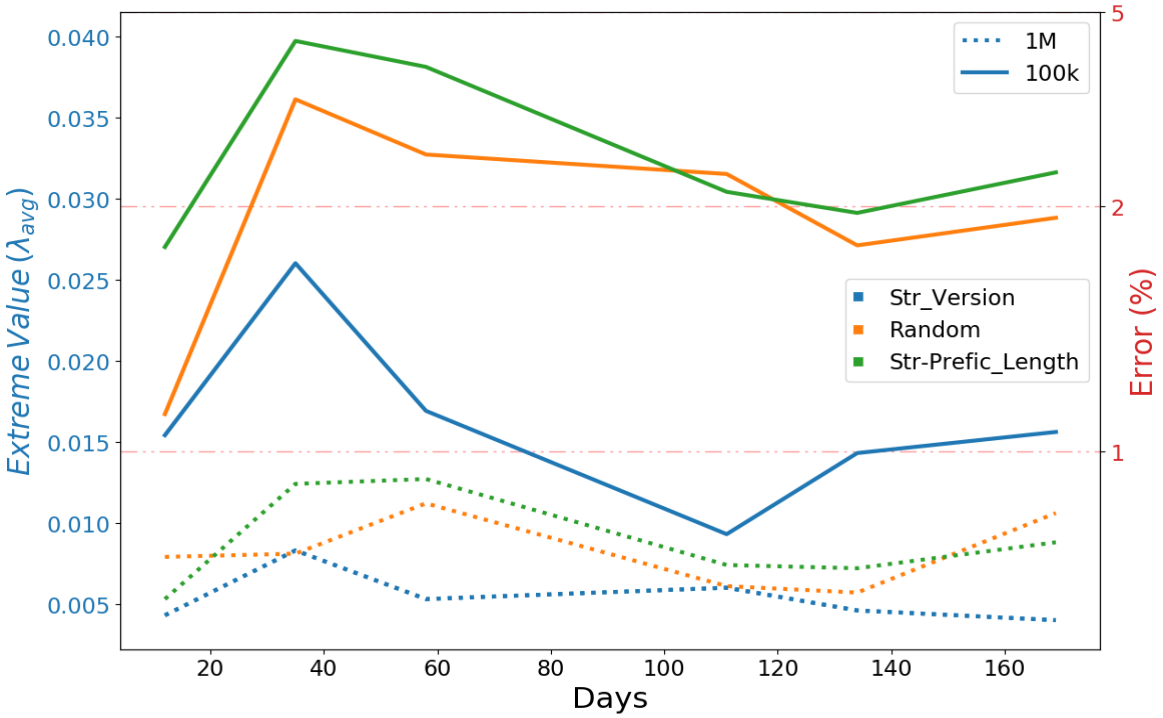
Figure 5.5: Cross-Protocol Responsiveness

where, R_{xy} = Fraction of hosts that is responsive to protocol 'x' is also responsive to protocol 'y'
N_{xy} = Number of hosts that support protocol 'x' also supporting protocol 'y'
N_x = Total number of hosts that support protocol 'x'

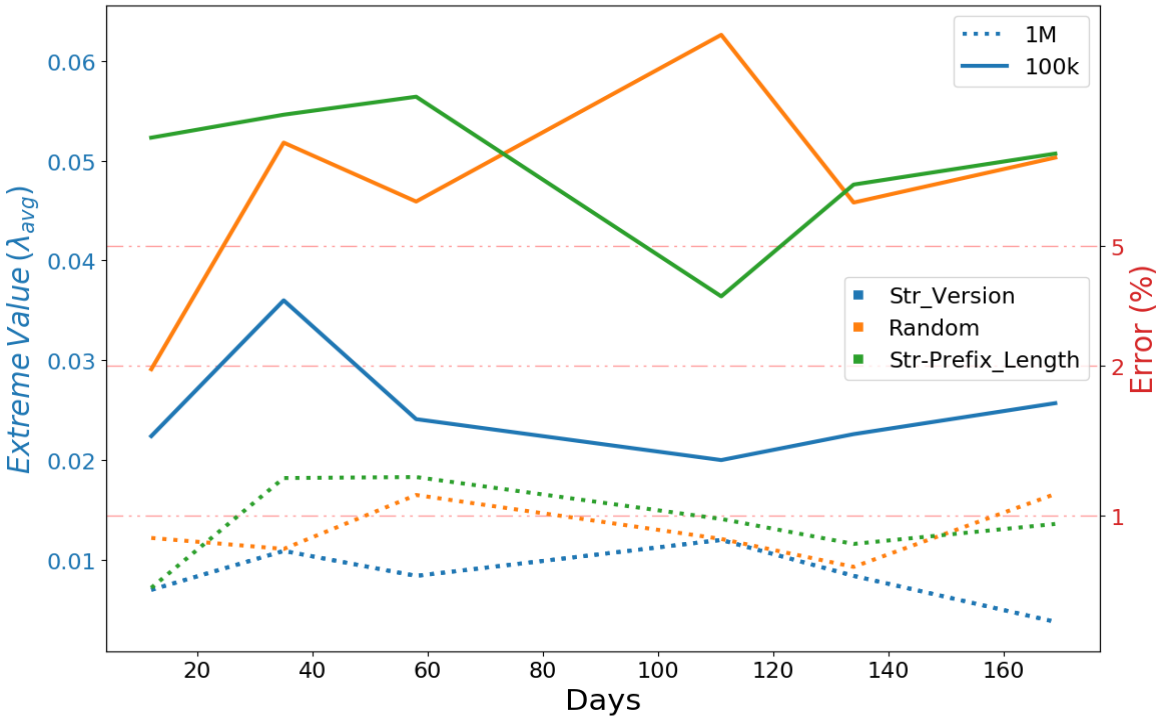
All the sampling techniques perform equally well, with a minimum discrepancy of $\sim$1% and the accuracy of prediction being almost indistinguishable. Stratified sampling and random sampling techniques perform almost similar as none of the attributes based on which stratification is performed have a significant influence on the responsiveness of the hosts. Figure 5.5b provides the results of a randomly selected sample of size 1.5K, and we can observe that the results obtained is similar to the parent population. Random selection of representatives is an ideal approach for this requirement as it demands very minimal effort to generate a hitlist and also because many factors constitute the responsiveness of end-hosts.

5.4 Stability Test

As we have identified the best sampling technique, we conduct a longitudinal study to ensure that the chosen sampling approach performs uniformly over a period of time. Additionally, this experimentation aids in determining how long a scanned population be used as an input for sampling. Due to time constraints, the data collected from the five scans are interchangeably utilized to represent different time intervals (days). Appendix B discloses all the pair of scanned outputs that are considered to represent a particular time interval.



(a) All Protocol Version Included



(b) Protocol Version except TLS_Null

Figure 5.6: Longitudinal Performance of Sampling Techniques based on TLS Protocol Version

5.4.1 Protocol Version

Figure 5.6 represents the longitudinal behavior of the top three sampling techniques that generalizes the protocol version characteristic. All the sampling approaches with a sample size of 1M is stable in terms of the error. Similarly, for any sample size, the hitlist developed by stratification using protocol version remains stable in terms of the error value.

With increasing time interval between the sample and evaluation dataset, the sampling result gradually deteriorates initially and gets better after a certain period. The point at which the sample gets better is when the hitlist becomes stale. From this point in time onwards, ∼90% of the stable IP hosts feature in the hitlist and the parent population. This saturation is due to the presence of dynamic addressing and the existence of middleboxes, especially firewalls. This observation reconfirms the same argument made by Fan et al. [4].

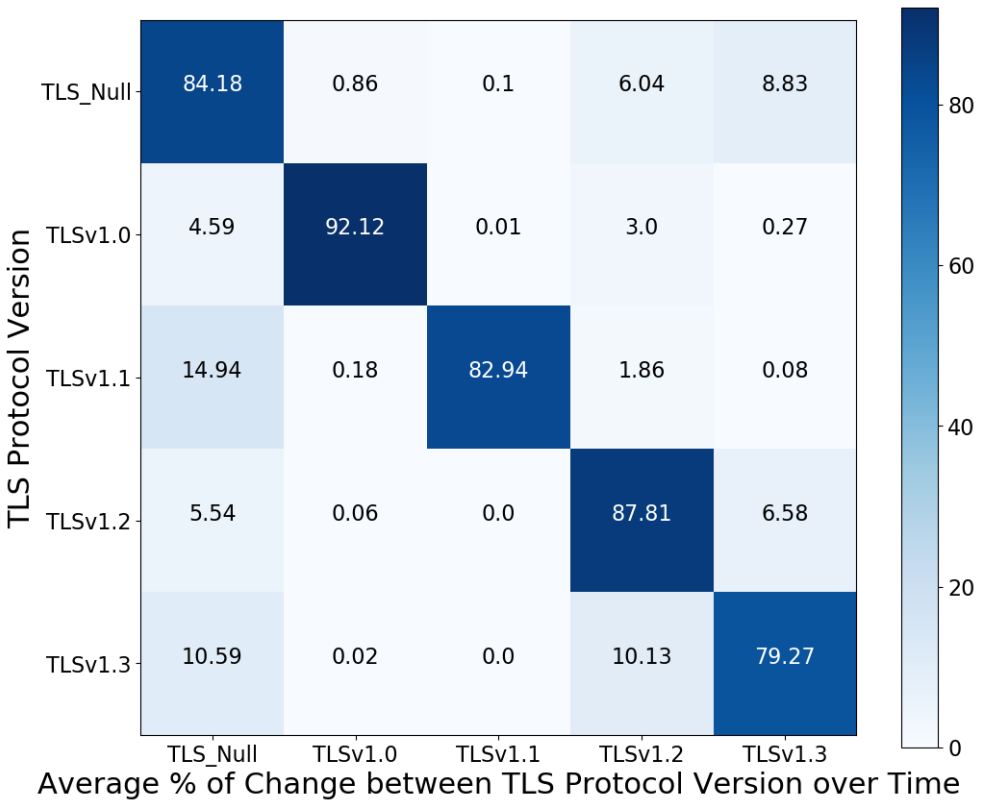


Figure 5.7: Average Change of each TLS protocol version over time

Apart from the dynamic property of the IP hosts, the change in protocol version support by the end host over time affects the hitlist's performance to a certain extent. The average rate at which a portion of end-hosts supporting a particular version is likely to upgrade/downgrade to another protocol version is explained using a matrix in Figure 5.7. When there is a notable variation in protocol versions between two

populations from the average, it affects the overall proportion that compromises the hitlist's performance. This change could be due to the up-gradation of the end-hosts or possibly misconfigured.

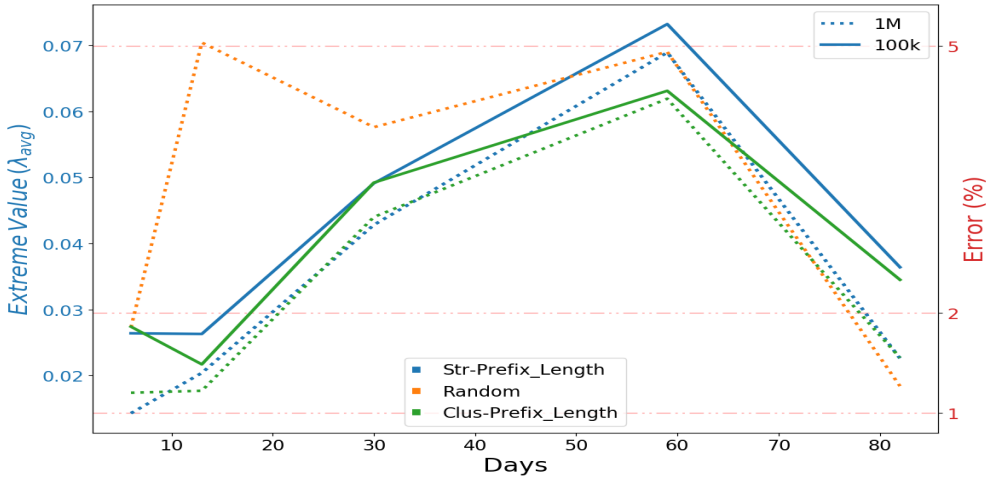
Based on Table B.2 in Annexure B, we observe that the TLSv1.3 gradually increases, and TLSv1.2 decreases over time. TLS_Null shows a stochastic behavior in every scan. Upon observing Figure 5.7, we identify that a significant portion ($\sim$1.5M) of hosts that supports HTTPSv1.2 is upgraded in a short span of 12 days. This shift between the protocol versions ensures that the general pattern observed in the overall data is maintained. However, in certain instances, a considerable amount of hosts supporting HTTPS_Null shifts to HTTPSv1.2, resulting in the rise of HTTPSv1.2 and thus deviating from the general trend of the protocol versions.

5.4.2 Prefix Length

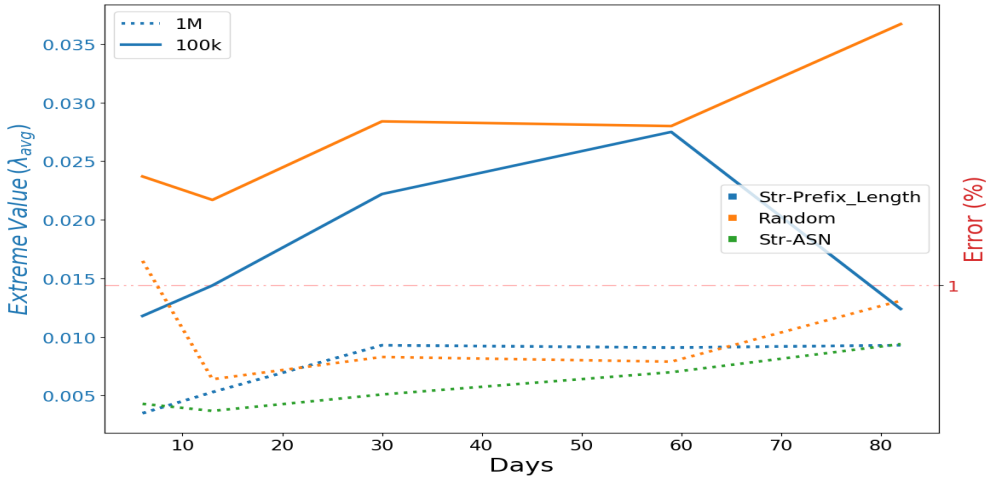
Figure 5.8 represents the hitlist behavior of HTTP deployment over prefix-length. When considering all the available prefix lengths, as in Figure 5.8a, the curve exhibits a similar trend to the protocol version but with a higher amplitude. The growth of hosts across prefix lengths and the amount of skewness determines the degree of performance degradation. The plot reassures the argument that the hitlist performance is influenced by the dynamic property of IP allocation and the presence of firewalls. A random sample of size 1M performs inferior to the cluster-based sampling of size 100K for almost the first 70 days.

Random sampling performance matches the stratified/cluster sampling only when the hitlist becomes stale, where the majority of the population comprises of stable IP hosts that are responsive. Figure 5.9 represents the population of different hosts' responsive behavior over time, and the data becomes stale after two months. However, an extended period of the longitudinal study is required to make any concrete statement in this regard. With more scans, it is possible to predict the time precisely when the data becomes stale.

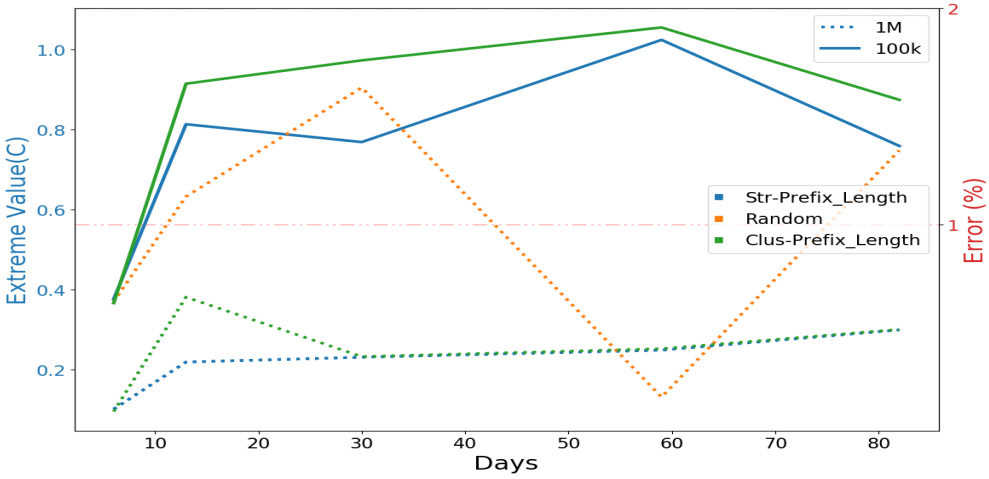
In Figure 5.8b,c, where we consider the HTTP deployment across routable prefix lengths and in /24 prefix length, the stratified sampling methods are much more stable as the data is evenly balanced and exhibit a linear growth. After the longitudinal observation, it is confirmed that stratified sampling is a better approach than the conventional random sampling technique to develop a stable hitlist that generalizes with better accuracy, precision, and minimal sample size.



(a) All 25 Prefix Length



(b) Routable Prefix Length



(c) Only /24 Prefix Length

Figure 5.8: Longitudinal Performance of Sampling Techniques on HTTP deployment across on Prefix Length

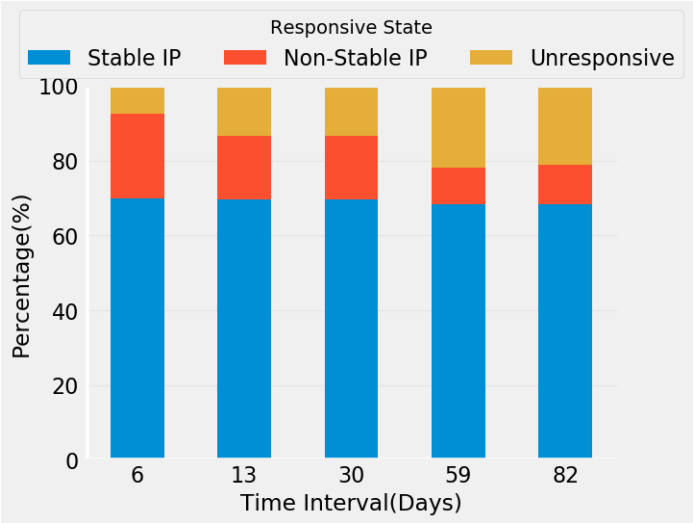


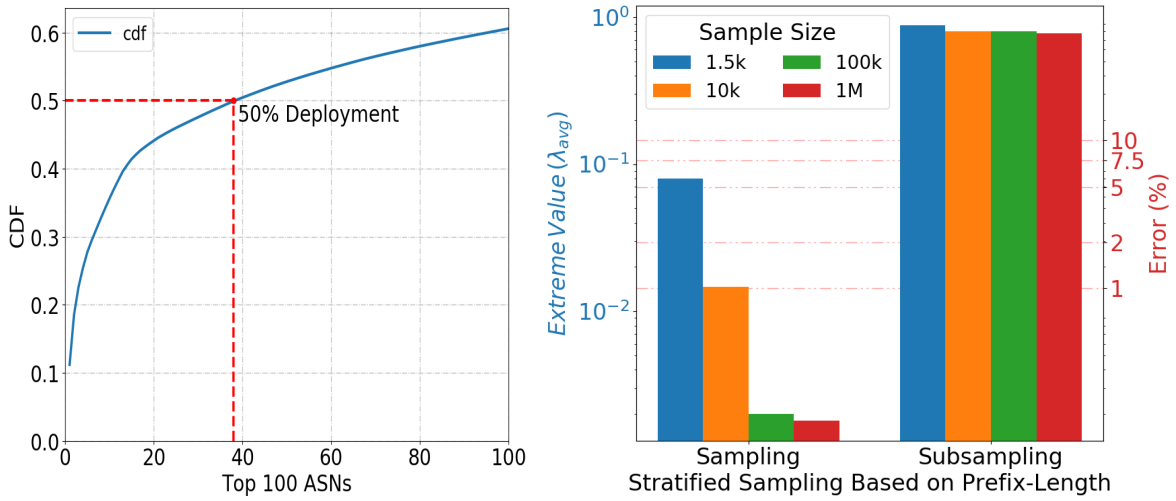
Figure 5.9: HTTP Relative Responsiveness Characteristics over Time

5.5 Impact of Internet Centralization

The main objective of this thesis is to minimize the traffic overhead while measuring the Internet. To do this, we develop a tool that takes the output of the scanner to provide a meaningful hitlist. In the previous section, we have already figured out the best sampling approach to develop a hitlist. As the tool requires input from the scanner, we attempt to minimize the scanning target space and observe if we can still capture the generalized overview of the Internet. As Internet Centralization is directly influencing the growth of the protocol [58], we are motivated to examine if we could generate an optimal hitlist by scanning only the top 'k' ASN deployments.

The same HTTPS population data from the previous Sub-Section 5.3.1 is used to conduct this experiment and compare the results. To accomplish this task, we identify the top 'k' ASN that constitutes to 50% of the global deployment. From the CDF curve presented in Figure 5.10a, we observe that the top 38 ASNs are responsible for half of the HTTPS deployment globally. We shortlist all the host that belongs to the top 38 ASNs and perform the sampling procedure. This two-stage process of sampling is called subsampling.

Figure 5.10b compares the performance between the sampling and subsampling process in generating a hitlist. For the brevity of this report, we discuss only the routable prefix-length characteristics using the stratified sampling based on prefix-length. The results suggest that the subsampling process of concentrating only the top 38 ASNs is not efficient way in comparison to using the entire parent population for generalization. To further analyze the reason behind such a degraded

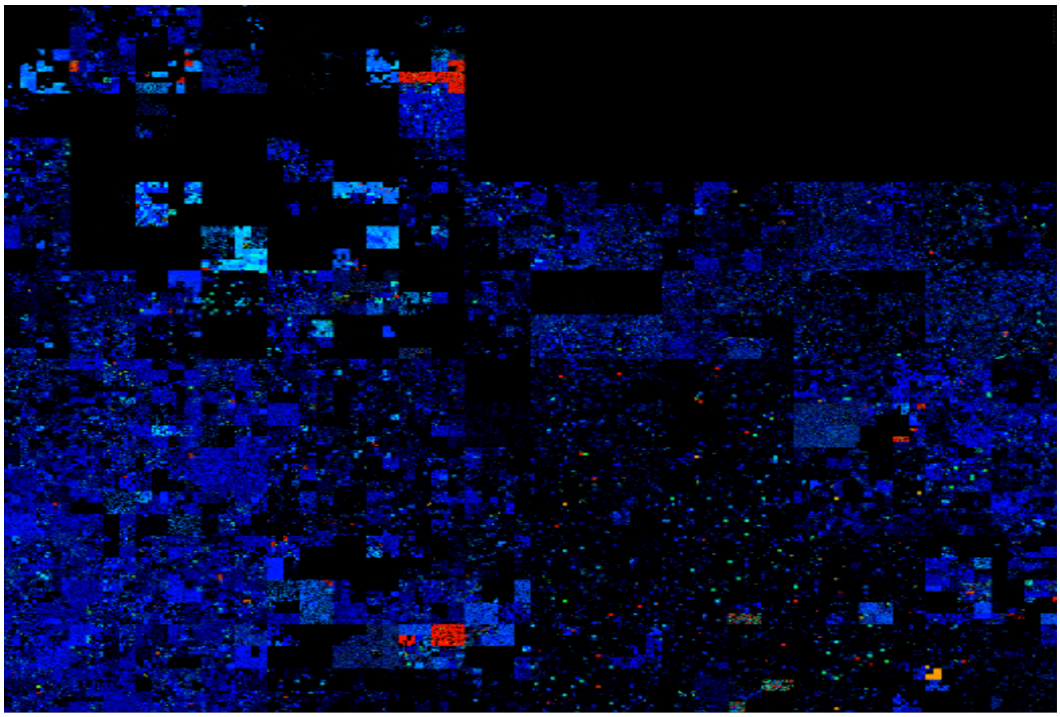


(a) CDF Curve on the Deployment of Top 100 ASNs *(b)* Conventional Sampling vs Two-Stage Subsampling

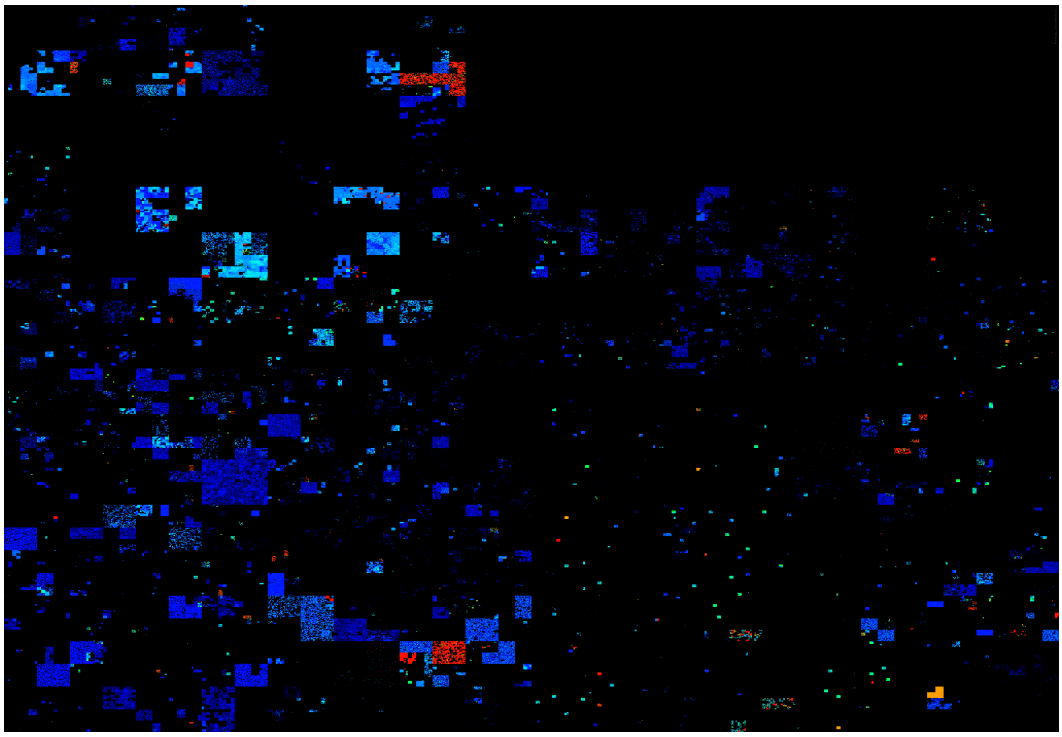
Figure 5.10: TLS Subsampling Based on Internet Centralization

performance, we plotted both the input data using the ipv4 heatmap[56]. Figure 5.11 reveals that in the first level of processing, the data trimming is ineffective. Hosts to the right of the heatmap and few central regions are extremely filtered. Certain IP hosts are skewed in the first level of the filtering process as they are under one ASN authority, leading to biased results. A much more effective first-level sampling would result in achieving better generalization.

[5]The IPv4-heatmap is a 2-D plot that visualizes the IPv4 address space using Hilbert-curve
[6]Heatmap tool: https://github.com/measurement-factory/ipv4-heatmap



(a) Complete Internet-wide TLS Deployment



(b) Top 38 ASNs TLS Deployment

Figure 5.11: IPv4 heatmap of TLS Deployment

Chapter 6

Conclusion

Research groups and experts measure the Internet to check the reachability, latency, routing stability, to conduct a lexical analysis and trends of end-hosts and services. A low response rate affects the effectiveness of Internet measurement. This low rate is due to dynamic address allocation and the middleboxes on the Internet, and it subsequently leads to the generation of excessive, needless traffic. However, when scanning only the specific targets prescribed by the hitlist mitigates the excessive traffic generation and also facilitates a global view of the Internet.

In recent times, the general practice is to pick 1% of IP hosts ($\sim$24k) randomly and scan in short intervals to gain an extended view. This research shows that such a random sampling approach is not ideal for capturing the Internet's characteristics. Sampling-based on stratification can provide more accurate generalization with minimal representatives. Unlike the random sampling approach, the performance of stratified sampling is precise, stable, and invariant of time. We tried to minimize our scans to top 'k' ASNs, such that the input required to capture the overall bigger view is minimal. However, the effect of Internet Centralization does not contribute to the betterment of the sampling techniques.

6.1 Answering Reasearch Questions

RQ1: What are the current strategies employed in generating a hitlist?

A probe list is a set of targets for Internet scans, and they are classified into two based on the target as hitlist and top list. A top list is a sample set of frequently visited domains, and these are only applicable in domain-based scans. Top lists like Alexa, Cisco Umbrella are quite familiar and well-renowned in the Internet Measurement Community. The commonly used top list sizes are 1K, 10K, 100K, 1M. These lists are subjected to the following shortcomings:

1. It is highly unstable with a 50% change in the list every day.

2. The generation method is not transparent and proprietary. Studies have proved that the use of a top list results in biased results [37].

Hitlist, on the other hand, is a set of IP addresses that are useful to scan a single port or service of the entire Internet. The hitlists are developed by selecting representatives randomly or prioritizing hosts based on weights, or by combining both the previous methods. TCP based scans provide $\sim$13% more responsiveness than ICMP based scans when scanning the Internet and thus TCP based scans are preferred to capture the snapshot of the golbal IP space [42]. Efforts are even made to generate a hitlist using top lists, where IP information is extracted from the domains that feature in the top lists. Random selection is the most convenient and preferred method to generate a hitlist for a generalized observation. The most common practice is to choose a sample size of 24K representatives, where one representative from each announced BGP prefix is randomly selected.

RQ2: What is the best sampling technique used to generate a hitlist? Whether each characteristic of a protocol that needs to be studied demands a different sampling approach?

From the previous Research Question **RQ1:**, we understand that the random sampling technique is the state-of-the-art approach to develop a hitlist that generalizes the Internet. We attempted to challenge the performance of random sampling with other probabilistic sampling techniques like stratified and cluster-based sampling. A statistical evaluation method was sketched by modifying the phi chi-square goodness of fit test to determine the measure of the discrepancy between the parent population and the hitlist. Upon conducting this experimentation, we observe that the best sampling approach varies for every desired characteristics, which the hitlist needs to express. Mostly, stratified sampling is the best approach in most of the cases. However, the input attribute based on which stratification is implemented needs to have a highly positive correlation with the expected characteristic from the hitlist. In some cases, when the desired feature possesses numerous categorical values, and also its distribution is skewed and unbalanced, then a cluster-based sampling is appropriate. Finally, random sampling is preferred when the desired characteristic depends on numerous factors and is complex to predict its correlation with single information.

RQ3: Is the generated hitlist stable and invariant of time?

A longitudinal study was conducted to study the stability property of the best three sampling results. It was evident that random sampling was quite unstable with time and exhibited a high discrepancy value for the first two months of

the parent population. Once the parent population is two months, it starts to saturate as ∼90% of the responsive hosts features only stable IP addresses, and this is the point where every sampling technique's performance become almost identical. Thus, it is recommended that a fresh Internet-wide scan is required to better study Internet behavior.

Stratified and cluster-based sampling shows a stable performance, and better accuracy is possible when the scanned data displays a linear growth/fall over time and is balanced without being skewed. The stability of the hitlist varies from protocol to protocol. For example, it is more likely that smaller servers turned on and off the HTTP services based on their requirement, whereas it is not the case for a DNS protocol.

RQ4: Can Internet centralization aid in influencing the hitlist generation tactics?

The rise of global players on the Internet influence the growth in protocol deployments, and thus resulting in Internet Centralization. It is mandatory to scan the complete Internet and use it as input for sampling. However, we attempted to use this effect and observe whether sampling only the deployments belonging to the top 'k' ASNs that constitutes to the 50% of the overall protocol deployment is sufficient. Primarily, we determined the number of ASNs that occupies half of the overall deployment. We understand that 40-60 ASNs capture half of the deployment. Upon sampling only these top ASNs, the hitlist were not able to generalize the Internet. This two-level of sampling is called subsampling. From this experiment, we understand that the inclusion of smaller ASNs is mandatory to obtain a generalized Internet behavior.

Main Question: "How to generate a sustainable hitlist that reputedly represents the general Internet behaviour?"

As a conclusion of the research, we can safely state that the ideology of "one solution fit all" should be discarded when it comes to hitlist generation. Every study has a unique objective and scope, so based on the requirement of the research, a hitlist generation technique needs to be inculcated. A fresh scan is required every two months as the dynamic property of the Internet is lost due to the temporal churn and filtering process. Based on all the experiments conducted, we present the following one-line command tool that provides a hitlist based on the requirement:

IPv4 Hitlist Tool- https://github.com/danish10499/hitlist-ipv4.git

6.2 Limitations

This thesis is subject to the following restrictions:

- **Yate's Correction:** When conducting a longitudinal study on the DNS deployment across all the prefix length using the hitlist, we observe a sudden rise in the discrepancy value after the 20th day, as in Figure C.6. This unexpected peak is caused due to the short-coming in the statistical evaluation when using Yate's correction. The number of hosts across prefix length is highly unbalanced, and a significant rise in one of the least populated prefix length is witnessed over time. The expected count of this growing prefix length was less than five in the hitlist and thus discarded from the calculation when estimating the discrepancy. Therefore the difference between the hitlist and parent population against the evaluation data is comparatively higher.

- **Reproducibility:** The networks and their location from where the Internet-wide scans are conducted influence the results [59]. Thus, different venues of the scanners produce a marginally different result due to regional access limitations, and the policy decisions of a local service provider can restrict a scanner's visibility. Additionally, we probed all the announced BGP prefixes

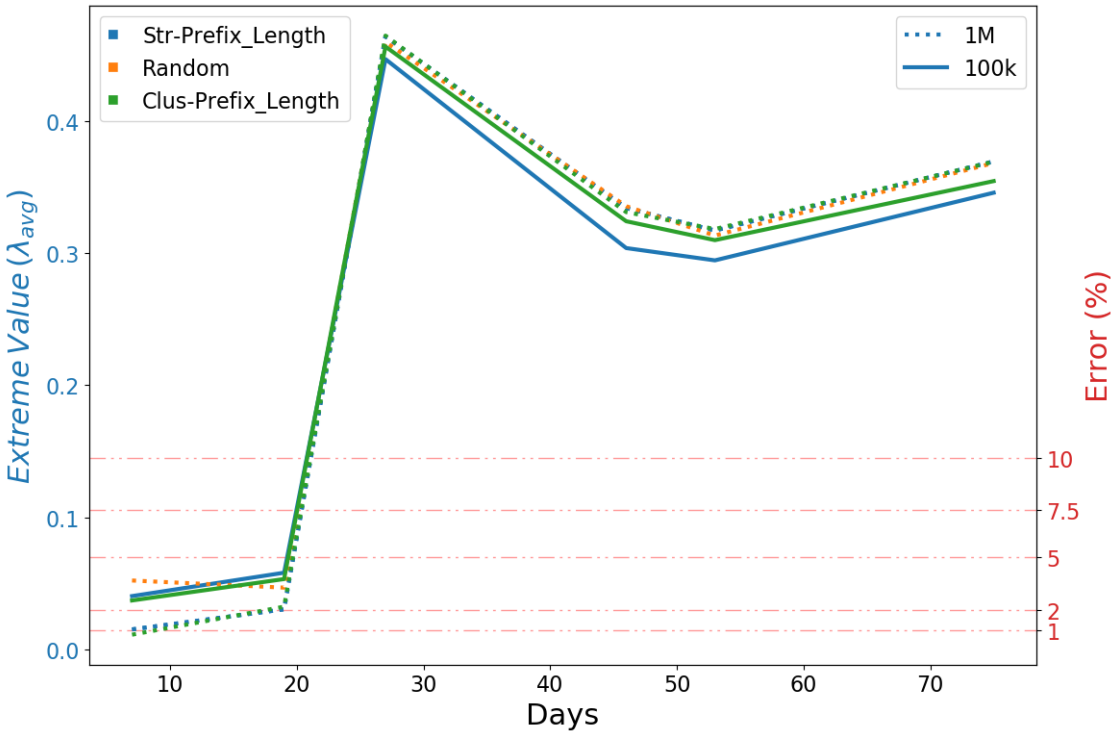


Figure 6.1: Longitudinal Performance of Sampling Techniques on DNS deployment across on Prefix Length

during the time of the scan to capture the present global snapshot, and also, we updated the blacklist before every scan.

6.3 Recommendations

Based on the results obtained and observation of all the sampling techniques' performance, we propose the following recommendations:

- Table 6.1 recommends the type of sampling and the preferred hitlist size to consider based on the acceptable error value.
- Most of the research works aim at developing a complete hitlist by selecting a representative from each announced BGP prefix. The results are compared between a hitlist using random sampling and a hitlist where one representative from each prefix is selected to describe the deployment across /24 prefix length in Figure 6.2. From the results, it is evident that this method yields inaccurate results and yields a biased result. Even when a study attempts to produce an optimized hitlist with high responsiveness, this technique compromises these properties. Selecting a single representative from each group/cluster is an inappropriate technique to capture the general characteristics of the Internet. Also, the use of random sampling needs to be minimized wherever possible as this hitlist generation technique is highly unstable with regard to time.

6.4 Future Work

Several aspects that can be looked upon and researched on this topic in the future are:

- Due to time constraints in this thesis work, we confined our longitudinal study and used scans in the best way possible. However, an extensive longitudinal study can reconfirm our observations and granularly examine the dynamic property of IP responsiveness.
- Reproducing the previous hitlist related studies and comparing the results obtained using the random sampling and other probabilistic sampling methods like stratified and cluster-based sampling.
- Fine-tune the statistical evaluation method to nullify the limitation caused due to Yate's correction.

Protocol	Error	Sample	Protocol Verion		Prefix Length			Cross-Responsiveness
			All	Errorless	All	Routable	/24	
TLS	1%	Type	Stratified_Version	Stratified_Version	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	1M	1M	1M	100K	100K	1.5k
	2%	Type	Stratified_Version	Stratified_Version	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	100K	100K	100K	100K	100K	1.5k
	5%	Type	Stratified_Version	Stratified_Version	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	10K	10K	100K	10K	10K	1.5k
HTTP	1%	Type	NA	NA	Cluster_Prefix Length	Stratified_ASN	Stratified_Prefix Length	Random Sample
		Size	NA	NA	1M	1M	1M	1.5k
	2%	Type	NA	NA	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	NA	NA	1M	100K	100K	1.5k
	5%	Type	NA	NA	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	NA	NA	100K	100K	10K	1.5k
DNS	1%	Type	NA	NA	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	NA	NA	1M	1M	1M	1.5k
	2%	Type	NA	NA	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	NA	NA	1M	100K	100K	1.5k
	5%	Type	NA	NA	Cluster_Prefix Length	Stratified_Prefix Length	Stratified_Prefix Length	Random Sample
		Size	NA	NA	100K	10K	10K	1.5k

Table 6.1: Recommendation on Sampling Size and Technique Based on Acceptable Error

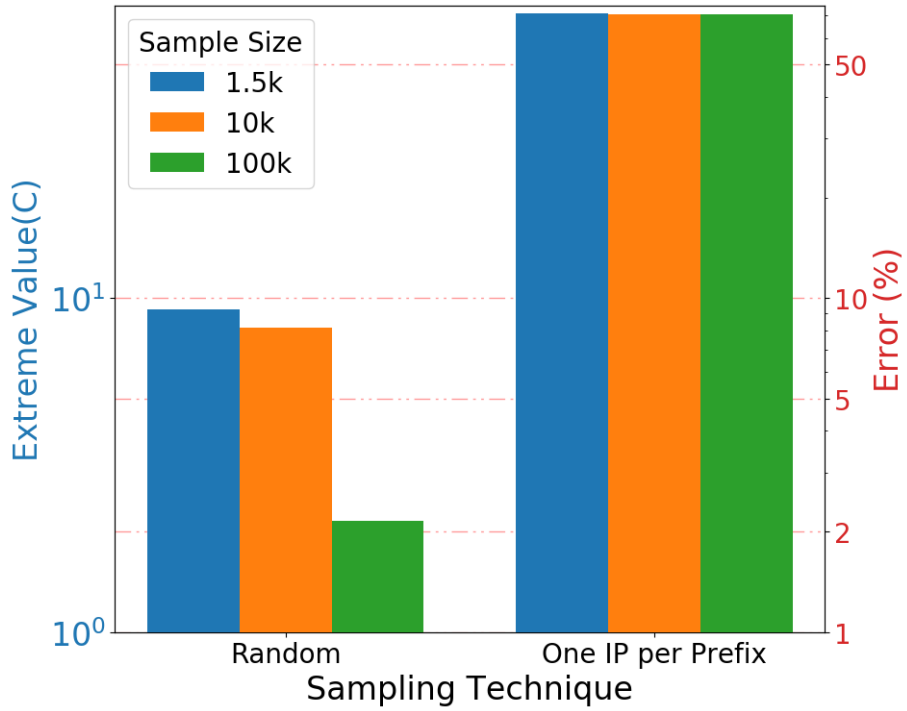


Figure 6.2: Comparison between Random Sampling and Per Prefix Sampling on HTTP's /24 prefix length

- In this thesis, we implemented univariate testing. However, it is necessary to extend the analytical method by including a multi-variate testing technique, so we could generate hitlists that capture multiple characteristics of the Internet.
- Look for alternative approaches to achieve better generalization using the sub-sampling process. Examine the performance of stratified sampling based on geographical information. Finally, look into a new set of requirements that a researcher could consider and figure out a suitable hitlist generation methodology.
- In the context of this research, we evaluate only the scans based on the vantage point located in Australia. But a proper ground truth may have to be compiled based on many vantage points. Thus, it is important to replicate the exercise using the data collected from different vantage points across the globe.

Bibliography

[1] M. Rabinovich and M. Allman, “Measuring the internet,” *IEEE Internet Computing*, vol. 20, no. 04, pp. 6–8, jul 2016.

[2] J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister, “Census and survey of the visible internet,” in *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’08. New York, NY, USA: Association for Computing Machinery, 2008, p. 169–182. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/1452520.1452542

[3] X. Cai and J. Heidemann, “Understanding block-level address usage in the visible internet,” *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, p. 99–110, Aug. 2010. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/1851275.1851196

[4] X. Fan and J. Heidemann, “Selecting representative ip addresses for internet topology studies,” in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 411–423. [Online]. Available: https://doi.org/10.1145/1879141.1879195

[5] Z. Durumeric, E. Wustrow, and J. A. Halderman, “Zmap: Fast internet-wide scanning and its security applications,” in *Proceedings of the 22nd USENIX Conference on Security*, ser. SEC’13. USA: USENIX Association, 2013, p. 605–620.

[6] D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman, “Zippier zmap: Internet-wide scanning at 10 gbps,” in *8th USENIX Workshop on Offensive Technologies (WOOT 14)*. San Diego, CA: USENIX Association, Aug. 2014. [Online]. Available: https://www.usenix.org/conference/woot14/workshop-program/presentation/adrian

[7] “The zmap project.” [Online]. Available: https://zmap.io/

[8] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, “A search engine backed by internet-wide scanning,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 542–553. [Online]. Available: https://doi.org/10.1145/2810103.2813703

[9] J. Klick, S. Lau, M. Wählisch, and V. Roth, “Towards better internet citizenship: Reducing the footprint of internet-wide scans by topology aware prefix selection,” in *Proceedings of the 2016 Internet Measurement Conference*, ser. IMC ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 421–427. [Online]. Available: https://doi.org/10.1145/2987443.2987457

[10] O. Gasser, Q. Scheitle, S. Gebhard, and G. Carle, “Scanning the ipv6 internet: Towards a comprehensive hitlist,” *CoRR*, vol. abs/1607.05179, 2016. [Online]. Available: http://arxiv.org/abs/1607.05179

[11] “Scans.io.” [Online]. Available: https://scans.io/

[12] “Zoomeye.” [Online]. Available: https://www.zoomeye.org/

[13] “fofa.io.” [Online]. Available: https://fofa.so/

[14] Y. Lee and Y. Lee, “Toward scalable internet traffic measurement and analysis with hadoop,” *SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 1, p. 5–13, Jan. 2012. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/2427036.2427038

[15] X. Cai and J. Heidemann, “Understanding block-level address usage in the visible internet,” in *Proceedings of the ACM SIGCOMM 2010 Conference*, ser. SIGCOMM ’10. New York, NY, USA: Association for Computing Machinery, 2010, p. 99–110. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/1851182.1851196

[16] G. Bartlett, J. Heidemann, and C. Papadopoulos, “Understanding passive and active service discovery,” in *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC ’07. New York, NY, USA: Association for Computing Machinery, 2007, p. 57–70. [Online]. Available: https://doi.org/10.1145/1298306.1298314

[17] V. E. Paxson, “Measurements and analysis of end-to-end internet dynamics,” Ph.D. dissertation, USA, 1998.

[18] “Masscan: Mass ip port scanner.” [Online]. Available: https://github.com/robertdavidgraham/masscan

[19] “Scanrand,” 2008. [Online]. Available: http://www.vulnerabilityassessment.co.uk/scanrand.html

[20] “Unicornscan,” 2013. [Online]. Available: https://www.aldeid.com/wiki/Unicornscan

[21] “shodan.io.” [Online]. Available: https://www.shodan.io/

[22] L. Quan, J. Heidemann, and Y. Pradkin, “When the internet sleeps: Correlating diurnal networks with external factors,” in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 87–100. [Online]. Available: https://doi.org/10.1145/2663716.2663721

[23] A. Schulman and N. Spring, “Pingin’ in the rain,” in *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, ser. IMC ’11. New York, NY, USA: Association for Computing Machinery, 2011, p. 19–28. [Online]. Available: https://doi.org/10.1145/2068816.2068819

[24] J. Naab, P. Sattler, J. Jelten, O. Gasser, and G. Carle, “Prefix top lists: Gaining insights with prefixes from domain-based top lists on dns deployment,” in *Proceedings of the Internet Measurement Conference*, ser. IMC ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 351–357. [Online]. Available: https://doi.org/10.1145/3355369.3355598

[25] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. A. Halderman, “A search engine backed by internet-wide scanning,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS ’15. New York, NY, USA: Association for Computing Machinery, 2015, p. 542–553. [Online]. Available: https://doi.org/10.1145/2810103.2813703

[26] “University of oregon. route views project.” [Online]. Available: http://www.routeviews.org.

[27] A. E. D. Awduche, A. Chiu, I. Widjaja, and X. Xiao, “Overview and Principles of Internet Traffic Engineering,” Informational, RFC 3272, May 2002. [Online]. Available: https://tools.ietf.org/html/rfc3272

[28] K. C. Claffy, G. C. Polyzos, and H.-W. Braun, “Application of sampling methodologies to network traffic characterization,” vol. 23, no. 4, p. 194–203, Oct. 1993. [Online]. Available: https://doi.org/10.1145/167954.166256

[29] F. Begtaševiüand and P. Mieghem, “Measurements of the hopcount in internet,” 01 2002. [Online]. Available: http://resolver.tudelft.nl/uuid:79c4c1f2-a443-4466-9cc2-67c7cea17508

[30] “Alexa. top 1m sites,” *Alexa*, 2020. [Online]. Available: https://www.alexa.com/topsites

[31] “Cisco. umbrella top 1m list.” [Online]. Available: https://umbrella.cisco.com/blog/cisco-umbrella-1-million

[32] “Majestic.” [Online]. Available: https://majestic.com/reports/majestic-million/

[33] “Quantcast.” [Online]. Available: https://www.quantcast.com/top-sites/US/1.

[34] “Statvoo.” [Online]. Available: https://statvoo.com/top/sites

[35] “Google. chrome user experience report.” [Online]. Available: https://developers.google.com/web/tools/chrome-user-experience-report/

[36] “Similarweb top websites ranking.” [Online]. Available: https://www.similarweb.com/top-websites.

[37] Q. Scheitle, O. Hohlfeld, J. Gamba, J. Jelten, T. Zimmermann, S. D. Strowes, and N. Vallina-Rodriguez, “A long way to the top: Significance, structure, and stability of internet top lists,” in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 478–493. [Online]. Available: https://doi.org/10.1145/3278532.3278574

[38] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczynski, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” *Proceedings 2019 Network and Distributed System Security Symposium*, 2019. [Online]. Available: http://dx.doi.org/10.14722/ndss.2019.23386

[39] L. Alt, R. Beverly, and A. Dainotti, “Uncovering network tarpits with degreaser,” in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC ’14. New York, NY, USA: Association for Computing Machinery, 2014, p. 156–165. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/2664243.2664285

[40] G. P. Moore, David S.; McCabe, *Introduction to the Practice of Statistics*, 4th ed. New York;: W.H. Freeman and Co.„ 2003.

[41] O. Gasser, Q. Scheitle, P. Foremski, Q. Lone, M. Korczyński, S. D. Strowes, L. Hendriks, and G. Carle, “Clusters in the expanse: Understanding and unbiasing ipv6 hitlists,” in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 364–378. [Online]. Available: https://doi.org/10.1145/3278532.3278564

[42] S. Bano, P. Richter, M. Javed, S. Sundaresan, Z. Durumeric, S. J. Murdoch, R. Mortier, and V. Paxson, “Scanning the internet for liveness,” *SIGCOMM Comput. Commun. Rev.*, vol. 48, no. 2, p. 2–9, May 2018. [Online]. Available: https://doi.org/10.1145/3213232.3213234

[43] P. Richter, R. Padmanabhan, N. Spring, A. Berger, and D. Clark, “Advancing the art of internet edge outage detection,” in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 350–363. [Online]. Available: https://doi.org/10.1145/3278532.3278563

[44] C. Testart, P. Richter, A. King, A. Dainotti, and D. Clark, “Profiling bgp serial hijackers: Capturing persistent misbehavior in the global routing table,” in *Proceedings of the Internet Measurement Conference*, ser. IMC ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 420–434. [Online]. Available: https://doi.org/10.1145/3355369.3355581

[45] C. Lu, B. Liu, Z. Li, S. Hao, H. Duan, M. Zhang, C. Leng, Y. Liu, Z. Zhang, and J. Wu, “An end-to-end, large-scale measurement of dns-over-encryption: How far have we come?” in *Proceedings of the Internet Measurement Conference*, ser. IMC ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 22–35. [Online]. Available: https://doi.org/10.1145/3355369.3355580

[46] M. Allman, “Comments on dns robustness,” in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC ’18. New York, NY, USA: Association for Computing Machinery, 2018, p. 84–90. [Online]. Available: https://doi.org/10.1145/3278532.3278541

[47] M. Nawrocki, J. Blendin, C. Dietzel, T. C. Schmidt, and M. Wählisch, “Down the black hole: Dismantling operational practices of bgp blackholing at ixps,” in *Proceedings of the Internet Measurement Conference*, ser. IMC ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 435–448. [Online]. Available: https://doi.org/10.1145/3355369.3355593

[48] P. Foremski, O. Gasser, and G. C. M. Moura, “Dns observatory: The big picture of the dns,” in *Proceedings of the Internet Measurement Conference*,

ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 87–100. [Online]. Available: https://doi.org/10.1145/3355369.3355566

[49] P. Kotzias, A. Razaghpanah, J. Amann, K. G. Paterson, N. Vallina-Rodriguez, and J. Caballero, "Coming of age: A longitudinal study of tls deployment," in *Proceedings of the Internet Measurement Conference 2018*, ser. IMC '18. New York, NY, USA: Association for Computing Machinery, 2018, p. 415–428. [Online]. Available: https://doi.org/10.1145/3278532.3278568

[50] B. Anderson and D. McGrew, "Tls beyond the browser: Combining end host and network data to understand application behavior," in *Proceedings of the Internet Measurement Conference*, ser. IMC '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 379–392. [Online]. Available: https://doi.org/10.1145/3355369.3355601

[51] J. Greensmith and U. Aickelin, "Dendritic cells for syn scan detection," ser. GECCO '07. New York, NY, USA: Association for Computing Machinery, 2007, p. 49–56. [Online]. Available: https://doi.org/10.1145/1276958.1276966

[52] J. Amann, O. Gasser, Q. Scheitle, L. Brent, G. Carle, and R. Holz, "Mission accomplished? https security after diginotar," ser. IMC '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 325–340. [Online]. Available: https://doi.org/10.1145/3131365.3131401

[53] Y. Zhu, J. Rexford, S. Sen, and A. Shaikh, "Impact of prefix-match changes on ip reachability," ser. IMC '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 235–241. [Online]. Available: https://doi-org.ezproxy2.utwente.nl/10.1145/1644893.1644922

[54] N. Duffield, "Sampling for passive internet measurement: A review," *Statist. Sci.*, vol. 19, no. 3, pp. 472–498, 08 2004. [Online]. Available: https://doi.org/10.1214/088342304000000206

[55] K. Carling and X. Meng, "Confidence in heuristic solutions?" vol. 63, no. 2, p. 381–399, Oct. 2015. [Online]. Available: https://doi.org/10.1007/s10898-015-0293-4

[56] N. G. Hall, S. Ghosh, R. Kankey, S. Narasimhan, and W. Rhee, "Bin packing problems on one dimension: Heuristic solutions and confidence intervals," *Comput. Oper. Res.*, vol. 15, no. 2, p. 171–177, Feb. 1988. [Online]. Available: https://doi.org/10.1016/0305-0548(88)90009-3

[57] Hyoung-Kee Choi and J. O. Limb, “A behavioral model of web traffic,” in *Proceedings. Seventh International Conference on Network Protocols*, 1999, pp. 327–334.

[58] R. Holz, J. Amann, A. Razaghpanah, and N. Vallina-Rodriguez, “The era of tls 1.3: Measuring deployment and use with active and passive methods,” 2019.

[59] G. Wan, L. Izhikevich, D. Adrian, K. Yoshioka, R. Holz, C. Rossow, and Z. Durumeric, “On the origin of scanning: The impact of location on internet-wide scans,” in *Proceedings of the ACM Internet Measurement Conference*, ser. IMC ’20. New York, NY, USA: Association for Computing Machinery, 2020, p. 662–679. [Online]. Available: https://doi.org/10.1145/3419394.3424214

[60] T. Dierks and C. Allen, “Rfc2246: The tls protocol version 1.0,” USA, Tech. Rep., 1999.

[61] “Plan for change: Tls 1.0 and tls 1.1 soon to be disabled by default.” [Online]. Available: https://blogs.windows.com/msedgedev/2020/03/31/tls-1-0-tls-1-1-schedule-update-edge-ie11/

[62] “Tls 1.0 and tls 1.1.” [Online]. Available: https://www.chromestatus.com/feature/5759116003770368

Appendix A

Overview on TLS Protocol Version

TLS is the abbreviation for Transport Layer Security, is a successor to SSL protocol. TLS ensures privacy and data integrity by providing secure communication between applications. The communication link is secured by encrypting the data transferred using a cryptographic algorithm. The main objectives of the TLS protocol are to offer: Cryptographic security, Interoperability, Extensibility, and Relative efficiency [60].

TLS has constantly evolved with every new attack and vulnerability identified. To date, four different TLS versions have been released, with the latest and most secured being the TLSv1.3. TLSv1.0 and TLSv1.1 are out-dated protocol versions as they do not support state-of-the-art cryptographic algorithms and possess numerous vulnerabilities that can be easily compromised. Internet Explorer 11 disabled these two versions from September 8, 2020 [61], and Chrome has already started to deprecate these versions in most of its products. TLSv1.2 has enjoyed wide adoption for almost a decade [62]. However, ever since the release of TLSv1.3, after going through 24 drafts, it has been significantly growing. Global players like Akamai, Amazon, Cloudflare adopted this version, which results in such a high deployment. An additional variant that is obtained while scanning is TLS_NULL. This Null cipher offers authenticity and integrity but no confidentiality.

Appendix B

Details about the scans

B.1 Generic Overview of the Scan

Scan	*Scan 1*	*Scan 2*	*Scan 3*	*Scan4*	*Scan5*	*Scan6*
Type	Active	Active	Active	Active	Active	Active
Port	80	80	80	80	80	80
Epoch Time	1595469983	1597812841	1599453211	1600601328	1602036129	1602575398
Date	July 23, 2020	August 19, 2020	September 7, 2020	September 20, 2020	October 7, 2020	October 13, 2020
Total Host	63,245,221	62,255,730	62,118,306	59,552,881	61,910,562	62,018,486
Total Announced Prefix	492,300	493,312	494,173	487,878	497,389	499,346
Total Allocated Prefix	366,485	367,036	367,703	364,576	370,004	371,058
Total ASN	60,717	60,794	60,878	60,404	61,268	61,411

Table B.1: General Overview of HTTP portocol

Scan	***Scan 1***	***Scan 2***	***Scan 3***	***Scan4***	***Scan5***
Type	Active	Active	Active	Active	Active
Port	443	443	443	443	443
Epoch Time	1585720500	1595310742	1597298851	1599288337	1600331255
Date	April 1, 2020	July 21, 2020	August 13, 2020	September 5, 2020	September 17, 2020
Total Host	53,259,200	53,720,295	52,454,494	52,134,641	52,818,060
Total Announced Prefix	463,122	464,206	464,457	466,160	468,018
Total Allocated Prefix	351,224	353,285	353,344	354,566	355,848
Total ASN	60,846	61,831	61,813	61,971	62,086
Host supporting TLS_Null	13,031,045	14,744,744	13,411,043	11,660,067	12,964,400
Host supporting TLSv1.0	4,385,821	3,785,076	3,707,711	3,834,115	3,578,834
Host supporting TLSv1.1 TLSv1.1	282,420	204,111	202,058	253,037	216,526
Host supporting TLSv1.2	27,840,581	26,230,288	25,930,999	25,947,234	25,463,760
Host supporting TLSv1.3	7,719,333	8,756,076	9,202,683	10,440,188	10,594,540

Table B.2: General Overview of TLS portocol

Scan	*Scan 1*	*Scan 2*	*Scan 3*	*Scan4*	*Scan5*
Scan	scan 1	scan 2	scan 3	scan 4	scan 5
Type	Active	Active	Active	Active	Active
Port	53	53	53	53	53
Epoch Time	1595996550	1597885251	1599543508	1601879119	1602465816
Date	July 29, 2020	August 20, 2020	September 8, 2020	October 5, 2020	October 12, 2020
Total Host	10,708,736	10,761,941	10,456,291	10,473,497	10,646,159
Total Announced Prefix	281,827	279,431	280,168	281,614	280,292
Total Allocated Prefix	224,379	222,770	223,244	224,571	224,570
Total ASN	45,237	45,001	45,170	45,559	45,831

Table B.3: General Overview of DNS portocol

B.2 Scans Used to Represent Days in Stability Test

Sampled Using	*Evaluated Against*	*Days*
Scan 5 October	Scan 6 October	6
Scan 3 September	Scan 4 September	13
Scan 3 September	Scan 5 October	30
Scan 1 July	Scan 4 September	59
Scan 1 July	Scan 6 October	82

Table B.4: Combination of Data used to represent the time interval for HTTP

Sampled Using	*Evaluated Against*	*Days*
Scan 4 September	Scan 5 September	12
Scan 3 August	Scan 5 September	35
Scan 2 July	Scan 5 September	58
Scan 1 April	Scan 2 July	111
Scan 1 April	Scan 3 August	134
Scan 1 April	Scan 5 September	169

Table B.5: Combination of Data used to represent the time interval for TLS

Sampled Using	*Evaluated Against*	*Days*
Scan 4 October	Scan 5 October	7
Scan 2 August	Scan 3 September	19
Scan 3 September	Scan 4 October	27
Scan 2 August	Scan 4 October	46
Scan 2 August	Scan 5 October	53
Scan 1 July	Scan 5 October	75

Table B.6: Combination of Data used to represent the time interval for DNS

Appendix C

Plots for all three protocol

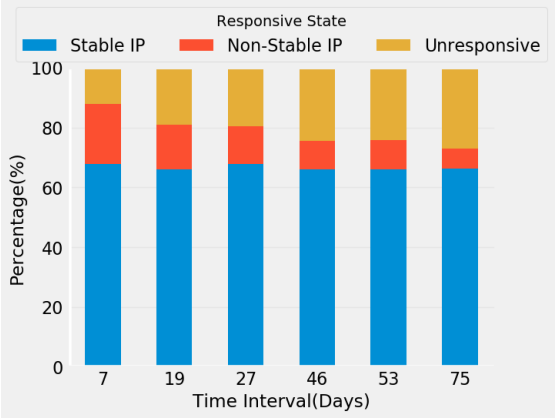


Figure C.1: DNS Relative Responsiveness Characteristics over Time

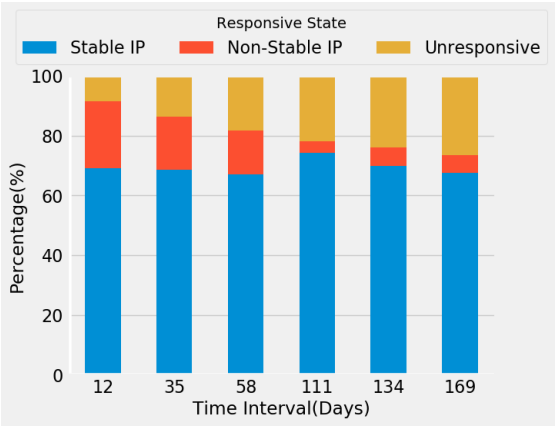
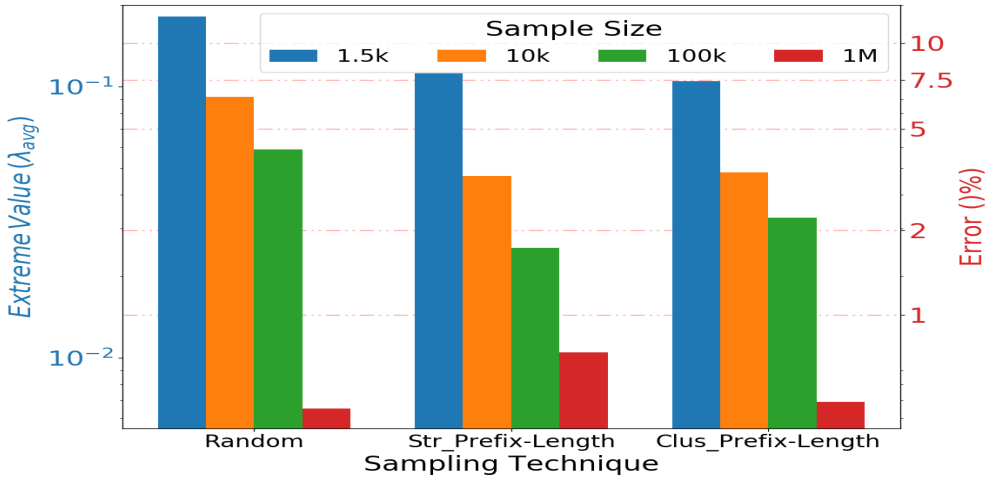
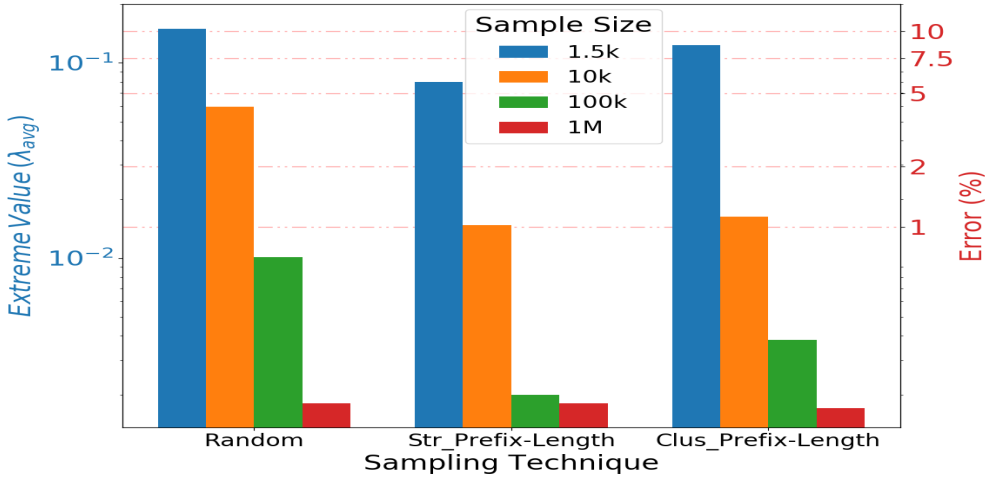


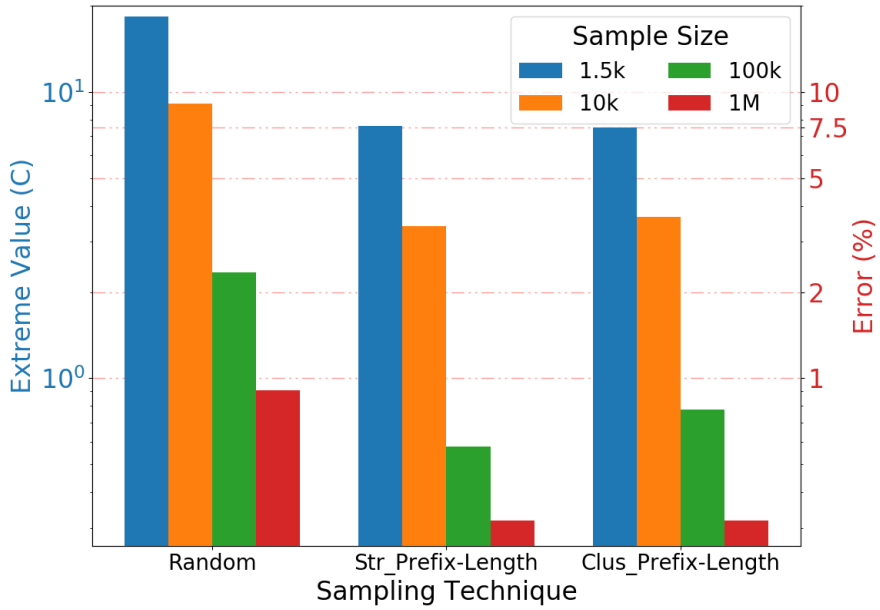
Figure C.2: TLS Relative Responsiveness Characteristics over Time



(a) All 25 Prefix Length

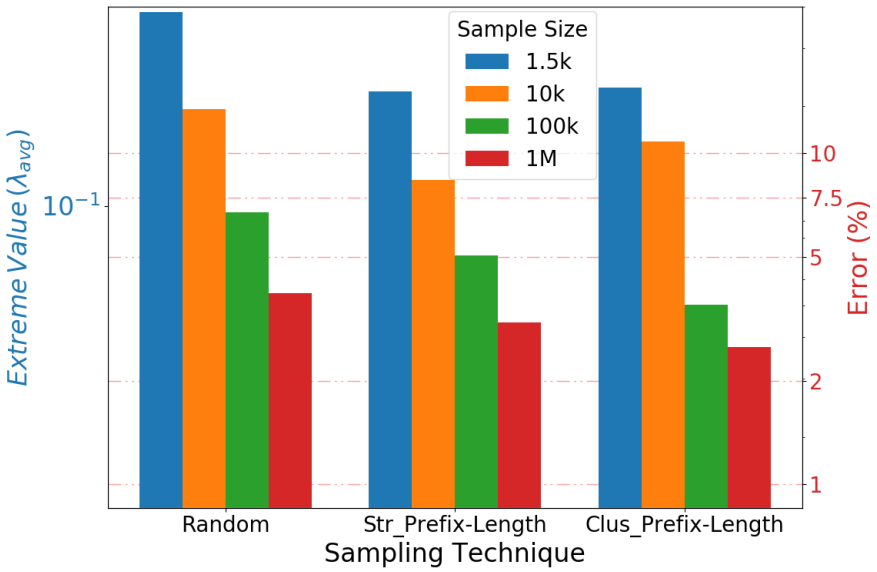


(b) Routable Prefix Length

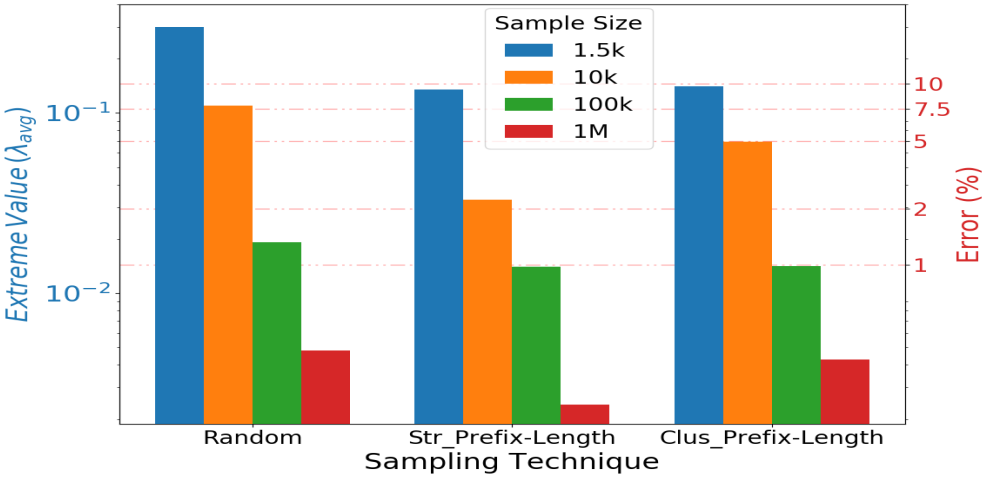


(c) Only /24 Prefix Length

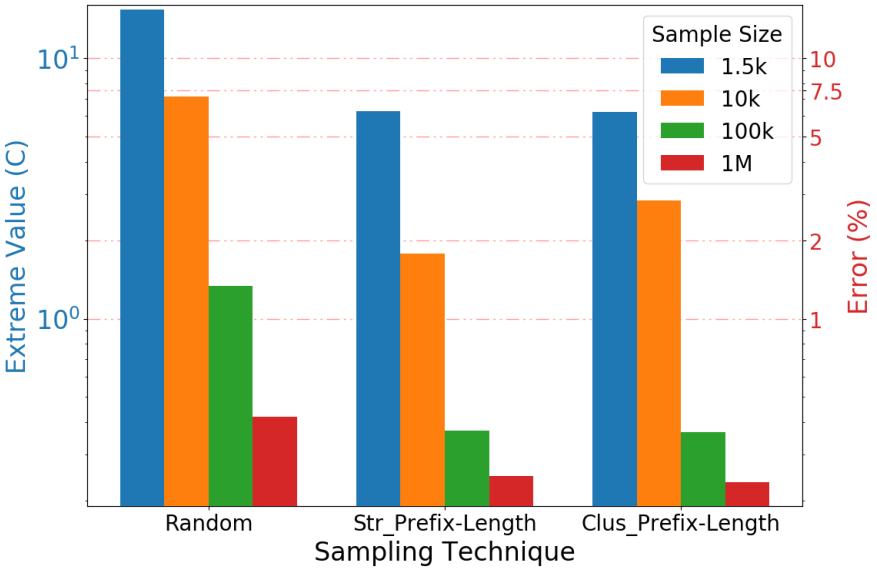
Figure C.3: Performance of Sampling Techniques based on TLS Deployment across Prefix Length



(a) All 25 Prefix Length

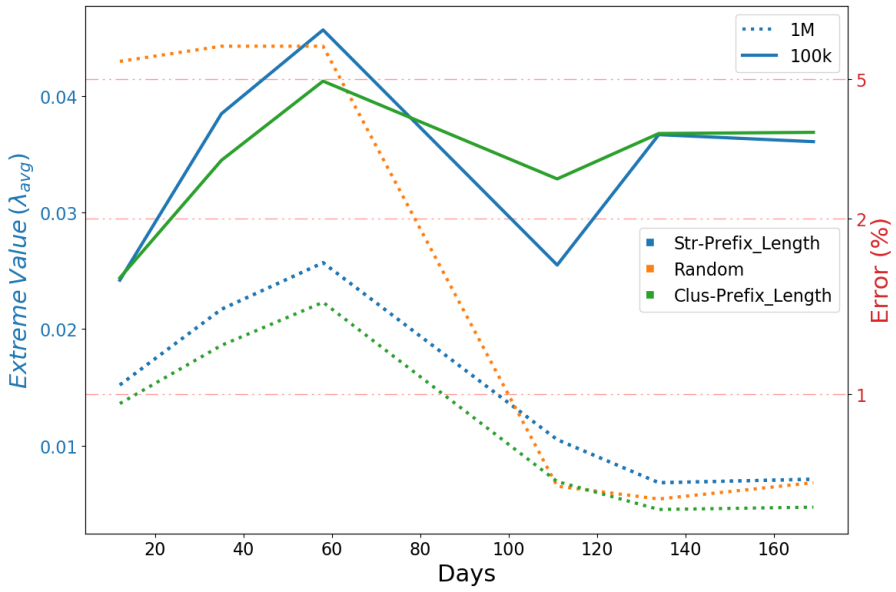


(b) Routable Prefix Length

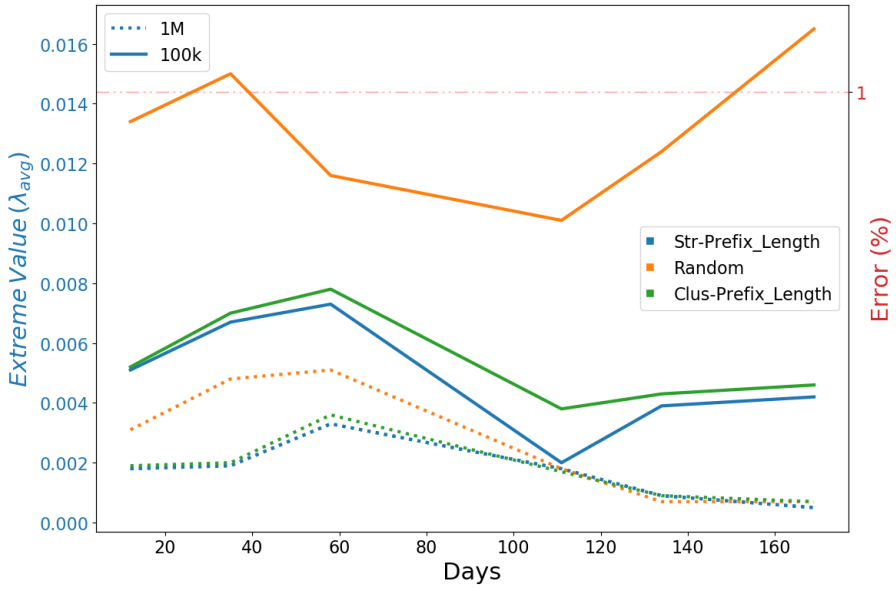


(c) Only /24 Prefix Length

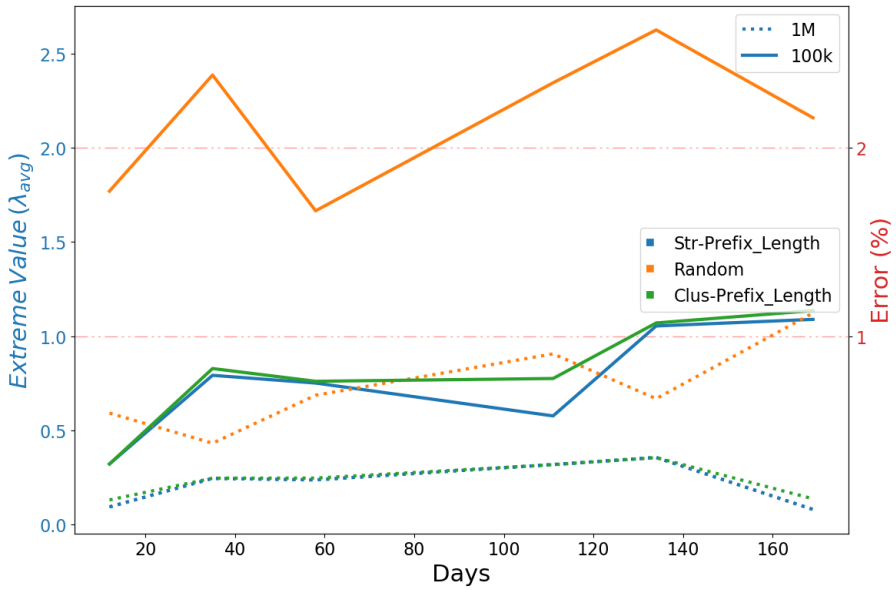
Figure C.4: Performance of Sampling Techniques based on DNS Deployment across Prefix Length



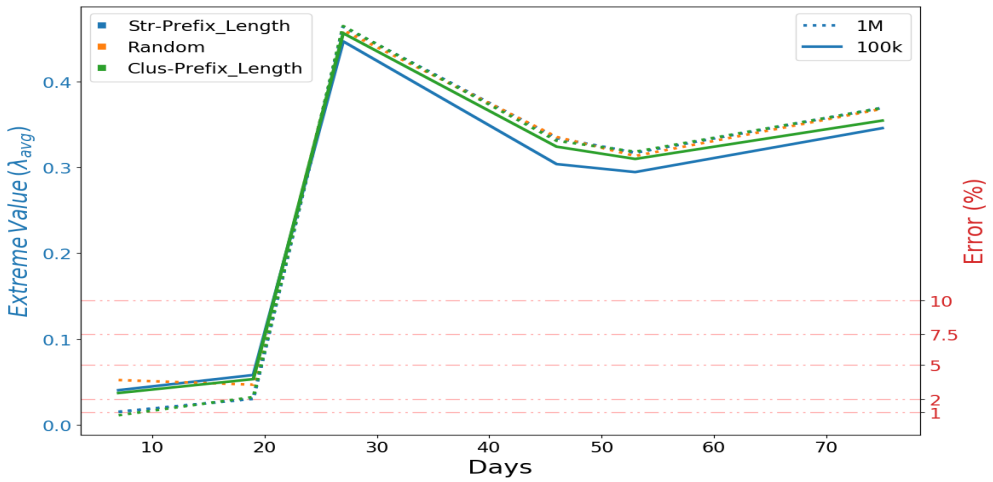
(a) All 25 Prefix Length



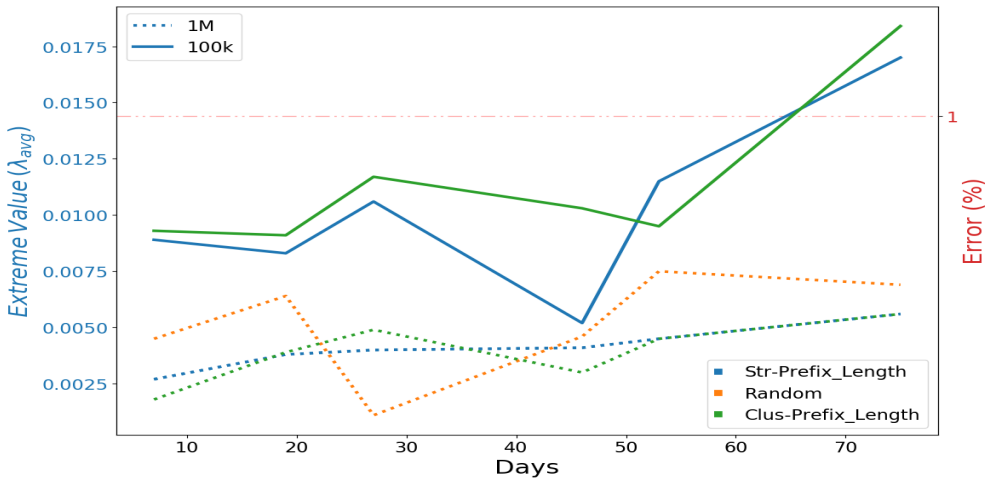
(b) Routable Prefix Length



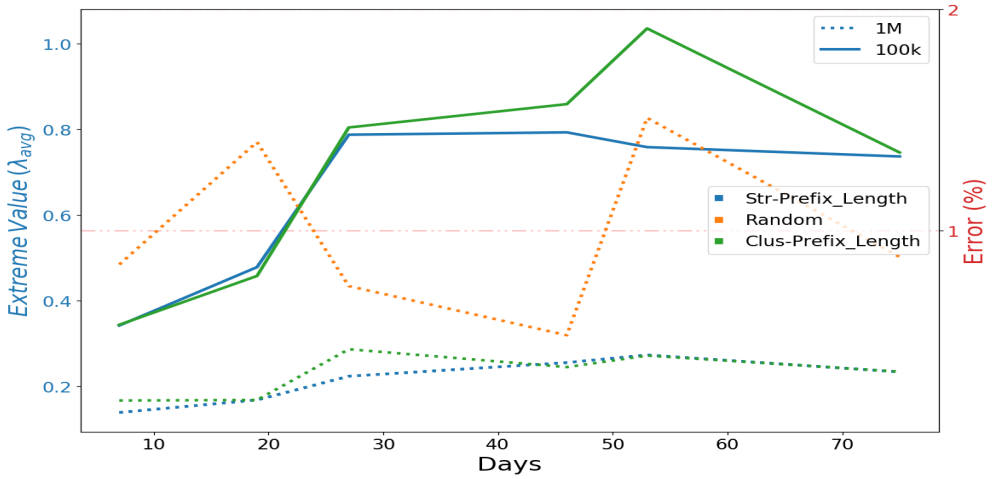
(c) Only /24 Prefix Length



(a) All 25 Prefix Length

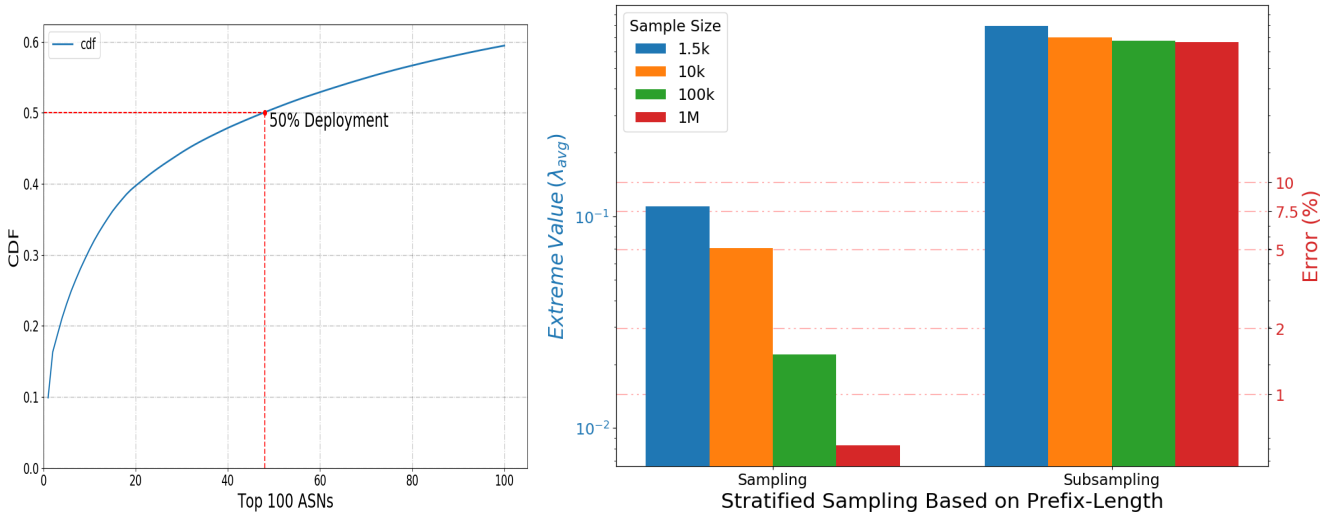


(b) Routable Prefix Length



(c) Only /24 Prefix Length

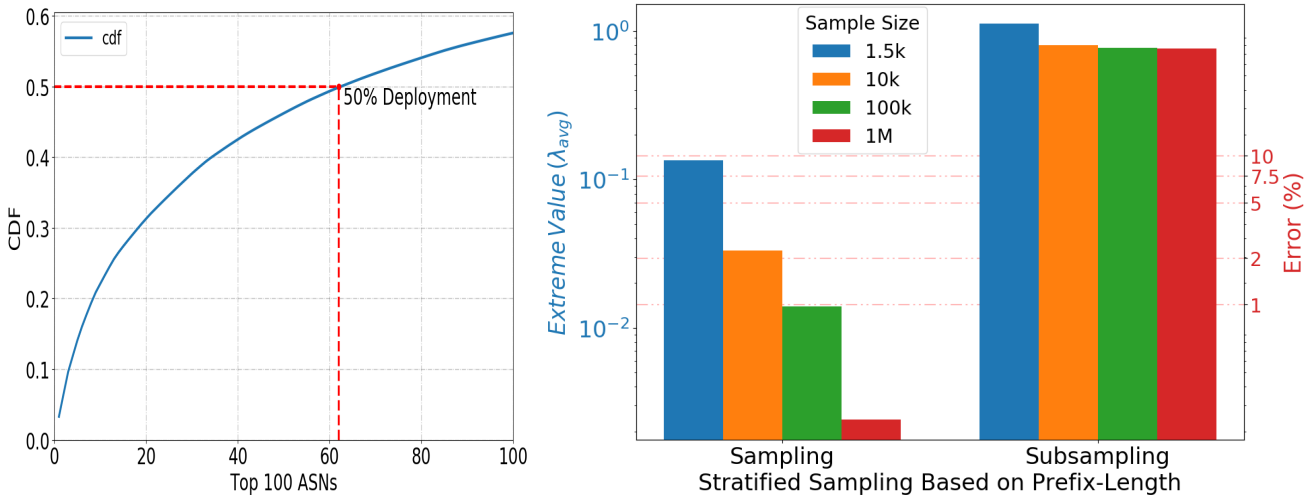
Figure C.6: Longitudinal Performance of Sampling Techniques on DNS deployment across on Prefix Length



(a) CDF Curve on the Deployment of Top 100 ASNs

(b) Conventional Sampling vs Two-Stage Subsampling

Figure C.7: HTTP Subsampling Based on Internet Centralization



(a) CDF Curve on the Deployment of Top 100 ASNs

(b) Conventional Sampling vs Two-Stage Subsampling

Figure C.8: DNS Subsampling Based on Internet Centralization