2015-09

Discovery of IPv6 router interface addresses via heuristic methods

Gray, Matthew D.

Monterey, California: Naval Postgraduate School

https://hdl.handle.net/10945/47265

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

DISCOVERY OF IPV6 ROUTER INTERFACE ADDRESSES VIA HEURISTIC METHODS

by

Matthew D. Gray

September 2015

Thesis Advisor:	Robert Beverly
Second Reader:	Arthur Berger

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

1. AGENCY USE ONLY *(Leave Blank)*	2. REPORT DATE 09-25-2015	3. REPORT TYPE AND DATES COVERED Master's Thesis 09-30-2013 to 09-25-2015

4. TITLE AND SUBTITLE DISCOVERY OF IPV6 ROUTER INTERFACE ADDRESSES VIA HEURISTIC METHODS	5. FUNDING NUMBERS CNS-1111445
6. AUTHOR(S) Matthew D. Gray	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Science Foundation 4201 Wilson Blvd., Arlington, VA 22230	10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

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

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

With the assignment of the last available blocks of public IPv4 addresses from Internet Assigned Numbers Authority, there is continued pressure for widespread IPv6 adoption. Because the IPv6 address space is orders of magnitude larger than the IPv4 address space, researchers need new methods and techniques to accurately measure and characterize growth in IPv6. This thesis focuses on IPv6 router infrastructure and examines the possibility of using heuristic methods in order to discover IPv6 router interfaces. We consider two heuristic techniques in an attempt to improve upon current state-of-the-art IPv6 router infrastructure discovery methods. The first heuristic examines the ability to generate candidate IPv6 addresses by finding the most common lower 64 bit patterns among IPv6 router interface address observed in historical probing data. The second heuristic generates candidate IPv6 addresses by assuming that an IPv6 address seen in historical probing data is one end of a point-to-point link, and uses the corresponding end's IPv6 address. Using a distributed active topology measurement system, we test these heuristic methods on the IPv6 Internet. We find that our first heuristic is successful in discovering a non-trivial number of new router interfaces, while the second heuristic is more efficient.

14. SUBJECT TERMS IPv6, Discover of Router Infrastructure, Heuristics	15. NUMBER OF PAGES 61
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU

THIS PAGE INTENTIONALLY LEFT BLANK

DISCOVERY OF IPV6 ROUTER INTERFACE ADDRESSES VIA HEURISTIC METHODS

Matthew D. Gray
Lieutenant, United States Navy
B.S., Purdue University, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
September 2015

Author:	Matthew D. Gray
Approved by:	Robert Beverly Thesis Advisor
	Arthur Berger Second Reader
	Peter J. Denning Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the assignment of the last available blocks of public IPv4 addresses from Internet Assigned Numbers Authority, there is continued pressure for widespread IPv6 adoption. Because the IPv6 address space is orders of magnitude larger than the IPv4 address space, researchers need new methods and techniques to accurately measure and characterize growth in IPv6. This thesis focuses on IPv6 router infrastructure and examines the possibility of using heuristic methods in order to discover IPv6 router interfaces. We consider two heuristic techniques in an attempt to improve upon current state-of-the-art IPv6 router infrastructure discovery methods. The first heuristic examines the ability to generate candidate IPv6 addresses by finding the most common lower 64 bit patterns among IPv6 router interface address observed in historical probing data. The second heuristic generates candidate IPv6 addresses by assuming that an IPv6 address seen in historical probing data is one end of a point-to-point link, and uses the corresponding end's IPv6 address. Using a distributed active topology measurement system, we test these heuristic methods on the IPv6 Internet. We find that our first heuristic is successful in discovering a non-trivial number of new router interfaces, while the second heuristic is more efficient.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

AAAA	Quad-A Record
APAR	Analytic and Probe-based Alias Resolver
Ark	Archipelago Measurement Infrastructure
AS	Autonomous System
BGP	Border Gateway Protocol
CAIDA	Center for Applied Internet Data Analysis
CDN	Content Distribution Network
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
IANA	Internet Assigned Numbers Authority
ICMPv6	Internet Control Message Protocol Version 6
IP	Internet Protocol
IPv4	Internet Protocol Version 4
IPv6	Internet Protocol Version 6
ISP	Internet Service Provider
MAC	Media Access Control
NAT	Network Address Translation
NLSDE	National Lab of Software Development Environment
RFC	Request for Comment
RIR	Regional Internet Registries

RSI	Recursive Subnet Inference
RTT	Round Trip Time
SCP	Subnet Centric Probing
SLAAC	Stateless Address Autoconfiguration
ToD	Topology on Demand

Acknowledgments

First, I would like to thank Dr. Robert Beverly. His knowledge and input was extremely valuable in getting this thesis to the state that it is in today. I appreciate the numerous hours he spent with me discussing the research, how to improve or better characterize the results based on the network measurements, and reviewing and revising my writing. Without his gracious help, I'm not sure where my thesis would be today. I also want to thank him for the time he spent as my professor for numerous courses during my time in the Computer Science Department at NPS. He was one of my favorite professors, and I always felt like I learned a lot in his classes.

Second, I would like to thank Dr. Arthur Berger for his insightful comments regarding the clarity of the thesis and research. He also suggested various additional analysis ideas that helped to strengthen the analysis of the experimental data in Chapter 4.

Finally, I would like to thank my wife, Stephanie, and our families for their immense support during my time in graduate school. Without their support and encouragement, I don't think I could have survived all the stress of graduate school and a wedding.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

There are currently two types of Internet Protocol (IP) addresses assigned to network interfaces connected to the Internet. The first and most common type are Internet Protocol Version 4 (IPv4) addresses, which are 32-bit unsigned integers and are usually expressed in "dotted-decimal" notation (e.g., `74.125.196.147`). The second type are Internet Protocol Version 6 (IPv6) addresses, which are 128-bit unsigned integers and are usually expressed in hexadecimal notation (e.g., `2607:f8b0:4002:c09::63`). The address spaces represented by both IPv4 and IPv6 are divided into IP allocation blocks by the Internet Assigned Numbers Authority (IANA). IANA will assign an allocation block to one of the five Regional Internet Registries (RIR) based on geographical location. The RIRs then provides sub-allocations out of their assigned IP allocation blocks to an entity. However, the last unallocated IPv4 blocks were allocated by IANA in February 2011 [1], [2]. With the inability of at least two of the five RIRs [3]–[5] to provide sub-allocations from their allocated IPv4 address blocks, the availability of globally routable IPv4 addresses is quickly running out. This exhaustion of IPv4 address space has put increasing pressure on Internet Service Providers (ISPs), content providers, organizations, and individuals to adopt IPv6 technologies due to the larger address space and greater availability of IPv6 addresses. Major ISPs, e.g., Comcast, recently have been experiencing shortages of IPv4 addresses to assign to their customers. These major ISPs are slowly adopting IPv6 as the larger address space of IPv6 enables them to better cope with the IPv4 address shortages. As ISPs and content providers continue to adopt IPv6, it becomes more advantageous for the end user to adopt IPv6 due to the possibility that, in the future, certain content will only be available to end users via IPv6.

The specifications for IPv6 were adopted in December 1998 and defined in Request for Comment (RFC) 2460. While IPv6 has been standardized for the past 15 years, it has not seen appreciable deployment until the late 2000s. There are several reasons that have caused IPv6 to not be widely adopted until recently. The primary reason for the lack of IPv6 adoption is that IPv6 is not backward compatible with IPv4. The lack of backward compatibility between IPv4 and IPv6 means that there is a need for increased complexity in the network and additional resources needed for purchasing required equipment upgrades to support IPv6. Additionally, the adoption of IPv6 has been slow due to security concerns. While IPv6 was engineered to address certain security issues found in IPv4 [6], it is believed that malicious actors will find ways exploit IPv6 as an attack vector. For example, researchers are already seeing malicious actors using IPv6 to bypass network security devices due to the lack of IPv6 support or configuration in these devices (e.g., network firewalls, network management, etc.). Finally, as adoption of IPv6 continues, malicious actors will discover exploitation methods specific to IPv6 and create new attacks using these newly discovered exploits [7], [8]. Thus, while the RFC specifies several motivations that lead to the creation of IPv6, the primary driver today is the exhaustion of usable IPv4 address space [6].

Network Address Translation (NAT) technology provides an interim solution to the issue of IPv4 address exhaustion by extending the useful life of IPv4. As a result of NAT, wide-scale adoption of IPv6 has been slow [9]. Specifically, large-scale NAT technologies are being proposed by ISPs as an alternative to IPv6. These large-scale NAT technologies, often referred to as carrier-grade NAT, allow an ISP to use private IPv4 address space within that ISP's internal network. This allows an ISP to share a single public IPv4 address among multiple subscribers [10]. However, NAT technologies do not solve the fundamental issue of IPv4 address space exhaustion. In fact, they introduce a new set of problems, including inhibiting end-to-end reachability, single points of failure in the network, and the requirement to maintain a large amount of network state [1], [8].

While the larger address space in IPv6 can support continued growth in the Internet, it also presents challenges to the efforts of researchers who are attempting to map and understand the topology of the IPv6 Internet. For example, exhaustive active scanning techniques that were feasible for IPv4 are not feasible given the size of the IPv6 address space.

The primary goal of this thesis is to develop alternative and efficient methods to discover IPv6 infrastructure, specifically router interfaces. By improving upon current IPv6 infrastructure discovery methods, we hope to enable better insight into the nature of the IPv6 transition and more wholly understand the topology of IPv6.

1.1 Motivation

With the continued exponential growth of IPv6 since 2008, approximately a thirty-fold increase as observed by Google, one can conclude that IPv6 is becoming more widely adopted. In addition, researchers are continuing to see growth in the core of the network with respect to support for IPv6. However, it is still unclear how widespread the adoption is and where this growth is occurring [8], [11], [12].

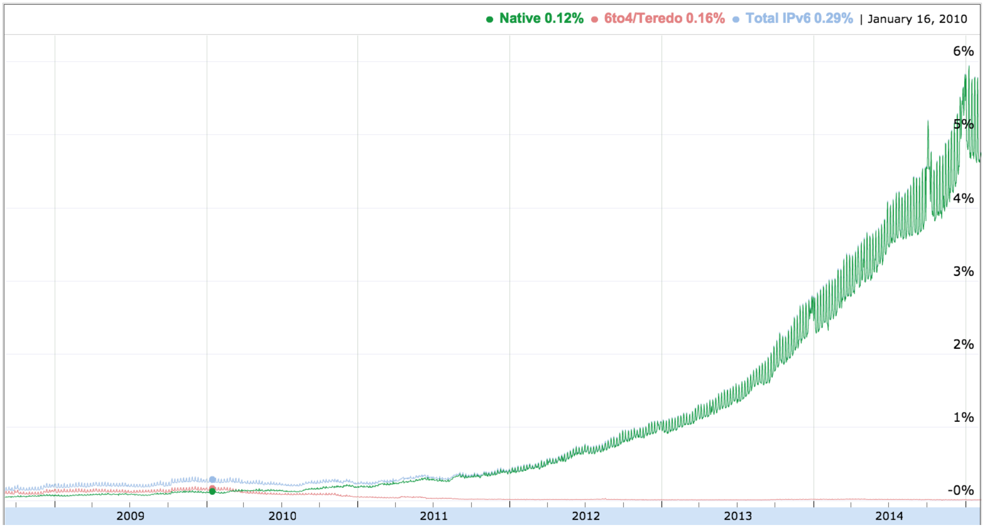

Figure 1.1: Google's IPv6 Adoption Statistics as of February 2015, from [11]

To understand the need for alternative and more efficient ways to discover IPv6 infrastructure, one must comprehend the size of the address space provided by IPv6 and the infeasibility of trying to probe all possible IPv6 addresses using current technologies. IPv6 uses a 128-bit unsigned integer to indicate the address of an endpoint, providing IPv6 with 2^{128}, approximately 3.4×10^{38}, possible unique addresses. In contrast, IPv4 uses 32-bit addresses, providing 2^{32}, or approximately 4.3×10^{9}, possible unique IP addresses. Thus, IPv6 provides an address space almost thirty orders of magnitude larger than IPv4. Given that the IPv6 address space is orders of magnitude larger than IPv4, it is currently infeasible to actively probe all possible addresses in the IPv6 address space.

For the purposes of discussion, assume that we have access to all the servers in a single data center (roughly 100,000 servers [13]) and that each server can probe IP addresses at a rate of 20 addresses per minute. Using these assumptions, it would take approximately 2,148 minutes or just under 36 hours to probe the entire IPv4 address space. This is quite feasible and there has been at least one instance of a botnet operating in a similar fashion to conduct a complete scan of the IPv4 address space [14]. However, using the same assumptions to probe the entire IPv6 address space, it would take 1.7×10^{32} minutes or approximately 3.2×10^{26} years to complete, which is an unrealistic timeframe. Thus, a current challenge faced by researchers and malicious actors alike is to find intelligent and efficient probing methods in IPv6 for discovering hosts and infrastructure.

1.2 Research Questions

The focus of this thesis is finding alternative and efficient methods to discover IPv6 infrastructure. To narrow the scope of our research, we focus on the ability to use heuristic methods to discover IPv6 router interfaces. A heuristic is a form of problem solving that uses a practical methodology in order to find a sufficient solution to a problem in a reasonable amount of time when finding the optimal solution to the problem is either impossible or impractical. Examples of common heuristic methods include using a rule of thumb to solve a problem, making an educated guess, and using common sense. The optimal solution to the problem of discovering IPv6 infrastructure, in particular router interfaces, is to exhaustively probe the entire address space. However, this optimal solution has been shown to be impossible (Section 1.1). Therefore, we seek to show that by using heuristic techniques, we can discover IPv6 router interfaces in a reasonable amount of time.

This thesis begins by using historical IPv6 probe data from the Center for Applied Internet Data Analysis (CAIDA) Archipelago Measurement Infrastructure (Ark) and a large scale Content Distribution Network (CDN) as inputs into our proposed heuristic techniques. The output from our heuristic techniques are candidate IPv6 addresses. We then actively probe the path to these candidate addresses, also using the Ark infrastructure, to determine the ability of our heuristics to discover new IPv6 router interfaces.

In our research into the feasibility of using heuristic techniques to discover IPv6 router interfaces, we seek to answer the following questions:

- Does historical data reveal patterns in IPv6 addressing via the host portion of an IPv6 address?
- If there are patterns in the historical data of IPv6 addressing, is it possible to leverage

these patterns in order to discover previously unknown IPv6 router interfaces?

- Do the discovered IPv6 router interfaces correspond to interfaces on previously known or new routers?
- Assuming the existence of point-to-point links, is it possible to leverage this assumption in order to discover previously unknown IPv6 router interfaces?

1.3 Contributions

Our research efforts into the feasibility of using heuristic methods to discover IPv6 infrastructure yielded the following findings:

- Although the IPv6 address space is very large, there is low-entropy in the host bits of router IPv6 interface addresses. The host bit values of `::1` and `::2` account for almost a third of all host addresses observed in historical data.
- A heuristic based probing approach can be successful in discovering a non-trivial number of new IPv6 router interfaces.
- Performing Internet-wide probing using a heuristic method based off of the 10 most common host bits of an IPv6 address yielded the discovery of approximately 5,500 previously unseen router interfaces.
- Performing Internet-wide probing using a heuristic method based off of generating IPv6 addresses by inferring the existence of point-to-point links yielded the discovery of approximately 10,150 previously unseen router interfaces. Additionally, this heuristic was more efficient than our other heuristic method. This heuristic produced the maximum number of new router interfaces with the least amount of candidate IPv6 addresses probed.

1.4 Thesis Structure

The remainder of this thesis is organized as follows:

- Chapter 2 discusses other IPv6 measurement and topology work, previous related work on using heuristics to discover IPv6 hosts, and IPv6 alias resolution techniques.
- Chapter 3 outlines two heuristic methods that generate candidate IPv6 addresses and describes our methodology for large-scale probing of these candidates.

- Chapter 4 provides results from our analysis of historical data on IPv6 addresses, results from probing using our most common lower-64 bit host heuristic and our point-to-point link heuristic.
- Chapter 5 details our research conclusions and provides recommendations for future research areas related to this work.

CHAPTER 2:
Background and Related Work

With the current ongoing transition from IPv4 to IPv6, researchers and content providers are interested in measuring the deployment of IPv6. Various content providers and organizations, including Akamai [15], Google [11], [16], and the U.S. government [17], all have web pages dedicated to providing near real-time statistics on the deployment of IPv6 from their respective vantage points. In addition, researchers are actively conducting experiments and measurements to characterize the adoption, use, and evolution of IPv6 using a variety of metrics and techniques. This chapter reviews features of IPv6 that are relevant to this thesis, as well as describing related research.

2.1 Overview of IPv6

As discussed in Chapter 1, an IPv6 address is a 128-bit unsigned integer. Its string presentation format is expressed in hexadecimal notation with the form `x:x:x:x:x:x:x:x` where each `x` represents 16-bits of the address as four hexadecimal values. In order to shorten the length of an IPv6 address, two shorthand notations have been adopted for IPv6. The first shorthand notation involves dropping all leading zeros in each sub-portion of an IPv6 address (e.g., `2607:f8b0:4002:0c09:0000:0000:0000:0063` is equivalent to `2607:f8b0:4002:c09:0:0:0:63`). The second shorthand notation uses “`::`” to represent one variable length run of zeros (e.g., `2607:f8b0:4002:0c09:0000:0000:0000:0063` is equivalent to `2607:f8b0:4002:c09::63`) [18]. For the purposes of our research, we define the “host bits” as the 64 lower, or least significant, bits of the 128-bit address. We term the upper, or most significant, 64 bits of the address as the “network bits.”

In order for a client, router or server to be able to communicate on the network via IPv6 it first needs to be assigned a globally unique IPv6 address. There are three primary ways to assign an IPv6 address to a device. The first method is via Stateless Address Autoconfiguration (SLAAC). SLAAC allows the host to generate a unique IPv6 address with the network prefix provided in router advertisement messages. The host creates a unique set of host bits by using the Media Access Control (MAC) address of its interface. To form the full 64-bit host bits, SLAAC inserts the hexadecimal values `0xFFFE` in between the upper

24 bits and the lower 24 bits of the MAC address [19]. The second method used for IPv6 address assignment is Dynamic Host Configuration Protocol (DHCP) in which the host requests an IP address from a DHCP server running on the network. The DHCP server then assigns the host an IPv6 address to use; often this assigned address is the next available IPv6 address in a block of values predefined by the network administrator [20]. In the third method, the host is manually configured by the network administrator with an unused IPv6 address. IPv6 address assignment is important to this thesis because it has a significant effect on our ability to develop heuristic techniques for intelligently probing IPv6.

2.2 IPv6 Deployment Measurement Studies

Significant prior research has sought to measure the deployment and growth of IPv6. One of the major challenges faced by IPv6 researchers is that many of the techniques developed for measuring IPv4 do not translate well, or at all, to IPv6 due to protocol differences and the much larger address space. As a result, researchers have developed new techniques and methods to accurately measure and characterize the growth in IPv6. The research discussed in this section focuses on IPv6 infrastructure deployment measurements; other research not discussed focuses instead on client adoption of IPv6.

Previous research by researchers from CAIDA [8], [12], focused on using data from publicly available Border Gateway Protocol (BGP) datasets in order to characterize trends in the growth of IPv6 and compare the growth of IPv6 to the growth of IPv4. Their research showed that IPv6 is experiencing an exponential growth trend while IPv4 growth is currently increasing gradually and linearly. They believe that the gradual linear growth in IPv4 is associated with the exhaustion of the address space. At the time of their data collection, the majority of the growth observed in IPv6 was in the core of the network, driven primarily by transit and content providers. One specific hypothesis that CAIDA researchers wanted to address was whether the maturing IPv6 topology was becoming more or less congruent with the current IPv4 topology. They analyzed AS level path data over an eight year period to test their hypothesis and determined that the similarity between IPv4 and IPv6 Autonomous System (AS) level paths increased from 10-20% to 40-50% during that timeframe. Thus, they showed that as IPv6 matures, it is becoming more congruent with the current IPv4 topology.

While the researchers from CAIDA focused on BGP to measure growth in IPv6, research by Czyz *et al.* [21] took a broader view and examined BGP data, CAIDA traceroutes, traffic data from an ISP, and several other datasets in order to draw conclusions regarding the growth of IPv6. In their study, they focused on sixteen different metrics to measure the growth in IPv6. Specific metrics examined included the number of IPv6 address block allocations, ability to resolve hostnames using the Domain Name System (DNS) and number of queries being made for DNS Quad-A Record (AAAA) resource records, and the current usage and traffic of IPv6 as viewed from an ISP. The researchers noticed orders of magnitude differences in the results from each metric, indicating that no one metric can at the moment accurately measure and characterize the growth in IPv6. However, they were able to conclude that IPv6 is experiencing a large amount of growth and that the performance of IPv6 in now comparable to that of IPv4.

Older work from Xiao *et al.* examined the IPv6 AS-level topology [22]. They focused on studying IPv6 as a complex network and wanted to know if they could categorize IPv6 as a scale-free network. It should be noted that this research was done in 2009 before any major adoption of IPv6 had occurred. However, they were able to show IPv6 was indeed a scale-free network similar to IPv4 but that the topology of the network was less uniform than the topology in IPv4.

The previously discussed IPv6 deployment measurement studies focused mainly on using historical BGP and Ark data to measure the growth in IPv6. Instead, our research focuses on the ability to use historical active traceroute probing data and heuristic methods to conduct experimental probing of the IPv6 address space attempting to discover new IPv6 router interfaces. If we are successful in determining the feasibility and effectiveness of using heuristic methods to discover new IPv6 router interfaces, then we believe that other researchers will be able to use our heuristic methods to improve their data collection techniques in IPv6. Thereby, increasing their ability to accurately measure and characterize the growth in IPv6.

2.2.1 IPv6 Measurement Infrastructure

With the ongoing transition to IPv6, and interest by many in measuring the transition, researchers require some form of dedicated measurement infrastructure. Ideally, this measurement infrastructure would be distributed, thereby providing multiple vantage points into the network and offering researchers autonomy and flexibility in their data collection. An infrastructure able to collect data from multiple vantage points would also provide researchers a more representative sampling of the IPv6 network. Two such major infrastructures have been used to measure IPv6 deployment.

The first infrastructure currently being used to measure IPv6 deployment, and the infrastructure we used in our research, is CAIDA's Ark [23]. Ark was the evolution from CAIDA's previous `skitter`-based measurement infrastructure. Ark uses the `scamper` program to perform topology probing in both IPv4 and IPv6. `scamper` provides researchers the capability to perform `ping` and `traceroute` network measurements; additionally, `scamper` provides support for `Paris traceroute`, `Multi-path Detection Algorithm traceroute`, and various alias resolution techniques [24]. As of February 2015, Ark consisted of 106 monitors, or vantage points, with 39 IPv6 capable monitors [25]. Currently, CAIDA's Ark performs topology measurement in IPv6 by probing a random IPv6 address and the `::1` in every advertised BGP prefix from each vantage point per cycle of probing [25]. The topology measurement or probe data contains `traceroute` information from a given vantage point to a destination address. This data contains the IPv6 addresses of router interfaces traversed during the `traceroute`, Round Trip Times (RTTs), and other data from the Internet Control Message Protocol Version 6 (ICMPv6) messages returned from the `traceroute`. In addition to Ark's automatic collection of topology data, Ark allows researchers to use it in an on-demand mode. This on-demand mode allows a researcher to request Ark to perform either a `ping` or `traceroute` from a requested vantage point to a specified destination IP address. In our research, we test the ability of our heuristic methods to discover router interfaces by utilizing the topology on-demand mode of Ark.

The second infrastructure that had been used to measure IPv6 deployment is BeiHang University National Lab of Software Development Environment (NLSDE)'s Dolphin [26]. Dolphin was developed solely to collect topology information and performance information in IPv6. Unlike CAIDA at the time, Dolphin could conduct near-real time measurements of IPv6. Dolphin used a modified version of `traceroute` to collect topology data for IPv6. However, it appears that this project ceased in 2010 and that CAIDA's Ark is the only IPv6 measurement infrastructure that is currently active and in use today.

2.3 Methods to Discover IPv6 Addresses

One of the major issues faced by researchers in measuring IPv6 deployment and determining the network topology in IPv6 is developing methods to intelligently probe the IPv6 address space. While some researchers primarily focus on measuring and characterizing IPv6 growth, other researchers are working to develop new techniques to probe active portions of the IPv6 address space. This is a challenge given the size of the IPv6 address space.

A study of insecurities in IPv6 by Heuse [27] proposed a method for probing in IPv6 that we used for the basis of our heuristic method described in Section 3.2.1. Part of Heuse's research was on the feasibility of performing remote alive probing in IPv6. During the course of his research, he realized that by combining information found from search engines, IPv6 address databases, and DNS records, he could possibly determine commonly used addresses in IPv6. Using data from various IPv6 databases and DNS records, he was able to determine that, from his dataset of unique IPv6 addresses, the vast majority of addresses (approximately 60-70%) shared common host addresses. Analyzing the host addresses, Heuse determined that if a host's IPv6 address was either manually configured or provided from a DHCP server, he could leverage this information to brute force discover additional IPv6 addresses. Using this theory, he was able to brute force candidate IPv6 addresses and successfully discovered new alive hosts. Our research seeks to perform a similar technique, but focuses instead on discovering *router interface* addresses.

In their research Bellovin *et al.* postulated possible methods for worms to propagate in IPv6 and divided these methods into local versus wide area propagation methods [28]. The local area methods of propagation primarily rely upon the ability of the worm to perform network reconnaissance using an infected host machine. These local propagation methods are not relevant to our research because researchers often do not have access to the remote networks they are probing. However, several of the wide area methods of propagation could form the basis for possible heuristic methods to intelligently probe the IPv6 address space. One method discussed the fact that IPv6 servers often have low-numbered addresses to enable easy memorization by system administrators. This method supports the work performed by Heuse and again leads us to believe that we can leverage this information to find a heuristic method to discover router interfaces. A second method suggested that an IPv6 worm could perform a dictionary search of hostnames using DNS to collect candidate IPv6 addresses from the returned AAAA records. A third method proposed was that the worm could use peer-to-peer networks to learn IPv6 addresses of the hosts within the peer-to-peer network. To accomplish this, the worm would have to participate in the topology maintenance of the peer-to-peer network, watching and listening for responses to queries, and occasionally sending queries of its own in an attempt to learn host addresses.

We utilize the techniques described by Heuse and Bellovin in our IPv6 router interface discovery work.

2.4 IPv6 Alias Resolution

Router alias resolution provides researchers another way to look at the topology of a network. While the focus of large scale active topology probing is to discover router interface addresses, alias resolution seeks to determine which interfaces belong to the same physical router. Thus, alias resolution permits researchers to infer the router-level topology of a network as opposed to the interface-level topology.

Suppose one was to perform two traceroutes to the same destination from different vantage points. During the first traceroute, at some point along the path interface A is seen followed by interface C. On the second traceroute interface B is seen followed by interface C. Alias resolution seeks to show that interfaces A and B are actually different interfaces on the same router (i.e., aliases) and not interfaces on two different routers (see Figure 2.1).

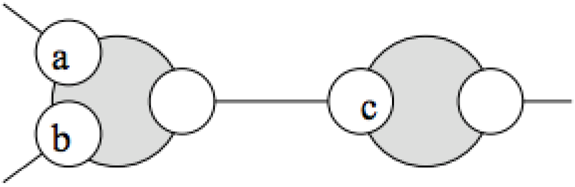

Figure 2.1: Diagram of an Alias Resolution Instance [29]

Keys [29] surveyed and discussed various methods for performing alias resolution in IPv4 and the ability to use those methods to perform Internet-scale alias resolution. Keys categorized these alias resolution techniques into two main categories: (a) fingerprinting techniques and (b) analytical techniques. He defined fingerprinting techniques as those that send probe packets to different IPv4 addresses and use identifying characteristics from the responses to infer if the responses came from the same router or not. In general, fingerprinting techniques are more accurate for alias resolution but are dependent upon the routers being configured to respond to probe packets. Analytical techniques instead attempt to draw inferences about the underlying topology of a network by analyzing the IP address graph. Analytical techniques rely upon many assumptions and, as a result, are often less accurate than fingerprinting. Some of the well known and used IPv4 alias resolution techniques discussed by Keys included `Ally`, `RadarGun`, `Analytic and Probe-based Alias Resolver (APAR)` and `kapar`. However, none of these techniques can be used in IPv6 either because they have not been, or cannot be, adapted to IPv6.

Currently, `speedtrap` is the only large-scale alias resolution technique for IPv6 [30]. The previously mentioned techniques for IPv4 alias resolution all rely on characteristics of IPv4, such as the identification (ID) field in the IPv4 header, that do not exist in IPv6. To develop IPv6 alias resolution techniques researchers needed to find unique IPv6 protocol features to exploit for the purposes of alias resolution, similar to IPv4 fingerprinting-based alias resolution techniques. Researchers discovered that by forcing a router to perform packet fragmentation, the IPv6 fragmentation extension header could be used for the purpose of alias resolution. Normally, IPv6 does not perform in-network packet fragmentation, instead placing the responsibility of fragmentation and reassembly on the end points. `speedtrap` performs alias resolution in IPv6 by inducing a router to send fragmented IPv6 packets from its control plane. `speedtrap` is then able to extract information from the fragmentation identification field in the fragmentation header to perform alias resolution using a fingerprinting technique. The functionality of `speedtrap` has been implemented into `scamper`. We use `speedtrap` to perform IPv6 alias resolution on our newly discovered IPv6 router interfaces to provide insight into the router infrastructure we are finding via our heuristics.

2.5 Subnet Inference via Router Topology Studies

A third concept that is useful in determining the underlying topology of a network is the ability to infer subnet information about the network. Gunes *et al.* [31] studied the relationship between collected IPv4 addresses from path traces of a network and the ability to infer subnet information from the collected data. Allowing researchers the ability to infer subnet information from collected path trace data providing them another means, like IP alias resolution, to generate an accurate and complete topology map of a given network. The goal of subnet inference is to determine whether seemingly separate links discovered via path traces can be merged into their single hop representation (e.g., point-to-point, multi-access, etc.). In order to infer subnet relations, Gunes *et al.* began by grouping IP addresses from the collected path trace data into candidate subnets based on the IP addresses having the same maximum *x* bit prefix. From this maximum */x* subnet their technique would then recursively form increasingly smaller candidate subnets. These candidate subnet relationships next needed to be pruned in an attempt to correlate the inferred subnet relationships to the actual subnet relationships that exist in the Internet. Gunes *et al.* proposed a set of four complementary conditions that assist in pruning down the candidate subnet relationships. We rely on some of the high-level concepts for inferring subnet information in a network discussed by Gunes *et al.* to guide the development of our inferred subnet based heuristic technique for IPv6 router interface discovery (see Section 3.2.2).

2.6 Recursive Subnet Inference (RSI) Probing Algorithm

While there is currently active research in discovering more intelligent probing primitives for IPv6, there has been similar work in discovering IPv4 intelligent probing primitives. An example of an intelligent IPv4 probing primitive is the RSI algorithm. RSI was rooted in concepts from the Subnet Centric Probing (SCP) algorithm, but went in a new direction to overcome some of the limitations of SCP. In general, RSI works by performing a binary search tree over a given input prefix. RSI will use the input prefix to determine the probing search space and divide the search space in half. The algorithm will generate a candidate address for probing at the midpoint address in each half of the search space. Based on the results from the probe, RSI will decide whether to continue recursively dividing the search space in half and probing additional candidate IP addresses or terminate the search on that branch of the binary search tree [32].

CHAPTER 3:
Methodology

As discussed previously in Section 1.1, it is feasible to probe all possible IPv4 addresses to determine the network infrastructure in IPv4. However, in IPv6 such exhaustive probing of the entire address space is unrealistic. Instead, we need more intelligent methods to discover IPv6 infrastructure and understand the topology in IPv6. One intelligent method of probing that is currently being researched is the RSI algorithm. Research into an IPv6 version of RSI has yet to be successful but has provided additional insight into subnetting in IPv6 [33]. This study instead focuses on determining the feasibility of using heuristic methods to discover IPv6 router interfaces. We used heuristic techniques to generate candidate IPv6 addresses for probing instead of performing a binary search in a given prefix to recursively generate candidate IPv6 addresses for probing.

3.1 Datasets

Our research into heuristic techniques for discovering IPv6 router interfaces utilized two unique datasets. The first set of data included all of CAIDA's Ark IPv6 topology probing from the month of July from the years 2009 to 2014 [34]. General information summarizing this data is given in Table 3.1. Although this dataset was not used to generate candidate IPv6 addresses for probing, it did provide insight into how the distribution of the host bits has changed over a period of six years. Analysis on the historical distribution of the host bits can be seen in Figures 4.1 and 4.2 and Table 4.1 with further discussion in Section 4.1.1.

The second set of data included all Ark topology probing results from January to August 2014 and a set of IPv6 router interface addresses collected by a large CDN. Table 3.2 summarizes this data. The data provided by the large CDN was only a list of IPv6 addresses and did not contain any information regarding the number of vantage points or number of traces used to generate the list. This second set of data was used to generate our list of candidate IPv6 addresses for experimental probing to test our hypotheses about our two heuristic techniques.

Name of Dataset	Number of Vantage Points	Number of Traces	Number of Unique Router Interfaces	Number of Unique Network Masks
CAIDA Ark July 2009	8	195,678	6,372	3,008
CAIDA Ark July 2010	10	331,968	9,342	4,282
CAIDA Ark July 2011	27	2,245,170	24,980	10,903
CAIDA Ark July 2012	26	3,503,595	39,630	17,716
CAIDA Ark July 2013	31	14,055,506	68,037	34,252
CAIDA Ark July 2014	35	17,044,334	76,452	36,637

Table 3.1: CAIDA Ark IPv6 Topology Datasets from July 2009 to July 2014

Name of Dataset	Number of Vantage Points	Number of Traces	Number of Unique Router Interfaces	Number of Unique Network Masks
CAIDA Ark January to August 2014	40	118,043,837	144,199	77,068
CDN	Unknown	Unknown	51,327	21,108
Combined	Unknown	Unknown	164,026	85,021

Table 3.2: CAIDA Ark and CDN Datasets used to Determine Most Common Lower-64 Bits

3.2 Heuristic-Driven Discovery

3.2.1 Heuristic #1: Frequency of Lower-64 Bits

Our first heuristic method is based off of the research previously performed by Heuse as discussed in Section 2.3. The intuition for this heuristic is that the host bits of IPv6 addresses associated with router interfaces have low-entropy. Because of this low-entropy, there exists a set of more commonly used host bits. Low-entropy in the host bits is frequently due to IPv6 addresses being statically assigned by network administrators in such a way as to ease network management. Often the assigned IPv6 address will be an address that is easily numbered and remembered by a human and facilitates association. Our hypothesis is that we can use this non-uniform distribution of host bits as a heuristic to more intelligently probe and discover IPv6 router interfaces. As a reminder, we define the "host bits" as the 64 lower, or least significant, bits of the 128-bit address. We term the upper, or most significant, 64 bits of the address as the "network bits."

The common host bits heuristic requires two distinct steps:

1. Empirically gathering common IPv6 router interface host bits. For this, we analyze historical IPv6 probing data from CAIDA's Ark measurement infrastructure and IPv6 addresses collected from a large CDN.
2. Generate candidate IPv6 addresses, based on the previously determined most common lower-64 or host bits, for use in experimental probing.

Determining Most Common Lower-64 bits of IPv6 Addresses

First, we examine the general distribution of IPv6 router interface host bits. If the host bits are uniformly distributed, then this heuristic method will not be a useful technique. To this end, we examine the Ark and CDN datasets.

The pseudo-code for our algorithm to determine the most common lower-64 bits of an IPv6 address can be seen in Algorithm 1. We first find the set of unique IPv6 addresses parsed from our datasets, while also filtering out addresses within any of the special use ranges in IPv6. Filtered IPv6 special use ranges we filtered included multicast, link and site local, private address space, and IPv6 6to4. From this set of unique IPv6 addresses, we extract the lower-64 bits of each IPv6 address and maintain a count for each unique lower-64 bit value. We then rank the lower-64 bits in order of decreasing frequency of occurrence.

Generating Candidate Addresses to Probe

The top *N* lower-64 bit values from this sorted list are used to generate valid candidate IPv6 addresses for experimental probing. As shown in Algorithm 2, we obtain a set of unique IPv6 network masks from the set of globally advertised IPv6 BGP prefixes in Routeviews [35]. For each advertised prefix, we form a candidate address by combining the advertised BGP prefix (regardless of size) with one of the most common host bit value as the lower-64 bits of the address. As an example, give the BGP prefix `2a00:1b00::` and a most common host bit value of `::1:1` we would combine them together to get the following

Algorithm 1: Histogram of Lower-64 Host Bits Among Set of IPv6 Addresses

```
Input: $Interfaces$
Output: $Lower$
$Unique \leftarrow \emptyset$
$Lower[] \leftarrow \emptyset$
for $i \in Interfaces$ do
    if $(i \notin Unique) \wedge (i \notin Special)$ then
        $Unique = Unique \cup \{i\}$
        $host = \left(i\& \left(2^{64} - 1\right)\right)$
        $Lower[host] = Lower[host] + 1$
```

candidate IPv6 address of `2a00:1b00::1:1`.

We create candidate probing lists for the top 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 150, 200, 250, 300, 350, 400, 450, and 500 most common lower-64 bit values. Note that this method of generating candidate IPv6 addresses could contain a subset of the IPv6 addresses already present in the historical data. In Section 4.1.2, we address this issue in our analysis of the experimental probing data.

Algorithm 2: Algorithm to Generate Candidate IPv6 Addresses for Experimental Probing

```
Input: $BGPPrefixes$
Input: $MostCommonLower$
Output: $TargetProbeAddresses$
for $i \in BGPPrefixes$ do
    for $j \in MostCommonLower$ do
        $TargeProbeAddress = BGPPrefix[i] || MostCommonLower[j]$
```

3.2.2 Heuristic Method #2: Inferring via `/126` Point-to-Point Links

The second heuristic examines the possibility of using historical IPv6 probing data in order to infer the existence of point-to-point links. The intuition for this heuristic is that point-to-point links are used in IPv4 to connect one router to another router and are assigned the smallest subnet necessary. Figure 3.1 provides a diagram for what we refer to as a point-to-point link in this research. Router A has an interface connected directly to an interface on Router B.

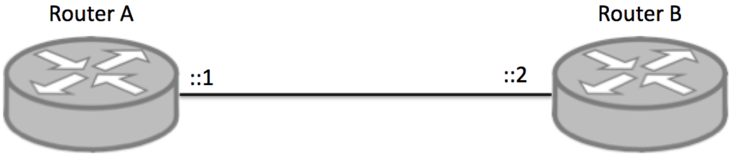

Figure 3.1: Diagram of a Point-To-Point Link Instance

In IPv4, point-to-point links are usually /30 or /31 [36]. Based on the existence of point-to-point links and their usage in connecting routers in IPv4, we posit IPv6 routers will be similarly connected. We hypothesize that given an IPv6 address and the assumption that the address is one end of a point-to-point link on a /126 subnet, we can discover new topology by probing the complementary end of the point-to-point link.

The pseudo-code to determine the corresponding IPv6 address assuming a /126 point-to-point subnet is given in Algorithm 3. We take each IPv6 address from our historical data and determine if we have not seen that address before and that it is not in any of the special use ranges in IPv6, using the same steps as discussed in Section 3.2.1. Given a unique global address, we take the IPv6 address and divide by four, which is the number of unique addresses in a /126 subnet. Based on the value of the remainder from the division operation, we either add or subtract one from the IPv6 address. This operation provides the IPv6 address corresponding to the other end on a given /126 point-to-point link. We store both the original IPv6 address and its point-to-point complement (ensuring no duplicate IPv6 addresses are stored). Once we have exhausted the IPv6 addresses from the datasets we generated our candidate IPv6 addresses for probing. To generate our candidate IPv6 addresses, we take the set of stored original IPv6 addresses and the calculated point-to-point complement IPv6 addresses and removed the set of IPv6 addresses from the original datasets.

3.3 Experimental Probing

Once we generated our candidate lists of IPv6 addresses to probe based on both heuristic techniques. We used CAIDAs Ark Topology on Demand (ToD) service to probe each of the candidate IPv6 addresses [37]. To send our probe requests into the Ark infrastructure, we

Algorithm 3: Algorithm to Infer `/126` Point-to-Point Links in IPv6

```
Input: $Interfaces$
Output: $TargetProbeAddresses$

$Unique \leftarrow \emptyset$
$PointtoPoint \leftarrow \emptyset$

for $i \in Interfaces$ do
    if $(i \notin Unique) \wedge (i \notin Special)$ then
        $Unique = Unique \cup \{i\}$
        $PointtoPoint = PointtoPoint \cup \{i\}$
        if $i \bmod 4 == 1$ then
            $PointtoPoint = PointtoPoint \cup \{i+1\}$
        else if $i \bmod 4 == 2$ then
            $PointtoPoint = PointtoPoint \cup \{i-1\}$

$TargetProbeAddress = PointtoPoint \setminus Unique$
```

feed as input into `todclient` our candidate list of IPv6 addresses. The results from each set of probing from the Ark infrastructure was stored into an output file for later analysis. By performing our experimental data collection using Ark we were able to conduct probing from various vantage points around the world using `scamper`'s implementation of `IPv6 paris-traceroute`. In our experimental probing we used 16 IPv6 vantage points, of these 9 were located in North America, 5 were located in Europe, 1 was located in Asia and 1 was located in Oceania. Prior to conducting our experimental probing we ensured all 16 vantage points were up and operational. After conducting the probing we verified that if we issued X number of traces we had X number of results before moving on to the analysis of the data. In an effort to reduce the probing load on Ark, we limited our probing rate to a maximum of 500 probe requests being processed by Ark at any given time.

3.4 Performing Alias Resolution on Experimental Results

In order to provide deeper insight into the results of our experimental probing, we performed alias resolution to determine how much new infrastructure we our discovering when probing our candidate addresses. Alias resolution allows us to determine whether newly-discovered interfaces are merely different interfaces on previously discovered routers (i.e., interfaces previously unknown that belong to a known router) or are new interfaces on previously unknown routers. To perform alias resolution on our collected data, we generate an input file of IPv6 addresses that contain all the unique IPv6 addresses from the historical data and all newly-discovered IPv6 addresses from our probing.

This list of addresses is used as input into `scamper`'s implementation of the `speedtrap` alias resolution technique, previously discussed in Section 2.4. The output from the alias resolution is pairs of IPv6 addresses that are different interfaces on the same physical router. We take each pair of aliased IPv6 addresses and convert them into a listing of all IPv6 addresses associated with a given router. The results from our alias resolution analysis can be found in Section 4.1.2 for Heuristic #1 and Section 4.2 for Heuristic #2.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 4:
Experimental Results

In Chapter 3, we introduced and discussed two potential heuristic methods for intelligently discovering IPv6 router interfaces. This chapter initially discuss the results of our historical analysis of the most commonly used lower-64 bits in router IPv6 addresses. Next, we discuss the results of our live network probing using the heuristic methods to generate candidate IPv6 addresses. Finally, we compare the relative performance of the two heuristics.

4.1 Analysis of Heuristic #1: Frequency of Lower-64 Bits

The intuition for this heuristic method was based on the fact that IPv6 addresses associated with router interfaces often have low-entropy due to manual configuration by network administrators. Our analysis of this heuristic method is divided into two separate sections; in the first one, we will discuss our analysis regarding the frequency of the lower-64 bit values of router IPv6 addresses from historical data. In the second section we will discuss the results of our experimental network probing based on the most common host bit values of an IPv6 address.

4.1.1 Analysis of the Lower-64 Bits in IPv6 Addresses

Before we were able to generate candidate IPv6 addresses based on the most common host bit values and test our hypothesis, we needed to show that there was indeed low-entropy in the host bits associated with router interfaces in IPv6. We also sought to determine what host bit values occurred more frequently than others. We initially began our analysis by observing the frequency in which host bit values occurred based on CAIDA Ark probing data as collected in the month of July over a six year period. This allowed us to determine if certain host bit values occur more frequently than others and if so, how they changed over a six year period.

Figure 4.1 summarizes the observed behavior in the frequency of host bit values from the month of July from the period of 2009 to 2014. We observed that a very small number of unique host bit values accounted for approximately 60% of all the host bit values observed in the datasets. However, due to this behavior of host bit values, the data we are most interested in is compressed against the y-axis in Figure 4.1. Figure 4.2 adjusts the plot axes to focus in on the area of interest. By focusing on the area of interest near the y-axis, we concluded that over the six years of data that about 100 unique host bit values accounted for approximately 50-60% of all the host bit values observed in the datasets.

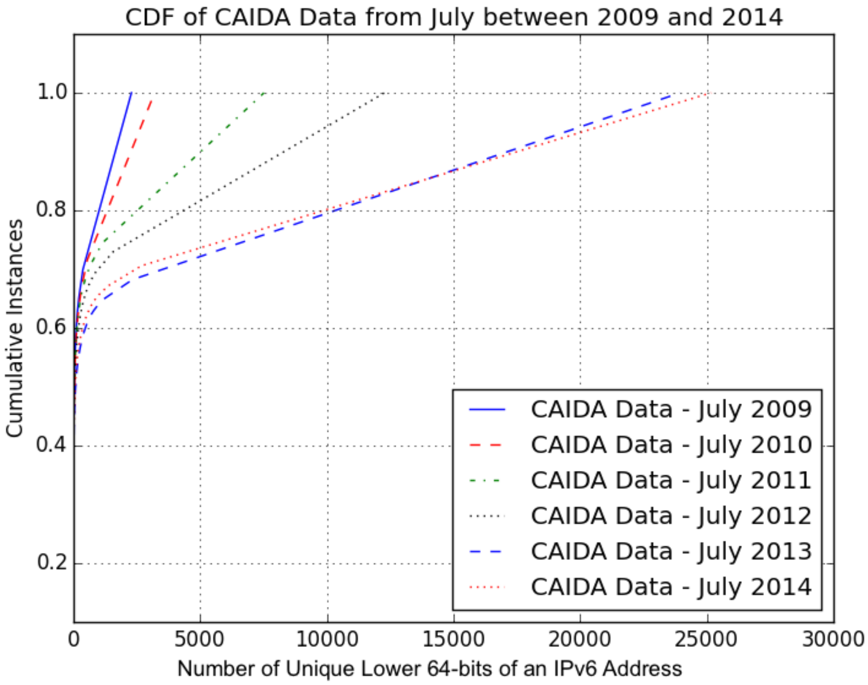

Figure 4.1: Cumulative Distribution of Historical Lower-64 Bits of IPv6 Address from CAIDA Datasets from July 2009 to July 2014

Table 4.1 contains the top 10 most common host bit values from the CAIDA July 2009 and CAIDA July 2014 datasets. In both datasets the host bit values of ::1 and ::2 represented on average about 35% of all host bit values in the CAIDA data. The next most common host bit values on average individually comprised less than 1% of all host bit values contained in the data. It is also clear from our analysis that over time the most common host bit values do not vary much each year; for the most part the most common host bit values in July 2009 were the most common host bit values in July 2014. A final observation is that the majority of the top 30 most common host bit values seem to use only the eight least significant host bits in an IPv6 address.

Additionally, we observed that each dataset shown in Figure 4.1 has an inflection point in

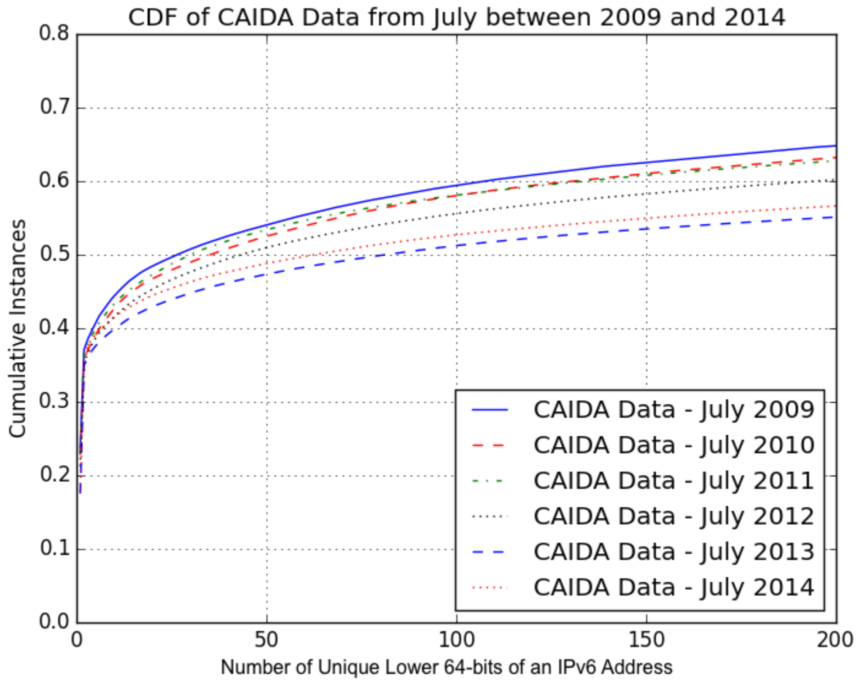

Figure 4.2: Cumulative Distribution of Historical Lower-64 Bits of IPv6 Address from CAIDA Datasets from July 2009 to July 2014 (Zoomed In)

	CAIDA July 2009 Dataset			**CAIDA July 2014 Dataset**		
Top #	IPv6 Host Bits	Frequency of Host Bits	Percentage of Dataset	IPv6 Host Bits	Frequency of Host Bits	Percentage of Dataset
1	::2	1,516	23.79%	::2	13,627	17.82%
2	::1	849	13.32%	::1	13,429	17.57%
3	::6	91	1.43%	::3	1,774	2.32%
4	::3	70	1.10%	::6	602	0.79%
5	::a	66	1.04%	::a	475	0.62%
6	::5	64	1.00%	::5	437	0.57%
7	::12	47	0.74%	::12	355	0.46%
8	::e	45	0.71%	::4	336	0.44%
9	::9	44	0.69%	::11	307	0.40%
10	::16	39	0.61%	::9	307	0.40%

Table 4.1: Top Ten Lower-64 bits of an IPv6 Address from CAIDA Datasets from July 2009 and July 2014

the curvature of the graph around 60% to 70% of all the host bit values observed. These inflection points become more pronounced each year. While we did not pursue any further investigation regarding the significance of these inflection points. We hypothesize that the reason these inflection points are becoming more pronounced each year is because of the growth of IPv6. Coupled with the growth of IPv6 is the need to add additional IPv6 infrastructure to the network. The increase in IPv6 infrastructure would require the addition of new routers and router interfaces in the network. As router interfaces are added into the network, one must assign a unique IPv6 address to the interface. We surmise that as network administrators address these new router interfaces they first will do so using addresses from the set of common host bit values. However, once they have used up the common host bit values, they begin to assign address to interfaces using another addressing scheme. This addressing scheme appears to be different for each network based on the presence of the tail in each graph.

To remove the potential bias due to examining traceroute probe data from a single source, we additionally analyzed data from a large CDN and compared it to CAIDA's data. Figures 4.3 and 4.4 both summarize the observed behavior in the frequency of host bit values from the CAIDA 2014 and CDN datasets. The behavior observed in these two datasets is very similar to the behavior observed in our earlier analysis. We observed that a small number of unique host bit values comprised 60% of all the host bit values, and that each graph has a distinct inflection point.

We then analyzed the combined data sets in order to obtain the most representative view of IPv6 router addressing. Table 4.2 contains the 10 most common host bit values from the combined CAIDA and CDN datasets. Once again we observed that the most common host bit values are `::1` and `::2`, accounting for 31% of all the unique host bit values. Similar to our earlier analysis from above, the next most common host bit values on average individually comprised less than 1% of all the host bit values.

From our analysis on the frequency of host bit values, we concluded that there is indeed low-entropy in the host bits associated with IPv6 router interfaces. Due to the low-entropy we were able to show the existence of a set of more commonly used host bit values used to address IPv6 router interfaces. With this knowledge, we were then able to perform experimental testing of a heuristic method that uses the most common host bit values to

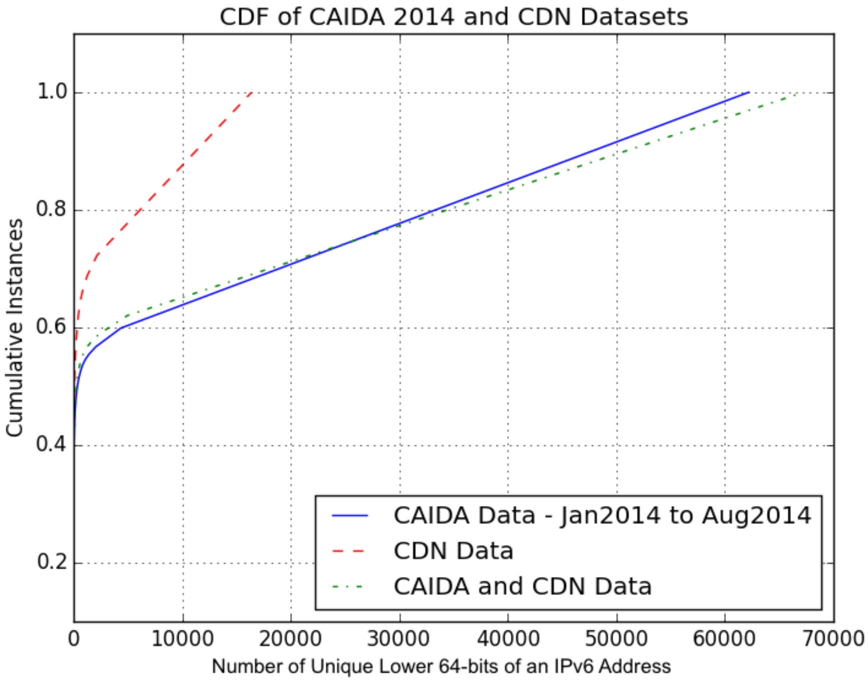

Figure 4.3: Cumulative Distribution of Historical Lower-64 Bits of IPv6 Address from Combined CAIDA and CDN Datasets

generate candidate IPv6 addresses for probing.

4.1.2 Analysis of Experimental Probing Results

The preceding analysis found an inflection point in the distribution of router host addresses where approximately 50% of all addresses use one of 500 different host bit values. We therefore use the combined data from the Ark topology probing results from January to August 2014 and a large CDN to determine the 500 most common host bit values used by IPv6 router interfaces. Next, we combined the 500 most common host bits with the19,441 advertised BGP prefixes (as of September 2014) to generate our candidate IPv6 addresses used for out experimental probing. While we could have conducted the experimental probing by probing all 500 most common host bit values at once, we broke the probing down into smaller sets of probing. We created this subdivision for two reasons: first, it allowed for more granularity in the results allowing us to better observe the effect of increasing the number of most common host bit values probed to the number of router interfaces discov-

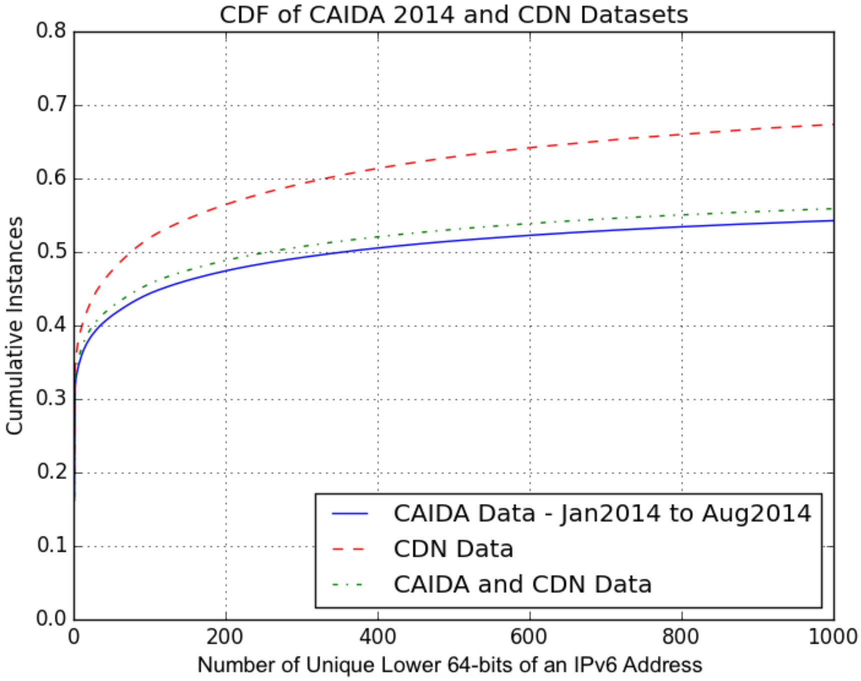

Figure 4.4: Cumulative Distribution of Historical Lower-64 Bits of IPv6 Address from Combined CAIDA and CDN Datasets (Zoomed In)

ered. Second, by splitting our probing into multiple rounds we reduced the workload on the Ark infrastructure and limited the impact on our probing if one of the monitors failed during a round of probing thereby causing us to restart that round of probing again.

Our experimental probing sets were divided such that for the top 100 most common host bit values we would probe the top 10 most common host bit values appended to the advertised BGP prefixes, then we would probe the top 11-20 most common host bit values appended to the advertised BGP prefixes, and so forth. Once we finished the top 100 most common host bit values, our probing technique changed slightly such that we then probed the 101-150 most common host bit values, followed by the 151-200 most common host bit values, and so forth until we conducted probing for all 500 most common host bit values.

Once we completed our experimental probing, we began our analysis by creating a list containing the unique IPv6 address hops observed in the collected `traceroute` data. To generate this list of unique IPv6 addresses observed, we parsed the IPv6 address for each

	CAIDA 2014 and CDN Dataset		
Top #	IPv6 Host Bits	Frequency of Host Bits	Percentage of Dataset
1	::1	27,306	16.65%
2	::2	25,376	15.47%
3	::3	2,519	1.54%
4	::6	980	0.60%
5	::a	839	0.51%
6	::5	835	0.51%
7	::4	642	0.39%
8	::11	573	0.35%
9	::12	563	0.34%
10	::9	555	0.34%

Table 4.2: Top Ten Lower-64 bits of an IPv6 Address from Combined CAIDA and CDN Datasets

hop in the `traceroute` output using a similar methodology as the one used for processing the historical CAIDA Ark topology data. By then removing the IPv6 address that we originally observed in the CAIDA and CDN data from our list of unique IPv6 addresses, we are able to determine the new IPv6 router addresses discovered as a result of our heuristic based experimental probing. These newly discovered IPv6 router addresses include both the probing target IPv6 addresses and the intermediate router address seen on the `traceroute` path.

The results of our experimental probing using the top 500 most common host bit values is shown in Figure 4.5. From the top 10 most common host bit values we discovered approximately 5,500 new router interface addresses. Additionally, as we increased the number of most common host bit values probed, we continue to see a gradual increase in the number of new router interfaces. However, there was a single anomaly in our experimental data in which we observed a large jump in the number of interfaces discovered. This large jump in our results occurred between the top 300 and top 350 most common host bit values. This anomaly was most likely caused by a several month gap in our experimental probing caused by multiple failures in the Ark infrastructure that required us to restart our probing of the top 301-350 most common host bits after each failure.

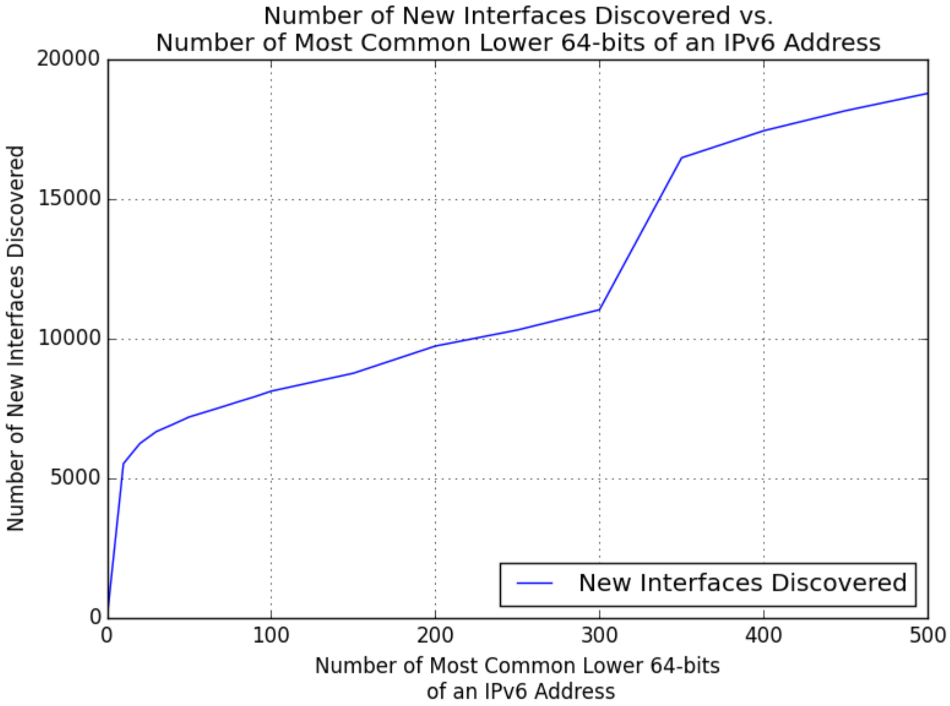

Figure 4.5: Number of New Interfaces Discovered using Heuristic #1

Using the speedtrap alias resolution technique, we conducted alias resolution using the 164,026 unique IPv6 addresses from our combined data set along with the 18,077 new router interfaces discovered from our heuristic. We found that 17% of the newly discovered router interfaces from our probing were interfaces on previous unseen router infrastructure. Another 2% of the newly discovered router interfaces were interfaces on previously seen router infrastructure. The remaining 81% consisted of previously discovered router interfaces on previously seen router infrastructure.

In our analysis of the experimental probing results, we wanted to see how the probing order impacts the rate of new router interfaces discovered. To answer this question, we investigate three ordering strategies: i) decreasing popularity (e.g., Top 1-10 host bits, followed by Top 11-20 host bits, followed by Top 21-30 host bits, etc.); ii) increasing popularity (e.g., Top 251-300 host bits, followed by Top 201-250 host bits, followed by Top 151-200 host bits, etc.); and iii) random. Figure 4.6 plots the rate of new interfaces discovered according to each of these orderings. For this portion of our analysis, we only considered the top

300 most common host bit values to avoid tainting our analysis with the large jump in new interfaces discovered due to the several month gap in experimental probing. From Figure 4.6 we observe that there is a significant effect on the initial rise of newly discovered router interfaces by selecting the Top 300 host bits in increasing order of popularity vice decreasing order of popularity. However, there seems to be no significant effect on the initial rise when comparing the randomly chosen order of popularity to the decreasing order of popularity. In general, we would expect the number of new interfaces discovered by randomly choosing the order of popularity to have as an upper bound the number of new interfaces discovered by decreasing order of popularity and have as a lower bound the number of new interfaces discovered by increasing order of popularity. These results suggest that further investigation into the effect of probing order on topology discovery is warranted.

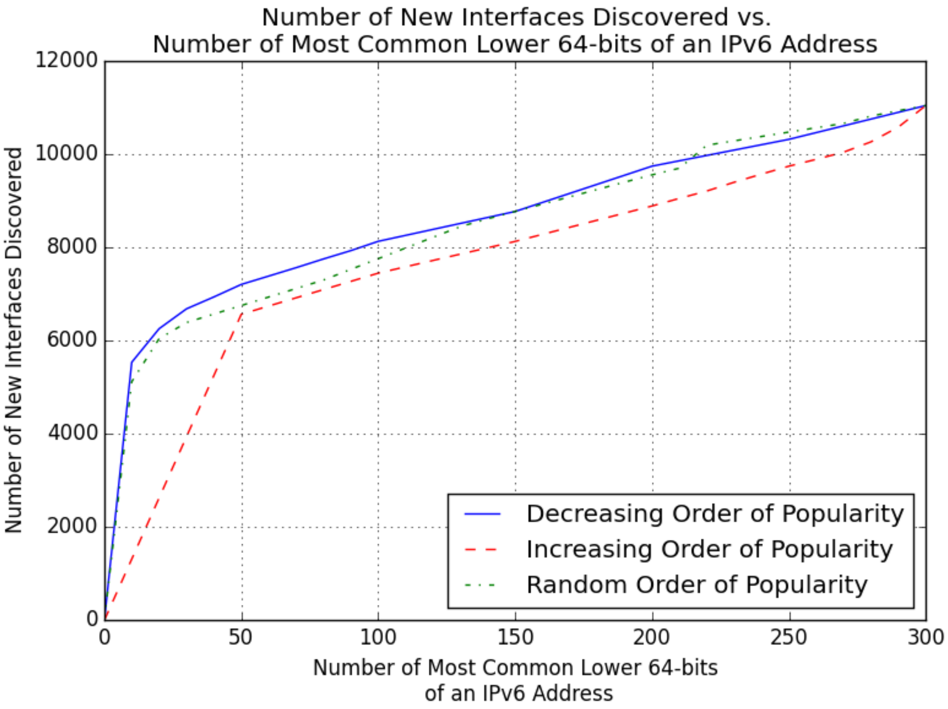

Figure 4.6: Effects of Popularity Order on Number of New Interfaces Discovered using Heuristic #1

Finally, we sought to determine the fraction of newly discovered router addresses that were intermediate hops along the path versus the target itself (since our targets are presumably router interfaces). To do this we first examined the fraction of target IPv6 addresses responding to our probe request. For the Top 10 most common host bit values appended to a given BGP prefix only 6.5% of the 194,420 target addresses probed responded to the probe request. As the number of Top *N* most common host bit values increased the fraction of responding target addresses steadily decreased. Next, we looked at what fraction of the new interfaces discovered were the target. Of the 6.5% of target addresses that responded to the probe request, about 5.6% were new interfaces that were discovered. Table 4.3 contains the percentages of target addresses that responded to our probe requests and the percent of those that did respond that are newly discovered interfaces for the Top 100 most common host bit values.

Top #	Number of Target IPv6 Addresses	Percentage of Target Addresses Responding to Probe	Percentage of Target Addresses that Responded to Probe that are Newly Discovered Interfaces
10	194,420	6.52%	5.66%
20	388,840	4.19%	8.41%
30	583,260	3.26%	9.96%
40	777,680	2.68%	11.09%
50	972,100	2.30%	12.05%
60	1,166,520	2.05%	12.87%
70	1,360,940	1.86%	13.88%
80	1,555,360	1.71%	14.72%
90	1,749,780	1.58%	15.35%
100	1,944,200	1.46%	15.77%

Table 4.3: Percentages of Top 100 Target Addresses that Responded to Probing and are Newly Discovered Interfaces

4.2 Analysis of Heuristic Method #2: Inferring via /126 Point-to-Point Links

The intuition for this heuristic method is based on the assumption that point-to-point links are used in IPv6 to connect routers to each other and that network administrators often will assign these point-to-point links the smallest subnet necessary. To test this heuristic, we used the same dataset used to test our first heuristic method. By using the same input data for both heuristics, we can meaningfully compare the ability of each heuristic method to discover router interfaces. Using our heuristic as described in Section 3.2.2, we generate 127,748 candidate IPv6 addresses for use in our experimental probing. We analyzed the probing results and found that we had discovered 10,157 new IPv6 router interface addresses.

While we were able to discover a non-trivial number of new interfaces via this heuristic method, we performed additional research regarding the subnet sizes associated with IPv6 point-to-point links. Our research suggests that there does not yet appear to be a standard subnet size associated with point-to-point links in IPv6. Some of the literature suggests using a /127 subnet for point-to-point links [38]; other literature suggests using a /64 for point-to-point links [39]. The effectiveness of this heuristic at discovering new router interfaces could be improved by additional research using different subnet sizes for inferring the endpoint IPv6 addresses for a given point-to-point link.

Using the speedtrap alias resolution technique, we conducted alias resolution using the 164,026 unique IPv6 addresses from our combined data set along with the 10,157 new router interfaces discovered from our heuristic. We found that 16% of the newly discovered router interfaces from our probing were interfaces on previous unseen router infrastructure. Another 2% of the newly discovered router interfaces were interfaces on previously seen router infrastructure. The remaining 82% consisted of previously discovered router interfaces on previously seen router infrastructure.

As before in Heuristic #1, we sought to determine whether our experimental probing was discovering new router interfaces at our target probing address or were we the new interfaces discovered simply new intermediate router interfaces. To do this we first looked at what fraction of the target addresses probed to responded to our probe request. Of the

127,748 target addresses probed about 40.7% of those target addresses responded to the probe request. Next we looked at what fraction of the new interfaces discovered were target address that responded to our probe request; in our experiential probing 31.7% of the new interfaces we discovered were the target address used for probing.

4.3 Comparison of Heuristic Methods

In this section, we compare our two heuristic methods and their ability to discover new IPv6 router infrastructure. While each heuristic method did yield a non-trivial number of router interfaces discovered, each method required a significant amount of experimental probing. One way to compare these two heuristic methods is to evaluate them by the relative measure of number of new interfaces discovered to the number of candidate IPv6 addresses probed. For the first heuristic method, using the top 500 most common host bits it was able to discover 18,773 router interfaces but required experimental probing of 9,720,500 candidate IPv6 addresses, a ratio of 0.002. However, if we consider only the top 10 most common host bits for the first heuristic we see significant improvements in the number of new interfaces discovered compared to the number of candidate IPv6 addresses probed. In this case, the heuristic was able to discover 5,532 router interfaces with only probing 194,410 candidate IPv6 addresses, a ratio of 0.028. The second heuristic method instead was able to discover 10,157 router interfaces while requiring only 127,748 candidate IPv6 addresses, a ratio of 0.080.

Overall, the second heuristic method was more effective at discovering the maximum number of new router interfaces with the least amount of candidate IPv6 addresses probed. Both heuristic methods performed equally as well with regards to the alias resolution results and the discovery of previously unseen router infrastructure.

CHAPTER 5: Conclusion

This thesis investigated the feasibility of using heuristic techniques to efficiently discover router infrastructure in IPv6. While we considered numerous possible methods to study, our research focused on two heuristics.

The first heuristic method relied upon finding a set of the most commonly used lower-64 bits in IPv6 router interface addresses and appending these most common lower-64 bit values to all advertised BGP prefixes to generate a list of candidate IPv6 addresses for probing. We show that, even though there are approximately 1.84×10^{19} possible lower-64 bit values, only a small number of these are used in the deployed Internet as inferred from our historical data. Additionally, we observed that this set of commonly used lower-64 bit values remained fairly constant over a six year period. From our experimental probing using the top 500 most common host bit values, we were able to discover a non-trivial amount of previously undiscovered IPv6 router infrastructure. By probing only the top 10 most common host bit values, this heuristic yielded the largest number of new IPv6 router interfaces discovered in a single round of experimental probing.

The second heuristic method relied on the assumption that point-to-point links in IPv6 use `/126` subnets. Similar to our results from the first heuristic, we again were able to discover a non-trivial amount of IPv6 router interfaces from our experimental probing. However, unlike our first heuristic method this method discovered the greatest number of IPv6 router interfaces with the least amount of experimental probing.

In conclusion, we showed that simple heuristic techniques are a feasible and effective solution to the problem of discovering router infrastructure in IPv6.

5.1 Future Work

This section presents suggestions for future work that will build upon the starting point of our research into using heuristic methods for discovering router interfaces in IPv6.

5.1.1 Research into Other Heuristic Methods

While we only studied two heuristics in this thesis, additional research into other heuristic techniques needs to be performed:

- One possible heuristic method that could be studied involves completing the sequence between known IPv6 address. As an example, assume that the following IPv6 addresses exist `2001:500:3::42`, `2001:500:3::45`, and `2001:500:3::46`. We could logically assume that there may exist network devices that would respond to probing at the following two IPv6 addresses `2001:500:3::43` and `2001:500:3::44`. Barnes *et al.* have previously conducted research into this sequence completion heuristic [40]. Their work from 2012 showed they had limited success discovering IPv6 infrastructure using a sequence completion heuristic. With the exponential growth currently being experienced in IPv6 we recommend that the sequence completion heuristic should be reinvestigation.
- Another heuristic that could be studied involves searching DNS records associated with known IPv6 router interfaces and looking for patterns in the hostnames assigned to the router interfaces. We could then use the observed patterns in the hostnames of router interfaces to query for associated AAAA DNS records that may return candidate IPv6 addresses that would respond to probing. As an example, given the following IPv6 address of `2001:1900:29::a` corresponding to a router interface. Performing a reverse DNS lookup with the given IPv6 address yields a DNS PTR record of `vl-5.car1.phoenix1.level3.net.`. The returned PTR record indicates several possible patterns used when providing the hostname to this particular router interface. In this example, we could try requesting the AAAA DNS record for `vl-5.car2.phoenix1.level3.net.`. The returned AAAA DNS record provides a candidate IPv6 address of `2001:1900:29::e` that may respond to experimental probing.
- Finally, a technique that generates candidate IPv6 addresses for probing by participating in peer-to-peer networks, as suggested in the Bellovin *et al.* [28], may reveal previously unknown infrastructure.

5.1.2 Integration of Heuristic #1 into CAIDAs Ark

As discussed in Section 2.2.1, CAIDAs Ark currently only probes a random IPv6 address in a given BGP prefix per round of probing. We suggest that CAIDA, in addition to their current method of probing the IPv6 address space, add probing for the top 10 most common lower-64 bit values into each round of probing. Based on our results, we believe that this additional probing will provide additional useful topology data without incurring a significant amount of overhead in time or processing to complete a round of probing.

THIS PAGE INTENTIONALLY LEFT BLANK

List of References

[1] Internet Corporation for Assigned Names and Numbers. (2011, February). Available pool of unallocated ipv4 internet addresses now completely emptied. [Online]. Available: https://www.icann.org/en/system/files/press-materials/release-03feb11-en.pdf.

[2] American Registry for Internet Numbers. (2011, February). The iana ipv4 address free pool is now depleted. [Online]. Available: https://www.arin.net/announcements/2011/20110203.html.

[3] Asia Pacific Network Information Centre. (2011, April). Apnic ipv4 address pool reaches final /8. [Online]. Available: https://www.apnic.net/publications/news/2011/final-8.

[4] Réseaux IP Européens. (2012, September). Ripe ncc begins to allocate ipv4 address space from the last /8. [Online]. Available: https://www.ripe.net/internet-coordination/news/announcements/ripe-ncc-begins-to-allocate-ipv4-address-space-from-the-last-8.

[5] G. Huston. (2015, February). Ipv4 address report. [Online]. Available: http://www.potaroo.net/tools/ipv4/index.html.

[6] S. Deering and R. Hinden, "Internet protocol, version 6 (ipv6) specification," Internet Requests for Comments, RFC Editor, RFC 2460, December 1998, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc2460.txt.pdf.

[7] M. H. Warfield *et al.*, "Security implications of ipv6," *Internet Security Systems*, vol. 4, no. 1, pp. 2–5, 2003, [Online]. Available: http://www.blackhat.com/presentations/bh-federal-03/bh-federal-03-warfield/bh-fed-03-warfield.pdf.

[8] K. Claffy, "Tracking ipv6 evolution: Data we have and data we need," *ACM SIGCOMM Computer Communication Review (CCR)*, vol. 43, no. 3, pp. 43–48, Jul 2011.

[9] P. Srisuresh and K. Egevang, "Traditional ip network address translator (traditional nat)," Internet Requests for Comments, RFC Editor, RFC 3022, January 2001, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc3022.txt.pdf.

[10] S. Perreault, I. Yamagata, S. Miyakawa, A. Nakagawa, and H. Ashida, "Common requirements for carrier-grade nats (cgns)," Internet Requests for Com-

ments, RFC Editor, RFC 6888, April 2013, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc6888.txt.pdf.

[11] Google. (2015, February). Google ipv6 adoption. [Online]. Available: http://www.google.com/intl/en/ipv6/statistics.html.

[12] A. Dhamdhere, M. Luckie, B. Huffaker, K. Claffy, A. Elmokashfi, and E. Aben, "Measuring the deployment of ipv6: Topology, routing and performance," in *Internet Measurement Conference (IMC)*, Nov 2012, pp. 537–550.

[13] S. Ballmer. (2013, July). Steve ballmer: Worldwide partner conference 2013 keynote. [Online]. Available: http://news.microsoft.com/2013/07/08/steve-ballmer-worldwide-partner-conference-2013-keynote/.

[14] A. Dainotti, A. King, K. Claffy, F. Papale, and A. Pescapè, "Analysis of a "/0" stealth scan from a botnet," in *Proceedings of the 2012 ACM Conference on Internet Measurement Conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 1–14.

[15] Akamai. (2015, February). Akamai ipv6 adoption. [Online]. Available: http://www.akamai.com/ipv6.

[16] L. Colitti, S. H. Gunderson, E. Kline, and T. Refice, "Evaluating ipv6 adoption in the internet," in *PAM 2010*, 2010.

[17] National Institute of Standards and Technology. (2015, February). Estimating usg ipv6 & dnssec external service deployment status. [Online]. Available: http://fedv6-deployment.antd.nist.gov/cgi-bin/generate-gov.

[18] R. Hinden and S. Deering, "Internet protocol version 6 (ipv6) addressing architecture," Internet Requests for Comments, RFC Editor, RFC 3513, April 2003, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc3513.txt.pdf.

[19] S. Thomson and T. Narten, "Ipv6 stateless address autoconfiguration," Internet Requests for Comments, RFC Editor, RFC 2462, December 1998, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc2462.txt.pdf.

[20] R. Droms, J. Bound, B. Volz, T. Lemon, C. Perkins, and M. Carney, "Dynamic host configuration protocol for ipv6 (dhcpv6)," Internet Requests for Comments, RFC Editor, RFC 3315, July 2003, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc3315.txt.pdf.

[21] J. Czyz, M. Allman, J. Zhang, S. Iekel-Johnson, E. Osterweil, and M. Bailey, "Measuring ipv6 adoption," in *Proceedings of the 2014 ACM conference on SIGCOMM*. ACM, 2014, pp. 87–98.

[22] B. Xiao, L.-d. Liu, X.-c. Guo, and K. Xu, “Modeling the ipv6 internet as-level topology,” *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 4, pp. 529–540, 2009.

[23] K. Claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov, “Internet mapping: From art to science,” in *Conference For Homeland Security, 2009. CATCH’09. Cybersecurity Applications & Technology*. IEEE, 2009, pp. 205–211.

[24] M. Luckie, “Scamper: A scalable and extensible packet prober for active measurement of the internet,” in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*. ACM, 2010, pp. 239–245.

[25] Center for Applied Internet Data Analysis. (2015, March). Archipelago measurement infrastructure. [Online]. Available: http://www.caida.org/projects/ark/.

[26] X. Lang, G. Zhou, C. Gong, and W. Han, “Dolphin: The measurement system for the next generation internet,” in *Communications, Internet, and Information Technology*, 2005.

[27] M. Heuse, “Recent advances in ipv6 insecurities,” in *Chaos Communications Congress*, 2010.

[28] S. M. Bellovin, B. Cheswick, and A. Keromytis, “Worm propagation strategies in an ipv6 internet,” *LOGIN: The USENIX Magazine*, vol. 31, no. 1, pp. 70–76, 2006.

[29] K. Keys, “Internet-scale ip alias resolution techniques,” *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 50–55, 2010.

[30] M. Luckie, R. Beverly, W. Brinkmeyer, and K. Claffy, “Speedtrap: Internet-scale ipv6 alias resolution,” in *Internet Measurement Conference (IMC)*, Oct 2013, pp. 119–126.

[31] M. H. Gunes and K. Sarac, “Inferring subnets in router-level topology collection studies,” in *Proceedings of the 7th ACM SIGCOMM Conference on Internet measurement*. ACM, 2007, pp. 203–208.

[32] G. Baltra, R. Beverly, and G. Xie, “Ingress point spreading: A new primitive for adaptive active network mapping,” in *Passive and Active Measurement*. Springer, 2014, pp. 56–66.

[33] B. LaFever, “Methods for intelligent mapping of the ipv6 address space,” master’s thesis, Naval Postgraduate School, March 2015.

[34] Center for Applied Internet Data Analysis. (2015, March). The ipv6 topology dataset. [Online]. Available: http://www.caida.org/data/active/ipv6_allpref_topology_xdataset.xml.

[35] D. Meyer. (2014, September). The ipv6 bgp routeviews dataset. [Online]. Available: http://routeviews.org/route-views6/bgpdata/2014.09/RIBS/.

[36] A. Retana, R. White, V. Fuller, and D. McPherson, “Using 31-bit prefixes on ipv4 point-to-point links,” Internet Requests for Comments, RFC Editor, RFC 3021, December 2000, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc3021.txt.pdf.

[37] Center for Applied Internet Data Analysis. (2015, March). Caida topology on demand. [Online]. Available: http://www.caida.org/projects/ark/.

[38] M. Kohno, B. Nitzan, R. Bush, Y. Matsuzaki, L. Colitti, and T. Narten, “Using 127-bit ipv6 prefixes on inter-router links,” Internet Requests for Comments, RFC Editor, RFC 6164, April 2011, [Online]. Available: http://www.rfc-editor.org/pdfrfc/rfc6164.txt.pdf.

[39] C. Grundermann, A. Hughes, and O. DeLong. (2011, October). Best current operational practices - ipv6 subnetting. [Online]. Available: http://bcop.nanog.org/index.php/IPv6_Subnetting.

[40] R. Barnes, R. Altmann, and D. Kerr. (2012, February). Mapping the great void: Smarter scanning for ipv6. [Online]. Available: http://www.caida.org/workshops/isma/1202/slides/aims1202_rbarnes.pdf.

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California