ABSTRACT

Title of dissertation: IMPROVING ACCURACY AND EFFICIENCY OF NETWORK MEASUREMENT BY IDENTIFYING HOMOGENEOUS IPV4 ADDRESSES

Youndo Lee, Doctor of Philosophy, 2017

Directed by: Professor Neil Spring
Department of Computer Science

Active Internet measurement relies on responses to active probes such as ICMP Echo Request or TCP SYN messages. Active Internet measurement is very useful in that it enables researchers to measure the Internet without privileged data from ISPs. Researchers use active measurement to study Internet topology, route dynamics and link bandwidth by sending many packets through selected links, and measure RTTs and reliability through probing many addresses. A fundamental challenge in active measurement design is in allocating and limiting measurement traffic by carefully choosing where measurements are sent and how many samples are taken per measurement. It is important to minimize measurement loads because heavy measurement traffic may appear malicious. If network operators consider measurement traffic as attacks, then they can blacklist the sources of measurement

traffic and thus affect the completeness and accuracy of the measurement. Another challenge of active measurement is that biases can occur due to no responses from or biased selection of destinations. Biases can cause misleading conclusions and thus should be minimized.

In this dissertation, I develop a general approach to reducing measurement loads and biases of active Internet measurement based on the insight that they can be reduced by letting Internet addresses represent larger aggregates. I first develop a technique that identifies and aggregates topologically proximate addresses. The technique called Hobbit compares traceroute results to measure topological proximity. Hobbit deals with load-balanced paths that can cause incorrect inferences of topological proximity by distinguishing between route differences due to load balancing and due to distinct route entries. Hobbit also makes a unique contribution that it can aggregate even discontiguous addresses. This contribution is important in that fragmented allocations of IPv4 addresses are common in the Internet.

I apply Hobbit to IPv4 addresses and identify 0.51M aggregates of addresses (i.e. Hobbit blocks) that contain 1.77M /24 blocks. I evaluate the homogeneity of Hobbit blocks using RTTs and show that Hobbit blocks are as homogeneous as /24s even though their sizes are generally larger than /24s. I then demonstrate that Hobbit blocks improve the efficiency of Internet topology mapping by comparing strategies that select destinations from Hobbit and /24 blocks. I also quantify the efficiency improvement of latency estimation that can be achieved by using Hobbit blocks. I show that Hobbit blocks tend to be stable over time and analyze the measurement cost of Hobbit block generation.

I finally demonstrate that Hobbit blocks can improve the representativeness of network measurement. I develop a methodology that measures the representativeness of measurement and show that active Internet measurement may not be representative even if the entire IPv4 space is probed. By using Hobbit blocks, I adapt weighting adjustment, which is a common bias correction technique in surveys, to active Internet measurement. I evaluate the weighting adjustment using various kinds of samples and show that the weighting adjustment reduces biases in most cases. If Hobbit blocks are given, the weighting adjustment incurs no measurement cost. I make Hobbit blocks publicly available and update them every month for researchers who want to perform weighting adjustment or to improve the efficiency of network measurement.

IMPROVING ACCURACY AND EFFICIENCY OF NETWORK MEASUREMENT BY IDENTIFYING HOMOGENEOUS IPV4 ADDRESSES

by

Youndo Lee

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2017

Advisory Committee:
Professor Neil Spring, Chair/Advisor
Professor Bobby Bhattarcharjee
Professor Rama Chellappa
Professor David Levin
Professor David Mount

Acknowledgments

I am truly grateful to my advisor, Neil Spring. He encouraged me to work on hard problems and gave me very insightful comments nearly every time we talked. I could learn the ways of finding and evaluating the significance of research problems and general approaches to conducting research by reflecting on his comments. I could also realize the importance of brevity and clarity in writing and presentations. Without the lessons I learned from him, this dissertation would never be possible.

Dave Levin was a great collaborator. He taught me the difference between the goals of engineers and researchers. I was close to an engineer, and after talking to him, I could adjust my attitude and eventually identify some research problems that I have addressed in this dissertation.

Bobby Bhattacharjee has been an inspiring teacher and advisor. He raised my expectations of answering questions in presentations and data logging practice. He also gave me useful comments on my thesis proposal draft.

Rama Chellappa, Amol Deshpande and David Mount gave me helpful feedback on my thesis proposal and dissertation. I thank them for their services and time.

I owe my gratitude to my parents. They have been very supportive of pursuing a PhD and giving me constant love. My wife, Hyeeun Chung, has shown amazing devotion to me and my daughter. Words cannot express my gratitude and love to her.

Aaron Schulman taught me basic tools and methodologies for network measurement research. Ramakrishna Padmanabhan and Zhihao Li gave me detailed feedback on my dissertation presentation. I would also like to thank other colleagues, Yunus Basagalar, Vassilios Lekakis, Matt Lentz, James Litton, Stephen Herwig, Mahmoud F. Sayed, Katura Harvey and Richard Roberts.

Table of Contents

List of Tables

List of Figures

Chapter 1: INTRODUCTION

Active measurement of the Internet, which is a major field of network measurement, probes Internet addresses by active probes such as ICMP, TCP and UDP probes. Active Internet measurement enables researchers to measure or track network attributes and events without access to privileged information. Many researchers rely on active Internet measurement to study the Internet's core [1–9] as well as edge networks [10–16]. The results of active Internet measurement provide information for diverse networks in the Internet and thus can have significant implications not only for researchers but also for Internet Service Providers (ISPs) that typically do not have information for other ISP networks. Although active Internet measurement is an essential method, it has a couple of issues.

First, active Internet measurement can incur heavy measurement loads because it sends probes to each of destination Internet addresses. Heavy measurement loads can lead to high use of measurement monitor resources or slow measurement speed. The resource usage in measurement monitors should be minimized because many public monitors such as planetlab nodes [17] and residential RIPE Atlas probes [18] have limited resources. Slow measurement speed also needs to be addressed because network properties such as routes and latency can change during measurement time.

Another issue with heavy measurement loads is that, when ingress routers of destination networks receive heavy measurement traffic, they may consider measurement traffic as malicious scans or attacks and thereby block the traffic.

Second, since active Internet measurement extracts information from responses to active probes, it tends to over-represent hosts that always respond to probes and under-represent hosts that intermittently respond. The issue is that the response rates of Internet hosts are highly correlated with geographic locations [16] and network prefixes [19]. This suggests that active Internet measurement results may be biased toward some geographic locations and networks. In addition, the locations of measurement monitors can also affect the representativeness of measurement results in that networks near the monitors are likely to be over-represented. Measurement monitors are typically located in the institutions of researchers who perform measurement and the institutions are mostly in the US and Europe. Popular public monitors such as planetlab nodes and RIPE Atlas probes are also concentrated in the US or Europe. Therefore, it is an important issue to develop a technique that reduces the bias in active Internet measurement.

In this dissertation, I address heavy measurement loads and potential bias of active Internet measurement, with the insight that they can be alleviated by letting Internet addresses represent larger aggregates of addresses. I develop a set of techniques for identifying and exploiting homogeneous groups of IPv4 addresses and demonstrate that the techniques can reduce measurement loads and bias of active Internet measurement. My work in this dissertation can be directly used in active Internet measurement and thus can have an impact on research and practice that perform active Internet measurement.

1.1 Challenges

One may think probing a large number of addresses (and thus incurring heavy measurement loads) is not an issue in that an optimized scanning tool called ZMap [20] probes the entire public IPv4 addresses in about 10.5 hours (according to the ZMap ICMP Echo Request dataset [21]). ZMap is very useful for reducing measurement execution time, but does not address all the challenges caused by heavy measurement loads. First of all, ZMap traffic may appear to be malicious attacks because ZMap generates traffic at very high speeds. Also, measurement execution time can still be a challenge. ZMap uses a single probe for each destination by default but several measurement studies require multiple probes for a single destination. Studies that use traceroute [2–6, 8, 9] are representative examples. It is hard to generalize the number of probes required for traceroute to a single destination because it largely depends on the hop counts between source and destination. But, according to the traceroutes of randomly sampled 100k destinations measured on a single machine at UMD using an exhaustive mode of Paris-traceroute (also known as MDA) [22], the average number of probes for each destination is 141. This implies that measurement execution time can be very long (2 months) even with ZMap. These challenges can be naturally addressed by a reduction in measurement loads.

Measurement loads can be reduced by sampling of Internet addresses. Sampling, however, is a challenging problem. Samples should represent diverse networks that have various networking technologies, administrative policies and geographic locations. The challenge is that Internet addresses are very unevenly distributed across countries and administrative domains. For example, while the US has about 36% of IPv4 addresses, most of other countries have less than 1% of the addresses [23]. This suggests that simple random sampling may generate unrepresentative samples that do not contain the addresses that are in the minority. An increase in sample size will improve the representativeness of a sample but it will also increase measurement loads.

Stratified sampling can deal with uneven distribution of Internet addresses. However, stratified sampling requires a grouping of Internet addresses. A traditional grouping approach is to aggregate addresses by network prefixes such as BGP and /24 prefixes. This method is simple and practical but only aggregates addresses that are numerically adjacent. Given that route aggregation and fragmented allocations of addresses are common in the Internet [24], the aggregation of numerically adjacent addresses may not be appropriate. BGP prefixes can comprise diverse addresses and discontiguous /24 prefixes may be (topologically) closer to each other than contiguous /24 prefixes. To address these challenges, I exploit topological information obtained by traceroute rather than relying on network prefixes in identifying the groups of Internet addresses.

1.2 Thesis

In this dissertation, I defend the following thesis: *IPv4 addresses can be grouped into homogeneous blocks that enable researchers to improve efficiency and accuracy of active Internet measurement.* I define "homogeneous blocks" to be aggregates of IPv4 addresses that are topologically proximate. I define "improving efficiency" as reducing the number of probes required for obtaining the same amount of information, and "improving accuracy" as enhancing the representativeness of measurement results from samples without increasing sample sizes.

I support the thesis based on the insight that topologically proximate addresses are likely to have similar properties and the aggregates of topologically proximate addresses can be identified by using traceroute. My insight enables stratified sampling of IPv4 addresses (that draws sample points from each aggregate). I expect the stratified sampling will improve efficiency of active measurement compared to random sampling. The stratified samples can also be more representative than random samples (in that the aggregates of topologically proximate addresses are likely to widely vary in size).

The aggregates of topologically proximate addresses can also be used for post bias correction. My insight is that I can leverage weighting adjustment [25] that is one of the most common bias correction techniques in surveys, and that the aggregates of topologically proximate addresses can be used as input to weighting adjustment. Active Internet measurement and surveys are similar in that they obtain information about individual units (i.e., people and Internet addresses) by sending requests (i.e., questionnaires and active probes). The causes of biases are also similar. Samples can be biased or no responses may lead to bias. This suggests that it is possible to adapt weighting adjustment to active Internet measurement. Weighting adjustment requires a grouping of individuals (i.e., IPv4 addresses) like stratified sampling. I support that the aggregation of topologically proximate addresses is a proper grouping for weighting adjustment.

1.3 Contributions

A technique for identifying the aggregates of topologically proximate IPv4 addresses. I develop a technique that measures topological proximity of IPv4 addresses and aggregates proximate addresses. The technique called Hobbit, distinguishes between route differences due to load balancing and distinct route entries and can even aggregate discontiguous addresses. These are unique characteristics that can greatly improve the quality of the aggregates in that load-balancing is prevalent in the Internet as well as fragmented allocations of IPv4 addresses. I verify the correctness of Hobbit by using information from various sources including Whois databases, reverse DNS names, P2P crawlers and latency characteristics.

Identification and analysis of the aggregates of topologically proximate addresses. I identify 0.51M blocks of topologically proximate addresses that cover 1.77M /24 prefixes (and thus 256*1.77M individual addresses) by using Hobbit. The blocks, which I call Hobbit blocks, can be used for improving the efficiency of network measurement. I analyze the measurement cost of topology discovery that uses Hobbit blocks in selecting destinations and show that Hobbit blocks improve the efficiency of topology discovery (compared to an existing approach). I quantify the similarity of RTTs of the addresses within the same Hobbit blocks. Hobbit blocks are highly correlated with RTTs and thus can be used for latency prediction. I actually show that Hobbit blocks improve the coverage of an existing latency prediction technique, resulting in improved efficiency. I analyze the similarity of geographic locations within Hobbit blocks. Hobbit blocks are mostly homogeneous in geographic locations. This suggests that Hobbit blocks can help active Internet measurement to cover more diverse geographic locations, in other words, improve the representativeness. (I actually use Hobbit blocks in developing a technique for improving the representativeness of active Internet measurement.) I make Hobbit blocks publicly available.

A technique that improves the representativeness of network measurement. I motivate, develop and evaluate a technique for improving the representativeness of active Internet measurement. I develop a methodology that quantifies the representativeness of network measurement results. With this methodology, I demonstrate that active measurement results may not be representative even when the entire public IPv4 addresses are probed, and thus a technique for improving the representativeness is needed. I develop a technique that improves the representativeness of measurement results by adapting weighting adjustment to active Internet measurement. I identify key factors that influence the performance of the technique and evaluate the performance with various combinations of the factors. The results show that the technique improves representativeness in most cases.

1.4 Organization

In Chapter 2, I describe related work with a focus on existing approaches to identifying aggregates of IPv4 addresses and their applications. In Chapter 3, I present a technique for identifying topologically proximate addresses (Hobbit) and describe the blocks generated by using Hobbit (Hobbit blocks) focusing on their sizes and the characteristics of largest and smallest Hobbit blocks. In Chapter 4, I describe ways of using Hobbit blocks to improve the efficiency of network measurement and demonstrate that Hobbit blocks improve the efficiency of topology discovery and latency estimation. In Chapter 5, I detail a methodology for quantifying the representativeness of measurement results and present a description and evaluation of a technique that improves the representativeness of active Internet measurement. In Chapter 6, I conclude with a summary of contributions and implications for future research.

Chapter 2: RELATED WORK

2.1 Identifying the aggregates of addresses

The IP prefixes advertised through BGP provide useful information about topological proximity. Researchers often group IP addresses by their BGP prefixes and treat the groups as a unit for reducing measurement load [3] or inferring the properties of Internet addresses such as geographic locations [26] and access network types [27]. I also aggregate addresses but use the last IP hop routers as a criterion. My work is also unique in that I use the aggregates of addresses for improving the accuracy of network measurement as well as the efficiency.

BGP prefixes may be aggregates of smaller (i.e., more specific) prefixes. This motivates researchers to disaggregate large BGP prefixes into small prefixes, typically /24s, and use the addresses that have common /24 prefixes as a unit [15, 16, 28, 29]. Although the blocks identified by Hobbit, i.e., Hobbit blocks are likely to be smaller than BGP blocks, Hobbit is different from the disaggregation based on network prefixes in that Hobbit aggregates even discontiguous addresses. In addition, I show that the use of Hobbit blocks improves the efficiency of Internet topology mapping compared to when using /24s (Section 4.1).

The iPlane [3] clusters network interfaces by geographic locations and AS numbers, and performs measurement at the granularity of clusters in order to reduce measurement load. The iPlance also uses BGP prefixes in selecting destination addresses for topology mapping. My approach in this dissertation is similar to iPlane in that I also aggregate addresses with an purpose of reducing measurement loads. However, the aggregation by last-hop routers yields better homogeneity than the aggregation by geographic locations and BGP prefixes (as will be shown in Section 4.3.1).

RTT measurements are often used for determining [26, 30–35] or validating [36, 37] the geographic locations of Internet hosts. The intuition behind the use of RTT measurement for geolocation is that the addresses that have similar RTTs will be geographically proximate. Hobbit blocks also can be used for geolocation because the addresses within the same Hobbit blocks are likely to be geographically co-located (Section 4.3.2). A drawback of using RTT measurements for geolocation is that RTTs can vary widely depending on queuing delays [11,38,39], IP paths [40] and connection types [41]. On the other hand, last hop routers of IPv4 addresses tend to be very stable (Section 4.1.3).

2.2 Assessing the aggregates of addresses

Freedman et al. [24] have measured the geographic locality of BGP prefixes and found that many prefixes comprise geographically diverse addresses. Chen et al. [42] have also measured the geographical locality of the clusters generated by using local domain name servers as a criterion. I also assess the homogeneity of aggregates of addresses (generated by using BGP prefixes and geographic locations). But, I measure the correlation of RTTs rather than geographic co-locality (Section 4.3.1).

Gharaibeh et al. [35] have also quantified the geographic co-locality of IP addresses. They have focused on disaggregated BGP prefixes, that is, /24 blocks. They have inferred the co-locality by clustering the addresses that have similar RTTs and counting the number of distinct clusters within /24 blocks. Hobbit also measures the homogeneity of /24 blocks but Hobbit uses last hop routers as a metric. Last-hop routers are more likely to be stable than RTTs (as I describe in Section 2.1).

2.3 Improving the representativeness of network measurement

I use Hobbit blocks in adapting weighting adjustment to network measurement. The weighting adjustment improves the representativeness of network measurement by post-error correction. There have been alternative approaches to improving the representativeness.

One of the main causes of unrepresentativeness is sampling bias. Many studies have addressed the sampling bias issue by developing techniques that obtain representative samples mostly focusing on P2P networks [43, 44] and online social networks [45–48]. My approach is different from these work in that I use post-processing (i.e., weighting adjustment) rather than creating unbiased samples. Weighting adjustment and generating representative samples complement each other because the performance of weighting adjustment is maximized with representative samples, and that active measurements may not be representative even with representative samples (due to no response) and weighting adjustment can improve the representativeness (Section 5.3.3).

Kandula et al. [49] have developed a post-correction technique called Broom that corrects bias in network path measurements. Whereas Broom focuses on biased selection of source addresses, I focus on unrepresentativeness caused by no response or biased selection of destination addresses. The biases in sources and destinations can be dealt with separately and thus Broom and my study can complement each other. He et al. [50] have developed a framework that supplies information missing from existing datasets. Weighting adjustment also can be viewed as supplying information missing from measured data. However, whereas weighting adjustment is a general technique for improving the representativeness, their framework is tailored to the detection of missing AS links.

2.4 Predicting latency between Internet hosts

I use Hobbit blocks for latency estimation (that is, to estimate the RTT between two Internet hosts without direct measurement). Latency estimation has be been studied over a decade. A common approach is to build a network coordinate system that assigns each host synthetic coordinates in a coordinate space such that the distance in the space corresponds to the RTT between the hosts [51–58].

There also have been techniques that do not employ network coordinate systems. King [59] exploits local DNS servers and Ting [60] uses Tor [61] protocol. All these existing techniques apply to certain subsets of IP addresses. The network coordinate based techniques can only predict RTTs between addresses participating the coordinate system. King is only applicable to addresses of which local DNS servers are open recursive resolvers and Ting can only predict RTTs between Tor relays. Hobbit blocks complement the existing techniques by extending their coverage (as I will show in Section 4.2.2).

2.5 Measuring and dealing with load-balancing

Several studies have observed path diversity due to load balancing. My work is relevant to these studies in that I also have observed and dealt with the path diversity. Augustin et al. [4] have observed that 39% of source destination pairs traversed per-flow or per-packet load balancers, and 70% traversed per-destination load-balancers. Flach et al. [7] have quantified violations of destination-based forwarding due to load-balancing. Pelsser et al. [40] observed a significant difference in latency between flows for the same source destination pair, which implies the existence of per-flow load-balancers.

Paris-traceroute MDA [22] enumerates all load-balanced paths between source-destination pairs, i.e., enumerates all "per-flow" load-balanced paths. Hobbit uses Paris-traceroute MDA to enumerate per-flow load-balanced paths. MDA can be also used for enumerating per-destination load-balanced paths. However, Hobbit does not use MDA for enumerating per-destination load-balanced paths because MDA just enumerates all the paths towards /24s based on the assumption that the paths towards /24s will be the same unless they are per-destination load-balanced paths. Hobbit does not assume the homogeneity of /24s and concludes that the paths are per-destination load-balanced paths only when their relationships are non-hierarchical.

Chapter 3: IDENTIFYING HOMOGENEOUS ADDRESSES

In this chapter, I describe a technique for identifying the aggregates of homogeneous IPv4 addresses. I first measure the homogeneity of /24 blocks (i.e., the aggregates of addresses having common /24 prefixes) and then identify larger blocks by aggregating homogeneous /24 blocks. /24 blocks are used as a unit in several measurement studies and systems. An Internet outage detection system called Trinocular [15] tracks outages for /24 blocks, and a recent study on the availability of Internet hosts have focused on the availability of /24 blocks [16]. The IPv4 topology dataset of CAIDA [28] is constructed by probing the destinations randomly chosen from each routed /24 prefix. The EDNS-Client-Subnet DNS extension [29] strongly encourages recursive resolvers to truncate the IPv4 addresses of users to 24 bits, for the purpose of protecting the privacy of users. Whereas these systems use /24 blocks with an implicit assumption of the homogeneity of /24 blocks, I validate the homogeneity of /24 blocks and only consider homogeneous blocks.

I focus on topological proximity in estimating the homogeneity. Topological proximity is closely related with the operations of several existing measurement systems. For example, if the addresses within /24s are topologically distant, they are unlikely to have identical traceroute results (thus affecting topology discovery by CAIDA), concurrent outages (affecting Trinocular) and identical corresponding front-end servers (affecting the EDNS extension).

The measurement of homogeneity in terms of topological proximity may seem to be a trivial problem that can be simply solved by using traceroute. However, due to the prevalence of load-balancing, comparing the traceroute results is not straightforward. I detail the challenges in measuring the topological proximity and present a technique that identifies homogeneous /24s in the presence of load-balancing. By applying the technique, I actually find homogeneous address blocks and describe the characteristics of the blocks.

3.1 Measuring topological proximity

A straw-man proposal for measuring the homogeneity of /24 blocks is to obtain IP-level routes of all the addresses within /24 and conclude that a /24 is homogeneous if all the IP-level routes are identical. An underlying assumption of this approach is that the routes towards co-located addresses are identical. However, in today's Internet where path diversity due to load-balancing is prevalent, this is not true for many addresses. Even probes between the same source-destination pairs often take different paths [4]. I first describe how to deal with load-balanced paths.

3.1.1 Paris-traceroute is helpful but not enough

Paris-traceroute, which is a variant of traceroute, has been proposed to correct inaccurate inferences of paths due to load-balancing. It tunes the values of the packet header fields that affect the path selection by load-balancers, so that all probes towards a destination follow the same path. Paris-traceroute can also be extended to a tool[1] that enumerates all paths between a source-destination pair.

I use Paris-traceroute MDA in comparing (IP-level) routes of different addresses to prevent from falsely classifying identical routes as being different. If the numbers of routes towards destinations are more than one, identifying a single route for each destination may cause false classifications. For example, if destinations A and B both have routes $\{r_1, r_2\}$, and I find only a single route r_1 for A and r_2 for B, then A and B will appear to have different routes which is not true. To prevent this from happening, I enumerate all routes using Paris-traceroute MDA and compare the sets of routes.

Based on the methodology described above, I perform a preliminary analysis on the homogeneity of /24 blocks. I first identify active IPv4 addresses using ZMap ICMP Echo Request scan dataset [21, 62]. This dataset is generated by sending ICMP Echo request probes to all public IPv4 addresses, and recording the reply messages (if exist). I only consider IPv4 addresses that responded with ICMP Echo reply messages to be active. Given the list of active addresses, I select an active address from each /26 block while excluding /24 blocks that have no active address in any of the /26s within them. In other words, I only select /24s that have at least one active address in every /26 block within them to increase the confidence

---

[1] The extended version is called Multipath detection algorithm (MDA). In this dissertation, I use the term "Paris-traceroute MDA" because MDA is often considered as a subcomponent of Paris-traceroute.

of my result to represent /24s not /25s nor /26s. For each chosen address, I enumerate all the routes between a source located at UMD and the chosen address. I consider that addresses have identical routes if they share at least one route. A /24 block is regarded as being homogeneous if all of the (four) addresses within the block have identical routes. To my surprise, 88% of the /24 blocks were *heterogeneous*. Considering that I address per-flow load-balancing using Paris-traceroute MDA and that I am generous in determining whether addresses have identical routes by requiring only one route to be identical, 88% is unexpectedly high. With a doubt that ICMP rate limiting can be a confounding factor, I try to eliminate the effect of ICMP rate limiting. I use unresponsive hops as wildcards that can represent any address in comparing routes. For example, routes <A.A.A.A, B.B.B.B, C.C.C.C>, <A.A.A.A, *, C.C.C.C> and <*, B.B.B.B, C.C.C.C> are all considered to be identical where * represents unresponsive hop. This change to the route comparisons reduces the percentage of heterogeneous /24 blocks, but very slightly: The percentage of heterogeneous blocks decreases to 87% from 88%.

3.1.2 Per-destination load-balancing matters

The unexpectedly high ratio of heterogeneous /24 blocks implies that there can be other confounding factors than per-flow load-balancing addressed by Paris-traceroute MDA and ICMP rate limiting. One possibility is that load balancing is performed by destination, not by flow. I estimate how significant the effect of per-destination load-balancing can be.

Although Paris-traceroute MDA is used to discover per-destination load-balanced paths [4], it just enumerates all distinct paths towards the addresses within /24 blocks, assuming that paths towards the addresses within /24s are "identical" unless they are load-balanced paths. However, my goal in this chapter is to verify whether /24 blocks are homogeneous and thus I cannot rely on MDA. Instead, I make an assumption that is much more likely to be true than the assumption of Paris-traceroute MDA. I assume that the addresses within "/31" blocks have identical routes unless per-destination load-balancing occurs. Based on this assumption, I select two addresses that are within a /31 block from each /24, and then discover routes between a source (located at UMD) and the selected addresses using Paris-traceroute MDA. If the addresses within /31s have distinct routes, I consider that the /24s they are chosen from are affected by per-destination load-balancing. About 77% of the /31s have distinct routes. This shows that per-destination load-balancing is prevalent and can be a significant confounding factor in determining the homogeneity by comparing routes.

3.1.3 Dealing with per-destination load-balancing

Per-destination load-balancers can take different paths even for topologically co-located addresses. Hence, in the presence of per-destination load-balancing, homogeneity cannot be measured by simply comparing routes. A remedy is to focus on last-hop routers[2] instead of the entire routes. If routes are different due to load-

---

[2] Last-hop routers are the last routers in the paths to the destinations. Their addresses may not be identified by traceroute if they do not respond to traceroute probes.

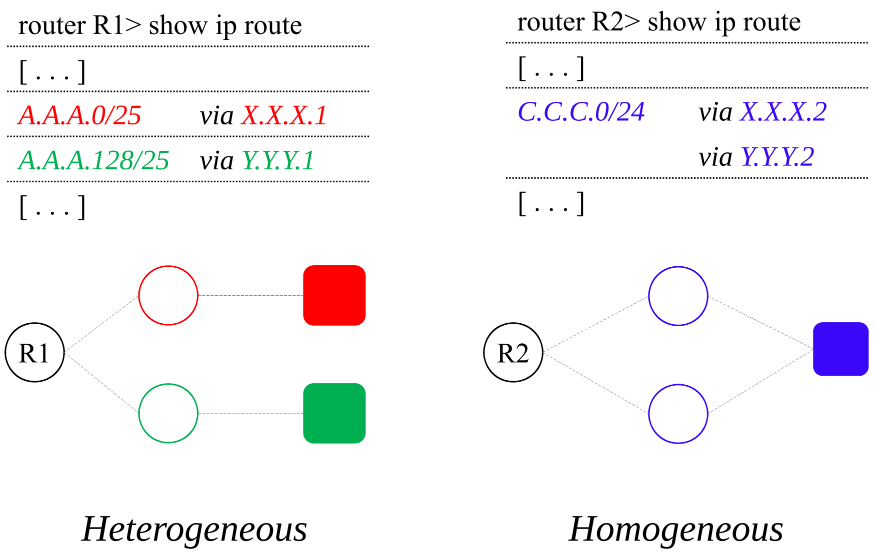
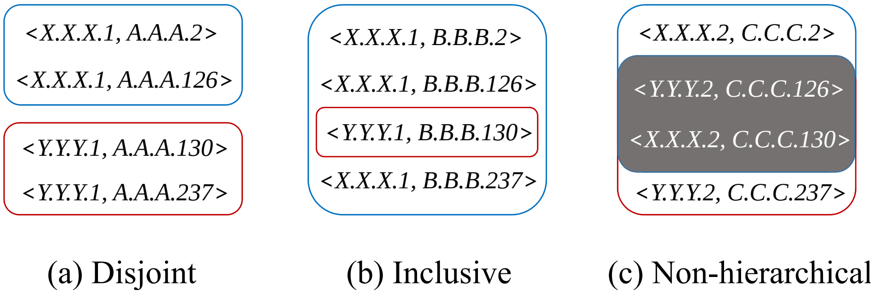


Figure 3.1: Different last-hop routers due to distinct route entries (left) and per-destination load-balancing (right).



(a) Disjoint (b) Inclusive (c) Non-hierarchical

Figure 3.2: The relationship between the sets of the addresses grouped by last-hop routers. $< X, Y >$ denotes X is a last-hop router of a destination Y.

balancing but eventually converge, last-hop routers will be identical. If routes are identical, last-hop routers are obviously identical. One missing case is when routes are different due to load-balancing but do not converge. In other words, last-hop routers are different due to load-balancing. It might be questionable how often this happens. According to the traceroutes dataset I collected for the addresses within /31s, about 30% of the address pairs within /31s have distinct last-hop routers. These differences are likely due to load-balancing (under the assumption that addresses within /31s are unlikely to have different routes without load-balancing).

The question is how to distinguish whether the difference in last-hop routers is caused by load-balancing or heterogeneity. I consider that addresses are heterogeneous (in terms of topological proximity) if their last-hop routers are different due to distinct route entries rather than load-balancing[3] (figure 3.1). Route entries are typically generated for subnets of which network prefixes do not overlap each other, unless one subnet includes the other. Therefore, the relationships between distinct route entries will be hierarchical. To be specific, every pair of the entries will be either mutually disjoint (a sibling relationship[4]), or one includes the other (a parent-child relationship). Hence, if last-hop routers are different due to distinct route entries, when grouping addresses by their last-hop routers and representing each group by the range from the numerically smallest address in the group to the largest one, the relationships between the ranges also will be hierarchical (Figure 3.2a and 3.2b). The contrapositive of this statement, which should be also true, is that the addresses within /24 blocks are not heterogeneous (i.e., homogeneous), if any of the addresses is not hierarchical when grouped by their last-hop routers (Figure 3.2c). Combining this with that /24 blocks are homogeneous if their addresses have identical last-hop routers, I determine that /24s are homogeneous if any of the addresses within them does not a have hierarchical relationship with others, or they all have common last-hop routers. I call this methodology *homogeneous block identification technique* (Hobbit).

---

[3] Per-destination load-balancing is often implemented by installing route-cache entries for each of destinations [63]. I do not consider them to be distinct. I only consider route entries for different destination networks to be distinct.

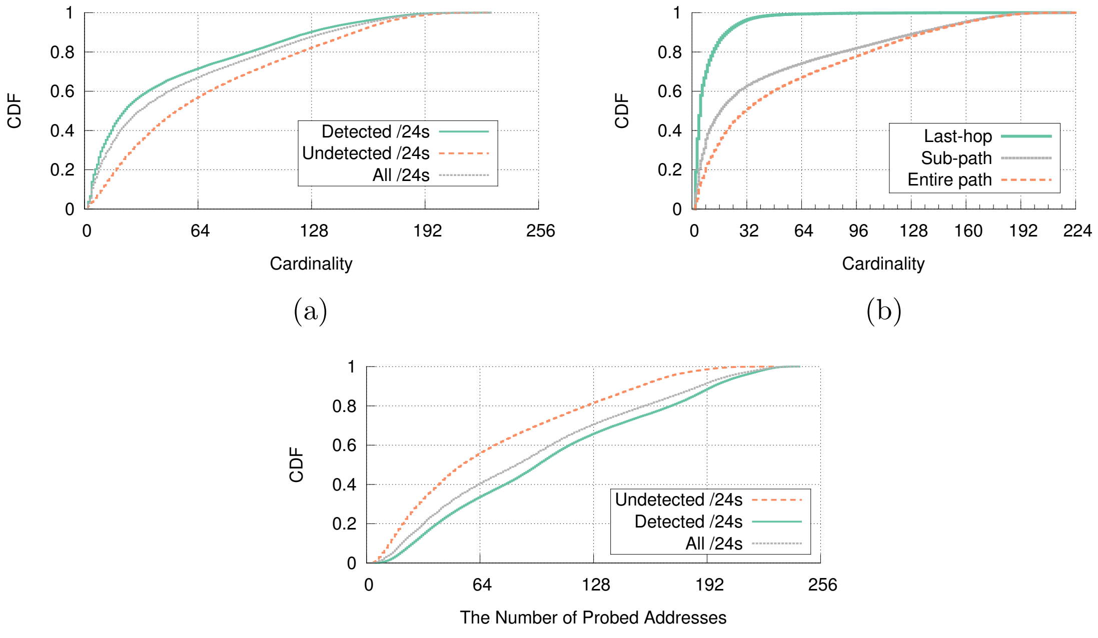
[4] I use the term "sibling" in that distinct subnets within a /24 subnet have a common /24 prefix (i.e., a common parent).

3.2 Elaboration on Hobbit

3.2.1 Last-hop vs entire traceroute

The basic idea of Hobbit is to examine whether the addresses within /24s have hierarchical relationships. This idea is applicable not only to last-hop routers but also to entire traceroutes. (I can group addresses having common traceroutes and check the relationships between the groups.) Nevertheless, I focus on last-hop routers. Reducing measurement loads (as I describe in Section 3.2.4) is not the only reason. More importantly, the coverage of Hobbit is enhanced when applied to last-hop routers compared to when applied to entire traceroutes. I compare how many homogeneous /24s Hobbit finds in each case. /24 blocks that have /31s of which traceroutes are different are likely to be homogeneous. Among these, I only select the /24s having different last-hop routers for fair comparison. If all the last-hop routers of a /24 are the same, I can conclude that it is a homogeneous block without checking the relationships. This is an advantage for the case of when applied to last-hop routers. I collect the traceroutes of all the active addresses within the chosen /24s (from a machine at UMD using Paris-traceroute MDA). I then apply Hobbit using two metrics, last-hop routers and entire traceroutes. In terms of traceroutes, only 70% of the /24s were determined to be homogeneous. 70% is quite low considering that I only selected /24s that are likely to be homogeneous. On the other hand, 92% of the /24s were homogeneous in terms of last-hop routers. I investigate what causes the difference.

Load-balancers use hashing to determine the next hop. Thus there is a chance



(a)

(b)

(c)

Figure 3.3: CDF of (a) cardinality and (c) the number of probed addresses for detected and undetected homogeneous /24 blocks by my methodology. (b) CDF of cardinality in different metrics, entire traceroute, last-hop router and sub-path that indicates the path from a common router to destination.

that load-balanced paths appear to have hierarchical relationships. If this false hierarchy appears, Hobbit may fail to recognize the homogeneity. The question is how often hashing falsely suggests hierarchy, what it is related to, and how Hobbit can control it. I observe that its probability is closely related to cardinality, that is, the number of distinct traceroutes (or last-hop routers) towards the addresses within /24. Figure 3.3a shows the CDF of the cardinalities (in terms of traceroutes) of the homogeneous /24s that were detected and undetected by Hobbit (along with those of all the homogeneous /24s). I can see that the undetected homogeneous /24s tend to have higher cardinalities compared to the detected and all homogeneous /24s. This implies that cardinality influences the probability of failures. The cardinality of /24s varies a lot depending upon the metrics that define cardinality. Figure 3.3b shows the CDF of the cardinalities of all the homogeneous /24s in terms of traceroutes, last-hop routers and sub-paths that indicate the paths from the routers that are common to all the destinations within /24 and closest to the /24. As I use smaller parts of traceroutes, cardinality tends to decrease. One reason could be that there are multiple load-balancers on the paths. The cardinality multiplicatively increases as the number of load-balancers increases. For example, if load-balancers L_1 and L_2 distribute traffic across N_1 and N_2 paths, the total number of distinct paths can be up to $N_1 * N_2$. In comparison to the cardinalities of entire traceroutes, those of last-hop routers are very small, and this is why the coverage of Hobbit is enhanced by 22% when using last-hop routers compared to using traceroutes.

3.2.2 How many destinations need to be probed?

Although Hobbit may fail to detect some homogeneous blocks depending on the cardinality, the probability of failures can be controlled by probing more destinations (because the probability is related to the number of probed addresses as shown in Figure 3.3c). The question is how many destinations need to be probed for a certain confidence level. I decide the number of destinations, by computing the probability of failures for each <cardinality, number of probed addresses> pair. Although the probability function could be theoretically developed, I rely on empirical analysis in this dissertation. I use the traceroute dataset collected for all active addresses within homogeneous /24s (as described in Section 3.2.1). For every combination of the destinations within a homogeneous /24, I can predict whether Hobbit will determine the /24 to be homogeneous if it only probes the destinations corresponding to the combination (simply by applying Hobbit to the partial information corresponding to the combination). All the combinations that would be determined not to be homogeneous are failures (and the others are successes), because all the combinations are chosen from homogeneous /24s. By classifying combinations by the number of destinations within them and cardinality (and computing the failure ratio in each category), I can obtain the probability of failures for each <cardinality, number of probed addresses> pair. One issue is that the total number of combinations is excessive. (The number of combinations for each /24 is $\sum_{i=1}^{n} \binom{n}{i}$ where n is the total number of active addresses within the /24, and I have data for more than 150k /24s.) To deal with this, I choose a random sample of all combinations such that most of the <cardinality, number of probed addresses> pairs have at least 16,588 sample points[5]. Figure 3.4 depicts[6] the distribution of degree of confidence, that is, 1 - failure ratio. As expected, the confidence tends to increase as the number of probed addresses increases and cardinality decreases. I use this data in deciding when to stop probing (as detailed in Section 3.2.5).

3.2.3 How to select destinations?

Hobbit requires at least 4 active addresses to be effective. It is because the relationships between less than 4 addresses are always hierarchical no matter how they are grouped. I also require that every /26 within /24 has at least one active address, so that my result represents the entire /24. I identify all active addres-

---

[5] I obtain this number by computing the number of samples required for 99% confidence level, 1% margin of error, 50% sample proportion estimate and infinite population size [64].

[6] The values of some pairs were not depicted because they have less than 16,588 sample points at a chosen sampling rate.

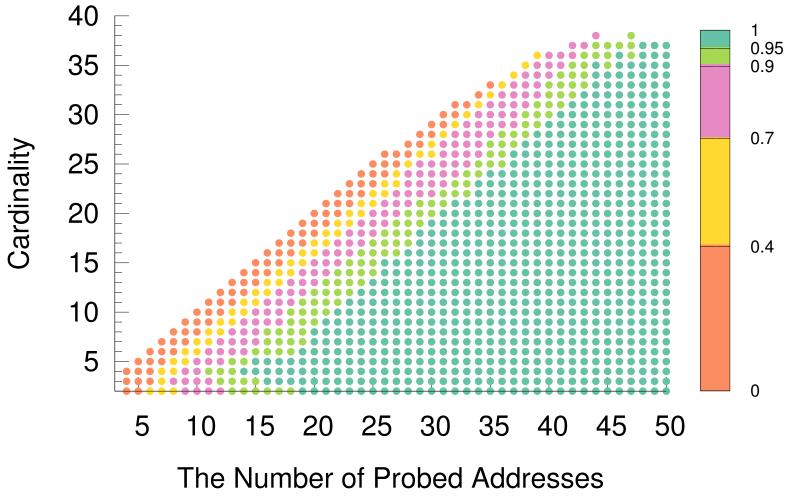


Figure 3.4: Degree of confidence that Hobbit will recognize a homogeneous /24 block per <cardinality, the number of probed addresses> pair.

ses using ZMap ICMP Echo Request dataset, and only select /24s that meet the criterion. For each chosen /24, I group the active addresses within it by their /26 prefixes, and then probe each /26 group in a round-robin fashion. I shuffle the order of the /26s to probe at the end of each round.

3.2.4 How to identify last-hop routers?

The only information I need to gather by probing the destinations are their last-hop routers. In order to efficiently identify last-hop routers, I try to infer a hop count between source and a last-hop router. I send an ICMP Echo Request to a destination and inspect the response's TTL field. If I know a default TTL value of the destination host (that is, the initial TTL value written by the destination host), I can compute the hop count between source and destination. Although default TTL values are different for different operating systems, the values of 64, 128 and 255 are commonplace [65–67]. So I consider that a default TTL is 64 if the TTL value of the response (TTL_{res}) is less than 64. If $64 \leq TTL_{res} < 128$, $128 \leq TTL_{res} < 192$ or

$192 \leq TTL_{res}$, a default TTL is considered to be 128, 192 or 255, respectively. Once I identify the default TTL value of a destination, I compute the hop count between source and the last-hop router by subtracting the TTL_{res} from the default TTL value. I then run Paris-traceroute MDA with the $first\_ttl$ configured to the hop count. The inferred hop count value may be inaccurate if routers use customized default TTL values or the hop counts of the forward and reverse path are different. If the hop count is an underestimate, I will find some more routers than the last-hop router. If the hop count is an overestimate, I will fail to identify the last-hop router. If it happens, I halve the $first\_ttl$ and run again Paris-traceroute MDA. This is repeated until the last-hop router is identified or the $first\_ttl$ becomes 1.

3.2.5 When to terminate?

Hobbit determines that a /24 is homogeneous if all the addresses have a common last-hop router, or any of them have a non-hierarchical relationship when grouped by their last-hop routers. Hence, I terminate probing if the non-hierarchical relationship is found or I can confirm that all the addresses have a common last-hop router. To determine with a high degree of confidence whether or not a /24 has a single last-hop router, I exploit the analysis of Paris-traceroute MDA. That is, a router has a single nexthop interface (for a certain destination) at the probability of 95% if 6 probes are responded by a single nexthop interface [22]. I can view the number of interfaces as a random variable and thus substitute it with the number of last-hop routers. Therefore, I determine that a /24 has a single last-hop router (and stop probing), if I only find a single last-hop router having probed 6 destinations.

I also terminate probing when I have probed as many destinations as required for 95% confidence level (figure 3.4). If no confidence value is present for the current <cardinality, number of probed addresses> pair, I probe all the active addresses.

3.3 Measurement results

I measure the homogeneity of /24 blocks using Hobbit. I choose 3.37M /24 blocks based on the ZMap data (Section 3.2.3), and probe each of them from a machine located at UMD. In this section, I present and analyze the measurement results.

3.3.1 How homogeneous are /24 blocks?

Table 3.1 shows a summary of measurement results. There have been /24 blocks that were not analyzable by Hobbit. Although I only choose /24s having at least 4 active addresses using the ZMap data, some blocks had less than 4 active addresses when I probed them. Even when blocks have at least 4 active addresses, if the number of active addresses are less than required for achieving a desired confidence level, that is, 95% (figure 3.4), I classify the blocks as "Not analyzable". These two cases account for about 25% of the /24s I probed. Despite the large enough number of active addresses, 16.8% of the /24s were not analyzable because none of their last-hop routers were responsive.

I have found 1.77M homogeneous /24 blocks. About 0.62M blocks had common last-hop routers, and 1.15M blocks had different last-hop routers but their relationships were non-hierarchical. This result reinforces that per-destination load-

Classification		# of /24 blocks
Not analyzable	Too few active	840,258 (24.9%)
	Unresponsive last-hop	567,439 (16.8%)
Homogeneous	Same last-hop router	616,719 (18.2%)
	Non-hierarchical	1,153,628 (34.2%)
Different but hierarchical		198,292 (5.9%)

Table 3.1: Measurement results of the homogeneity of /24

balancing is prevalent and it even changes last-hop routers of destinations, and thus simply checking whether addresses have a common last-hop router is not enough for determining homogeneity. The remaining 0.2M blocks consist of the addresses that have different last-hop routers of which relationships are hierarchical. Since I probed as many addresses as required for 95% confidence level, the probability of these blocks being homogeneous is less than or equal to 5%. If I consider all these blocks as heterogeneous, I can conclude that 1.77M out of 1.97M /24s, that is, 90% of the /24s are homogeneous.

3.3.2 Analyzing heterogeneous /24s

Strictly speaking, the last category in table 3.1, a set of /24s that have different last-hop routers but the relationships of their addresses appear to be hierarchical is a mixture of homogeneous and heterogeneous /24 blocks. There is a non-negligible chance (5%) that the /24 blocks in the category are homogeneous. I have examined this category to discover /24s that are “very likely” to be heterogeneous, and found the criteria that define a certain class of /24s that are “very likely” to be heterogeneous.

The first criterion is that, when the addresses within /24 are grouped by their last-hop routers, the relationship between any pair of the groups is disjoint (i.e., not inclusive). Second, the groups are aligned. To be specific, when each group is represented by a subnet whose network prefix is the longest common prefix of the addresses within group, every subnet contains only the addresses that are within the corresponding group. For example, if I observe that the addresses <X.Y.Z.2, X.Y.Z.125> and <X.Y.Z.129, X.Y.Z.254> have common last-hop routers respectively, then I will consider that X.Y.Z.0/24 is a heterogeneous block, because the two groups are disjoint and the two corresponding subnets, X.Y.Z.0/25 and X.Y.Z.128/25 only contain the addresses within each group. If the second group were <X.Y.Z.127, X.Y.Z.254>, I would not consider this /24 to be heterogeneous because the groups would be disjoint but not aligned. /24 blocks that satisfy this criteria are very likely to be heterogeneous. I verified that homogeneous /24 blocks meet the criteria at the probability of less than 0.1%. Based on this criteria, I found 17,387 heterogeneous /24 blocks (in other words, the other 198,292 - 17,387 /24 blocks were either inclusive or disjoint but not aligned). These blocks consist of homogeneous sub-blocks. Table 3.2 shows the distribution of sub-block compositions. More than half of the /24s are composed of two homogeneous /25 blocks. One /25 along with two /26s and four /26s are also common compositions. /27 and /28 are also present although they are not as common as /25 and /26.

Given that at least 90% of the /24s are homogeneous, it could be considered unusual to split /24s into smaller sub-blocks and treat them differently. In order to discover who is splitting /24 blocks and why, I obtain AS numbers, organization

Composition	Ratio
{/25, /25}	50.48%
{/25, /26, /26}	20.65%
{/26, /26, /26, /26}	15.79%
{/25, /26, /27, /27}	5.92%
{/26, /26, /26, /27, /27}	4.63%
{/26, /26, /27, /27, /27, /27}	1.13%
{/25, /26, /27, /28, /28}	0.81%
{/25, /27, /27, /27, /27}	0.58%

Table 3.2: The distribution of homogeneous sub-blocks within heterogeneous /24 blocks

names and geolocations of all the heterogeneous /24s using the Maxmind GeoLite database [68]. I then group the /24 blocks by the ASN they belong to. Table 3.3 shows the top 10 ASes with the most number of heterogeneous /24 blocks, along with organization names, countries the /24s have been allocated to, and the types of organizations I figured out from their websites. The top 2 ASes, which are both from Korea, include about 60% of the heterogeneous /24s. Other countries also tend to have more than one AS. France, Denmark and Georgia each have two. The US has one AS of which organization type is a hosting company; the rest are under the control of broadband ISPs.

To further analyze heterogeneous /24 blocks, I make WHOIS queries to KRNIC [69] that is a Korean national Internet registry maintaining specific information about the addresses allocated to Korea. I focus on the top AS, Korea Telecom, because it keeps assignment information current. I made a query for each of the heterogeneous /24s and could verify that they are actually being split into sub-blocks. Table 3.4 shows an example. The /24 block 220.83.88.0/24 is divided into

Rank	# of Heterogeneous /24s	ASN	Organization	Country	Type
1	8207	AS4766	Korea Telecom	Korea	Broadband ISP
2	1798	AS9318	SK Broadband		
3	499	AS15557	SFR	France	
10	106	AS35632	IRIS 64		
4	486	AS3292	TDC A/S	Denmark	
6	172	AS9158	Telenor A/S		
5	242	AS4788	TM Net	Malaysia	
8	115	AS28751	Caucasus	Georgia	
9	108	AS20751			
7	125	AS36352	ColoCrossing	US	Hosting

Table 3.3: Top 10 ASes having the most number of heterogeneous /24 blocks

IPv4 Address	:	220.83.88.0/25	220.83.88.128/26	220.83.88.192/26
Organization Name	:	KT	Chungbukbonbujang	Donghajeongmil
Network Type	:	CUSTOMER		
Address	:	Cheongwon-Gu	Jincheon-Eup	Munbaek-Myeon
		Cheongju-Si	Jincheon-Gun	Jincheon-Gun
Province	:	Chungcheongbuk-Do		
Zip Code	:	360172	365-800	365-860
Registration Date	:	20160112	20150317	20150317

Table 3.4: WHOIS responses from KRNIC for a /24

220.83.88.0/25, 220.83.88.128/26 and 220.83.88.192/26, each of which is allocated to different customers located at different addresses. Although Korea has more than 100 million IPv4 addresses [70], considering that nearly all the heterogeneous blocks including the example block have been registered in 2015 or later, IPv4 address depletion might be a reason for splitting the /24 blocks.

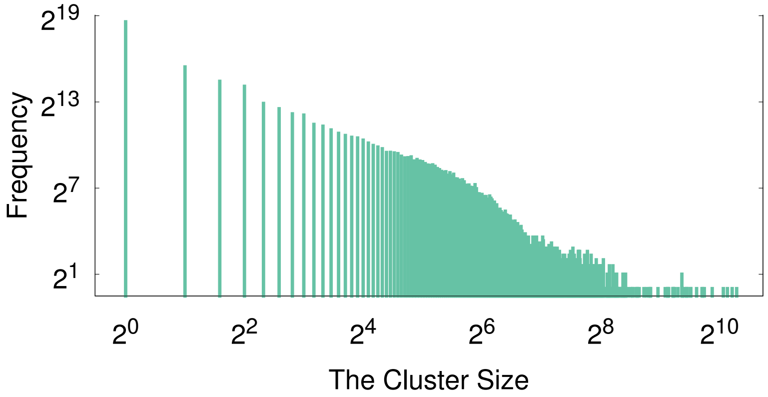


Figure 3.5: The size distribution of aggregated homogeneous blocks in terms of /24 blocks they contain

3.4 Aggregating homogeneous /24s

A natural extension of the measurement of the homogeneity of /24s is to find homogeneous sub-blocks within heterogeneous /24s and to find larger homogeneous blocks than /24s (by aggregating them) if they are homogeneous. In this section, I focus on the aggregation of homogeneous /24 blocks.

I associate each homogeneous /24 with the set of last-hop routers of the addresses within the /24. The set can be a singleton if all the addresses within a /24 have a single common last-hop router, but can instead include multiple last-hop routers if the addresses have different last-hop routers due to load-balancing. My approach to aggregation is to merge /24s having the identical[7] sets of last-hop routers. This method enables the aggregation of discontiguous /24 blocks. I present the aggregation results made using this method.

---

[7] I consider two sets are identical if their sizes are equal and every last-hop router in one set is also in the other set.

3.4.1 How large are the aggregated blocks?

By aggregating homogeneous blocks that have identical sets of last-hop routers, the total number of homogeneous blocks has been reduced from 1.77M to 0.53M (including not aggregated homogeneous /24s). Figure 3.5 depicts the distribution of size; that is, the number of /24s within the aggregated blocks. About 0.39M blocks have the size of 1. This indicates that they have not been aggregated. Still, many blocks have the size greater than 1. Although the number of blocks with the size x decreases as x increases, 21,513 blocks consist of at least 16 /24s, and 2,430 blocks contain at least 64 /24s. There are even blocks that include more than 1024 /24s. This result demonstrates that, even though /24 blocks are mostly homogeneous, they are not necessarily the largest homogeneous block. Therefore, using /24s could be inefficient. For example, since traceroutes towards homogeneous addresses are likely to be the same, selecting destinations for topology discovery from each /24 might be less efficient than choosing the destinations from the homogeneous blocks I have identified.

3.4.2 Who are the biggest homogeneous blocks?

In the presence of IPv4 address exhaustion, assigning a large number of addresses to the machines located at the topologically same location may seem unexpected. To understand why it happens, I characterize top 15 largest homogeneous blocks. I identify their ASNs, organization names and geolocations using the Maxmind GeoLite databases, and the types of organizations from their websites. Table 3.5 summarizes the identification results. With respect to their types, "Hosting"

Rank	Size	ASN	Organization	Geo-location	Type
1	1251	18779	EGI Hosting	US	Hosting
2	1187	1257	Tele2	Sweden	Broadband ISP
3	1122	16509	Amazon	Japan	Hosting/Cloud
4	1071	2914	NTT America	US	Hosting/Cloud
5	940	32392	OPENTRANSFER	US	Hosting
6	857	1257	Tele2	Sweden	Broadband ISP
7	840	4713	OCN	Japan	Broadband ISP
8	835	16509	Amazon	US (San Jose)	Hosting/Cloud
9	783	4713	OCN	Japan	Broadband ISP
10	732	9506	SingNet	Singapore	Broadband ISP
11	731	17676	SoftBank	Japan	Broadband ISP
12	703	26496	GoDaddy	US	Hosting
13	699	22394	Verizon Wireless	US	Broadband ISP
14	698	32392	OPENTRANSFER	US	Hosting
15	679	22773	Cox	US (Arizona)	Residential ISP

Table 3.5: Top 15 largest homogeneous blocks

indicates a hosting company. I add the suffix "/Cloud" to "Hosting" if the website describes their hosting services as cloud computing services. Although Amazon is well-known for electronic commerce, I classify it as "Hosting/Cloud" because the reverse DNS names of the addresses within the corresponding blocks begin with "ec2" that is the name of its cloud computing service. "Broadband" denotes an ISP that provides both mobile and fixed broadband services. Verizon Wireless (also known as Cellco Partnership) and Cox are classified as "Mobile Broadband" and "Fixed Broadband", respectively, because they provide each of the services only.

7 of the 15 blocks are being used by hosting companies. It is understandable

that hosting companies allocate many addresses to the same region because they run datacenters for their services. The addresses within each block might have been assigned to the servers in a datacenter. Actually, the two blocks of Amazon appear to be allocated to their datacenters. The reverse DNS names of the addresses within each block have the common keyword "ap-northeast-1" and "us-west-1", respectively, that indicate the endpoints of their datacenters located in Japan and US west [71].

6 blocks have been classified as "Broadband". Since "Broadband" ISPs provide both mobile and fixed broadband services, the addresses within these blocks could be allocated to cellular networks. A recent study on timeouts has observed that, if an initial probe to a destination experiences a higher delay than subsequent probes, then the destination is likely a cellular wireless device [72]. I use this observation to identify whether the addresses within each block are assigned to cellular devices. I randomly choose 200 /24s from each block, and then send 20 ping probes to every active address within the chosen /24s. For each address, I compute the difference between the RTT of the first ping and the maximum RTT of the rest of the pings. If the addresses within a block tend to have higher first RTTs than the maximum RTTs of the rest (i.e., if the differences tend to be positive), then the block is likely being used for a cellular network. Figure 3.6 depicts the distributions of the differences of the 6 "Broadband" blocks plus the Verizon wireless block that I add for reference. Tele2 and OCN each have two blocks and the differences tend to be high in all the blocks. About 50% of the addresses within the blocks have the differences greater than 0.5s and the differences of at least 10% of the addresses are greater than or

equal to 1s. Verizon wireless also has a similar distribution. Therefore, the Tele2 and OCN blocks as well as the Verizon wireless block are likely being assigned to cellular networks. SingTel and SoftBank are very different from the others. Most of the differences are nearly zero. This indicates that they are not being used for cellular devices.

Recent studies have shown that major US cellular carriers connect their cellular networks with the Internet through a few infrastructure locations (so-called ingress points) [27,73]. This means that probes for many cellular devices traverse a common ingress point, and thus they would appear to be co-located on the Internet topology. This explains why Verizon wireless has a large homogeneous block. I suspect that the ingress points of Tele2 also cover a wide area because the addresses within the Tele2 blocks are located across three countries– Sweden, Croatia and Netherlands, according to the Maxmind GeoLite databases and their reverse DNS names. I are not certain that OCN ingress points also cover a wide area but it appears likely, considering that the OCN blocks are as large as Tele2 and Verizon wireless blocks. Therefore, my result may imply that not only US cellular carriers but also European (Tele2) and Asian (OCN) carriers deploy only a few ingress points.

The last block is owned by Cox that provides fixed broadband service to residential and business customers. Most of the addresses within the block are located in Phoenix, Arizona according to the Maxmind GeoLite databases and their reverse DNS names. They do not seem to be residential addresses, considering that most of their reverse DNS names begin with “wsip” whereas Bitcoin nodes in the Cox network (that are likely to be residential) mostly have the reverse DNS names be-

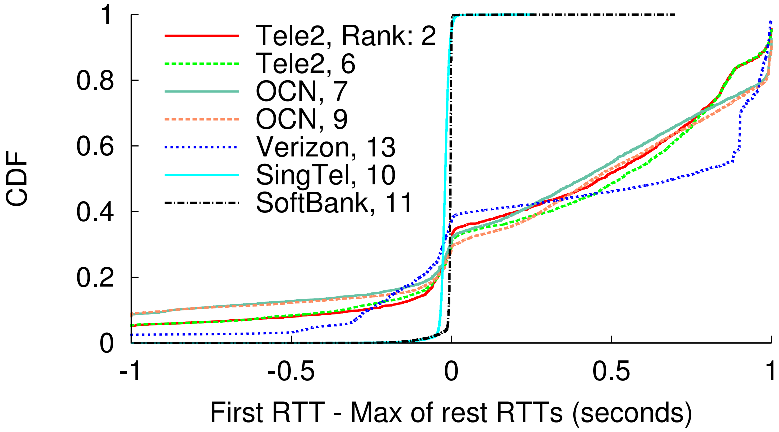


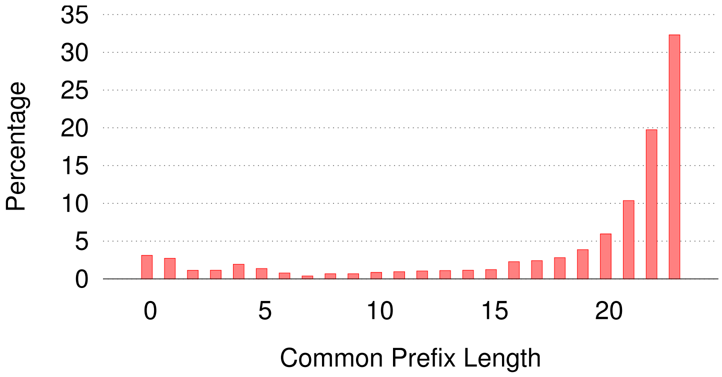
Figure 3.6: The CDF of the differences between the first RTT and the maximum of the rest RTTs for “broadband” blocks

ginning with “ip” [74]. Cox operates a large datacenter in Phoenix for business customers [75]. It could be the location where the addresses within the Cox block are allocated to. Singtel and SoftBank also provide datacenter services. The Singtel and SoftBank blocks might also be assigned to datacenters, considering that their RTTs were very stable (Figure 3.6).

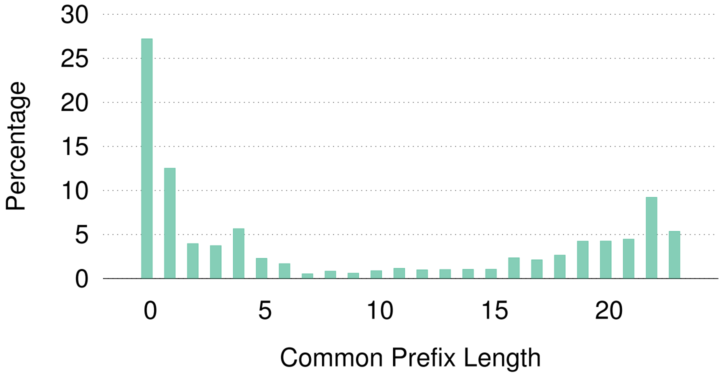
3.4.3 Are the addresses within blocks numerically adjacent?

Topologically co-located addresses may be expected to be numerically adjacent because routing decisions are usually based on prefixes rather than the entire address. In this section, I analyze the numerical adjacency of the /24 blocks within the homogeneous blocks I have identified. I estimate the degree of adjacency between a /24 pair by computing the longest common prefix length of the pair. Since I compare /24s (rather than entire addresses), the length ranges from 0 to 23, and high length represents high degree of adjacency.

I numerically sort the /24s within each homogeneous block, and then compute



(a)



(b)

Figure 3.7: The length distribution of the longest common prefixes between (a) adjacent /24s within homogeneous blocks (b) the smallest and the largest /24s

the common prefix length between the /24s that are right next to each other. Figure 3.7a shows the distribution of the lengths. More than 30% of the /24 pairs have the length 23, and the lengths of about 70% are at least 20. This implies that many /24s are contiguous within the blocks. However, this does not necessarily mean that the blocks mostly consist of a single contiguous block. I next measure the common prefix length between the smallest and the largest /24s within each block (figure 3.7b). About 40% of the pairs have the length 0 or 1 whereas only about 5% have the length 23. This, in combination with the above result that many /24s are contiguous, implies that homogeneous blocks often consist of multiple contiguous

#1. EGI Hosting (cluster size - 1251)

#2. Tele2 (1187)

#3. Amazon (1122)

#4. NTT America (1071)

#5. OPENTRANSFER (940)

#6. Tele2 (857)

#7. OCN (840)

#8. Amazon (835)

Figure 3.8: Visualization of numerical adjacency of /24s within the top 8 homogeneous blocks

sub-blocks that are separated from each other.

I verify that homogeneous blocks consist of contiguous sub-blocks by visualizing the adjacency of the /24s within the top 9 largest homogeneous blocks in Figure 3.8. For each block, given a sorted list of /24s $\{p_1, p_2, ..., p_n\}$, I draw a vertical line at x_i such that

$$x_i = \begin{cases} 1 & \text{if } i \text{ is } 1 \\ x_{i-1} + (24 - LCP\_LEN(p_{i-1}, p_i)) & \text{if } i > 1 \end{cases}$$

where $LCP\_LEN(p_i, p_j)$ denotes the longest common prefix length of p_i and p_j. The gap between the vertical lines represents the degree of adjacency. A large gap indicates low degree because the gap becomes larger as the length of the corresponding longest common prefix decreases. Most of the blocks contain large contiguous segments, none of which covers the entire block. This demonstrates that large homogeneous blocks mainly consist of several contiguous sub-blocks that are separated from each other.

3.5 Summary

In this chapter, I presented the design and implementation of a technique called Hobbit that measures topological proximity of IPv4 addresses and aggregates topologically proximate addresses. Hobbit deals with path diversity due to per-destination load-balancing by distinguishing between route differences due to load-balancing and different route entries. I have identified 1.77M homogeneous /24 blocks using Hobbit and aggregated them into 0.51M homogeneous aggregate blocks. I have characterized the top 15 biggest blocks, and found that most of them have been allocated to datacenters or cellular networks.

I also have discovered that addresses within many homgeneous /24s have different last-hop routers due to load-balancing and most of Hobbit blocks consist of numerically discontiguous addresses. These results suggest that the unique characteristics of Hobbit greatly help to identify more homogeneous addresses. Hobbit can aggregate even discontiguous addresses by focusing on last-hop routers rather than network prefixes, and can deal with load-balancing by checking the relationship of addresses within /24s. In the next chapters, I demonstrate that Hobbit blocks are as homogeneous as /24s and thus can improve the efficiency of Internet topology mapping that draws destinations from /24 blocks (like like CAIDA's topology discovery [28]). I also show that Hobbit blocks can be used for improving the representativeness of active Internet measurement.

Chapter 4: IMPROVING EFFICIENCY OF ACTIVE MEASUREMENT

In this chapter, I demonstrate that Hobbit blocks can help to improve the efficiency of active Internet measurement. I quantify the efficiency improvement achieved by using Hobbit blocks for two specific applications: 1) Internet topology mapping and 2) latency prediction.

I compare an Internet topology mapping strategy that selects destinations from each /24 block (like CAIDA's topology discovery [28]) with that selects destinations from each Hobbit block. I quantify the measurement load of each strategy and show that the strategy using Hobbit blocks discovers more links when the same number of destinations are probed.

To demonstrate that Hobbit blocks can be used for improving the efficiency of latency prediction, I first quantify the degree of correlation of RTTs within Hobbit blocks. I show that RTTs are highly correlated within Hobbit blocks.I then demonstrate that Hobbit blocks can actually help existing latency estimation techniques to predict RTTs between more number of address pairs without additional measurements, in other words, to improve the efficiency of latency prediction.

I finally compare Hobbit with a specific system that aggregates Internet addresses for improving efficiency of network measurement, that is, iPlane [3]. Whereas

Hobbit uses last-hop routers as a criterion of aggregation, iPlane uses BGP prefixes and geographic locations. I compare these criteria by quantifying the correlation of RTTs within the aggregates generated by different criteria.

4.1 Improving the efficiency of Internet topology mapping

Internet topology mapping has a wide range of applications including the inference of routing policy [76], realistic network simulation [77] and network neutrality inference [78]. It is important for Internet topology mapping to reduce the number of probe destinations, because Internet topology mapping typically uses traceroute that requires tens of probes for many destination unlike ping style probing methods that only send a few probes to each destination. Reduced measurement loads can help to increase measurement frequency or prevent measurement traffic from being confused with malicious attacks. I demonstrate that measurement loads for Internet topology mapping can be reduced by using Hobbit blocks.

4.1.1 Comparing Internet topology mapping strategies that use /24 prefixes and Hobbit blocks

A typical approach to selecting measurement destinations for Internet topology mapping is to choose a destination from each BGP prefix with large prefixes disaggregated into /24 prefixes [2, 28]. I compare this approach with an alternative approach that selects a destination from each Hobbit block. Due to the prevalence of per-destination load-balancing, it may not be enough to select a single destination from each /24 or Hobbit block for comprehensive topology discovery. I thus perform the following analysis. I sample 10,000 /24 blocks and perform traceroute to all the

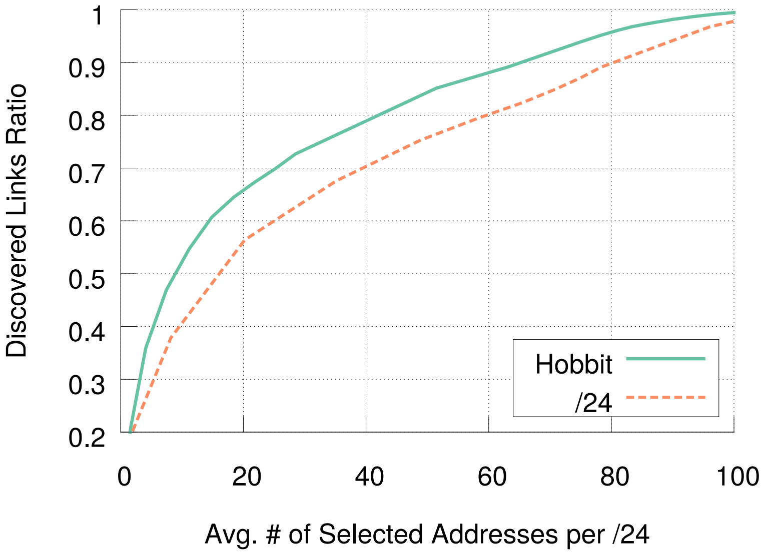


Figure 4.1: The ratio of the links discovered by two different approaches: To select addresses from 1) each Hobbit block and 2) each /24.

active addresses within the sampled /24 blocks[1]. I then choose destinations (and extract the corresponding traceroutes) from the collected traceroute dataset in two different approaches, that is, to select a destination from 1) each /24, and 2) each Hobbit block. I compute and compare a discovered links ratio, that is, the number of distinct links within the chosen traceroutes divided by the total number of distinct links in the dataset. To enumerate all the links, I repeat to select more destinations from each block until the discovered links ratio nearly becomes 1. Figure 4.1 shows the ratios achieved by the different approaches as a function of the average number of selected destinations per /24 (that is, the number of selected destinations divided by the total number of /24 prefixes in the dataset). Selecting destinations from the Hobbit blocks always results in the discovery of more links compared to selecting

---

[1] The total number of probed addresses is 1.53M.

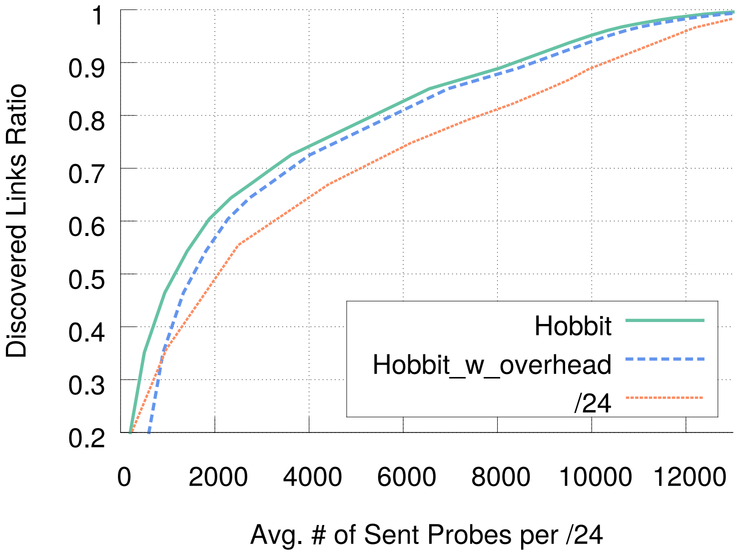


Figure 4.2: The ratio of the links discovered as a function of the number of sent probes.

destinations from /24. This demonstrates that the use of the Hobbit blocks can improve the efficiency of topology mapping. This result also shows that probing a single destination per /24 results in incomplete topology discovery considering that, even with 20 destinations per /24 probed, only about 40% and 55% of the links are discovered when /24 and Hobbit blocks are used in destination selection.

4.1.2 Analyzing the effect of Hobbit block generation overhead

Hobbit block generation incurs a substantial amount of active measurements (whereas /24 prefixes can be directly identified from IP addresses without any measurement). I analyze the measurement overhead of Hobbit block generation and the effect of the overhead on the efficiency improvement of Internet topology mapping. Although Hobbit probes each destination using traceroute like topology mapping, the number of probed destinations may not be an appropriate metric for measuring the Hobbit block generation overhead. It is because Internet topology mapping usually probes all intermediate hops between source and destination whereas Hobbit tries to probe only the last hop. I thus use the number of sent probes as a metric for overhead measurement. I count the number of probes generated by Hobbit and normalize it by the total number of /24 prefixes probed by Hobbit, obtaining the result of 396.45 probes per /24. To quantify the effect of the overhead on the efficiency of topology mapping, I compute the measurement load of topology mapping by the average number of sent probes per /24 (instead of the average number of probed destinations as I did in Section 4.1.1). I then add the overhead of Hobbit block generation to the measurement load of topology mapping that uses Hobbit blocks. Figure 4.2 shows the discovered link ratios as a function of the average number of sent probes per /24. When the Hobbit block generation overhead is not considered, the topology mapping strategy that selects destinations from Hobbit clusters shows better efficiency than the strategy that uses /24 blocks. Even when the overhead is included, the strategy using Hobbit blocks discovers more links using the same amount of probes except when the discovered link ratio is less than 35%. This result shows that Hobbit blocks can improve the efficiency of topology mapping even considering the block generation overhead. Note also that the overhead can be amortized over multiple vantage points in that topology mapping is typically performed from multiple vantage points whereas Hobbit uses a single vantage point. In addition, Hobbit blocks can be used for other purposes than topology mapping as I will show in Section 4.2 and Chapter 5).

The Number of Changes	Ratio (%)
0	83.27846
1	10.81293
2	3.62084
3	1.06284
4	0.72537
5	0.32779
6	0.13655
7	0.03235
8	0.00255
9	0.00033

Table 4.1: The distribution of the number of changes in last-hop routers of each /24

4.1.3 Updating Hobbit blocks periodically

The maintenance of Hobbit blocks may require periodic updates to deal with changes in topology. Hobbit blocks are based on last-hop routers. If last-hop towards Hobbit destinations frequently change, Hobbit blocks also need to be frequently updated. I analyze the frequency of the last-hop changes to figure out how often Hobbit blocks need to be updated. I collected the last-hop routers of all Hobbit-eligible /24 blocks (that was described in Section 3.2.3) every month from November 2016 to August 2017 obtaining 10 snapshots. For each /24 in the dataset, I count the number of changes in last-hop routers. For example, if the last-hop routers of a /24 in the snapshots are $< A, A, A, B, B, A, A, A, C, C >$, the number of changes is 3.

Table 4.1 shows the distribution of the number of last-hop router changes of the /24s. About 83.3% of the /24s did not change their last-hop routers for 9 months. 10.8% changed last-hop routers only once. Only about 0.5% of the /24s changed last-hop routers more than 4 times. This result shows that last-hop routers tend to be very stable over time. This result also reflects the expected duration of last-hop routers remaining unchanged. For example, the number of changes 2 indicates the expected duration 9/(2+1) = 3 months (because the dataset spans 9 months). Hence, the expected duration is at least 9/4 for about 98.8% of the /24s. Given this, I consider that a month is a reasonable period of Hobbit blocks update. I thus perform Hobbit every month and publish the Hobbit blocks.

4.1.4 Can Hobbit improve the efficiency of Trinocular?

Trinocular tracks outages of each /24 block. If the outages identified by Trinocular are correlated within Hobbit blocks, the efficiency of Trinocular can possibly be improved by using Hobbit blocks (i.e., by probing each Hobbit block instead of /24). The Trinocular dataset [15] records each outage that Trinocular detects. Using this dataset, I measure the correlation between outages and Hobbit blocks. For each Hobbit block, I identify the number of /24 blocks with correlated outages (i.e., /24s with time-overlapped outages) and then divide it by the number of total /24s within the Hobbit block.[2] This fraction value indicates the degree of correlation of outages within Hobbit blocks. For example, the fraction 1 indicates that all the /24s within the Hobbit block experienced concurrent outages. Since the sizes of

---

[2] I exclude Hobbit blocks having only a single /24 and blocks with no outages.

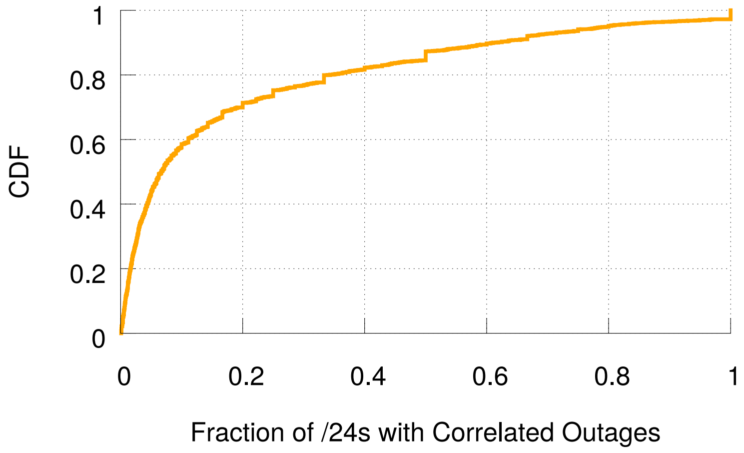


Figure 4.3: The CDF of the fractions of /24 blocks with correlated outages.

Hobbit blocks are various, I assign different weights to different fraction values. I give higher weights to larger Hobbit blocks because larger blocks can be more useful in improving the efficiency of Trinocular. For example, if the sizes of Hobbit blocks are 2 and 3 and their fraction values are 10 and 20, I generate 5 data points, <10, 10, 20, 20, 20> (and draw the CDF of the datapoints).

Figure 4.3 shows the weighted CDF of the fraction values. The fractions tend to be very small. The median value is 0.067 and the 90th percentile is 0.615. This result shows that Hobbit blocks are unlikely to be useful for enhancing the efficiency of Trinocular.

4.2 Improving the efficiency of latency estimation

Latency estimation is to predict the RTTs between Internet hosts without directly measuring it. It enables to avoid the cost of measurement [79] or to estimate the RTTs between arbitrary Internet addresses. Latency estimation has a wide range of applications such as online game matchmaking [80] and improving

Tor's performance [60]. I demonstrate that Hobbit blocks can help existing latency estimation techniques to predict RTTs between more addresses without additional measurement.

4.2.1 Measuring the correlation of RTTs within Hobbit blocks

I quantify the correlation of RTTs within Hobbit blocks to show that Hobbit blocks can be used for latency prediction. If RTTs are highly correlated within Hobbit blocks, RTTs to some addresses in a Hobbit block can be imputed from RTTs to other addresses in the same block. Since imputation does not incur additional measurement, it can lead to improved efficiency.

Obtaining RTT measurements To measure the correlation of RTTs, I obtain RTT measurements using ZMap ICMP Echo Request scan dataset [21]. The dataset contains the results of the entire IPv4 space scan by ICMP Echo Request probes. I only select the addresses that responded with ICMP Echo Reply messages[3] and compute RTTs using the timestamps recored in the dataset when the packets were sent and received. Since RTTs can vary widely depending on network conditions, I obtain 8 consecutive snapshots and calculate median RTTs.

Quantifying the correlation of RTTs I estimate the similarity of RTTs of the addresses within Hobbit blocks by using the coefficient of variation (CV), that is, the ratio of the standard deviation to the mean. I aim to quantify the spread of RTTs within Hobbit blocks and CV is a reasonable measure for that purpose. Given the RTT measurements (obtained from the ZMap ICMP Echo

---

[3] About 376M addresses responded with ICMP Echo Reply messages.

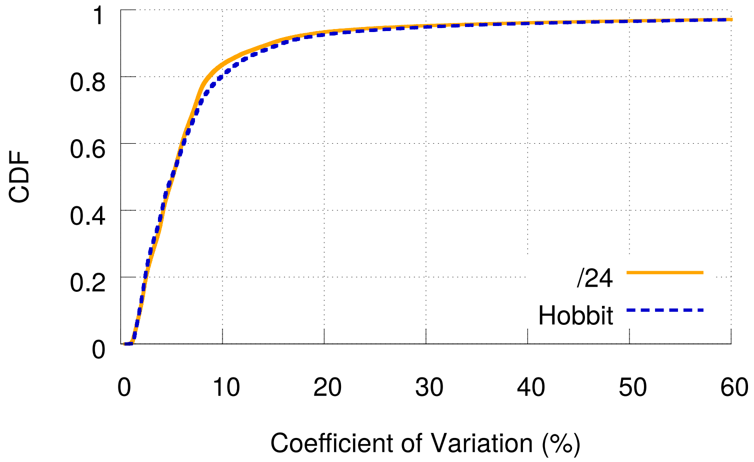


Figure 4.4: The CDF of coefficient of variation (CV) of RTTs within /24 and Hobbit blocks.

Request dataset), I compute CV for each Hobbit block by calculating the mean and standard deviation of the RTTs of all the addresses within the block. A higher CV indicates a lower degree of similarity (or correlation) of RTTs. I compare the CVs of Hobbit blocks with those of /24 blocks (that is, the aggregates of addresses having common /24 prefixes). /24 blocks are known to be mostly homogeneous in several aspects including RTTs [16, 19, 24, 35, 81]. If the CVs of Hobbit blocks are similar to those of /24 blocks, then I can consider that RTTs are strongly correlated within Hobbit blocks. Figure 4.4 shows the CDF of the CVs (in percentages) of Hobbit blocks[4] and /24 blocks. The distributions are nearly identical. About 70% of the blocks have at most 7% CV for both Hobbit and /24. Although the percentage of /24 blocks that have at most 10% CV is greater than that of Hobbit blocks, the difference is small (83% for /24 and 80% for Hobbit).

The above analysis is simple and easy to understand but can be misleading if

---

[4] I do not include Hobbit blocks that contain only a single /24 prefix because I am comparing to /24 blocks.

very large Hobbit blocks[5] have high CVs (i.e., low degrees of correlation) because the analysis does not consider the size of the blocks. To address this, I design and perform an alternative analysis. The basic unit of Hobbit clustering is a /24 block. In other words, Hobbit blocks consist of /24 blocks. For each /24 within Hobbit blocks, I compute the CV of the /24 and the CV of the Hobbit block it belongs to, and then subtract the CV of the /24 from that of the Hobbit block. This CV difference value can be interpreted as how much the correlation of RTTs is degraded when a /24 block is aggregated into larger blocks (i.e., Hobbit blocks). A higher value represents a higher degree of degradation. This analysis creates a data point for each /24 block, which is a fixed size block, rather than for each Hobbit block. Therefore, the size of Hobbit blocks is not an issue in this analysis. Figure 4.5 shows the CDF of the CV differences. The difference values tend to be very small. About 64% of the differences are less than 1, and 86% of the differences are at most 10. This result reinforces the high degree of correlation of the RTTs within Hobbit clusters.

4.2.2 Improving the efficiency of existing latency prediction techniques

4.2.2.1 Quantifying the efficiency improvement

High correlation of RTTs within Hobbit blocks suggests that Hobbit blocks can be useful for inferring RTTs and thus can improve the efficiency of existing latency estimation techniques. The efficiency improvement can be achieved by im-

---

[5] Unlike /24 blocks, the sizes of Hobbit blocks vary widely. Several blocks contain more than a thousand /24 blocks while many have one or two.

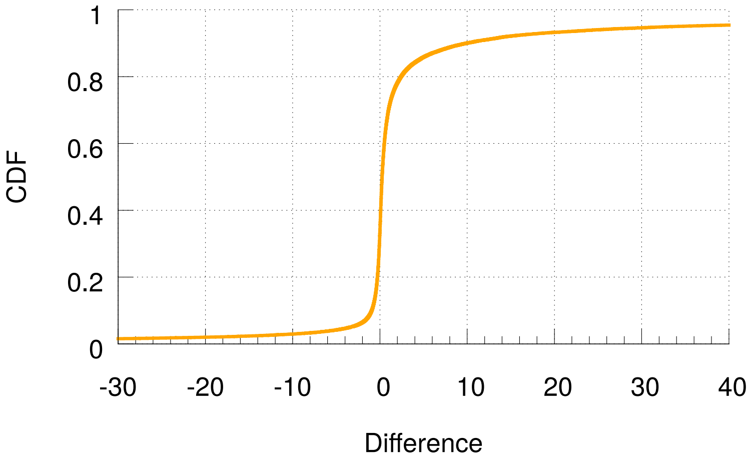


Figure 4.5: The CDF of the differences between the CVs of /24 blocks and those of the Hobbit blocks that the /24s belong to.

puting the RTTs between unmeasured addresses that are in Hobbit block A and B from RTTs between measured addresses in A and B. I quantify the efficiency improvement achieved by the imputation by measuring the number of additional latency estimations with the imputation. Specifically, I compare the number of address pairs between which RTTs can be estimated by an existing latency prediction technique called Ting [60] without the imputation against that with the imputation. Ting can only estimate RTTs between Tor [61] relays. With the imputation, the RTTs between any two addresses that belong to two different Hobbit blocks containing any Tor relay can be estimated. The RTTs between the addresses within the same Hobbit block can be estimated only when the block contains more than one Tor relays. Hence, the number of address pairs of which RTTs can be estimated with the imputation will be:

$$\sum_{i=1}^{C-1} \sum_{j=i+1}^{C} n_i * n_j + \sum_{i=1}^{C} b_i * n_i * (n_i - 1)/2$$

where C, n_i, b_i denote the total number of the Hobbit block (that contain any Tor relay), the number of /24 blocks within the i-th cluster and a binary variable that becomes 1 if the i-th Hobbit block contains more than one Tor relays and 0 otherwise. Note that I denote the number of /24 blocks by n_i rather than the number of individual addresses, in order not to over-estimate[6] the efficiency improvement by Hobbit blocks. Tor had 7,023 active relays at the time of my measurement. Given a list of active Tor relays, I identified Hobbit blocks that contain any Tor relay and calculated the above formula. The number of the address pairs obtained by the above formula was 66 times greater than the number of the address pairs that Ting can resolve without the imputation (i.e., the total number of Tor relay pairs, 7,023*7,022/2). This result demonstrates that Hobbit blocks can improve the efficiency of latency estimation.

4.2.2.2 Evaluating estimation error

The imputation may cause increased estimation error, that is, the difference between measured and estimated values. I evaluate the estimation error caused by the imputation. I use the ZMap ICMP Echo Request dataset. I obtain the RTT measurements (from the source of the ZMap scan) to Tor relay nodes and all the addresses in the Hobbit blocks having any Tor relay node. For each of the Hobbit blocks, I compute the median of RTTs of all the Tor relays within the block and regard it as the estimated RTT of all the other addresses in the block (i.e., the addresses that are not Tor relays). For each address with the estimated RTT, I

---

[6] It may not be useful to estimate the RTTs between the addresses within the same /24.

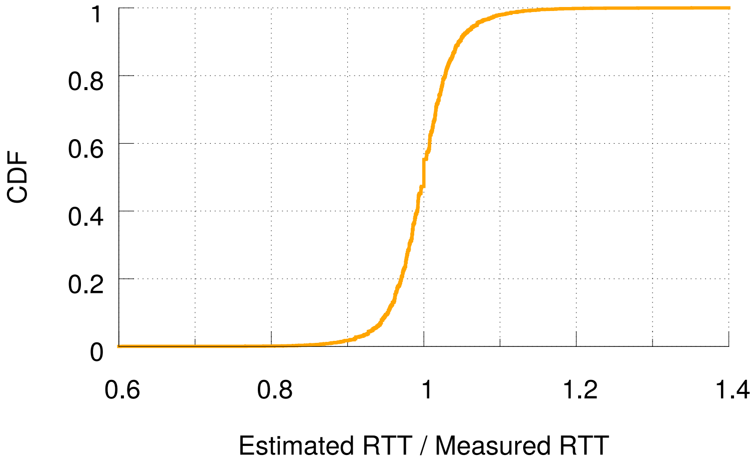


Figure 4.6: The distributions of the latency estimation error.

calculate the estimation error as the estimated RTT divided by the measured RTT (that I obtain from the ZMap dataset). Figure 4.6 shows the CDF of the estimation error. About 96.15% of the addresses have the estimation error between 0.9 and 1.1 and only about 0.3% of the addresses have the error greater than 1.2 or less than 0.8. This estimation error result is comparable with those of existing latency estimation techniques [59, 60, 79, 82]. This result reinforces that Hobbit blocks can be used for latency prediction and thus improve the efficiency of existing latency estimation techniques.

4.3 Comparing with alternative ways of improving efficiency

My approach to improving the efficiency of network measurement is similar with the approach of iPlane (that is, to aggregate Internet addresses to reduce measurement loads). However, iPlane aggregates addresses by either their geographic locations or BGP prefixes whereas Hobbit uses last-hop routers. I compare these different criteria to show that the last-hop router is a better criterion at least for latency prediction than the others. I also analyze their relationships.

4.3.1 Comparing the homogeneity of different aggregates

I compare the different criteria of aggregation by evaluating the correlation of RTTs within aggregates generated by using different criteria (that is, the aggregates of geographically co-located addresses, the aggregates of addresses having common BGP prefixes, and the aggregates of addresses having common last-hop routers, i.e., Hobbit blocks). I first describe how to obtain geographic locations of Internet addresses and then present the comparison results.

4.3.1.1 Obtaining geographic locations

Reverse DNS based geolocation Many ISPs encode several properties of Internet hosts (such as geographic locations, link type, IP allocation type, and subscription type) in their reverse DNS (rDNS) names. RDNS based geolocation techniques [2, 37] discover and interpret location hints in rDNS names to infer geographic locations. I use rDNS based techniques[7] for geolocation. There exist alternative approaches to geolocation such as inferring geographic locations from RTT measurements [26, 30–33] or relying on public databases (e.g., MaxMind GeoLite dataset [68]). However, I cannot use the RTT based techniques because one of my goals is to study the relationship between RTTs and geographic locations. The MaxMind dataset has shown to be much less accurate than rDNS based techniques at the city level [37]. Thus I mainly use a rDNS-based geolocation technique.

---

[7] Stale rDNS name may cause incorrect inferences of geographic locations and thus may affect my results by causing under- or over-estimation of the correlation of geographic locations. However, given that rDNS mis-naming only occurs in a small portion (0.5%) of IP addresses [83], my results still can be reliable especially if the degree of the correlation is very low or high.

I consider two representative rDNS-based geolocation techniques, undns [2] and DRoP [37, 84]. Undns extracts location information from rDNS names using a hand-crafted rule set that has been evolved over a decade. DRoP uses a machine learning technique (a decision tree) to extract location hints such as airport codes (IATA and ICAO [85]), CLLI [86], and UN/LOCODE [87] from rDNS names.

Selecting between DRoP and undns I first tried to apply both DRoP and undns and only take the common results in order to improve the reliability of geolocation. However, I observed that the results of DRoP and undns do not agree even at the country level for many rDNS names. This suggests that one of the techniques might have very low accuracy. To investigate this problem, I performed a simple experiment that compares the accuracy and coverage of DRoP and undns. First, I obtained all IPv4 PTR records (i.e., rDNS names) from project Sonar IPv4 rDNS dataset [88] that contains rDNS lookup results for all IPv4 addresses. I then applied DRoP and undns to all the rDNS names. I compared their coverage by counting the number of rDNS names they decoded. DRoP and undns decoded 56.13M and 294.75M rDNS names, respectively. To compare the accuracy, I used the MaxMind GeoLite dataset because MaxMind datasets are highly accurate at the country level [89]. Specifically, I compared the locations determined by undns and DRoP against those in the MaxMind GeoLite dataset at the country level. Undns yielded the same results with the Maxmind for about 97.8% of the rDNS names it resolved. On the other hand, DRoP agreed with the Maxmind for only about 51.1% of the rDNS names it resolved.

It may be meaningless to know the locations of not routed addresses, so I

Rank	Keyword	Rate (%)	Interpretation by DRoP
1	mta	22.67	IATA code of the airport in New Zealand
2	res	14.94	IATA, Argentina
3	sta	12.10	IATA, Denmark
4	asm	7.29	IATA, Eritrea
5	vic	6.44	IATA, Italy
6	ppp	6.41	IATA, Australia
7	cable	4.00	UN/LOCODE, Canada
8	kya	3.27	IATA, Turkey
9	cpe	3.17	IATA, Mexico
10	lns	2.75	IATA, USA
11	dsl	1.83	IATA, Sierra Leone
12	biz	1.81	IATA, Papua New Guinea
13	qld	1.30	IATA, Algeria
14	nas	1.29	IATA, Bahamas
15	stx	1.16	IATA, US Virgin Islands

Table 4.2: The top 15 keywords that are misinterpreted by DRoP.

performed the same experiment only on the rDNS names of routed addresses[8] instead of all rDNS names. The results were similar. Undns and DRoP resolved 288.3M and 54.05M rDNS names while achieving the accuracy of 98% and 52.3%.

To understand what causes the low accuracy of DRoP, I analyze the keywords within the rDNS names that were resolved to incorrect locations (at the country level) by DRoP. Table 4.2 shows the top 15 most common keywords along with their percentages and how they are interpreted by DRoP. Intuitively, many keywords do not appear to indicate geographic locations. For example, “res” and “biz” are likely

---

[8] I obtained a list of advertised BGP prefixes from RouteViews dataset [90], and classified addresses that match at least one BGP prefix as routed addresses.

to indicate “residential” and “business” customers, and “mta” and “ppp” may stand for “Mail Transfer Agent” and “Point-to-Point Protocol”. DRoP interprets these keywords as location hints such as UN/LOCODE and IATA codes, resulting in the incorrect inferences of geographic locations.

Finding geographically co-located addresses Since undns provides high coverage and accuracy than DRoP, I use undns for geolocation. Undns resolves 288.3M rDNS names to geographic locations when applied to all the routed IPv4 addresses. Among these, I only select the locations that are confirmed to be correct at the country level by the MaxMind GeoLite dataset. As a result, I end up obtaining the geographic locations of 282.4M IPv4 addresses. I consider addresses to be geographically co-located if they are in the same city. I aggregate geographically co-located addresses. I also consider AS numbers in aggregation because addresses that are in different ASes may have very different attributes even if they are geographically co-located. The iPlane also aggregates addresses that are in the same AS and location. I call the aggregates of addresses that are geographically co-located and are in the same ASes GeoASN blocks.

4.3.1.2 Comparing the correlation of RTTs

I measure the correlation of RTTs using a method in Section 4.2.1, that is, to compute the difference between the CV of /24 and the CV of the aggregates that the /24 belongs to. I compare the CV differences of Hobbit blocks, GeoASN blocks and BGP blocks that are the aggregates of addresses routed through common BGP

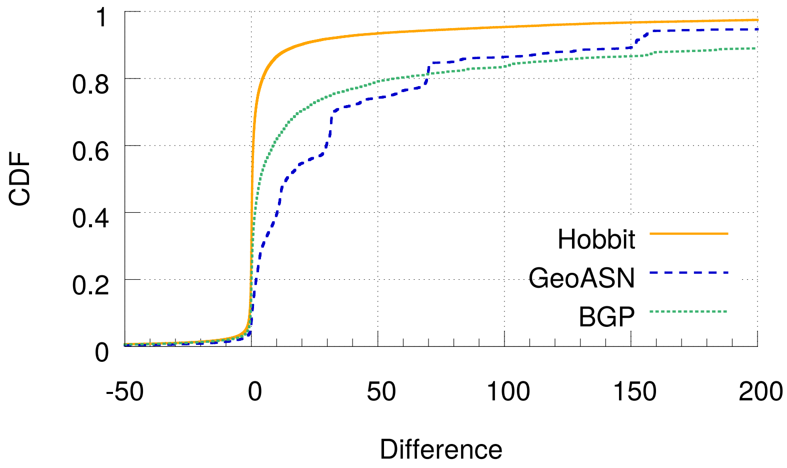


Figure 4.7: The CDF of the differences between the CVs of /24 blocks and those of the Hobbit, GeoASN and BGP blocks that the /24s belong to.

prefixes.[9]

Figure 4.7 shows the CDF of the CV differences for different aggregates. The difference values for Hobbit blocks tend to be much smaller than those for BGP and GeoASN blocks. About 90% of the differences are less than 20 for Hobbit blocks whereas only about 55% and 70% of the differences for GeoASN and BGP blocks are less than 20. This result demonstrates that only Hobbit blocks can be useful for latency estimation.

4.3.2 Analyzing the relationship between different aggregates

Hobbit blocks are generally finer-grained than GeoASN and BGP blocks in that Hobbit blocks have higher degrees of RTT correlations (as shown in the above section). I analyze whether addresses within the same Hobbit blocks have the same geographic locations or BGP prefixes.

Geographic co-locality within Hobbit blocks A simple method for quan-

---

[9] I identify BGP prefixes from RouteViews dataset [90]

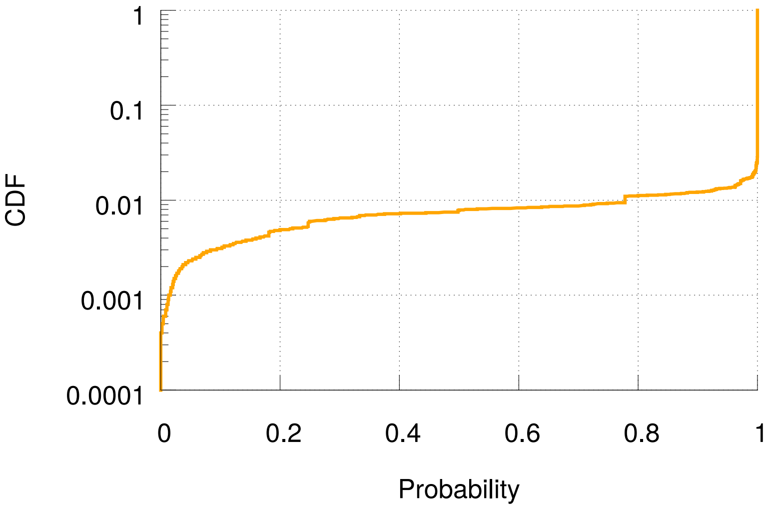


Figure 4.8: The CDF of the probability that an address will be in the same location to another address within the same Hobbit block.

tifying the correlation of geographic locations within Hobbit blocks is to count the number of distinct locations within Hobbit blocks. However, the number of distinct locations may not be an accurate measure of the correlation for a couple of reasons: 1) it does not reflect the sizes of Hobbit blocks and 2) the ratio of each location within Hobbit blocks having multiple locations. The sizes of Hobbit blocks need to be considered in that smaller blocks are more likely to have higher degrees of correlation (because smaller blocks have smaller numbers of addresses). The ratio of each distinct location also needs to be considered because different distributions of ratios may indicate different degrees of the correlation. For example, if two Hobbit blocks both have 3 locations X, Y and Z but the distribution of X, Y and Z is <90%, 9%, 1%> in one block, and <40%, 30%, 30%> in the other, the correlation within these two blocks need to be interpreted differently. For these reasons, I use an alternative metric, that is, the probability that an address will be in the same geographic location with another arbitrary address in the same Hobbit block. The

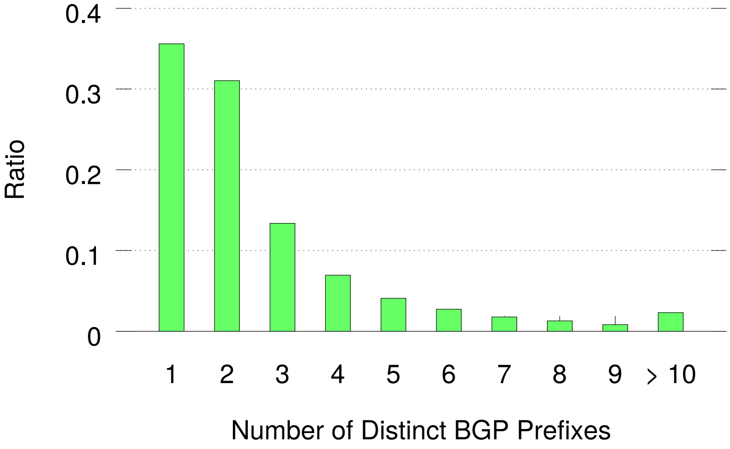


Figure 4.9: The distribution of the number of distinct BGP prefixes within Hobbit blocks.

probability for an address that is located in G and in Hobbit block B is calculated as: $(N_{B,G} - 1)/(N_B - 1)$ where $N_{B,G}$ represents the number of addresses that are located in G and in Hobbit block B, and N_B represents the total number of addresses in block B. For example, if a Hobbit block contains 4, 2 and 1 addresses located in X, Y and Z, respectively, then the probability for an address located in X will be 3/6. The probability is computed for each individual address and thus not affected by various sizes of Hobbit blocks. This method also better represents the ratios of the locations within blocks because the probability largely depends on the ratios of the locations.

Figure 4.8 shows the CDF of the probabilities (on a log scale). About 98% of the addresses have the probability 1. This result demonstrates that geographic locations are strongly correlated within Hobbit blocks.

The relationship between Hobbit blocks and BGP prefixes Hobbit blocks may be expected to have a single BGP prefix in that Hobbit blocks are finer-grained than BGP blocks. I count the number of BGP prefixes within Hobbit blocks. Figure 4.9 depicts the distribution of the numbers of distinct BGP prefixes. Only about 35.6% of the Hobbit blocks have a single BGP prefix. This result reinforces that fragmented allocations of IPv4 addresses are common in the Internet and thus the aggregation of even discontiguous prefixes is needed.

4.4 Summary

In this chapter, I showed that the efficiency of Internet topology mapping can be improved by selecting destinations from Hobbit blocks instead of /24 blocks. I analyzed the measurement overhead of Hobbit and analyzed its effects on the efficiency improvement. I also analyzed the changes in last-hop routers of IP addresses over time and showed that the last-hop routers (and thus Hobbit blocks) tend to be very stable over time. I compared the correlation of RTTs within Hobbit blocks to the correlation within aggregates of addresses having 1) common BGP prefixes and 2) geographic locations. Hobbit blocks have shown to be highly correlated with RTTs whereas the aggregations by BGP prefixes and geographic locations had low correlations with RTTs. I actually showed that Hobbit blocks can improve the efficiency of existing latency estimation techniques. I also measured the correlation of geographic locations within Hobbit blocks. The addresses within Hobbit blocks are very likely to be geographically co-located. This suggests that Hobbit blocks can be used for geolocation.

Hobbit blocks are as homogeneous as /24s and, at the same time, are larger than /24s. This means that Hobbit blocks are better aggregates than /24s as supported by topology mapping efficiency improvement by Hobbit blocks. Although Hobbit blocks are generally smaller than BGP blocks, the homogeneity of Hobbit blocks are much higher. I showed that Hobbit blocks can be used for latency estimation. Given that Hobbit blocks tend to be stable over time, the cost of maintaining Hobbit blocks is not high. For practical use of Hobbit blocks, I run Hobbit every month and make the results publicly available.

In the next chapter, I describe a different kind of application of Hobbit blocks. I use Hobbit blocks to improve the representativeness of network measurement.

Chapter 5: IMPROVING REPRESENTATIVENESS OF ACTIVE MEASUREMENT

Active measurement studies draw conclusions from the addresses measured by active probes. The issue is that measured addresses may not be representative of the Internet or destination networks in that addresses may not respond to active probes at the measurement time for various reasons. A lack of representativity is a significant problem because the importance of unexpected observations (e.g., higher RTTs than typical default timeout values [72]) can be under-estimated if addresses with such observations are under-represented, or very rare events may be considered more important than they actually are by over-representation.

In this chapter, I demonstrate that Hobbit blocks can be used for improving the representativeness of active measurement by developing a technique that improves the representativeness using Hobbit blocks as input.

I first develop a methodology that quantifies the representativeness of active measurement and show that active measurement results may not be representative even when the entire public IPv4 addresses are probed. I then describe a technique that improves the representativeness. My approach is to use weighting adjustment [25] that is one of the most common bias correction techniques in surveys. Weighting adjustment improves the representativeness of surveys by assigning different weights to different respondents depending on how many people they can represent. I adapt weighting adjustment to active measurement by using Hobbit blocks. I demonstrate that the weighting adjustment improves the representativeness by identifying key factors that influence the performance of the weighting adjustment and evaluating the performance with various combinations of the factors.

5.1 Motivation: Demonstrating unrepresentativeness

In this section, I demonstrate that the results of active measurements can be unrepresentative. I develop a methodology that quantifies the representativeness of active measurements and apply the methodology to specific examples.

5.1.1 The representativeness of Internet-wide scanning

An optimized scanning tool called ZMap [20] enables fast probing of the entire IPv4 addresses (when only a few probes are required for each destination). For example, ZMap can perform a full IPv4 ICMP Echo Request scan in only about 12 hours according to the ZMap dataset [21,62]. Thus, it may appear that representativeness is not an issue in active measurement. However, even if all the IPv4 addresses are probed, the result may not be representative. A key issue is various availability of IPv4 addresses. While there are many addresses that always respond to probes, there are also some other addresses that intermittently respond for reasons such as turning devices off at night and dynamic address assignment [16]. This implies that low available addresses may not respond at the time of measurement and no responses can lead to bias towards highly available addresses. I measure the representativeness of the entire IPv4 address space scan to show that it may not be representative.

5.1.2 Measuring representativeness

5.1.2.1 Methodology

The goal is to quantify the representativeness of a measurement instance. My general approach is to compare the result of the instance with that of a more representative measurement instance. Specifically, I compare the distribution of a target variable obtained from the instance (that I call an estimated distribution) with that from a more representative instance (that I call a reference distribution). I consider that the distance between the estimated and reference distribution indicates the degree of the representativeness of the instance. (A smaller distance[1] would indicate a higher representativeness.)

I aim to measure the representativeness of the entire IPv4 space scan. Thus I obtain an estimated distribution by probing all IPv4 addresses and using each responded address as a data point. In order to generate a reference distribution, I perform a more representative measurement than the entire IPv4 space probing. The idea is to repeat probing. A single scan only captures the IP addresses that respond at the measurement time. In other words, it does not measure[2] the addresses that are temporarily not occupied or whose occupying devices are turned off at the measurement time. This suggests that, if I repeat the entire scan (possibly

---

[1] I use the Area Test statistic [91] to quantify the distance between distributions as I will describe in Section 5.2.3.

[2] No response may be considered as a piece of information especially when measuring reachability or availability [15, 16, 19]. However, non-responding addresses only tell reachability or availability but not other information. So I consider non-responding addresses are not measurable.

at different times), more addresses (particularly, low available addresses) can be measured. Therefore, I obtain a more representative distribution, i.e., reference distribution from the aggregate of the multiple IPv4 space scans.

I do not perform (active) measurements to obtain distributions. I instead use existing dataset, ZMap ICMP Echo Request dataset [21]. The dataset contains the results of weekly ICMP Echo Request scan of full IPv4 address space. It records whether each address has responded to ICMP probes and thus I can identify all responding addresses[3] using this dataset. I use round-trips times (RTTs) as a target variable. In other words, I compare the distribution of RTTs from a single snapshot with that from the aggregate of multiple snapshots. RTTs can be obtained from the dataset itself because it captures the timestamps when packets are sent and received. When aggregating multiple snapshots, if an address has multiple measurements of RTTs, I randomly select one.

5.1.2.2 Results

Figure 5.1 shows the CDFs of RTTs from the single and aggregated snapshots with various aggregation window sizes. As the window size increases, the difference between the estimated and reference distribution (i.e., the distributions from the single and aggregated snapshots, respectively) also increases. While the two distributions are very similar when the window size is 2, the distinction between them becomes clear with the window size greater than or equal to 12. The estimated distribution tends to under-estimate RTTs. When the window size is 12, the ratios

---

[3] I only consider the addresses that respond with ICMP Echo Reply messages to be responding

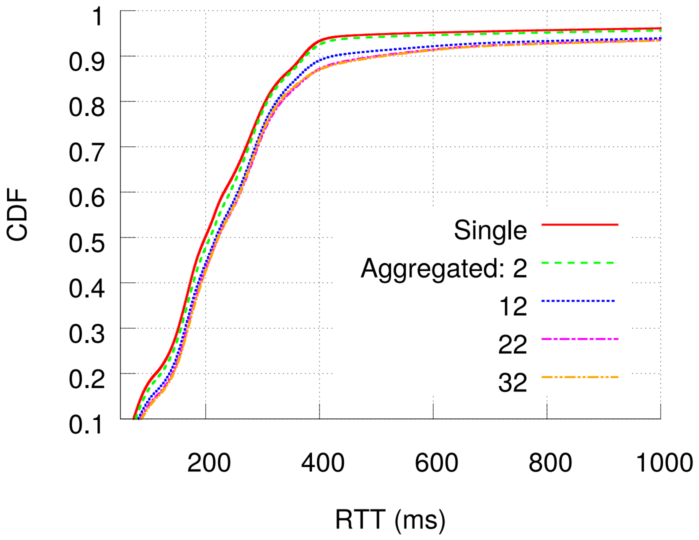
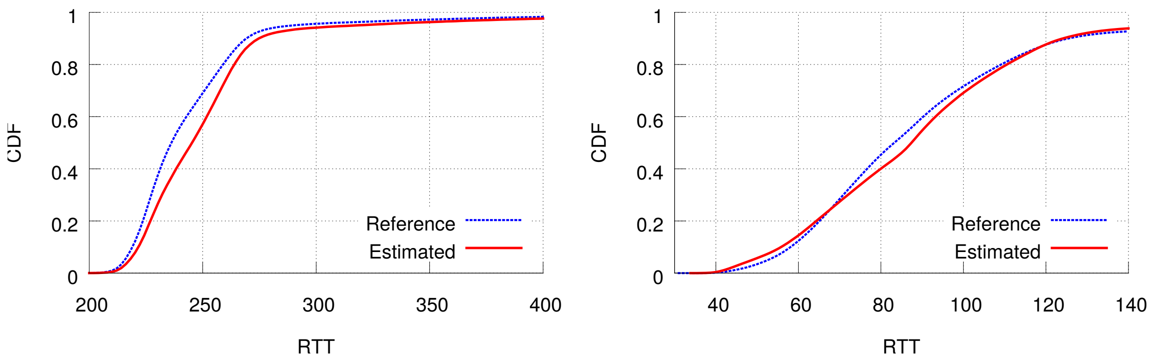


Figure 5.1: The CDFs of RTTs from a single and aggregated snapshots with various aggregation window sizes.

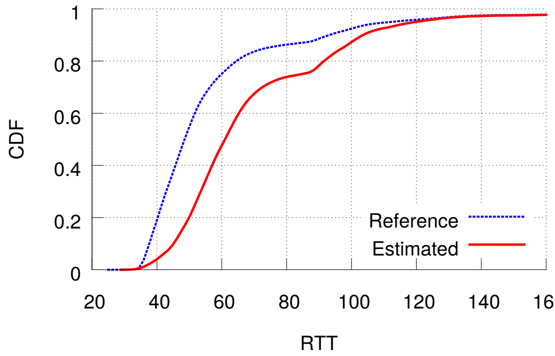
of the RTTs that are at most 100, 200 and 400ms are 18%, 50% and 93% in the estimated distribution whereas the ratios are 14%, 44% and 89% in the reference distribution. Considering that the source of RTT measurement (i.e., the source of the ZMap ICMP scan) is located at the University of Michigan, this may imply that the estimated distribution over-represents the addresses in the US and thereby under-represents the addresses outside the US. (I study in more detail the relationship between geographic locations and the representativeness in Section 5.5.2). This result shows that even the entire IPv4 space probing can be unrepresentative (possibly due to various availability of IPv4 addresses).

5.1.3 A challenge for improving the representativeness

It may appear that the representativeness of active measurements can be easily improved by additional measurements, e.g., by repeating probing. However, even with additional measurements, the representativeness may not improve. I describe



(a) AS 35805, Silknet (b) AS 5650, Frontier



(c) AS 6461, Abovenet

Figure 5.2: The RTT distributions from all the responding addresses (reference distribution) and the addresses having rDNS names (estimated distribution).

when additional measurements do not help with concrete examples.

5.1.3.1 The representativeness of subsets of addresses

Active measurement studies often measure or analyze only subsets of responding addresses. For example, many studies select addresses to probe or analyze based on reverse DNS (rDNS) names of the addresses [11, 14, 16, 19, 81, 92]. Since not all responding addresses have rDNS names and the addresses with rDNS names may not be representative of all the addresses, the selection based on rDNS names may generate a biased sample that undermines the representativeness. In general, the unrepresentativeness due to a biased sample can be mitigated by increasing the sample size. However, in the selection by rDNS names, it is not possible to increase the sample size because the selection is constrained by whether addresses have rDNS names. The repetitive measurement (that was described in Section 5.1.2.1) does not work either because the addresses not having rDNS names cannot be analyzed no matter how many times they are probed. Therefore, additional measurements cannot improve the representativeness of the results from the rDNS-based samples.

5.1.3.2 Motivating examples

I show that the rDNS-based samples can be unrepresentative. This motivates the development of a technique that improves the representativeness without additional measurements (because additional measurements do not alleviate bias in the rDNS-based samples). I measure the representativeness of some rDNS-based samples by using the methodology described in Section 5.1.2.1. I obtain an estimated distribution from the responding addresses having rDNS names and a reference distribution from all the responding addresses. I use RTTs as a metric, in other words, compare the distributions of RTTs. (I identify a list of responding addresses and their RTT measurements from the ZMap ICMP Echo Request dataset.) I consider each AS because many ISPs do not maintain rDNS names and thus the rDNS-based sample will obviously not be representative if I consider the entire IPv4 address space. To be specific, I compare the distribution of RTTs of all the responding addresses in an AS with that of the responding addresses that are in the same AS and have rDNS names.

Figure 5.2 depicts the reference and estimated distributions for three examples

ASes. The estimated distributions are different from the reference distributions in all the ASes. This suggests that the addresses having rDNS names do not well represent all the addresses, in other words, the rDNS-based samples are unrepresentative. Note again that additional measurements cannot improve the representativeness.

5.2 Applying weighting adjustment

Weighting adjustment is a common bias correction technique in surveys. I exploit weighting adjustment to improve the representativeness of active measurement. In this section, I describe how to apply weighting adjustment to active measurement with concrete examples. I also explain a methodology for quantifying the improvement of the representativeness by weighting adjustment.

5.2.1 How to perform weighting adjustment?

Weighting adjustment is to assign different weights to different entities (i.e., respondents in surveys and responding addresses in active measurement). Specifically, weighting adjustment can be considered as giving higher and lower weights to the addresses in under- and over-represented groups, respectively. Hence, the first step of weighting adjustment is to determine the groups of the addresses that can represent each other. I use Hobbit blocks as the address groups for weighting adjustment. Since the addresses within the same Hobbit blocks share common last-hop routers, they are likely to have similar characteristics (Section 4.2.1, 5.5).

Since the weights for the addresses that are in the same address groups are the same, I can consider that a weight is assigned to each address group. The weight of each group should be proportional to its share in a population and inversely proportional to its share in a sample. I define a population as all responding addresses[4] because I consider active measurement. A sample consists of all the measured addresses (i.e., the addresses that actually responded to the active probes of the measurement to which I apply weighting adjustment).

Hence, I compute the weight of each Hobbit block as the number of the addresses in the population and in the Hobbit block divided by the number of the measured addresses in the Hobbit block. Using the computed weights, I generate the weighted version of the estimated distribution, which I call an adjusted distribution.

5.2.2 Example Applications

I apply weighting adjustment to some example measurements. First, I apply weighting adjustment to the single snapshot of the entire IPv4 space scan that was described in Section 5.1.2. Specifically, I adjust the estimated distribution of RTTs (that is obtained from the single snapshot) by weighting adjustment obtaining an adjusted distribution. I then compare the adjusted distribution with the reference distribution (that is obtained from the aggregate of multiple snapshots) to see if weighting adjustment improves the representativeness of the estimated distribution. Figure 5.3 depicts the adjusted, estimated and reference distributions. The adjusted distribution is very close to the reference distribution. Considering that the estimated distribution is clearly different from the reference distribution, this result shows the improvement of the representativeness by weighting adjustment.

---

[4] I approximate the number of all responding addresses as the number of responding addresses in the aggregate of the multiple entire IPv4 space scans.

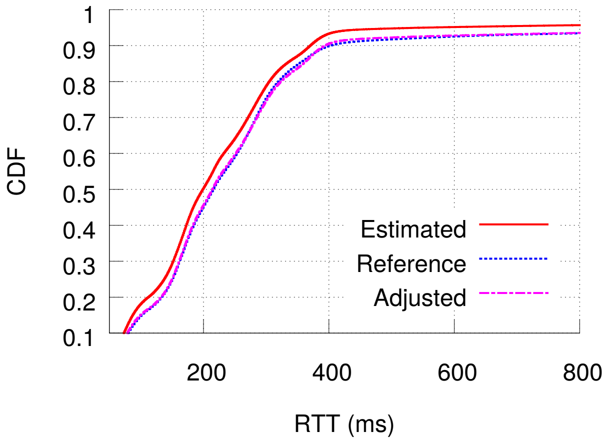
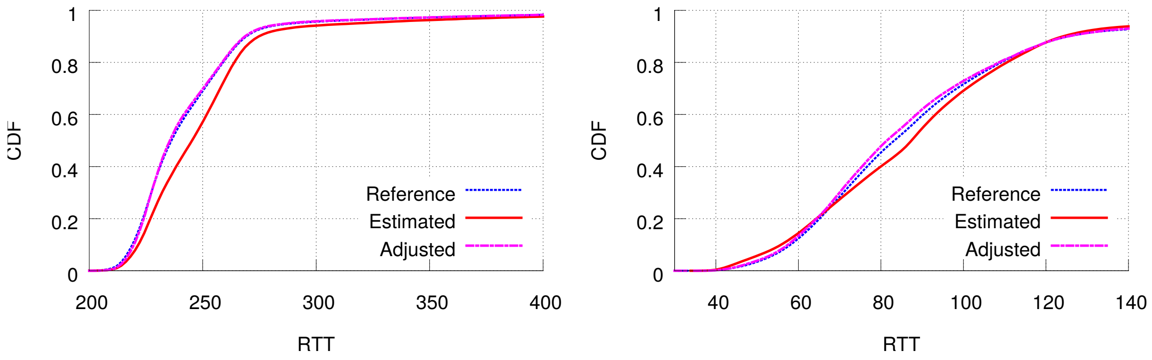


Figure 5.3: The adjusted, estimated and reference distributions of RTTs.

I also apply weighting adjustment to the three rDNS-based samples that I considered in Section 5.1.3. Figure 5.4 shows the estimated, reference and adjusted distributions for the three example ASes. Weighting adjustment substantially alters the estimated distributions. In the results of AS 35805, the adjusted distribution is nearly identical to the reference distribution. The adjusted and the reference distributions of AS 5650 are slightly different but the difference between them is much smaller than that between the reference and estimated distributions. For AS 6461, the adjusted distribution is quite different from the reference distribution particularly for RTT values less than 60ms. However, the adjusted distribution is still much closer to the reference distribution than the estimated distribution. These results indicate that weighting adjustment can actually improve the representativeness.

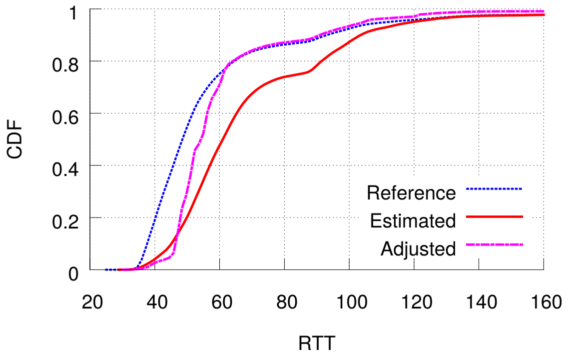
5.2.3 Quantifying the performance of weighting adjustment

I consider that weighting adjustment improves the representativeness if the distance between the reference and adjusted distributions is smaller than that between the reference and estimated distributions. In order to quantify the represen-



(a) AS 35805, Silknet

(b) AS 5650, Frontier



(c) AS 6461, Abovenet

Figure 5.4: The RTT distributions from all the responding addresses (reference distribution) and the addresses having rDNS names (estimated distribution), and the adjusted distributions by using Hobbit blocks.

tativeness improvement by weighting adjustment, I measure the distance between two distributions (specifically, CDFs) by using a summary statistic called an Area Test [91]. The Area Test statistic between two CDFs is defined as the area between the CDF curves normalized by the width of the CDF (that is, the difference between the maximum and minimum x values). I also considered the two-sample Kolomogorov-Smirnov (KS) Test [93]. However, the KS Test only uses a single x value that maximizes the difference between the corresponding y values whereas the Area Test covers all x values in that it computes area. The Area Test can be more appropriate for quantifying the difference between CDFs. For example, in

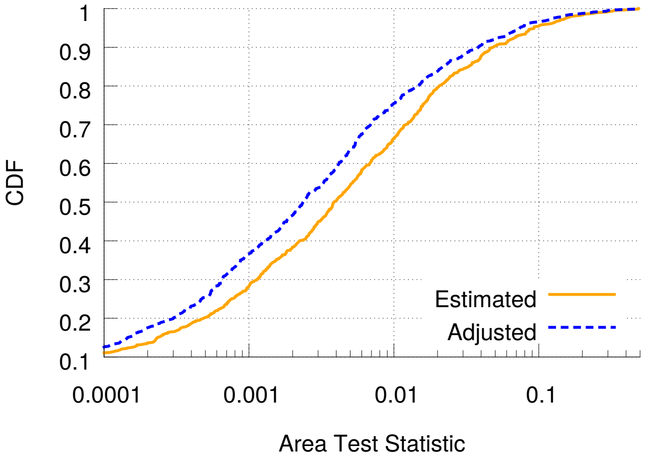


Figure 5.5: The CDFs of the Area Test statistics between 1) the reference and estimated distributions and 2) the reference and adjusted distributions.

Figure 5.4c, the adjusted distribution is much closer to the reference distribution than the estimated distribution. But the KS test statistic between the reference and estimated distributions (0.35) is similar to that between the reference and adjusted distributions (0.33). On the other hand, the Area Test statistic between the reference and estimated distributions is 0.13 whereas the statistic between the reference and the adjusted distributions is 0.06.

I have applied weighting adjustment to the rDNS-based samples of the three example ASes in Section 5.2.2. I now apply weighting adjustment for every AS and quantify the representativeness improvement by using the Area Test statistics. To be specific, for each AS, I compute the Area Test statistics for two pairs of the distributions: 1) the reference and estimated distributions and 2) the reference and adjusted distributions. Figure 5.5 depicts the CDFs of the statistics of all the ASes. Different lines represent distinct statistics, i.e., the statistics for distinct pairs of the distributions. The statistics between the reference and adjusted distributions tend to be smaller than those between the reference and estimated distributions. This suggests that weighting adjustment generally improves the representativeness.

Although the results in this section show the possibility of improving the representativeness by weighting adjustment, they do not thoroughly evaluate the effectiveness of weighting adjustment. For thorough evaluation and optimization of weighting adjustment, I identify key factors affecting the performance of weighting adjustment and analyze their effects on the performance in the next sections.

5.3 Analyzing the performance of weighting adjustment

Weighting adjustment is not a panacea. It may not improve or even worsen representativeness in certain cases. In this section, I describe the key factors that influence the performance of weighting adjustment and analyze the performance with various combinations of the factors.

5.3.1 What are the key factors affecting the performance?

Weighting adjustment is basically to estimate the measurement values of unobserved addresses (that is, the addresses that are not in a sample but a population) from those of observed addresses (that is, the addresses that are in the sample). Therefore, the properties of samples largely affect the performance of weighting adjustment. An obvious influential attribute is the size of a sample. Depending on the sample size (i.e., the number of the observed addresses), the unobserved addresses may or may not be well represented by the observed addresses. Another important factor is the distribution of sample points. In weighting adjustment, addresses are grouped into blocks (such as /24 blocks or Hobbit blocks) and the values of unobserved addresses are estimated from those of the observed addresses within the same blocks. This suggests that the distribution of sample points (i.e., observed addresses) across the blocks is an influential factor in determining the performance of weighting adjustment. For example, if sample points are concentrated in small blocks while large blocks have a few sample points, the effectiveness of weighting adjustment may be limited even if the sample size is large. I analyze both the effects of sample sizes and the distributions of sample points.

5.3.2 Methodology

5.3.2.1 Generating samples

I first describe from where I select sample points, i.e., the addresses to measure. In active measurements, the addresses that (always) do not respond to active probes cannot be measured and there are many such addresses. Hence, if I choose destination addresses from the entire IPv4 space, many of the chosen addresses may not be measurable. So I select destination addresses from a list of the addresses that have previously responded to probes. I obtain those addresses from the ZMap ICMP Echo Request dataset. This sort of destination selection scheme is actually used by researchers for the purpose of increasing a hit ratio [81, 94–97].

In order to evaluate the effects of the distribution of sample points across address groups, I select sample points in three different ways. First, I select an address group with the probability proportional to the size of the group[5], and then

---

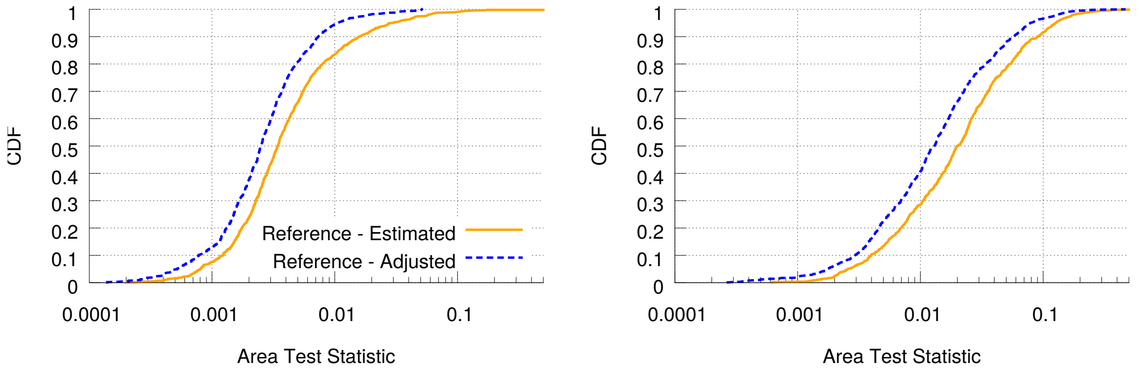
[5] I use Hobbit blocks as the address groups because they have shown a better performance than /24 blocks in Section 5.2.3. The size of a group is the number of responding addresses within the group.

randomly choose an address from the selected address group. I repeat this until I obtain a sample of a desired size. The samples generated in this way, which I call *good samples*, are favorable to weighting adjustment in that larger address groups are likely to have more sample points. Second, I generate samples in the opposite way. I select address groups with the probability “inversely” proportional to the address group size (and then randomly select sample points from the chosen address groups). In these samples, sample points are likely to be concentrated in small blocks. I call these samples *bad samples*. Lastly, I randomly select an address group and then choose a sample point from the group. In this way, sample points are likely to be uniformly distributed over the address groups and thereby I call the generated samples *uniform samples*. By using these diverse methods and varying sample sizes, I evaluate the performance of weighting adjustment with various kinds of samples in order to show that weighting adjustment generally improves the representativeness of samples.

5.3.2.2 Measuring representativeness

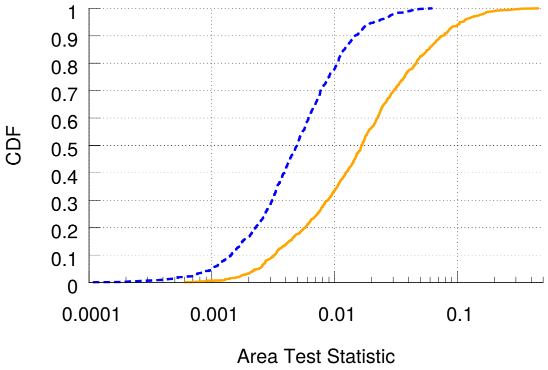
For each of the generated samples, I build an estimated distribution from the addresses within the sample and compare it to the reference distribution that I obtain using all the responding addresses. I again use RTTs as a metric, i.e., compare the distributions of RTTs. Note that some addresses in the sample may not be responding and thus not be included in the construction of the estimated distribution because the availability of addresses changes over time [16].

I apply weighting adjustment using Hobbit blocks to the estimated distribu-



(a) Good Samples

(b) Bad Samples



(c) Uniform Samples

Figure 5.6: The CDFs of the Area Test statistics of the ASes for three types of samples. Two different lines represent the statistics between the reference and 1) estimated distribution and 2) the adjusted distribution by using Hobbit blocks.

tions and obtain adjusted distributions. I quantify the differences between 1) the reference and estimated distributions and 2) the reference and adjusted distributions by calculating the Area Test statistics. I perform this analysis for each AS[6] and present the CDFs of Area Test statistics of the ASes.

5.3.3 The effects of the distribution of sample points

Figure 5.6 shows the CDFs of the Area Test statistics of the ASes for three kinds of samples. The sizes of the samples are all 20% of the population. Weighting

---

[6] I only consider the ASes that have at least 10,000 responding addresses so that I can analyze with small-sized samples, e.g., samples having only 1% of all the responding addresses.

adjustment improves the representativeness for all the kinds of samples. In the result of *good samples*, the Area Test statistics between the reference and estimated distributions (i.e., before weighting adjustment) are at most 0.01 for about 83.5% of the ASes whereas about 94.3% have at most 0.01 Area Test statistic between the reference and adjusted distributions (i.e., after weighting adjustment). *Good samples* are equivalent to randomly chosen samples because, if sample points are randomly chosen, the number of the sample points in each Hobbit block will be proportional to the size of the block. Thus, this result suggests that even random samples (containing 20% of the population) may not be representative. Again, this happens because of time-varying availability of addresses. Even if samples are randomly generated, the results may not be representative if some addresses in the samples do not respond at the measurement time.

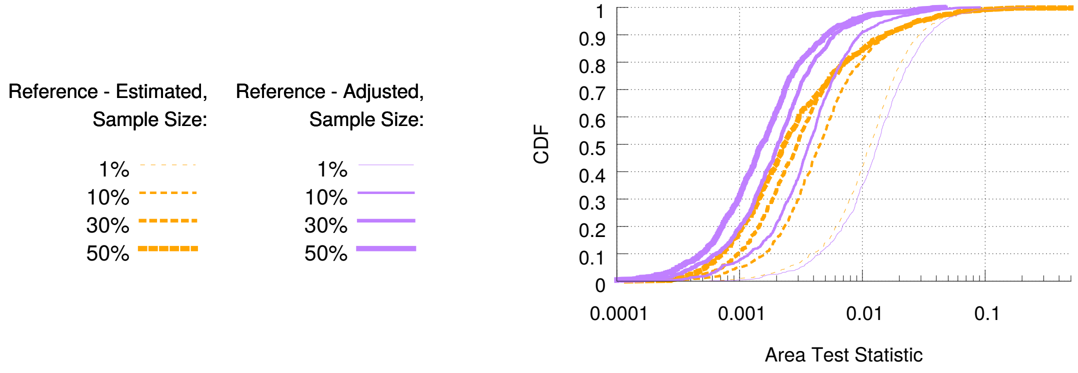
The Area Test statistics before weighting adjustment for *bad* and *uniform samples* are very similar. However, the statistics after weighting adjustment are much smaller in *uniform samples* than in *bad samples*. In other words, the degree of representativeness improvement by weighting adjustment is much larger for *uniform samples* than *bad samples*. This indicates that the distribution of sample points greatly influences the performance of weighting adjustment. Nevertheless, even for *bad samples*, weighting adjustment still improves the representativeness. About 28.6% and 91.4% of the ASes have at most 0.01 and 0.1 Area Test Statistic before weighting adjustment whereas about 40.5% and 96.6% have at most 0.01 and 0.1 after weighting adjustment.

5.3.4 The effects of the sample sizes

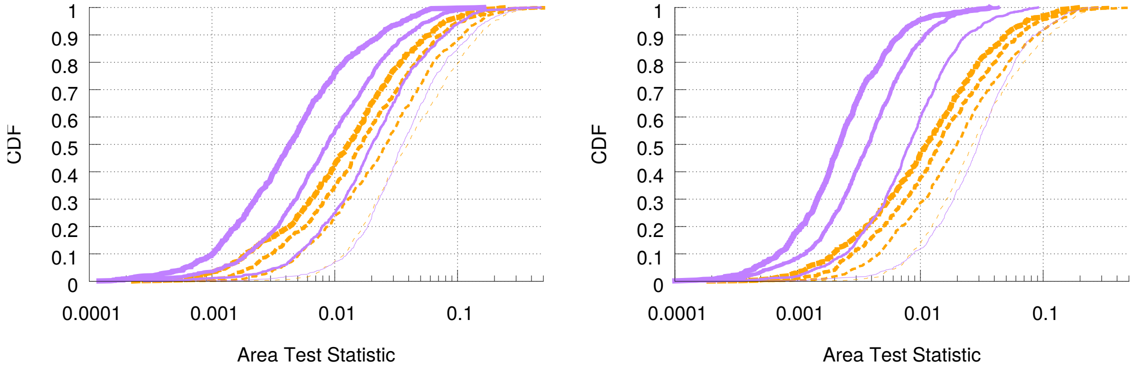
Figure 5.7 shows the CDFs of the Area Test statistics for the three kinds of samples with various sample sizes including 1%, 10%, 30% and 50% of the population. As the sample size increases, the statistics reduce for all the kinds of samples both with and without weighting adjustment. The degree of representativeness improvement by weighting adjustment also increases as the sample size increases. When the sample size is 50%, with weighting adjustment, about 95% of the ASes have at most 0.01 Area Test statistic for *uniform samples* whereas only about 46% have at most 0.01 without weighting adjustment. Weighting adjustment substantially decreases the Area Test statistics for *good* and *bad samples* as well when the sample sizes are large.

The Area Test statistics with weighting adjustment are even smaller than those obtained from larger samples without weighting adjustment. With weighting adjustment, the statistics for 30% samples are smaller than those for 50% samples without weighting adjustment in all cases. In *uniform samples*, even the statistics for 10% samples with weighting adjustment tend to be smaller than that for 50% samples without weighting adjustment.

When sample sizes are 1%, weighting adjustment does not perform well. For *bad samples*, the statistics with weighting adjustment are only slightly lower than those without weighting adjustment. The benefit of weighting adjustment is unclear for *uniform samples* in that the ratio of ASes having at most 0.01 and 0.1 Area Test statistic decreases and increases, respectively, after weighting adjustment. For *good*



(a) Good Samples



(b) Bad Samples

(c) Uniform Samples

Figure 5.7: The CDFs of the Area Test statistics of the ASes for samples with various sizes and types. Dashed lines indicate the statistics between the reference and estimated distributions and the solids lines indicate those between the reference and adjusted distributions. The thicker the line is, the larger the corresponding sample size is.

samples, weighting adjustment slightly worsen the representativeness. This could be because some Hobbit blocks might not have any sample point when the sample size is small. If this happens, weighting adjustment can result in under-representing the addresses in the blocks having no sample points and over-representing the addresses in the blocks having sample points (and thereby worsening the representativeness). This suggests that the properties of the address groups used in weighting adjustment such as the sizes of the groups influence the performance of weighting adjustment as well as the properties of the samples. I analyze the effects of the address groups on the performance of weighting adjustment in the next section.

5.3.5 Summary

The results in this section have shown that active measurements of sampled addresses may not be representative even if samples are randomly generated and large (namely, 50% of the population). Weighting adjustment improves the representativeness of samples in most cases. While the degree of the representativeness improvement is maximized when sample points are uniformly distributed over the address groups, the representativeness after applying weighting adjustment is maximized when samples are randomly selected (i.e., in *good samples*). This suggest that, when selecting a subset of IPv4 addresses and measuring them by active probes, the representativeness can be maximized by randomly selecting the destination addresses and applying weighting adjustment. Even if random sampling is not possible due to constraints on selecting destination addresses, e.g., the existence of rDNS names (as described in Section 5.1.3), it is beneficial to apply weighting adjustment. I have shown that weighting adjustment improves the representativeness even for *bad samples*. However, if samples are very small, weighting adjustment might not improve or even worsen the representativeness. I alleviate this issue in the next section by optimizing the selection of address groups in weighting adjustment.

5.4 The effects of address groups on weighting adjustment

In this section, I perform and compare the results of weighting adjustment using two different types of address groups. By this comparison, I analyze the effects of the address groups on the performance of weighting adjustment and develop a way of combining different types of address groups to improve the performance.

5.4.1 What are the key properties affecting the performance of weighting adjustment?

Weighting adjustment estimates the values of unmeasured addresses from those of the measured addresses that are in the same address groups. Therefore, grouping of addresses is a significant factor in determining the performance of weighting adjustment. The two key attributes of address groups affecting the performance are the sizes of the groups and the similarity of the addresses within the groups. Ideally, the address groups should have both large sizes and high degrees of similarity because they are both beneficial for weighting adjustment. If the group sizes are small, some groups can have no measured addresses and thus the values of the unmeasured addresses in those groups cannot be estimated, resulting in reduced representativeness. The similarity of the addresses is also an important factor. If the addresses within the same groups have very different characteristics, the measured addresses may not well represent the unmeasured addresses in the same group.

The challenge is that there is a trade-off between the group sizes and the similarity of the addresses within the groups. If address groups are fine-grained, the degree of the similarity will be high but the sizes of the groups will be small.

The sizes can be increased by coarser-grained grouping but then the degree of the similarity may be decreased. For example, Hobbit blocks and BGP blocks that refer to the groups of the addresses that are routed through common advertised BGP prefixes have their own advantage. The addresses within the same Hobbit blocks have common IP-level last-hop routers whereas the addresses within BGP blocks have common AS-level paths. Hence, the degrees of the similarity of the addresses within Hobbit blocks are generally higher than those within BGP blocks. On the other hand, the sizes of Hobbit blocks tend to be smaller than those of BGP blocks. By comparing weighting adjustment that uses Hobbit BGP blocks, I analyze the effects of the properties of the address groups on the performance of weighting adjustment.

5.4.2 Methodology

I obtain a list of advertised BGP prefixes from RouteViews dataset [90]. I identify the most specific matching prefix for each address and cluster the addresses with common corresponding prefixes into address groups, namely, BGP blocks.

I evaluate the performance of weighting adjustment that uses Hobbit BGP blocks by the methodology described in Section 5.3.2. To summarize, I draw samples from the previously responded addresses and obtain the estimated distributions from the samples. (I focus on the distribution of RTTs and obtain RTT measurements from the ZMap ICMP Echo Request dataset.) By applying the weighting adjustment to every sample, I derive the adjusted distributions. I quantify the differences in the estimated and adjusted distributions from the reference distributions by using the Area Test statistics. I compute the statistics for each AS and compare the distributions of the statistics of all the ASes.

In order to accurately measure the effects of the address groups, I need to use common samples for weighting adjustment using Hobbit blocks and that using BGP blocks because the performance of weighting adjustment is largely affected by the distributions of samples points across address groups (as shown in Section 5.3.3). I follow the sample generation scheme in Section 5.3.2.1 that creates *good*, *bad* and *uniform samples*. The issue is that the sample generation scheme is dependent on address groups.[7] If I generate samples using either Hobbit or BGP blocks, samples may be biased towards either of them. For example, if I use Hobbit blocks in generating *uniform samples*, the sample points will be uniformly distributed over Hobbit blocks but not necessarily over BGP blocks, and vice versa. To deal with this, I define the address groups to be the groups of addresses that are in the same Hobbit blocks "and" BGP blocks[8] and then select the address groups uniformly or depending on the sizes of the address groups, followed by the selection of sample points from the chosen address groups. This method can generate more balanced samples compared to when using either Hobbit or BGP blocks.

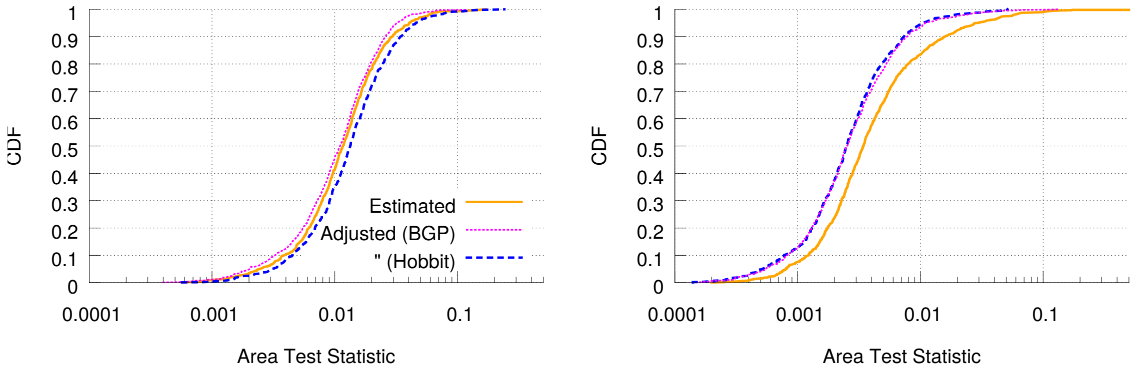
5.4.3 Results

Figure 5.8 shows the CDFs of the Area Test statistics of the ASes for *good samples* with various sample sizes (1%, 20% and 50%). The weighting adjustment

---

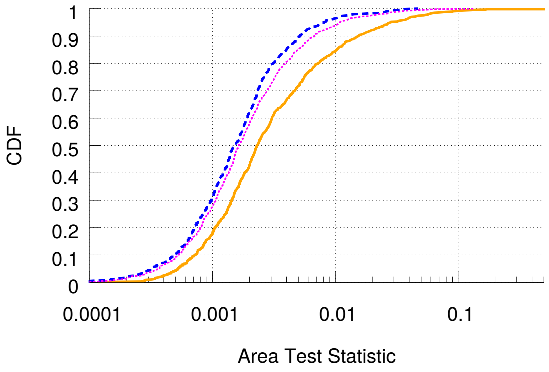
[7] Note that the generation of *good samples* is independent of address groups because *good samples* are equivalent to random samples, that is, the samples generated by the random selection of sample points.

[8] Note that I do not use this definition in weighting adjustment



(a) Sample Size: 1%

(b) 20%



(c) 50%

Figure 5.8: The CDFs of the Area Test statistics of the ASes for good samples. Three lines represent the statistics between the reference and 1) estimated distribution and 2) the adjusted distribution by using Hobbit blocks and 3) BGP blocks.

that uses BGP blocks effectively improves the representativeness of the samples. The Area Test statistics between the reference and the adjusted distributions by using BGP blocks tend to be smaller than those between the reference and the estimated distributions for all the sample sizes. Even when the sample size is 1%, weighting adjustment using BGP blocks improves the representativeness unlike that using Hobbit blocks. However, as the sample size increases, the performance of weighting adjustment using Hobbit blocks becomes better than that using BGP blocks. When the sample size is 50%, about 96.4% of the ASes have at most 0.01 statistic between the reference and the adjusted distributions when using Hobbit

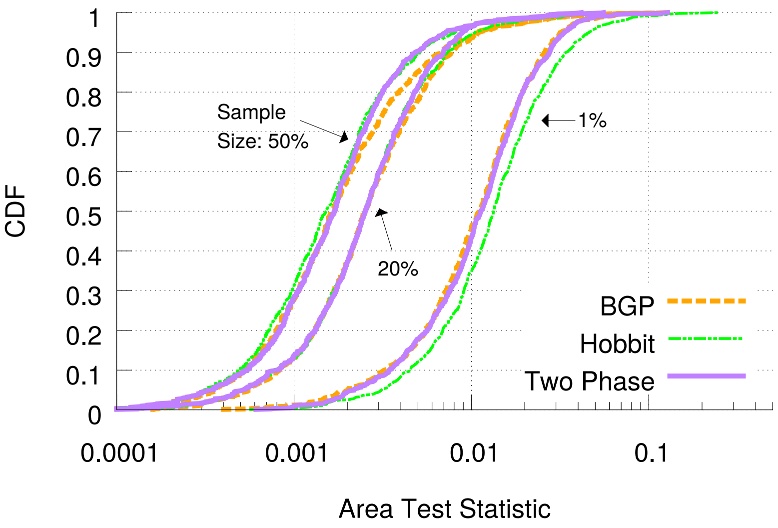


Figure 5.9: The CDFs of the Area Test statistics for good samples with various sizes. Different types of lines represent the adjusted distributions by using Hobbit blocks, BGP blocks and the two-phase approach.

blocks whereas 93.7% have at most 0.01 statistic when using BGP blocks. This could be because, as the sample size increases, more Hobbit blocks are likely to have some sample points. Since Hobbit blocks are finer-grained than BGP blocks, unobserved addresses can be better represented by the observed addresses within the same Hobbit blocks rather than those within the same BGP blocks.

I observe similar results for *uniform* and *bad samples*: 1) the weighting adjustment with BGP blocks as well as that with Hobbit blocks improves the representativeness of the samples and 2) the use of BGP blocks yields better results when the sample size is 1% whereas it is better to use Hobbit blocks when the sample size is 20% or 50%.

5.4.4 Exploiting both Hobbit and BGP blocks

Hobbit BGP blocks have their own advantage. The weighting adjustment using Hobbit blocks performs well on large samples whereas that using BGP blocks performs well on small samples. I develop a weighting adjustment strategy that works well for both small and large samples. A straightforward approach is to set a threshold for the sample size, and use Hobbit blocks if the given sample size is greater than the threshold value and use BGP blocks otherwise. This may be an effective approach but it can be very challenging to find a right threshold value. I instead develop a two-phase approach that uses Hobbit blocks in the first phase and relies on BGP blocks in the second phase.

It can be confusing to compute weights in two phases by using two different types of address groups. Instead, I perform an equivalent alternative method of weighting adjustment in two phases. The alternative method is to, for each of unmeasured addresses, randomly select a measured address from the address group it belongs to and use the value of the chosen address as its value in constructing a distribution. I have confirmed that this method generates identical distributions with weighting adjustment. It is straightforward to perform this method in two phases. In the first phase, I impute the values of unmeasured addresses from those of (randomly chosen) measured addresses in the same Hobbit blocks. If there is no measured address in the Hobbit block, I resort to BGP blocks as the second phase (i.e., impute the values of unmeasured addresses from those of measured addresses in the same BGP blocks).

I evaluate the performance of the two phase approach by applying it to *good samples*. Figure 5.9 depicts the CDFs of the Area Test statistics of the ASes for the distributions adjusted by weighting adjustment that uses either Hobbit or BGP blocks, and by the two-phase approach. I can see that the two-phase approach overcomes the weakness of weighting adjustment using either Hobbit or BGP blocks. When the sample size is 1%, the two-phase approach performs better than weighting adjustment that uses Hobbit blocks. At the same time, its performance is better than that of weighting adjustment using BGP blocks for the sample size 50%. It even performs better than both weighting adjustment using Hobbit blocks and that using BGP blocks when the sample size is 20%. These results demonstrate that the two phase approach performs well irrespective of the sample size and thus is better than weighting adjustment using either Hobbit or BGP blocks.

5.5 For which properties is weighting adjustment effective?

5.5.1 A criterion determining suitability

It would be ideal to identify specialized address groups to each metric for maximizing the performance of weighting adjustment. However, identifying the address groups that are generally applicable to different metrics can be more practical and useful. Hobbit and BGP blocks are representative examples of such address groups. They can be suitable for other metrics than RTT that I have already considered. The question is what determines whether or not they are usable. Hobbit and BGP blocks both consist of topologically proximate addresses although there is a difference in degree. (The addresses in the same Hobbit and BGP blocks have common

IP-level last-hop routers and AS-level paths, respectively.) This suggests that the correlation with topological proximity can be a key factor in determining whether Hobbit and BGP blocks are suitable. For example, RTTs are correlated with topological proximity and thus could be analyzed with Hobbit and BGP blocks. On the other hand, the characteristics of end hosts themselves such as OSes, the types of devices and certificate vulnerability may not be correlated with topological proximity. Even topologically proximate end hosts can have very different such characteristics and thus they may not be analyzable with Hobbit and BGP blocks.

In order to demonstrate that Hobbit and BGP blocks are suitable for metrics that are correlated with topological proximity, I evaluate the performance of weighting adjustment (specifically, the two-phase approach described in Section 5.4.4) using more metrics, that is, geographic locations and the prevalence of load-balancers.

5.5.2 Evaluation with geographic locations

I use the methodology described in Section 5.1.2.1. Hence, I compare the distribution of geographic locations[9] of measured addresses (i.e., the responding addresses in a single snapshot of the Internet-wide scan) with that of all responding addresses (i.e., the responding addresses in the aggregate of multiple snapshots). The distributions from the single and the multiple snapshots correspond to the estimated and reference distributions. Note that I do not split addresses by their AS number but use all the addresses in constructing the distributions. In some ASes,

---

[9] I use the Maxmind dataset [68] to obtain geographic locations

all the addresses are geographically co-located and thus it may not be meaningful to build the distributions of geographic locations within such ASes.

I adjust the estimated distribution by applying weighting adjustment, specifically, the two-phase approach described in Section 5.4.4. Since I use city names in determining the geographic locations of the addresses, geographical location is a discrete variable. Therefore, the Area Test statistics that I used for quantifying the difference between the distributions of RTTs may not be suitable for geographic locations. I instead use L1 distance [98] to quantify the differences in distributions. To be specific, for each value of a target variable, I compute the absolute difference between its ratio in one distribution and that in the other distribution, and add all the absolute differences. I compute the L1 distances between 1) the reference and estimated distributions and 2) the reference and adjusted distributions. These distances indicate the representativeness (of the estimated and adjusted distributions). A smaller distance indicates a higher degree of the representativeness.

The L1 distance between the reference and estimated distributions was 7.05674. This may imply that addresses in many locations are being over- or under-represented. I have actually analyzed regions where over- or under-representation occurs. Most of over-represented locations are in the US and Western Europe whereas under-represented locations are mostly in the other regions, particularly Asia and South America. The US and Western Europe have 19 of the top 20 most over-represented cities. While they have only 2 of the top 20 most under-represented cities, Asia and South America have 15 of the cities. Considering that the main cause of the unrepresentativeness is no response, this result is consistent with a previous study that has shown the availability of Internet addresses is very high in the US and Western Europe and diurnal in much of Asia and South America [16].

The weighting adjustment substantially improves the representativeness. The L1 distance between the reference and adjusted distributions is only 0.38057. This result demonstrates that my weighting adjustment strategy is suitable for geographic locations.

5.5.3 Evaluation with load-balancing existence

A straightforward way of determining whether weighting adjustment using Hobbit and BGP blocks are suitable for a metric is to apply weighting adjustment to the estimated distribution of the metric and compare the adjusted distribution against the reference distribution. However, this method may not be feasible especially when reference distributions are hard to obtain. An alternative method is to measure the correlation between the target metric and Hobbit blocks. If they are correlated, i.e., the addresses in the same blocks have similar metric values, then weighting adjustment by Hobbit blocks are likely to be suitable for the metric. Using this method, I measure if it is suitable to use Hobbit or BGP blocks for the existence of load-balancers. In other words, I measure if the paths towards the addresses in the same Hobbit blocks commonly have load-balancers or not. This can be useful not only for verifying that weighting adjustment by Hobbit or BGP blocks can be used for metrics correlated with topological proximity but also for reducing measurement loads for enumerating load-balanced paths. If the presence of load-balancers is correlated with Hobbit blocks, then Paris-traceroute MDA [22], which is a tool for enumerating load-balanced paths, does not have to be performed for the addresses in the blocks having no load-balancers except for the first few addresses that need to be probed for determining the existence of load-balancers in the blocks. This can result in reduced measurement loads because Paris-traceroute MDA causes heavier loads than typical traceroute [22].

I randomly select 4 addresses from each routed /24 prefix and identify the presence of load-balancers[10] on the paths towards the addresses by performing Paris-traceroute MDA for each address. To quantify the correlation between Hobbit blocks and load-balancer existence, I compute the probability that an addresses will have the same result with another arbitrary address in the same block. For example, if there are 4 and 6 addresses with and without load-balancers in a certain block, then the probability for the addresses with and without load-balancers will be 3/9 and 5/9, respectively. If the probability is 1, it indicates that the addresses within the block are completely correlated. I compute the probability for each address and identify how many addresses have the probability 1. For Hobbit blocks, about 94.62% of the addresses have the probability 1 indicating a very high degree of correlation between Hobbit blocks and the existence of load balancers. BGP blocks also have a correlation with the load balancer existence. About 79.83% of the addresses have the probability 1. This results suggest that the two-phase approach that uses Hobbit and BGP blocks will be helpful for improving the representativeness of the measurement of the load-balancing prevalence.

---

[10] I only consider per-flow load-balancing.

5.6 Summary

In this chapter, I showed that active measurements may not be representative even if the entire IPv4 space is probed and there are situations where even additional measurements cannot improve the representativeness. I adapted weighting adjustment to active measurements by using Hobbit blocks and showed that the weighting adjustment can improve the representativeness of active measurements. To design a good strategy for weighting adjustment in active measurements, I identified and analyzed key factors affecting the performance of weighting adjustment. Although the performance is largely influenced by the properties of samples including the sample sizes and the distributions of sample points, the weighting adjustment generally improves the representativeness of various kinds of samples. I also analyzed the effects of the address groups used in computing weights by comparing Hobbit and BGP blocks.

Weighting adjustment has hardly been used in network measurement despite its commonness in other fields such as surveys. I have shown that weighting adjustment can apply to active network measurement and it actually improves the representativeness in most cases. Weighting adjustment is very easy to apply given that Hobbit blocks can be used as input to weighting adjustment. Since I publish Hobbit blocks every month, one can perform weighting adjustment at no measurement cost. Weighting adjustment thus can have an impact on the practice of network measurement.

Chapter 6: CONCLUSIONS AND FUTURE WORK

6.1 Thesis

In this dissertation, I developed a general approach to reducing measurement loads and biases that are fundamental issues for improving the reliability of active Internet measurement. Based on the insight that measurement loads and biases can be reduced by letting Internet addresses represent larger aggregates, I defended the following thesis: *IPv4 addresses can be grouped into homogeneous blocks that enable researchers to improve efficiency and accuracy of active Internet measurement.*

In Chapter 3, I developed Hobbit, a technique that identifies the aggregates of topologically proximate addresses. Unlike the previous work [4] that assumes that traceroute differences within /24s are all due to load-balancing, Hobbit distinguishes between traceroute differences due to load-balancing and distinct route entries. Hobbit is also unique in that it can aggregate even discontiguous addresses unlike traditional network prefix based approaches [3, 19]. I evaluated Hobbit using various information sources such as Whois databases, reverse DNS names and latency characteristics. By evaluation, I showed that the unique characteristics of Hobbit contribute to aggregating more homogeneous /24s and detecting heterogeneous /24s. I applied Hobbit to all eligible /24s and identified 0.51M homogeneous blocks (i.e., Hobbit blocks) that contain 1.77M /24s.

In Chapter 4, I demonstrated that the use of Hobbit blocks can improve the efficiency of network measurement. I compared an Internet topology mapping strategy that selects destinations from /24s (like CAIDA's topology discovery [28]) with a strategy that selects destinations from Hobbit blocks. I evaluated that the strategy using Hobbit blocks discovers more links when the same number of probes are sent. I also showed that Hobbit blocks improve the efficiency of latency estimation. I quantified the correlation of RTTs within Hobbit blocks and the efficiency improvement that can be obtained by using Hobbit blocks. When Hobbit is combined with Ting, which is an existing latency estimation technique, Ting can predict RTTs between 66 times more address pairs without additional measurements. Hobbit causes additional estimation error but the degree of the estimation error is comparable with that of existing latency estimation techniques.

In Chapter 5, I demonstrated that Hobbit blocks can be used for improving the accuracy of network measurement. I used Hobbit blocks in adapting weighting adjustment that is a common bias correction technique in surveys to active Internet measurement. To demonstrate the accuracy improvement, I developed a methodology that measures the representativeness of active Internet measurement. I identified the key factors influencing the performance of the weighting adjustment and evaluated the effectiveness of the weighting adjustment using various samples with different combinations of the factors. I showed that the weighting adjustment improves the representativeness for most samples.

6.2 Future Work

6.2.1 Aggregating IPv6 addresses

The IPv6 Internet is growing [99] and has been studied by many researchers [100–104]. One research direction in IPv6 is to adapt network measurement techniques designed for IPv4 to IPv6 [105] or develop new techniques for IPv6 in cases where the existing techniques do not apply to IPv6 [106,107]. Applying Hobbit to IPv6 can suggest several research problems. The key component of Hobbit is to deal with load-balancing by checking if addresses can be grouped by non-overlapping prefixes. This component will apply to IPv6 as long as IPv6 load-balancers use hashing to determine the next hop. But, its accuracy still needs to be verified. The usefulness also needs to be evaluated in that the degree of the prevalence of load-balancing is different in the IPv4 and IPv6 Internet [105]. With Hobbit adapted to IPv6, the study that identifies and analyzes Hobbit blocks of IPv6 addresses can be useful. The analysis of the Hobbit blocks can provide insights on IPv6 addressing schemes. For example, if Hobbit blocks have many discontiguous addresses, it may suggest the prevalence of fragmented allocations of IPv6 addresses. If many BGP prefixes comprise different Hobbit blocks, it may mean that route aggregation is common in IPv6.

6.2.2 Going beyond active measurement

In this dissertation, I focused on active Internet measurement. Accordingly, in computing weights for weighting adjustment, I defined a population to be all responding addresses (Section 5.2.1). This is accurate with respect to active measurement because active measurement can only measure responding addresses. However, focusing only on responding addresses is a limitation of active measurement, in that there are many invisible addresses (i.e., the addresses that are being used but intentionally do not respond to probes by, e.g., access control) [12] and the distribution containing both visible and invisible addresses may be different with the distribution of only visible addresses. Nevertheless, this does not mean that weighting adjustment has a fundamental limitation. Rather, it can help to overcome the limitation of active measurement. The idea is to use the number of both visible and invisible addresses in computing weights instead of the number of visible addresses. This idea may seem to be straightforward but needs to be studied in more detail. First of all, weighting adjustment can only be effective when address groups have both visible and invisible addresses. If most of the address groups have either visible or invisible addresses, then weighting adjustment will not be successful because weighting adjustment infers the values for invisible addresses from those for visible addresses in the same groups. It also needs to be verified whether visible and invisible addresses are correlated within the same address groups. If they have different characteristics even within the same address groups, then weighting adjustment may be inappropriate. Finally, the number of invisible addresses within each address group needs to be accurately estimated. This can be a challenging task in that Internet-wide traces are hard to obtain. Studying these research problems can potentially make a contribution towards extending the coverage of active measurement.

Bibliography

[1] Yin Zhang, Nick Duffield, Vern Paxson, and Scott Shenker. On the constancy of Internet path properties. In *IMW*, 2001.

[2] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with rocketfuel. In *SIGCOMM*, 2002.

[3] Harsha V Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iplane: An information plane for distributed services. In *OSDI*, 2006.

[4] Brice Augustin, Timur Friedman, and Renata Teixeira. Measuring load-balanced paths in the Internet. In *IMC*, 2007.

[5] Matthieu Latapy, Clémence Magnien, and Francois Ouedraogo. A Radar for the Internet. In *ICDMW'08*, 2008.

[6] Ítalo Cunha, Renata Teixeira, and Christophe Diot. Measuring and characterizing end-to-end route dynamics in the presence of load balancing. In *PAM*, 2011.

[7] Tobias Flach, Ethan Katz-Bassett, and Ramesh Govindan. Quantifying violations of destination-based forwarding on the internet. In *IMC*, 2012.

[8] Ramakrishnan Durairajan, Joel Sommers, and Paul Barford. Layer 1-Informed Internet Topology Measurement. In *IMC*, 2014.

[9] Ítalo Cunha, Pietro Marchetta, Matt Calder, Yi-Ching Chiu, Brandon Schlinker, Bruno VA Machado, Antonio Pescapè, Vasileios Giotsas, Harsha V Madhyastha, and Ethan Katz-Bassett. Sibyl: A Practical Internet Route Oracle. In *NSDI*, 2016.

[10] Karthik Lakshminarayanan and Venkata N. Padmanabhan. Some findings on the network performance of broadband hosts. In *IMC*, 2003.

[11] Marcel Dischinger, Andreas Haeberlen, Krishna P Gummadi, and Stefan Saroiu. Characterizing residential broadband networks. In *IMC*, 2007.

[12] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, Genevieve Bartlett, and Joseph Bannister. Census and survey of the visible Internet. In *IMC*, 2008.

[13] Daniele Croce, Taoufik En-Najjary, Guillaume Urvoy-Keller, and Ernst W Biersack. Capacity estimation of ADSL links. In *CoNEXT*, 2008.

[14] Aaron Schulman and Neil Spring. Pingin' in the rain. In *IMC*, 2011.

[15] Lin Quan, John Heidemann, and Yuri Pradkin. Trinocular: Understanding Internet reliability through adaptive probing. In *SIGCOMM*, 2013.

[16] Lin Quan, John Heidemann, and Yuri Pradkin. When the Internet sleeps: correlating diurnal networks with external factors. In *IMC*, 2014.

[17] PlanetLab. `https://www.planet-lab.org/`.

[18] RIPE NCC. RIPE Atlas. `http://atlas.ripe.net`.

[19] Xue Cai and John Heidemann. Understanding block-level address usage in the visible Internet. In *SIGCOMM*, 2010.

[20] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *USENIX Security*, 2013.

[21] ZMap. FULL IPv4 ICMP Echo Request. `https://scans.io/series/0-icmp-echo_request-full_ipv4`.

[22] Brice Augustin, Timur Friedman, and Renata Teixeira. Multipath tracing with paris traceroute. In *E2EMON*, 2007.

[23] MaxMind, Inc. Allocation of IP Addresses by Country. `https://www.maxmind.com/en/allocation-of-ip-addresses-by-country`.

[24] Michael J Freedman, Mythili Vutukuru, Nick Feamster, and Hari Balakrishnan. Geographic locality of ip prefixes. In *IMC*, 2005.

[25] Roderick JA Little. Survey nonresponse adjustments for estimates of means. *International Statistical Review/Revue Internationale de Statistique*, pages 139–157, 1986.

[26] Venkata N Padmanabhan and Lakshminarayanan Subramanian. An investigation of geographic mapping techniques for internet hosts. In *SIGCOMM*, 2001.

[27] Qiang Xu, Junxian Huang, Zhaoguang Wang, Feng Qian, Alexandre Gerber, and Zhuoqing Morley Mao. Cellular data network infrastructure characterization and implication on mobile content placement. In *SIGMETRICS*, 2011.

[28] CAIDA. The CAIDA UCSD IPv4 Routed /24 Topology Dataset. http://www.caida.org/data/active/ipv4_routed_24_topology_dataset.xml.

[29] Carlo Contavalli, Warren Kumari, and Wilmer van der Gaast. Client subnet in dns queries. 2016.

[30] Inja Youn, Brian L Mark, and Dana Richards. Statistical geolocation of internet hosts. In *Internatonal Conference on Computer Communications and Networks*, 2009.

[31] Brian Eriksson, Paul Barford, Joel Sommers, and Robert Nowak. A learning-based approach for ip geolocation. In *PAM*, 2010.

[32] Mohammed Jubaer Arif, Shanika Karunasekera, Santosh Kulkarni, Ajit Gunatilaka, and Branko Ristic. Internet host geolocation using maximum likelihood estimation technique. In *IEEE International Conference on Advanced Information Networking and Applications*, 2010.

[33] Brian Eriksson, Paul Barford, Bruce Maggs, and Robert Nowak. Posit: a lightweight approach for ip geolocation. *ACM SIGMETRICS Performance Evaluation Review*, 2012.

[34] Zi Hu, John Heidemann, and Yuri Pradkin. Towards geolocation of millions of ip addresses. In *IMC*, 2012.

[35] Manaf Gharaibeh, Han Zhang, Christos Papadopoulos, and John Heidemann. Assessing co-locality of ip blocks. In *IEEE Global Internet Symposium*, 2016.

[36] Matt Calder, Xun Fan, Zi Hu, Ethan Katz-Bassett, John Heidemann, and Ramesh Govindan. Mapping the expansion of google's serving infrastructure. In *IMC*, 2013.

[37] Bradley Huffaker, Marina Fomenkov, and kc claffy. Drop: DNS-based router positioning. *CCR*, 44(3):5–13, 2014.

[38] Gregor Maier, Anja Feldmann, Vern Paxson, and Mark Allman. On dominant characteristics of residential broadband Internet traffic. In *IMC*, 2009.

[39] Srikanth Sundaresan, Walter de Donato, Nick Feamster, Renata Teixeira, Sam Crawford, and Antonio Pescapè. Broadband Internet performance: A view from the gateway. In *SIGCOMM*, 2011.

[40] Cristel Pelsser, Luca Cittadini, Stefano Vissicchio, and Randy Bush. From paris to tokyo: On the suitability of ping to measure latency. In *IMC*, 2013.

[41] Igor Canadi, Paul Barford, and Joel Sommers. Revisiting broadband performance. In *IMC*, 2012.

[42] Fangfei Chen, Ramesh K Sitaraman, and Marcelo Torres. End-user mapping: Next generation request routing for content delivery. In *SIGCOMM*, 2015.

[43] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. Sampling techniques for large, dynamic graphs. In *INFOCOM*, 2006.

[44] Daniel Stutzbach, Reza Rejaie, Nick Duffield, Subhabrata Sen, and Walter Willinger. On unbiased sampling for unstructured peer-to-peer networks. *IEEE/ACM Transactions on Networking (TON)*, 17(2):377–390, 2009.

[45] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. A walk in facebook: Uniform sampling of users in online social networks. *arXiv preprint arXiv:0906.0060*, 2009.

[46] Minas Gjoka, Maciej Kurant, Carter T Butts, and Athina Markopoulou. Walking in facebook: A case study of unbiased sampling of osns. In *INFOCOM*, 2010.

[47] Maciej Kurant, Minas Gjoka, Carter T Butts, and Athina Markopoulou. Walking on a graph with a magnifying glass: stratified sampling via weighted random walks. In *SIGMETRICS*, 2011.

[48] Minas Gjoka, Carter T Butts, Maciej Kurant, and Athina Markopoulou. Multigraph sampling of online social networks. *IEEE Journal on Selected Areas in Communications*, 29(9), 2011.

[49] Srikanth Kandula and Ratul Mahajan. Sampling biases in network path measurements and what to do about it. In *IMC*, 2009.

[50] Yihua He, Georgos Siganos, Michalis Faloutsos, and Srikanth V Krishnamurthy. A systematic framework for unearthing the missing links: Measurements and impact. In *NSDI*, 2007.

[51] Manuel Costa, Miguel Castro, R Rowstron, and Peter Key. Pic: Practical internet coordinates for distance estimation. In *ICDCS*, 2004.

[52] Yongjun Liao, Pierre Geurts, and Guy Leduc. Network distance prediction based on decentralized matrix factorization. *NETWORKING*, 2010.

[53] Hyuk Lim, Jennifer C Hou, and Chong-Ho Choi. Constructing internet coordinate system based on delay measurement. In *IMC*, 2003.

[54] Cristian Lumezanu and Neil Spring. Measurement manipulation and space selection in network coordinates. In *ICDCS*, 2008.

[55] Yun Mao, Lawrence K Saul, and Jonathan M Smith. Ides: An internet distance estimation service for large networks. *IEEE Journal on Selected Areas in Communications*, 24(12):2273–2284, 2006.

[56] TS Eugene Ng and Hui Zhang. A network positioning system for the internet. In *USENIX ATC*, 2004.

[57] TS Eugene Ng and Hui Zhang. Predicting internet network distance with coordinates-based approaches. In *INFOCOM*, 2002.

[58] Liying Tang and Mark Crovella. Virtual landmarks for the internet. In *IMC*, 2003.

[59] Krishna P Gummadi, Stefan Saroiu, and Steven D Gribble. King: Estimating latency between arbitrary internet end hosts. In *IMW*, 2002.

[60] Frank Cangialosi, Dave Levin, and Neil Spring. Ting: Measuring and exploiting latencies between all tor nodes. In *IMC*, 2015.

[61] Roger Dingledine, Nick Mathewson, and Paul Syverson. Tor: The second-generation onion router. In *USENIX Security*, 2004.

[62] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A search engine backed by Internet-wide scanning. In *CCS*, 2015.

[63] Cisco Load-Balancing. `http://www.cisco.com/c/en/us/support/docs/ip/border-gateway-protocol-bgp/5212-46.html#perper`.

[64] Steven K Thompson. *Sampling*. John Wiley & Sons, 2012.

[65] Yi-Chao Chen, Yong Liao, Mario Baldi, Sung-Ju Lee, and Lili Qiu. OS fingerprinting and tethering detection in mobile networks. In *IMC*, 2014.

[66] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with paris traceroute. In *IMC*, 2006.

[67] Cisco. TTL Expiry Attack Identification. `http://www.cisco.com/c/en/us/about/security-center/ttl-expiry-attack.html`.

[68] MaxMind, Inc. Geolite databases. `http://dev.maxmind.com/geoip/legacy/geolite/`.

[69] KRNIC. WHOIS services. `http://whois.kisa.or.kr/eng/`.

[70] KRNIC. IPv4 address holdings. `http://krnic.or.kr/jsp/eng/ipas/statistics/ipV4.jsp`.

[71] Amazon. AWS regions and endpoints. `http://docs.aws.amazon.com/general/latest/gr/rande.html`.

[72] Ramakrishna Padmanabhan, Patrick Owen, Aaron Schulman, and Neil Spring. Timeouts: Beware surprisingly high delay. In *IMC*, 2015.

[73] Kyriakos Zarifis, Tobias Flach, Srikanth Nori, David Choffnes, Ramesh Govindan, Ethan Katz-Bassett, Z Morley Mao, and Matt Welsh. Diagnosing path inflation of mobile client traffic. In *PAM*, 2014.

[74] BITNODES. https://getaddr.bitnodes.io/.

[75] Cox. PHOENIX NAP. http://www.coxbusinessaz.com/assets/docs/BusinessContinuitySheet.pdf.

[76] Neil Spring, Ratul Mahajan, and Thomas Anderson. Quantifying the causes of path inflation. In *SIGCOMM*, 2003.

[77] NS-3 Topology Input Readers. https://www.nsnam.org/doxygen/group__topology.html.

[78] Zhiyong Zhang, Ovidiu Mara, and Katerina Argyraki. Network neutrality inference. In *SIGCOMM*, 2014.

[79] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: A decentralized network coordinate system. In *SIGCOMM*, 2004.

[80] Sharad Agarwal and Jacob R Lorch. Matchmaking for online games and other latency-sensitive p2p systems. In *SIGCOMM*, 2009.

[81] Youndo Lee and Neil Spring. Identifying and aggregating homogeneous ipv4 /24 blocks with hobbit. In *IMC*, 2016.

[82] Jonathan Ledlie, Paul Gardner, and Margo I Seltzer. Network coordinates in the wild. In *NSDI*, 2007.

[83] Ming Zhang, Yaoping Ruan, Vivek S Pai, and Jennifer Rexford. How dns misnaming distorts internet topology mapping. In *USENIX Annual Technical Conference, General Track*, 2006.

[84] CAIDA. Public DNS Decoding database. http://ddec.caida.org/.

[85] Wikipedia page on airport codes. http://wikipedia.org/wiki/Airport_code.

[86] Common Language Location Identifier (CLLI) code. http://en.wikipedia.org/wiki/CLLI_code.

[87] UN/LOCODE Code List. http://www.unece.org/cefact/locode/service/location.html.

[88] Project Sonar reverse DNS dataset. https://scans.io/study/sonar.rdns.

[89] MaxMind, Inc. The accuracy of MaxMind databses. https://support.maxmind.com/geoip-faq/geoip2-and-geoip-legacy-databases/how-accurate-are-your-geoip2-and-geoip-legacy-databases/.

[90] University of Oregon Route Views Archive Project. http://www.routeviews.org/.

[91] Arash Molavi Kakhki, Abbas Razaghpanah, Anke Li, Hyungjoon Koo, Rajesh Golani, David Choffnes, Phillipa Gill, and Alan Mislove. Identifying traffic differentiation in mobile networks. In *IMC*, 2015.

[92] Andreas Haeberlen, Marcel Dischinger, Krishna P Gummadi, and Stefan Saroiu. Monarch: a tool to emulate transport protocol flowsover the Internet at large. In *IMC*, 2006.

[93] Indra M Chakravarty, JD Roy, and Radha Govind Laha. Handbook of methods of applied statistics. *McGraw-Hill*, 1967.

[94] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Internet optometry: assessing the broken glasses in internet reachability. In *IMC*, 2009.

[95] Kimberly Claffy, Young Hyun, Ken Keys, Marina Fomenkov, and Dmitri Krioukov. Internet mapping: from art to science. In *Conference For Homeland Security, 2009. CATCH'09. Cybersecurity Applications & Technology*, 2009.

[96] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for internet map discovery. In *INFOCOM*, 2000.

[97] Xun Fan and John Heidemann. Selecting representative ip addresses for internet topology studies. In *IMC*, 2010.

[98] Paul E. Black. Manhattan distance. in *Dictionary of Algorithms and Data Structures* [online], Vreda Pieterse and Paul E. Black eds. Available from: https://www.nist.gov/dads/HTML/manhattanDistance.html.

[99] Jakub Czyz, Mark Allman, Jing Zhang, Scott Iekel-Johnson, Eric Osterweil, and Michael Bailey. Measuring ipv6 adoption. In *SIGCOMM*, 2014.

[100] Lorenzo Colitti, Steinar H Gunderson, Erik Kline, and Tiziana Refice. Evaluating ipv6 adoption in the internet. In *PAM*, 2010.

[101] KC Claffy. Tracking ipv6 evolution: data we have and data we need. *ACM SIGCOMM Computer Communication Review*, 41(3):43–48, 2011.

[102] Mehdi Nikkhah, Roch Guérin, Yiu Lee, and Richard Woundy. Assessing ipv6 through web access a measurement study and its findings. In *CoNEXT*, 2011.

[103] Amogh Dhamdhere, Matthew Luckie, Bradley Huffaker, Ahmed Elmokashfi, Emile Aben, et al. Measuring the deployment of ipv6: topology, routing and performance. In *IMC*, 2012.

[104] Robert Beverly, Matthew Luckie, Lorenza Mosley, and Kc Claffy. Measuring and characterizing ipv6 router availability. In *PAM*, 2015.

[105] Rafael Almeida, Osvaldo Fonseca, Elverton Fazzion, Dorgival Guedes, Wagner Meira, and Ítalo Cunha. A characterization of load balancing on the ipv6 internet. In *PAM*, 2017.

[106] Matthew Luckie, Robert Beverly, William Brinkmeyer, et al. Speedtrap: internet-scale ipv6 alias resolution. In *IMC*, 2013.

[107] Ramakrishna Padmanabhan, Zhihao Li, Dave Levin, and Neil Spring. Uav6: Alias resolution in ipv6 using unused addresses. In *PAM*, 2015.