COLLECTION AND ANALYSIS OF INTERNET TOPOLOGY DATA

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of Master of Science in
Computer Science and Engineering

by

Jay Dee Anthony Thom
Dr. Mehmet Hadi Gunes / Thesis Advisor
May 2017
We recommend that the thesis prepared under our supervision by JAY DEE ANTHONY THOM entitled COLLECTION AND ANALYSIS OF INTERNET TOPOLOGY DATA be accepted in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE

Mehmet H. Gunes, Ph.D. – Advisor

Lei Yang, Ph.D. – Committee Member

Hanif Livani, Ph.D. – Graduate School Representative

David Zeh, Ph.D. – Dean, Graduate School

May 2017
ABSTRACT

Consisting of billions of endpoints, the Internet was designed without a means of monitoring or measuring itself, leaving a difficult problem for design, maintenance, and research. The focus of this study is the acquisition and analysis of data provided by the public Internet topology measurement platforms. In particular, this thesis will focus on the router level probe datasets from USC/ISI Ant census, CAIDA Archipelago (Ark), UW Information Plane (iPlane), Measurement Lab (M-Lab), and RIPE NCC Atlas, and Autonomous System level BGP datasets from UCSD CAIDA, UCLA Internet Research Lab (IRL), and CIDR. Using tools such as Ping and Traceroute, these platforms have performed measurements utilized by researchers over several years, and have provided data useful for measurement, performance testing, and visualization. However, one of the problems encountered is that this data is stored in multiple formats, making it difficult to collect and use. We have developed tools to gather this data and restructure it using a common file system and format. Once collected, it is utilized in conjunction with data acquired by our own platform and will create a map of the Internet backbone. We provide insights into the characteristics of these public platforms and the properties of the data sets sampled by them. In addition, we compare the coverage of Autonomous Systems by these platforms based on several perspectives. Finally, we develop a network of single board computers (Odroids) which will provide a platform for generating future topology measurement data. We address issues regarding its architecture and deployment.
I dedicate this work to my family and friends who have been a great support to me.
ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Mehmet Hadi Gunes, and my committee members Dr. Lei Yang and Dr. Hanif Livani for their time and suggestions. I would also like to thank my partners in this project; Abdullah Canbaz, Ahmet Aksoy, Khalid Bakhshaliyev, Mustafa Solmaz, Zachary Newell, and others who came before us that contributed to this ongoing work. Finally, I would like to thank my family for their patience and support.

This thesis is based in part upon work supported by the National Science Foundation under grant numbers CNS-1321164 and EPS-IIA-1301726. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>i</td>
</tr>
<tr>
<td>Dedication</td>
<td>ii</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>iii</td>
</tr>
<tr>
<td>Table of Contents</td>
<td>iv</td>
</tr>
<tr>
<td>List of Tables</td>
<td>vi</td>
</tr>
<tr>
<td>List of Figures</td>
<td>vii</td>
</tr>
</tbody>
</table>

1 Introduction                                           1

2 Background

2.1 Preliminaries                                      4  
2.1.1 Autonomous System (AS)                          4  
2.1.2 Regional Internet Registry (RIR)                5  
2.1.3 Border Gateway Protocol (BGP)                   6  

2.2 Internet Topology Measurement                      7  
2.2.1 Ping                                             8  
2.2.2 Traceroute                                       8  

2.3 Network Measurement Platforms                     10  
2.3.1 UCSD Caida Archipalago Project (Ark)             10  
2.3.2 UW Information Plane (iPlane)                   11  
2.3.3 Measurement Lab (M-Lab)                         12  
2.3.4 Ripe Atlas (Ripe)                               13  
2.3.5 USC/ISI Ant Census                              13  
2.3.6 PlanetLab                                        14  
2.3.7 UCLA IRL                                        15  
2.3.8 CIDR                                            16  
2.3.9 UCSD Caida BGP Stream                           16

3 Related Work                                           17  
3.1 Internet Topology Measurement Issues               17  
3.1.1 Survey of Internet Topology Discovery Methods    17  
3.1.2 IP Alias Resolution                             18  
3.1.3 Survey of Network Performance Measurements      18  

3.2 Internet Topology Measurement Platforms             19  
3.2.1 mrinfo-rec                                      19  
3.2.2 Cheleby                                         20  
3.2.3 Archipelago                                     20  
3.2.4 Ant Census                                      21  
3.2.5 Measurement Lab                                 21  
3.2.6 PlanetLab                                       22  
3.2.7 Information Plane                               22  

4 Data Collection

4.1 Storage Formats ........................................ 24
  4.1.1 UCSD Caida Ark .................................... 25
  4.1.2 UW Information Plane ................................. 26
  4.1.3 M-Lab .............................................. 27
  4.1.4 Ripe ............................................... 27
  4.1.5 USC/ISI Ant Census .................................. 28

4.2 DataCollector ........................................... 28

4.3 Output Format .......................................... 29

5 Data Analysis ............................................. 31

5.1 30-day Router Level Topology Data ..................... 31
  5.1.1 Total IP, Trace, Edge, Source, and Destination Statistics ... 34

5.2 5-day Router Level Topology Data ........................ 35
  5.2.1 Total IP, Trace, Edge, Source, and Destination Statistics ... 36
  5.2.2 Data Analysis per Source ............................. 40
  5.2.3 Data Analysis per Destination ........................ 43

5.3 AS Coverage ............................................ 46
  5.3.1 BGP Coverage ...................................... 47
  5.3.2 AS Rank ........................................... 49

5.4 Unresponsive Routers ................................... 53

5.5 Routing Loops .......................................... 54

5.6 IP Address Observance in Traces vs. Pings .............. 55

5.7 BGP Level Topology Data ................................ 56
  5.7.1 BGP Announcements ................................ 57
  5.7.2 MOAS Prefixes .................................... 57
  5.7.3 Prefix Distribution .................................. 59

6 Deployment of Internet Measurement Platform ............. 61

6.1 The Odroid Device ...................................... 61

6.2 Network Architecture .................................. 62

6.3 Deployment of IM Nodes ................................ 63

7 Conclusion ................................................. 70

7.1 Summary ................................................. 70

7.2 Future Work ........................................... 71

Bibliography ................................................. 72
LIST OF TABLES

5.1 Coverage for 5 days data ........................................... 36
5.2 BGP coverage .......................................................... 48
5.3 IP addresses not seen in Ant Census .............................. 56
5.4 BGP announcements .................................................... 57
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Number</th>
<th>Figure Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Traceroute tool implementation</td>
<td>9</td>
</tr>
<tr>
<td>2.2</td>
<td>Probes used by the Archipelago (Ark) project. [2]</td>
<td>11</td>
</tr>
<tr>
<td>2.3</td>
<td>Probes used by Ripe NNC. [3]</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>An example of a Ripe NNC anchor. [4]</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>The PlanetLab slice architecture. [5]</td>
<td>15</td>
</tr>
<tr>
<td>4.1</td>
<td>DataCollector architecture</td>
<td>29</td>
</tr>
<tr>
<td>5.1</td>
<td>Daily unique data for 30 day data set</td>
<td>32</td>
</tr>
<tr>
<td>5.2</td>
<td>Cumulative unique data for 30 day data set</td>
<td>33</td>
</tr>
<tr>
<td>5.3</td>
<td>Sources, Destinations, IPs and Edges for 5 days data</td>
<td>37</td>
</tr>
<tr>
<td>5.4</td>
<td>Traces for 5 days data</td>
<td>39</td>
</tr>
<tr>
<td>5.5</td>
<td>Destinations per Vantage Point for 5 days data</td>
<td>41</td>
</tr>
<tr>
<td>5.6</td>
<td>Traces per Vantage Point for 5 days data</td>
<td>41</td>
</tr>
<tr>
<td>5.7</td>
<td>IPs per Vantage Point for 5 days data</td>
<td>42</td>
</tr>
<tr>
<td>5.8</td>
<td>Edges per Vantage Point for 5 days data</td>
<td>42</td>
</tr>
<tr>
<td>5.9</td>
<td>Sources per Destination (log-log scale)</td>
<td>44</td>
</tr>
<tr>
<td>5.10</td>
<td>Traces per Destination (log-log scale)</td>
<td>45</td>
</tr>
<tr>
<td>5.11</td>
<td>IPs per Destination (log-log scale)</td>
<td>46</td>
</tr>
<tr>
<td>5.12</td>
<td>Edges per Destination (log-log scale)</td>
<td>47</td>
</tr>
<tr>
<td>5.13</td>
<td>AS coverage of Sources</td>
<td>49</td>
</tr>
<tr>
<td>5.14</td>
<td>AS coverage of Destinations</td>
<td>49</td>
</tr>
<tr>
<td>5.15</td>
<td>AS coverage of IPs</td>
<td>50</td>
</tr>
<tr>
<td>5.16</td>
<td>AS rank of Sources</td>
<td>51</td>
</tr>
<tr>
<td>5.17</td>
<td>AS rank of Destinations</td>
<td>51</td>
</tr>
<tr>
<td>5.18</td>
<td>AS rank of IPs</td>
<td>52</td>
</tr>
<tr>
<td>5.19</td>
<td>AS rank of Edges</td>
<td>53</td>
</tr>
<tr>
<td>5.20</td>
<td>Loops identified in trace data</td>
<td>54</td>
</tr>
<tr>
<td>5.21</td>
<td>Trace IP addresses not seen in Ant Census (graph)</td>
<td>56</td>
</tr>
<tr>
<td>5.22</td>
<td>MOAS prefixes (log scale)</td>
<td>58</td>
</tr>
<tr>
<td>5.23</td>
<td>BGP announcements (log scale)</td>
<td>59</td>
</tr>
<tr>
<td>5.24</td>
<td>Prefix announcements vs Mask distribution (log scale)</td>
<td>60</td>
</tr>
<tr>
<td>6.1</td>
<td>Board details of the Odroid C-2. [6]</td>
<td>62</td>
</tr>
<tr>
<td>6.2</td>
<td>Block diagram of the Odroid C-2. [7]</td>
<td>63</td>
</tr>
<tr>
<td>6.3</td>
<td>Board details of the Odroid C-2. [6]</td>
<td>66</td>
</tr>
<tr>
<td>6.4</td>
<td>Block diagram of the Odroid C-2. [7]</td>
<td>66</td>
</tr>
<tr>
<td>6.5</td>
<td>Odroid connection</td>
<td>67</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

In the words of Eric Schmidt (Alphabet, Inc.), “The Internet is the first thing that humanity has built that humanity doesn’t understand; the largest experiment in anarchy that we have ever conducted.” The Internet as we know it today is an evolution of the original concept first published in a series of memos by J.C.R. Licklider of MIT in 1962, in which he discussed a “Galactic Network” [18]. He envisioned a globally interconnected set of computers through which anyone on the network could quickly access programs and data from any site. This idea evolved into a project at the Defense Advanced Research Projects Agency (DARPA), with Licklider as its first director. In 1962 the first paper on packet switching theory entitled “Information flow in large communication nets” by L. Kleinrock was published, and in 1964 it appeared as a book. This led to a shift away from circuit switching, which was the basis for the original telephone system, to the packet switching network that would become the ARPANET, and eventually the Internet we know. Initial nodes on the ARPANET were primarily government, research, and educational institutions, and as such the original design evolved without the need for much security or monitoring capability since there was a community of mutually trusting participants. Little provision was made for measuring or maintaining the network, as its founders were unaware of the eventual magnitude and importance of what they were creating.

The public Internet has grown continuously over the past 25 years with service providers ranging from networking hardware manufacturers to application developers. While there is a general spirit of cooperation that allows this network of networks to exist, ultimately each Autonomous System (AS) works independently to
optimize its own performance. In addition, business relationships between ASes change continually as each player works to maximize its own performance and profits, resulting in a dynamic topology. To further complicate matters, the Internet provides abundant opportunities for crime, terrorism, and espionage, making vulnerable much of our personal, financial, and business information. Understanding the dynamics of Internet topology is essential to assure its continued expansion and safe, efficient operation.

The importance of understanding the underlying structure of the Internet has been recognized and the research community has developed numerous approaches to measure the Internet topology. Platforms have been built for the collection of Internet topology data, and several groups have developed long-lasting measurement projects to continuously probe the Internet, and to act as repositories for topology data sets which are made available to network practitioners and researchers [28]. Current and past measurement studies span from active to passive probing approaches [20], [27], [28]. With the introduction of power law and rich club phenomena applied to Internet topology [37], research directions in the measurement community have expanded to include network characteristics analysis for both router-level and AS-level topologies [35].

The Internet Measurement Project seeks to contribute to this body of knowledge by utilizing existing measurements, as well as developing techniques for efficiently collecting new measurements, to gain a deeper understanding of the Internet backbone, and to create a map that can be made available to the community. In this thesis, we focus on the collection and classification of data from public Internet topology measurement platforms and compare the characteristics of their datasets. In particular, we focus on AS level data from UCSD Cooperative Association for
Internet Data Analysis (CAIDA) [15], UCLA Internet Research Lab (IRL) [13], and CIDR [8]; and router level data from USC/ISI Ant census [1], UCSD CAIDA Archipelago (Ark) [15], UW Information Plane (iPlane) [34], Measurement Lab (M-Lab) [29], and RIPE NCC Atlas [10], [12].

In addition to collecting data sets produced by other organizations, the Internet Measurement Project will generate its own topology data. To accomplish this, we have developed a software tool that utilizes servers (nodes) provided by the PlanetLab network to probe a large set of IP addresses collected from other measurement platforms. Recently, we have observed the decline of the PlanetLab system [20] from over 1100 nodes at its peak in 2008, to as few as 30 functioning nodes in February of 2017. A well-distributed network of vantage points is integral to our continued work. Hence, this thesis will also discuss the development and deployment of an Internet Measurement platform (IM). A network of Odroid-based single board computers that will provide a platform from which to collect future Internet measurements. We consider the architecture of this network, its connection to our measurement system, and a plan for distributing probes strategically to optimize our view of the Internet.

In addition to providing an analysis of measurements, the data we collect has been made public for use by others conducting research on Internet Topology at:

http://im.cse.unr.edu/data
In this chapter we will discuss some of the preliminary concepts on Internet measurement. We will cover details regarding Autonomous Systems (AS), the individual networks that make up the larger Internet. We will summarize the BGP protocol that is used to facilitate communications between networks and RIRs that administer interconnection. We will then discuss the network analysis tools ping and traceroute that can be used to discover the Internet topology and how they work. We will define the components we use to reference the structure of the Internet. Finally, we will discuss some of the public measurement platforms that are in existence, and will examine their methods for collecting measurement data.

2.1 Preliminaries

In this section, we present some key protocols, technologies, and organizations that serve to connect different networks to the Internet, and to govern its organization and operation.

2.1.1 Autonomous System (AS)

An Autonomous Systems (AS) is either a single network or a group of interconnected networks that are controlled by an independent network administrator. Each AS has its own routing policy which may differ from those of other ASes. An AS is categorized by one of two roles on the Internet; as a transit AS or a stub
AS. A *transit* AS is one which is connected to multiple other ASes and allows data traffic to pass through itself providing connectivity to the rest of the Internet. A *stub* AS is one which is connected only to one or more *transit* ASes. A *stub* AS said to be be multi-homed if it possesses multiple provider ASes as *transits* but does not allow any *transit* data traffic to pass through itself. In order to provide connectivity across the Internet, each AS needs to be provided with some information in order to know how to reach the rest of the network.

An Autonomous System Number (ASN) is a unique identifier that is assigned to an AS by the Regional Internet Registry (RIR). With an AS number, a single network or a group of networks administered by a single authority can be represented as an independent entity. When a router announces a new IP prefix to other ASes, it uses the ASN to identify itself. ASes receiving this announcement will in turn append their ASN to the AS path, and will propagate this information to its neighbors.

### 2.1.2 Regional Internet Registry (RIR)

Regional Internet Registries (RIR) are organizations that allocate IPs and ASes in distinct geographical areas world wide. There are five RIRs:

1. ARIN, which is responsible for the United States, Antarctica, and some parts of the Caribbean, North Atlantic Islands, and Canada
2. AfriNIC, which managing IPs and ASes for Africa
3. APNIC, which is responsible for the Asia Pacific region
4. LACNIC, which is responsible for some parts of the Caribbean and Latin America
5. RIPE NCC, which is responsible for Europe, the Middle East, and central Asia

Based on information from these organizations, we can determine what range of IP addresses belong to which AS.

2.1.3 Border Gateway Protocol (BGP)

While IP addresses are used to provide for the distinct identification of devices across the Internet, BGP is essential for communication in the network as a whole, as it is the only protocol that provides reachability and path information across the whole Internet.

Border Gateway Protocol (BGP) is the protocol used for implementing the interconnection between ASes on the Internet by providing mechanisms for the exchange of routing information (AS paths) between ASes [4]. BGP also allows ASes to enforce peering agreements with neighboring ASes, determining the proper flow of traffic between them, and thereby implementing the aggregate structure of the Internet.

*Internal BGP* (iBGP) protocol exchanges routing information within an AS, while *external BGP* (eBGP) protocols provide for the exchanges of routing information between neighboring ASes. The routers that exist at the margin between ASes must run both eBGP and iBGP, and are called the *border routers*. Each border-router has a direct link to another border-router in a neighboring AS. When new reachability information is discovered within an AS, or when advertisements for new paths are received from neighboring ASes, this information is appended to a table within the
router (BGP tapble), and is propagated to neighboring ASes. In this way, paths for access to all parts of the Internet are shared across the network.

2.2 Internet Topology Measurement

Internet topology measurements are generated thorough either passive or active means. Passive measurements are performed by measuring a network without generating any additional traffic. Passive measurement techniques avoid the possibility of altering traffic data by not generating additional packets on the measured network. For example, the Routeviews Project [16] collects data from BGP routing tables, which define how ASes are interconnected. Inferences can be made about the characteristics and topology of a network via passive measurements [38]. Other methods of passive measurement include tools that utilize packet sniffing techniques to capture network data.

Active measurement methods are used to gather information by injecting probing traffic into the target network. Using probe packets it is possible to obtain information about a network such as the number of nodes and edges it contains, the hop distance between nodes, and how nodes are inter-connected. It is also possible to find characteristics of probed devices such as host OS type, link bandwidth, and their connectivity.

The Two most widely used active measurement tools for Internet measurement are Ping and Traceroute.
2.2.1 Ping

Ping is useful for determining if a target node is reachable by utilizing the Internet Control Message Protocol (ICMP). It sends ICMP ECHO request packets to a destination IP address and waits for an ICMP ECHO reply message from the target. If the probed IP responds, then it measures the Round Trip Time (RTT) of each packet. RTT is a measure of the elapsed time between when a data packet is sent to a destination IP, and when the reply message from the target IP is received. RTT is useful for inferring the physical distance between Internet devices.

2.2.2 Traceroute

As we collect traceroute data we can identify these paths, and can infer the topology of the Internet. By resolving the subnet that a particular IP address belongs to, we can determine to which AS an IP address belongs, and can create a graph of these connections. For this study, we will be interested in determining how well each network measurement platform is able to cover the Internet by determining how many ASes it is able to see from its set of vantage points.

Traceroute is a tool that provides information about the route a packet follows as it makes its way through the Internet. Data packets of a traceroute probe can be crafted to utilize either ICMP, User Datagram Protocol (UDP), Transmission Control Protocol (TCP). Each packet has a field in its header to store the packet’s time to live (TTL). When a TCP packet is sent, its TTL is set, determining the maximum number of routers (hops) the packet will be allowed to traverse before it is discarded. As the packet passes through a router the TTL is decremented until it
reaches zero, at which time the packet is discarded and an ICMP time exceeded message is returned to the source address. The return message’s TTL is set by the terminating router when it creates the packet, and is decremented normally.

Trace Route works by setting the TTL for a packet to 1, sending it towards the requested destination host, and listening for the reply. When the initiating machine receives a “time exceeded” response, it examines the packet to determine where the packet came from, which identifies the router one hop away. Then the source generates a new packet with TTL 2, and uses the response to determine the router 2 hops away, and so on. Eventually the destination replies to ICMP, UDP, or TCP probes.

Figure 2.1: Traceroute tool implementation
Figure 2.1 presents the probing mechanism of the Traceroute tool. In this figure the source host initiates a traces toward destination $D$ with a TTL of 1. Once the packet reaches the first router (hop), its TTL is decremented to 0, which triggers an ICMP Time Exceeded message back to the host. This message reveals the IP address of router $A$. Then, source host sends probes toward destination IP $D$ with a TTL of 2. These packets are discarded at router $B$ as the TTL is decremented to 0, and a warning is sent to the host notifying it of the loss. Following this pattern, all routers on a path between the source host and destination $D$ are discovered.

### 2.3 Network Measurement Platforms

To make meaningful inferences about the topology of the Internet, one must be able to observe the network from a sufficient and well-distributed set of vantage points. A vantage point is essentially an IP on the Internet that can be utilized as a point of origin for performing traceroutes. We consider the number of vantage points in relation to network coverage to compare topology data sets.

#### 2.3.1 UCSD Caida Archipelago Project (Ark)

The UCSD Cooperative Association for Internet Data Analysis (CAIDA) introduced their Archipelago measurement infrastructure (Ark) in September 2007 to replace the Skitter platform which had been running since 1998 [33]. The goal of the Ark measurement infrastructure is to support large-scale measurements, allow collaborators to run vetted measurement tasks on a security-hardened distributed platform, and to collect and distribute datasets to support various research, engi-
neering, and operational interests.

Figure 2.2: Probes used by the Archipelago (Ark) project. [2]

As of August 2016 there were 162 monitors hosted in 40 countries. 41 billion traces (18 TB) had been accumulated, and grow by 766 million traces (approximately 316 GB) per month [15]. Originally utilizing a server-based probe, these were eventually replaced with a more lightweight Raspberry Pi styled unit that could be more easily hosted by willing participants. An example of an Ark node is shown in 2.2.

### 2.3.2 UW Information Plane (iPlane)

iPlane is a measurement platform implemented by the University of Washington [20]. The goal of this project is to provide a scalable service capable of generating accurate predictions of Internet path performance for emerging overlay services [34]. iPlane utilizes the PlanteLab infrastructure to probe the available IP
address space, along with various public Looking Glass/Traceroute servers for low-intensity probing. In addition to the raw traceroute data, iPlane provides lists of alias clusters, IP to PoP mapping, origin AS mapping, IP to AS mapping, access link bandwidths, inter-PoP links, inter-IP links and loss rates of inter-PoP segments. Unfortunately, as of August 2016 the iPlane infrastructure went offline and will likely not be restored. Archived historical data from this platform is accessible through Ripe NNC.

![Figure 2.3: Probes used by Ripe NNC. [3]](image)

### 2.3.3 Measurement Lab (M-Lab)

Measurement Lab (M-Lab) is an open, distributed server platform designed to allow researchers to deploy their active Internet measurement tools [9]. M-Lab was launched in 2009 by founding partners Google Inc., the PlanetLab Consortium, and New Americas Open Technology Group (OTI) with six measurement tools available, mainly for performance testing. Since that time several other organizations have partnered with M-Lab, and a host of tools for measuring performance, censorship, and topology have been added. Data is openly shared and is accessible
through Google Cloud Storage. M-Lab is run in conjunction with PlanetLab [20] with a few key differences in administration and software [22]. As of August 2016 there were 176 active M-Lab nodes available for use.

### 2.3.4 Ripe Atlas (Ripe)

Ripe Atlas is a more recent measurement platform deployed in January 2013 by the RIPE Network Coordination Centre (RIPE NCC). The RIPE measurement platform consists of a large number of probes (shown in figure 2.3) distributed across the globe, and a smaller number of servers known as anchors (shown in figure 2.4). As of August 2016, there were 13,554 probes [12] and 208 anchors [10] hosted by a variety of users and organizations. Probes are capable of ping, traceroute, DNS, SSL, NTP and HTTP measurements, and are used by RIPE NCC to conduct regularly scheduled base-line measurements to anchors for the purpose of continually measuring regional reachability [10]. Probes can also be programmed through the RIPE API to conduct user-defined measurements to selected IP addresses. Anchors are dedicated servers hosted by larger corporate or academic organizations [10]. Anchors can be used both as a source and a destination for measurement traces, and are capable of conducting a large number of measurements in parallel.

### 2.3.5 USC/ISI Ant Census

Ant Census Researchers at the University of Southern California (USC) Information Sciences Institute (ISI) began an effort in 2003 known as the Ant Census to collect data about the Internet address space. This is accomplished by conducting
a census of every IP address in the visible Internet [1]. Ant estimates that 3.6% of the allocated address space is occupied by visible hosts, with approximately 34 million addresses reported as stable and visible to their probes (16% of responsive addresses). From this, they estimate there are 60 million stable Internet-accessible computers on the network [30].

2.3.6 PlanetLab

PlanetLab is a global platform for deploying and evaluating network services [21]. It was launched in 2002 with 100 machines distributed at 40 independent sites, and eventually grew to include approximately 1100 nodes spanning 336 sites and 35 countries. It has hosted thousands of researchers affiliated with various projects. It has been used to evaluate a diverse set of planetary-scale network services, including content distribution, anycast, DHTs, robust DNS, large-file distribution, measurement and analysis, anomaly and fault diagnosis, and event notification [21] [24]. It has supported the design and evaluation of dozens of long-running
services that at one time transported an aggregate of 3-4TB of data daily, satisfying tens of millions of requests involving roughly one million unique clients and servers. Users were assigned a limited amount of access across the entire network of nodes called a "slice", as seen in Figure 2.5, allowing them to run their experiments. Unfortunately, in recent years PlanetLab has fallen into disrepair, and has degraded to the point where in February 2017 there were at times less than 30 active nodes available for use.

2.3.7 UCLA IRL

Internet Research Lab (IRL) at the University of California Los Angeles hosts an archive of historical Internet AS-level topology data for academic research. Historical BGP data collected by Route Views, RIPE RIS, PCH and Internet2 through 2015 are available on a daily and monthly basis [13]. This service is not currently collecting new data.
2.3.8 CIDR

CIDR is an online resource for network data such as BGP updates, IPv4 and IPv6 address space reports, AS number space reports, and a variety of useful information for Internet measurement [2].

2.3.9 UCSD Caida BGP Stream

UCSD Caida makes archived BGP data made available by the RouteViews project [16], which is collected from a group of 12 BGP routers. In addition, Caida provides a service they call BGPStream, an open-source software framework for the analysis of both historical and real-time BGP measurement data. BGPStream enables efficient real time collection of BGP data from a large group of routers through the use of an API designed to work with Python.
CHAPTER 3

RELATED WORK

Over the past two decades there has been much research on the topic of Internet topology. As the structure of the Internet is invisible, numerous approaches have been developed to discover and map its features, and measure its performance. In this chapter, we will give an overview of some works that are relevant to our study, and will discuss their similarity and differences to our research.

3.1 Internet Topology Measurement Issues

In this section we present a survey of related works outlining some of the issues associated with Internet topology measurement and solutions previously implemented by other researchers in this field.

3.1.1 Survey of Internet Topology Discovery Methods

Benoit Donnet et al conduct a survey of Internet Topology Discovery in 2007 [27]. In their work they seek to represent the Internet as a complex network, relating IP interfaces, routers, PoPs, or ASes as vertices depending on the level considered. From this perspective, one can study characteristics of the network in terms of average degree, degree distribution, clustering coefficient, or betweenness centrality to infer the nature of Internet topology. In this way they seek to discover the Internet’s theoretical resilience to failure, and to assess the efficiency of current routing patterns. They also consider the possibility of using collected data to create a vi-
visualization of the Internet for the purpose of simulation. This work is similar in some regards to ours, with the main difference being that it addresses the ideas of topology discovery, but collects no actual data to make inference about the existing topology.

3.1.2 IP Alias Resolution

In a later publication from CAIDA (2012), Bradley Huffaker et al discuss possible errors in network topology inference due to over-estimation of the number of routers on the Internet by failing to perform IP address alias resolution. Alias resolution is a method by which the various interfaces that can exist on a single router can be identified as being the same device, rather than mistakenly assuming each address is from a separate device [31]. By constructing a realistic view of the physical network, valuable metrics such as average node degree and degree distribution can be obtained and used to infer a more realistic topology.

3.1.3 Survey of Network Performance Measurements

Vaibhav Bajpai and Jurgen Schonwalder conduct a survey of Internet measurement platforms with a focus on performance measurements rather than topology discovery [22]. Metrics of this kind are motivated by a need to provide network operational support to fixed-line and mobile access networks [14]. While their survey also views some of the platforms we study (iPlane, Ripe Atlas), they do so from the perspective of performance measurement tools. They also survey a large number of other platforms with the same focus, and provide an overview of available
network performance resources.

3.2 Internet Topology Measurement Platforms

In this section we discuss related works covering the major public Internet measurement platforms. We discuss different approaches and their respective architectures and approaches in generating Internet measurements.

3.2.1 mrinfo-rec

Since the creation of the Internet a number of methods have been explored for discovering its hidden topology. Between 2004 and 2008 Pietro Marchetta et al developed a tool called mrinfo-rec [19]. This was a system by which an initial address would specified (the ‘seed’), and from this starting point mrinfo-rec would be invoked recursively to discover all neighbors of this address. The goal of this process was to find the largest multicast component reachable from a single starting point. This process was run on a daily basis from a single vantage point in Strasbourg, France, and was able to discover an average of 10,000 routers. Because of the limited number of vantage points available for the research on the Internet at the time, this was a significant effort in topology discovery. While interesting, both the Internet and methods/platforms for probing have advanced significantly since this paper was published.
3.2.2 Cheleby

Hakan Kardes and Mehmet Gunes submit a tool they call Cheleby to probe the Internet topology [33]. Using subnet data obtained from CIDR and the Planet-Lab system for vantage points, they use Paris traceroute to probe by dividing the address space into 1024 destinations. These blocks are then probed by 7 teams formed from functioning PlanetLab nodes which are divided based on their geographic location. From this data statistics are collected on unique nodes and edges discovered, and can be used to infer network topology. The data we utilize from the group of platforms outlined in our study have collected their data in a similar fashion.

3.2.3 Archipelago

In their 2009 publication “Internet Mapping: from Art to Science”, Kimberly Claffy et al describe their Archipelago (Ark) project, one of the measurement platforms addressed in our study. They seek to improve on CAIDA’s previous measurement effort, Skitter. They have several goals in mind as they make improvements. These include a development environment that provides for rapid prototyping to enable the implementation of new and risky measurement ideas without incurring high costs [28]. In this way they hope to encourage innovation leading to better and more useful measurements. They also hope to implement a higher degree of dynamism in measurement methods. Rather than probing a pre-configured set of measurements to a static set of targets, they seek to create a dynamic coordination between vantage points and target destinations to probe subnet prefixes in a binary search pattern to gain better results. They also seek to implement measurement
services in the form of archived data that can be accessed by others performing measurement research. Our study owes much to this effort.

3.2.4 Ant Census

John Heidemann et al of the USC Department of Computer Science describe their implementation of Ant, a tool to perform a census and survey of the visible network on a continual basis. Approximately every two months since 2006 the Ant census has made available to the research community a list of IP addresses that respond to ICMP echo requests, providing a list of viable target destinations for probing efforts, and to give a snapshot of how efficiently the address space is being utilized [30]. By using a multi-protocol probing method, their study discovers that previous estimates of the active address space were largely under-estimated. They are also able to probe the address space with far fewer complaints about scanning attacks than previous efforts. We consider the Ant census in our analysis of the IP address space.

3.2.5 Measurement Lab

Constantine Dovrolis et al extend an invitation to the measurement research community in their 2010 ACM SIGCOMM publication [29]. They describe the lack of well-provisioned, well-connected measurement servers in geographically distributed areas. Rather than every research effort needing to provide their own data, M-Lab seeks to collect useful measurement data of various kinds, and make that data available to the community through the use of the Google Cloud plat-
form. They run a variety of tools for the generation of topology and performance data, and allow for limited access to their system for the research community to conduct their own experiments.

3.2.6 PlanetLab

In 2003 Brent Chun et al proposed a global overlay network for developing and accessing broad-coverage network services [20]. While their system would eventually grow even larger than they hoped, at the time their goal was to grow to 1000 geographically distributed nodes that would have the capability to interconnect over a diverse collection of links. By allowing researchers to gain access to a ‘slice’ on PlanetLab (access to a small percentage of the total bandwidth available across all nodes) that would act as a virtual machine, giving them the ability to conduct experiments and measurements across a global network. PlanetLab nodes would be hosted by academic and research-oriented organizations, and would be activated outside of the organizations protective firewalls, allowing researchers to conduct experiments free from these limitations.

3.2.7 Information Plane

Finally, Harsha V. Madhyastha et al describe their efforts in the 2006 publication “iPlane: An Information Plane for Distributed Systems” [34]. Their goal was to provide a coarse-grained map of the Internet that would be sufficient to provide a utility that could improve overlay performance. Like M-Lab, iPlane utilized the PlanetLab network [20] to provide vantage points from which to generate tracer-
oute data, and have archived this data for consumption by the research community. Our study considers the performance of this platform and characteristics of its data set.
In this chapter we will discuss the methodology employed to collect and store data from the public measurement platforms. We will discuss the various formats in which the data is found, and will describe our accumulated data set. We will detail some of the challenges encountered in the process of data collection and organization, and will describe the tool we developed to perform the task on a periodic basis.

4.1 Storage Formats

Our efforts to create a map of the Internet backbone will rely on both data we generate ourselves and data collected by others. This said, one of the initial challenges for our project was to collect existing data from the public repositories that might also be helpful to our project, and create tools that could collect this data on a recurring basis. We developed the tool *DataCollector* that will perform this task, and will store the results in a manner suitable for our data analysis system. Trace data is gathered from UCSD Caida Archipelago (Ark), iPlane, Measurement Lab (M-Lab), Ripe NNC, and USC/ISI Ant Census on a periodic basis as it is needed. This task presented significant challenges, as the combined data from these sources is quite large, and is stored in various formats and file systems. Once collected, all data is stored on our server in a consistent manner.

The total size of a 30 day sample from Ark, iPlane, M-Lab, and Ripe is just under 1 TB uncompressed, and initial efforts to collect data were taking up to 45 days to
complete. As this was impractical, much effort was made to modify algorithms contained in the tool to optimize this process. At present a 30-day data set can be obtained in about 7 days, a significant improvement, and a one year data set in about 3 weeks. In the following some detail about collection of the specific data sets will be provided. We will address the format and file systems each platform utilizes, along with the basic methodology for collection and the particular challenges that were encountered in the process.

4.1.1 UCSD Caida Ark

The particular data set Ark that we are interested in is the topology data, IPv4 routed /24 (restricted). These traces are the most current, and require a user account and password to access. Caida probes their list of IP addresses in cycles, each cycle taking approximately 5 to 6 days to complete. Probes are divided into 3 teams, and traces to the address space are rotated across these teams to avoid the resemblance of port or network scanning on any particular network. Within each team, traces can be seen on a daily or hourly basis. Under daily traces there is a large archive of stored trace data from 2007 to present by year. For each year traces are stored by cycle, which correspond roughly (but not exactly) with the days of the month, since a cycle for a particular team can take slightly more than a day to complete. Under cycles, a list of roughly 60 probe names contain all traces from that probe, for a given cycle, for a given year, for a given team.

These can be downloaded in Gzip format. When unzipped, they are .warts files, which are binary files of the traces. At this point, a tool from Caida called Scamper must be utilized to convert the binary files into text files and stored in a temporary
file system to be parsed. Once Scamper is installed, this can be done from the Linux command line, and the results streamed to a file, with the following command:

\[ \text{warts2text} > \text{my\_temp\_file.txt} \]

When converted to text, a running list of traces containing the trace and various other bits of meta data appear, and can be parsed for the required information. We capture source IP, destination IP, hop numbers, and round trip times (RTT). We then reassemble this data into our preferred format.

### 4.1.2 UW Information Plane

After running for many years, the iPlane platform ceased creating new measurements as of August, 2016. Historical data is still available through the Ripe NNC mirror at:

https://labs.ripe.net/datarepository/data-sets/iplane-traceroute-dataset

As with the Ark data, Ripe requires a user account and password to access archived data. Archives are stored by year, and then by day for each day of the year. Files by day can be downloaded in Gzip format, and when unzipped are found to be in a binary format which requires a program written in the C language to convert them to text. The zipped files are quite large, 6-8 GB compressed, and create very large text files. These files are subdivided into files for each node, which are nothing more than a long stream of consecutive traces. The node ID must be converted to an IP address and placed at the head of the trace for completeness. Once this is done, traces can be parsed for desired information, and reassembled into the preferred format, similar to the Ark traces.
4.1.3 M-Lab

M-Lab utilizes the Google Cloud system for storage of paris-traceroute data, and stores data in text format. The XML structured file system used makes it relatively easy to find a particular trace for a particular day, but because of its layered nature, these are the most difficult to extract automatically. To collect all data for a given day, a recursive approach is required to get to the lowest layer of the file system and then work outward. Once the files are obtained, parsing is relatively easy as each trace is stored in a separate file, allowing for easy parsing of the desired information.

4.1.4 Ripe

Ripe NNC has about 220 anchors and roughly 10,000 probes. While Ripe performs a variety of measurements, their focus is on reachability which they provide as a service to network administrators seeking to monitor their network’s reachability. Ripe stores trace data which is performed automatically on a daily basis, but there is much redundancy in the data as many of the traces are performed in a repetitive manner. Probes send traces back toward the same 220 anchors repeatedly to check reachability. In addition, any user-defined measurements that are performed are also stored at the anchor and are part of the same data set.

Each Ripe anchor stores all of its historical data, which can be downloaded in .json format. To access data, probe ID, start time and end time must be entered in Unix epoch format. Because the files are very large, parsing can be costly in terms of memory usage and time. When first downloading these files, we attempted to
parse them by streaming the `.json`, but this method was very slow as each pass through the `.json` had to search from beginning to end of the file (some files being as large as 50 GB) to find matching symbols in the file. Finally, we downloaded the entire file and converted it from `.json` to `.csv` format. The file could now be read as a Python dictionary, seeking the desired fields from the file by key and value. This method reduced the time taken for reading 30 days data to about 3 days.

### 4.1.5 USC/ISI Ant Census

The Ant Census releases a new data set containing all responsive IP addresses from a scan about every 2.5 months, and makes the data available to the research community. This file contains roughly 400 million IP addresses that are known to respond to the ping requests. We compare this data set with IP addresses parsed from all traceroutes for comparison, and later will use these as targets when collecting trace data using our own system.

### 4.2 DataCollector

DataCollector driver is written in the C language. It prompts for a start date and number of days data to collect, which is then converted into Unix Epoch format. There are four subroutines written in Python 3.5 that download and parse files from each of the four platforms discussed above, intermittently calling both C and C++ routines to parse certain sections of the data for efficiency. A basic diagram of the architecture is seen in Figure 4.1. DataCollector stores a complete set, unique values only set, and counts of unique values in each complete set of traces, IP ad-
addresses, and edge-pairs for each of the platforms. Additional scripts combine these sets in various ways to give combines view of the data across platforms.

4.3 Output Format

The following is an example of a trace stored in our system:

```
91.184.204.124:171.36.195.140 91.184.204.126,1,0.344 <etc>
```
The first item in the string is delimited by a colon, and represents the source:destination pair, i.e. $<source>:<destination>$. This is followed by each router (interface) found in the trace, in the format: $<IP\ address><hop\ number><RTT>$. This format is consistent with the format we use when collecting and processing our own combined data. All data is then stored in individual files per data source in exactly the same way, using the same filenames within each directory to make the information easy to access later.
This thesis is a part of a much larger effort to create a map of the Internet backbone. The primary objective of this work was to collect existing data from the public Internet measurement repositories and store it in a format consistent with that of our own system. We also perform analysis on this data allowing us to make comparisons in the characteristics of these repositories, and to learn why some might be more effective in their efforts than others from different perspectives. The following is a report on the analysis of data collected between June 1, 2016 and June 30, 2016.

In this chapter, we analyze the probe data sets (i.e., ping and traceroute) from USC/ISI Ant census [1], UCSD CAIDA Archipelago (Ark) [15], UW Information Plane (iPlane) [20], Measurement Lab (M-Lab) [9], and RIPE NCC Atlas [10]. This data gives us a picture of the IP level topology, as some routers have multiple interfaces. We do not perform IP alias resolution in this study.

5.1 30-day Router Level Topology Data

We first analyze a 30-day set of data from these platforms and then perform in depth analysis on a single cycle of data. As Ark cycles recurred approximately every 5 days, we the choose the largest cycle from the 30-day set and compare it to data from the other sources for the same time period.
Figure 5.1: Daily unique data for 30 day data set
Figure 5.2: Cumulative unique data for 30 day data set
5.1.1 Total IP, Trace, Edge, Source, and Destination Statistics

We find total and unique IPs, source IP addresses (vantage points), destination IP addresses (targets), edges discovered, total traceroutes performed. We also identify loops, anomalies, and repeated addresses at the end of traceroutes.

Figure 5.1 and Figure 5.2 display the daily unique and cumulative, respectively, number of traces, source IP’s (vantage points), destination IP’s (targets) and discovered IPs and edges for each of the four data sources over the 30-day period (results are shown in log scale).

The Daily Cumulative Source IP Addresses figure shows that the number of vantage points on the first day is 80%, 55%, 88% and 98% of all vantage points used over the month for Ark, iPlane, Ripe, and M-lab, respectively. The Daily Cumulative Destination IP Address figure shows that the number of destinations do not change for iPlane and Ripe but the first day of Ark and Mlab is 6% of the combined destinations, for both. The number of destinations do not vary in iPlane and Ripe both seeing 100% of their total destinations visited on the first day, respectively. iPlane and Ripe both observe 89% and 88% of the overall IP addresses seen in the first day, while Ark and Mlab both only observe 6% of month-long IP addresses in the first day, respectively. Similarly, iPlane and Ripe observe 49% and 42% of overall edges in first day, whereas Ark and Mlab only observe 12% and 9% of month-long edges on the first day, respectively. Finally, we observe all methods have new traces over the time (i.e., there is a continually growing number of unique edges in the path trace over the month). Overall, Ark, iPlane, Ripe, and Mlab observe only 6%, 9%, 13%, and 3% of combined traces in the first day, respectively.

Ripe and iPlane each probe a roughly constant number of destinations each day,
while Ark and M-Lab progressively probe a list of destinations over the month. We note that Ripe performs measurements primarily between its own probes and anchors on a repetitive basis. This is because Ripe is focused on reachability, a service that it provides to its users to test their visibility to the rest of the Internet. iPlane, on the other hand, utilizes the PlanetLab system nodes, and does repeated measurements from these nodes. It is interesting to note that, while on one hand Ripe uses a small number of destinations and a large number of sources, iPlane does quite the opposite, using a small number of sources and a large number of destinations; yet they both see roughly the same number of traces over the course of the month. One explanation might be that both systems are very well distributed across the entire network. iPlane discovers more edges each month, probably because of the larger number of destinations it is measuring to, visiting a more diverse topology.

5.2 5-day Router Level Topology Data

We collect traceroute data from the four data sources (i.e., Ark, iPlane, M-Lab, and Ripe). We then perform analysis on this data to better understand the coverage of each platform, and to compare the IP addresses we found with the Ant Census IP data set. We first collected 30 days of data (June 1, 2016 to June 30, 2016) from each source, which totaled just under 1TB. We selected the first cycle of Ark trace data (June 1, 2016 to June 5, 2016) and gathered the data for the same time frame from the other sources. This smaller data set was comparatively consistent with the larger data set, as shown below.
5.2.1 Total IP, Trace, Edge, Source, and Destination Statistics

Table 5.1 and Figure 5.3 outline the number of sources, destinations, IPs and Edges for the June 1-5, 2016 data set (corresponding to one Ark data cycle). The figures show individual daily and combined data for all days for each of the measurement platforms. Additionally, we show unique data, i.e., data that were not observed in the other platforms, for each category.

Table 5.1: Coverage for 5 days data

<table>
<thead>
<tr>
<th></th>
<th>RIPE</th>
<th>M-Lab</th>
<th>iPlane</th>
<th>Ark</th>
</tr>
</thead>
<tbody>
<tr>
<td>Src</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>6,146</td>
<td>176</td>
<td>111</td>
<td>112</td>
</tr>
<tr>
<td>Unique</td>
<td>6,118</td>
<td>176</td>
<td>84</td>
<td>111</td>
</tr>
<tr>
<td>Dest</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>193</td>
<td>161,657</td>
<td>217,733</td>
<td>52,790,278</td>
</tr>
<tr>
<td>Unique</td>
<td>192</td>
<td>158,636</td>
<td>213,550</td>
<td>52,783,156</td>
</tr>
<tr>
<td>Trace</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>883,796</td>
<td>479,310</td>
<td>45,793,268</td>
<td>54,297,491</td>
</tr>
<tr>
<td>Unique</td>
<td>883,796</td>
<td>479,309</td>
<td>45,793,268</td>
<td>54,297,490</td>
</tr>
<tr>
<td>w/Repeat</td>
<td>169,018</td>
<td>236,965</td>
<td>654,454</td>
<td>43,932,702</td>
</tr>
<tr>
<td>w/Loop</td>
<td>777,476</td>
<td>201,262</td>
<td>16,844,293</td>
<td>32,152,569</td>
</tr>
<tr>
<td>IP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>6,154</td>
<td>161,833</td>
<td>563,313</td>
<td>54,359,540</td>
</tr>
<tr>
<td>Unique-Others</td>
<td>5,096</td>
<td>158,004</td>
<td>248,641</td>
<td>54,040,614</td>
</tr>
<tr>
<td>Unique-Ant</td>
<td>4,488</td>
<td>114,257</td>
<td>195,789</td>
<td>46,328,571</td>
</tr>
<tr>
<td>Edge</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>280,070</td>
<td>509,721</td>
<td>2,914,809</td>
<td>10,019,704</td>
</tr>
<tr>
<td>Unique Others</td>
<td>170,576</td>
<td>219,619</td>
<td>1,745,528</td>
<td>8,742,947</td>
</tr>
<tr>
<td>Bridge</td>
<td>61,876</td>
<td>80,136</td>
<td>468,472</td>
<td>683,681</td>
</tr>
</tbody>
</table>

We observe that Ark, iPlane, Mlab, and Ripe have 7122, 4183, 3021, and 1 desti-
Figure 5.3: Sources, Destinations, IPs and Edges for 5 days data
nation, respectively, that are traced by another platform as well (corresponding to 0.01%, 1.9%, 1.9%, and 0.5% of each platforms destinations, respectively). As Ripe traces to its own anchors, we observe that they utilize only 193 destinations. We get our data from anchors, so we might not see traces from other devices (probes). Overall, while others have traces from few sources to a large number of destinations, Ripe has traces from thousands of sources to less than two hundred destinations.

When analyzing the IP addresses, we observe that 99.4%, 44.1%, 97.6%, and 82.8% of IP addresses observed by Ark, iPlane, Mlab, and Ripe, respectively, are not observed by the other platforms during the same time frame. When compared to Ant census data that has 370.1 million ping responsive IP addresses (note that Ant census cycles take about two months to probe all IPv4 space), we observe Ark, iPlane, Mlab, Ripe, and all platforms, has 88.1%, 96.5%, 92.2%, 86.1%, and 88.1%, respectively, IP addresses that were not observed in the Ant census data. This most likely indicates that these devices ignored the ICMP ping requests but generated an ICMP TTL exceeded message.

Additionally, we observe that each platform is able to obtain unique edges compared to the rest. In particular, 89.9%, 96.3%, 98.4%, and 61.8% of edges sampled by Ark, iPlane, Mlab, and Ripe, respectively, unique to that platform and not observed by another platform during the same 5-day period. These findings indicate that vantage point location plays an important role in discovering new edges.

Figure 5.4 presents the number of unique traces (where traces that have the same sequence of IP addresses are ignored) observed by each method. We observe that Ark, iPlane, Mlab, and Ripe obtain 53.5%, 47.8%, 0.5%, and 0.9% of all traces, respectively.
The Figure 5.4 also shows how many of these traces contain a routing loop and repeated IPs at the end (most likely caused by a NAT box or firewall). We observe that 1.4%, 49.4%, 80.9%, and 9.6% of iPlane, Mlab, Ark, and Ripe traces, respectively, have a repeating IP addresses at the end. We believe Ripe and iPlane observe fewer border devices as they have selected destinations while Ark and Mlab trace toward arbitrary IP addresses that might contain a border device (NAT box or firewall) that causes these repetitions. Similarly, we observe 36.8%, 42.0%, 59.2%, and 44.0% of iPlane, Mlab, Ark, and Ripe traces, respectively, have a routing loop in them. Overall, we observe about half of traces have a routing loop in them.
5.2.2 Data Analysis per Source

In Figures 5.7 through 5.8 we perform an analysis of the relationships between vantage points and the data they are able to discover: Figure 5.6 shows the number of traces seen per vantage point, Figure 5.8 shows the number of edges discovered per vantage point, Figure 5.7 shows the total number of IP addresses visited per vantage point, and Figure 5.5 shows the number of destination addresses targeted per vantage point. We can see from the data that where Ripe is utilizing a much larger number of vantage points (6,146 in total for the 5-day period), they actually see a smaller portion of the network. This is likely because they perform repeated measurements of the same space, since they are primarily concerned with issues of reachability of Ripe Atlas nodes rather than the collection of data for a topology study. iPlane and M-lab both utilize the PlanetLab nodes for vantage points, but iPlane seems to see a much larger part of the network since it scans a much larger block of destination IP addresses. Ark by far outperforms both with only 112 vantage points, but targets a much larger number of IP addresses.

Figure 5.5 shows Ripe is utilizing the largest set of destination IPs, yet in Figure 5.7 it is apparent that they see the smallest number of unique IP addresses. This mainly because Ripe is performing repetitive traces between its own probes and anchors. Ark is targeting the largest number of destination IPs, followed closely by iPlane.

In Figure 5.6 we see that while Ripe has a few source IPs collecting many traces, it has many source IPs collecting only a few traces. While iPlane utilizes a smaller number of destination IPs than Ark, it collects a comparable number of unique traces overall. M-Lab and Ripe totals are similar, even though the M-Lab network
Figure 5.5: Destinations per Vantage Point for 5 days data

Figure 5.6: Traces per Vantage Point for 5 days data

is considerably smaller.

iPlane and M-Lab are both utilizing PlanetLab nodes as vantage points for their measurements; iPlane using 111 source IPs and M-Lab using 176. Even though
M-Lab has a larger set of sources, iPlane sees a substantially larger number of IPs, as shown in Figure 5.7. This is possibly because the destination IPs targeted by iPlane are larger in number and are generally located in backbone ISPs. Many M-Lab measurements are user defined, and are repetitive.
In Figure 5.8 we see that iPlane has a few vantage points that see a very small number of edges, possibly performing a small number of measurements. The bulk of iPlane vantage points see a very consistent number of edges, again likely due to the location of iPlane destination IPs in the backbone of the Internet. Ripe has the largest network, but sees the smallest number of unique edges.

We can see from the data in Figure 5.5 that Ripe nodes are tracing primarily to their anchors. We also see that most Ark monitors observe around 1 million IP addresses, while a few of them only observe tens of thousands of IPs. iPlane monitors observe on the order of a half million IPs, and seems to consistently target the same number of destinations. M-Lab monitors observe 25K to 500k IP addresses, and are significantly less consistent in the numbers of destinations they target. Ripe monitors seem to observe varying number of IP addresses depending on the vantage point. Similarly, Ark monitors observe around 1.4M to 30K edges, iPlane monitors observe approximately 750K to 30 edges, Mlab monitors observe approximately 30K to 500 edges, and Ripe monitors observe approximately 9K to 1 edge. We observe a similar pattern when the number of traces is considered, i.e., Ark and iPlane vantage points have similar workloads while Ripe and Mlab workloads vary.

5.2.3 Data Analysis per Destination

Figures 5.11 through 5.12 present the number of traces, sources, IPs, and edges per destination IP address. In Figure 5.9, we observe that only iPlane sources are targeting all destinations consistently. Similarly, Ripe vantage points are probing majority of destinations except couple of monitors that are lagging. As Ark sources target destinations with groups, only a few sources reach a destination.
M-Lab probes show a decreasing trend, which can be attributed to its measurement campaigns being triggered by users.

Figure 5.10 counts the unique traces to a given destination. Here we observe that some destinations targeted by M-Lab and Ark receive a relatively small number of traces. While there are many destinations that are probed only by ark monitors, they discover the fewest number of IPs and edges per destination (due to the team configuration of Ark). Overall, per destination, Ripe seem to observe the largest graph area as it has a much higher number of vantage points to work with.

![Figure 5.9: Sources per Destination (log-log scale)](image)

We can see from Figure 5.9 that Ripe uses by far the largest number of destinations of the four platforms, and sees a relatively highl number of IPs per destination as Ripe nodes are tracing back to Ripe anchors. We can see from the chart that iPlane is very consistent in the numbers of source IPs seen per destination, indicating a well distributed set of destination IP locations. We see that there appear to be four
main groups of destinations seeing between 1 and 4 sources per group. This is inconsistent with the 3 team architecture of Arks tracing strategy of dividing the nodes into 3 teams.

Figure 5.10: Traces per Destination (log-log scale)

Figure 5.10 shows that Ark traces are seen by one of four distinct groups of destinations, which is somewhat inconsistent with Arks strategy of probing from 3 teams. We also see that Ripe sees a high concentration of traces from the few destinations reported, as all probes are tracing back towards one of Ripes 202 anchors. M-Lab destinations generate an inconsistent number of traces per destination as many measurements are user defined, while iPlane destinations are very consistent in the number of traces they generate.

Again in Figure 5.11 we see a high number of IP addresses seen by Ripe destinations, as there are very few of them. Ark sees fewer IPs per destination, but uses a very large number of destinations for tracing. iPlane again shows a very consis-
Figure 5.11: IPs per Destination (log-log scale)

tent number of IPs discovered per destination, attributed to the careful selection of target destination IP addresses.

Figure 5.12 shows that while a few M-Lab destinations discover a large number of edges, there are many destination that see only a few edges. Ripe destinations are few, and discover a large number of edges per destination. Ark sees fewer edges per destination, but a larger number of destinations overall.

5.3 AS Coverage

In this section we perform analysis to understand how well each data set is able to discover the AS space. We compare the platforms by collecting BGP announcements which tell us which subnets belong to which AS. Note that we only utilize
BGP data sets from 2016. We then classify our IP data based on which subnet each address belongs to and determine how many ASes are covered by the data set. We compare by numbers of ASes covered with regard to numbers of source IPs, traces, total IPs, and edges (IP’s contained in an edge).

5.3.1 BGP Coverage

Table 5.3 presents the detailed coverage of each probe data set with respect to BGP data. IPs announced and IPs unannounced indicate the number of IP addresses from the probing data set that were covered and uncovered, respectively, in the BGP data sources. We observe that not all IP addresses found in probes are announced in the BGP data sets. CIDR, CAIDA, and combined, overall have 25.7%, 23.6%, and 11.2% of combined IP addresses not announced in their prefixes, re-
Table 5.2: BGP coverage

<table>
<thead>
<tr>
<th></th>
<th>RIPE</th>
<th>M-Lab</th>
<th>iPlane</th>
<th>Ark</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Src</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Announced</td>
<td>4,948</td>
<td>164</td>
<td>110</td>
<td>110</td>
</tr>
<tr>
<td>Missing</td>
<td>1,218</td>
<td>12</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>AS covered</td>
<td>611</td>
<td>30</td>
<td>53</td>
<td>45</td>
</tr>
<tr>
<td>Prefix covered</td>
<td>1,997</td>
<td>45</td>
<td>67</td>
<td>70</td>
</tr>
<tr>
<td><strong>Dest</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Announced</td>
<td>184</td>
<td>156,169</td>
<td>205,645</td>
<td>52,482,541</td>
</tr>
<tr>
<td>Missing</td>
<td>11</td>
<td>5,488</td>
<td>12,088</td>
<td>307,737</td>
</tr>
<tr>
<td>AS covered</td>
<td>156</td>
<td>7,165</td>
<td>42,469</td>
<td>46,068</td>
</tr>
<tr>
<td>Prefix covered</td>
<td>166</td>
<td>38,081</td>
<td>157,477</td>
<td>190,872</td>
</tr>
<tr>
<td><strong>IP</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Announced</td>
<td>4,947</td>
<td>156,689</td>
<td>531,951</td>
<td>48,225,987</td>
</tr>
<tr>
<td>Missing</td>
<td>1,207</td>
<td>5,144</td>
<td>31,362</td>
<td>6,133,553</td>
</tr>
<tr>
<td>AS covered</td>
<td>1,611</td>
<td>7,261</td>
<td>43,388</td>
<td>45,032</td>
</tr>
<tr>
<td>Prefix covered</td>
<td>2,002</td>
<td>39,637</td>
<td>172,076</td>
<td>467,050</td>
</tr>
</tbody>
</table>

spectively. Note that we identified the AS of the uncovered IP addresses using DNS lookup. Additionally, ASes and Prefixes indicate the number of covered ASes and Prefixes that had an IP address in the probe data set. We observe that CIDR has the fewest ASes and prefixes that are observed in the probe data sets.

The source row shows that 20% of Ripe vantage points did not show up in BGP prefix announcements but were located using DNS lookup. The destination row shows the total number of destination IPs within that data set.
5.3.2 AS Rank

Figures 5.13 through 5.15 show the combined BGP coverage of the respective characteristics. Note that we identified the AS of the uncovered IP addresses using DNS lookup. Figure 5.13 shows that 20% of Ripe vantage points did not show up in BGP prefix announcements but were located using DNS lookup. Figure 5.14 shows the total number of destination IPs within that data set. Destinations are grouped as either belonging to an AS (within a prefix) as covered or not covered in the combined 2016 prefix announcement data set.

Figure 5.13: AS coverage of Sources

Figure 5.14: AS coverage of Destinations
Figures 5.16 through 5.19 presents the coverage of each AS with respect to the number of sources that reached the AS, the number of trace destinations that either reached the AS or crossed the AS, and the number of IP addresses discovered per AS. Note that the figure ranks 42K observed ASes independent of data source while combined shows the cumulative discovery of measurement platforms for that AS. In figure 5.16, we observe the AS rank based on the number of vantage points it contains. For instance, there is one AS that has 384 Ripe vantage points in it. Also, Ripe has a large number of probes in the same AS, while Ark has very few probes located in the same AS.

Figure 5.16 shows the number of sources IP addresses seen by each AS. We see here that Ark sources are generally seen by only one AS, indicating that vantage points on the Ark network are very well distributed. Ripe vantage points are many, and follow a heavy tailed power law degree distribution. Many M-Lab vantage points are in one or two ASes, while some ASes contain many vantage points.

Figure 5.17 shows Ripe anchors are carefully distributed, with most appearing singularly in an AS. Ark uses a very large number of destinations and covers by far
Figure 5.16: AS rank of Sources

Figure 5.17: AS rank of Destinations
the largest number of ASes. M-Lab and iPlane show a roughly similar coverage of ASes with their destinations.

Figure 5.18: AS rank of IPs

Figure 5.18 Ark, iPlane, and M-Lab display a power law degree distribution of IPs per AS, while the distribution of IPs per AS by the Ripe platform is particularly heavy tailed. We notice that many ASes see only a few IPs in the Ripe data, possibly indicating that Ripe targets only the first router in an AS to determine reachability, and stops there to save resources.

In Figure 5.19 we see that while Ark sees by far the most edges per AS, the distribution is similar among all platforms.

In Figure 5.17, we observe that iPlane and Ark trace toward a similar number of ASes, while Ripe traces toward very few. The results show that with much fewer traces iPlane is able to match Ark’s AS coverage. This indicates iPlane is able to produce a good information plane for end-to-end communications, and is likely
due to the broader distribution of their vantage points on the PlanetLab system. In IP and edge distribution of ASes, we observe a power-law like distribution, where there are few ASes from which a large number of IP and edge are observed, while there are a large number of ASes from which a few number of IPs and edges are observed. Overall, we observe more edges than IPs per AS.

## 5.4 Unresponsive Routers

One of the issues encountered in the path traces is the problem of unresponsive routers. Some routers on the Internet are set not to respond with an ICMP time exceeded message, or may not respond for other reasons. These routers appear in the traceroute as a “*”, indicating that a router did not respond [48]. We assigned an ascending integer value to unresponsive hops so that each can be counted as an individual router, but their connection to neighboring routers are not counted as
5.5 Routing Loops

We find numerous anomalies in the traceroutes. For example, there may be a number of repeated IP addresses at the end of a trace, (i.e. A-B-C-D-D-D) which would likely indicate a NAT box that is repeating the IP at the entry to this part of the network several times for each translated IP address. Another common situation is one where the last IP address is repeated at the end of a trace with one unique IP in between (i.e. A-B-C-D-C) followed by a series of unresponsive routers. This likely indicates that there was a firewall, and no responses were possible for all routers between the firewall and the destination. There will be loops present in some of the traces (i.e. A-B-C-D-E-C-F-G), usually indicating some type of misconfiguration, malfunction, or error in the forwarding table of a router. There could also be a loop.
of the type (i.e. A-B-C-D-D-E), which seems to indicate a router looping to itself. This could be due to a router ignoring TTL checks, leading to the subsequent router generating two replies. Lastly, there will be some traces that end with a unique IP address, but are preceded by a repeating address (i.e. A-B-C-D-D-E). Figure 5.20 shows the number of these anomalies found per data source (Ark, iPlane, M-Lab, Ripe). All of these situations are counted, and comparisons will be made between the different platforms.

Figure 5.20 depicts the number of these anomalies that are found in the data provided by the indicated platform. Each trace is examined, and occurrences of identified patterns are counted and categorized as either a repeat or a loop.

5.6 IP Address Observance in Traces vs. Pings

In this section we analyze IP address coverage of trace platforms against replies to pings by Ant Census. Ant utilizes ICMP Echo Request messages (ping) to determine if routers are responsive. Since many routers are set not to respond to ping requests, but still return an ICMP Time Exceeded message, we observe a number of IP addresses that would not appear in the Ant census. Such addresses are useful in that they add to the body of responsive targets that can be used to generate new trace data. It will also be interesting to see if there is a difference in the number of these IPs found between the data platforms we are studying.

Figures 5.21 and ?? summarize the numbers of IP addresses identified in the indicated platform that did not appear in the Ant Census data.
Figure 5.21: Trace IP addresses not seen in Ant Census (graph)

<table>
<thead>
<tr>
<th></th>
<th>RIPE</th>
<th>M-Lab</th>
<th>iPlane</th>
<th>Ark</th>
<th>All Trace Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique vs. Trace Data</td>
<td>54,359,540</td>
<td>563,313</td>
<td>161,833</td>
<td>6154</td>
<td>54,771,284</td>
</tr>
<tr>
<td>Unique vs. Ant Data</td>
<td>10,062,583</td>
<td>1,792,066</td>
<td>83,919</td>
<td>8258</td>
<td>82,511,209</td>
</tr>
<tr>
<td>Ant Census Data</td>
<td>370,706,693</td>
<td>370,706,693</td>
<td>370,706,693</td>
<td>370,706,693</td>
<td>370,706,693</td>
</tr>
</tbody>
</table>

5.7 BGP Level Topology Data

In order to gain a comprehensive understanding of the BGP data sets, in this section, we analyze AS level prefix announcements of UCSD CAIDA [15], UCLA IRL [13], and CIDR [8]. AS level topological data can deliver insight into the connectivity and relationship among ASes. This includes information not only about their links to one-another, but also details about their inner topology.
5.7.1 BGP Announcements

Table 5.4 presents BGP data (i.e., announced ASes, announced prefixes, and total number of IPs in announced prefixes) provided by each platform. Among BGP data providers, UCLA IRL had no new data since 2015 and we were not able to obtain historical data from CIDR. We combined IRL and CIDR’s latest data sets with Caida’s BGP prefix announcement from the same time frame to obtain 2015 and 2016 data sets, respectively. While CIDR provides much fewer number of ASes, it has comparable number of IP addresses in the announced prefixes.

<table>
<thead>
<tr>
<th></th>
<th>ASes</th>
<th>Prefixes</th>
<th>IPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIDR16</td>
<td>1,911</td>
<td>145,734</td>
<td>2,694,269,384</td>
</tr>
<tr>
<td>CAIDA16</td>
<td>55,411</td>
<td>604,237</td>
<td>3,220,560,902</td>
</tr>
<tr>
<td>2016</td>
<td>55,563</td>
<td>745,507</td>
<td>5,321,550,144</td>
</tr>
<tr>
<td>IRL15</td>
<td>51,081</td>
<td>487,071</td>
<td>3,939,677,760</td>
</tr>
<tr>
<td>CAIDA15</td>
<td>49,061</td>
<td>356,171</td>
<td>3,087,590,262</td>
</tr>
<tr>
<td>2015</td>
<td>52,128</td>
<td>305,956</td>
<td>4,738,394,582</td>
</tr>
</tbody>
</table>

5.7.2 MOAS Prefixes

Figure 5.23 and Figure 5.22 present BGP data (i.e., announced ASes, announced prefixes, and total number of IPs in announced prefixes) provided by each platform. Interestingly, CIDR provides a much smaller number of ASes but has comparable number of IP addresses in the announced prefixes. We also consider the multiple MOAS prefix between AS prefixes, i.e. an IP range being announced as
part of two separate ASes. We observed that the CIDR and IRL data has prefix announcements that conflict with each other, while CAIDA has no conflicting announcements as shown in Figure 5.22. This indicates that while CAIDA filters such conflicting announcements, CIDR and IRL do not. When we consider pairwise conflicts between data sets, we observe IRL and CIDR also have conflicts with CAIDA announcements for respective years.

We then processed the prefix announcements to filter out duplicates and we merge consecutive smaller announcements into bigger ones. We identified multiple announcements for AS prefixes, i.e., an IP range being announced as part of two separate ASes, also called MOAS prefixes [47]. We observed that the IRL and CIDR data contain 51,554 and 946 prefix announcements, respectively, that overlap with each other involving 10,061 and 524 ASes. On the other hand, CAIDA had no such overlapping announcements indicating that such announcements are being
filtered. When we consider pairwise overlaps between the data sets for respective years, we observe IRL and CIDR announcements also overlap with the CAIDA announcements.

5.7.3 Prefix Distribution

In figure 5.24, we present how the subnet prefix announcements are distributed for each data set. Granularity of the prefix announcements give insight into how the longest prefix matching operations in the border routers result in higher AS coverage for each router level data source. We observe that a majority of the announced prefixes are /24. We observe a drop in the number of unique subnets smaller than /24. Note that we eliminated non-conflicting smaller prefix announcements over
the same region.

Figure 5.24: Prefix announcements vs Mask distribution (log scale)
To replace the PlanetLab network for our Internet measurement system, we will implement a network of small single board computers (Odroids) which will be distributed globally. This network will be connected by a virtual private network (VPN) to a server at the University of Nevada Reno. These devices will be implemented with a lightweight version of Debian Linux, and will provide the necessary functionality to perform traces and IP alias resolution for future study. Nodes will be distributed to Universities, network researchers, friends, and family both in the U.S. and abroad.

6.1 The Odroid Device

To implement a network of probes to be used as vantage points from which to collect measurement data, we have selected the ODROID-C2. Micro computing Odroid-C2 is a 64-bit quad-core single board computer that is both cost effective and powerful enough to meet our needs. It is built with an ARM architecture, and is capable of running a Linux operating system, making it a good match for our measurement system. The ARM processors small size, reduced complexity and low power consumption makes it very suitable for connection to a home network, and will have a small footprint in terms of network connection, power consumption, and physical dimensions so that it will be relatively friendly to potential hosts. We have chosen to install a lightweight version Linux, a 64-bit Armbian-Jessie distribution, with all non-essential services disabled to reduce maintenance issues and to increase security hardness. While the C2 can operate from an SD card, it
Figure 6.1: Board details of the Odroid C-2. [6]

also has the capability to use an eMMC memory device, making it more reliable and in the long term a bit faster. We use an 8 GB memory, and have encrypted the data to avoid compromising keys that would allow access to University systems that interact with the measurement nodes. In addition, we make the memory write-only, to prevent damaging the flash memory by repeated write operations, a problem faced by earlier platforms [17].

6.2 Network Architecture

To communicate with the Odroids once they are placed in a remote location, we have set up a Virtual Private Network (VPN) and a /24 private subnet to which future probes can be added. Our server at the university is also on the VPN, and we will run our measurement tool from this location. We use OpenVPN to implement the virtual network. A small script written in Python updates a log file written by the operating system in the server with information on the probes gateway.
6.3 Deployment of IM Nodes

In order to maximize our view of the Internet, we will attempt to distribute the probes globally in locations that will assure they are connected to different ASNs as much as possible. Effort will also be made to place them in different geographic regions. A brochure outlining some of the common questions and concerns we anticipate from hosts is delivered with the probes, and a spreadsheet of probe identities and locations is maintained.
The following is an example of the text used to recruit volunteers to host an IM node:

1. **WHO IS SPONSORING THIS PROJECT?**

   The Internet Measurement Project is sponsored by the NetLab at the University of Nevada-Reno, and is funded by a grant from the National Science Foundation (NSF grant number CNS-1321164). The project is led by Mehmet Gunes. This ongoing project has produced several scholarly articles, and has made significant contributions to our understanding of the Internet topology and its dynamics.

   ![Figure 6.3: Finding the “backbone” of the Internet.](image)

2. **WHY SHOULD I HOST A MEASUREMENT NODE?**

   By hosting a measurement node, you help understand and sustain a better Internet. Your help in hosting a node will make a difference, as it gives researchers a way to see into the Internet from different vantage points, and to make new discoveries about its behavior.

3. **WHAT IS THE MEASUREMENT NODE DOING?**

   Imagine trying to find all the roads that lead to Omaha, Nebraska. How would you do it? If you lived in Chicago, you could follow every road between Chicago
and Omaha, and write them all down. But that would only help you find the roads between those two places; what about the roads from Minneapolis, or San Francisco? To get a complete map, you would have to be able to follow the roads from these places to Omaha as well. In fact, the more starting points you could find, the better a map you would have. This is essentially what the measurement node is doing; giving a different view of the Internet.

The measurement node uses commonly used network diagnostic tools traceroute and ping and their variations to send out small probe packets to various addresses on the Internet, and receives replies. This information is then forwarded to a server at the University of Nevada, Reno (UNR). The node is basically a very simple computer with limited capabilities that maintains a VPN connection to UNR server. It waits for instructions on where to send its next probe, and then sends the results back to the UNR server.

4. HOW DOES THE MEASUREMENT NODE WORK?

The measurement node has a small program on its memory that connects it to a Virtual Private Network (VPN), which includes other measurement nodes around the world. Once plugged in to your network, it phones home to let the server know what its network IP address is, and then it waits for tasks. When idle, it simply sleeps, and waits for instructions. When active, it sends probe packets into the Internet and returns the results to the UNR server. It has a Linux operating system with very limited services. Its memory has been encrypted to prevent anyone from making changes, or using it for malicious purposes. If it loses power or is disconnected from Internet, it will re-establish its link to the home server at UNR automatically. It uses very little power, and has a minimal effect on the host network traffic.
5. WHAT IS INSIDE THE NODE?

Figure 6.4: Board details of the Odroid C-2. [6]

Figure 6.5: Block diagram of the Odroid C-2. [7]

6. How do I connect the measurement node to my network?

The measurement node kit includes an Odroid-C2 unit, a power supply, and an ethernet cable. Simply connect the supplied ethernet cable from the port on the
Odroid-C2 to one of the ethernet ports on the back of your home router as shown below (note that your router could look different). Then plug the power supply into a wall outlet (110v in US or 220v in Europe) and to the Odroid-C2 unit. You will see a red light and a flashing blue light on the Odroid-C2 indicating normal operation. At this time the Odroid-C2 will automatically set up its network connection and establish communication to the server at the University of Nevada Reno.

Figure 6.6: Odroid connection

7. How much traffic will the measurement node produce? Will it affect the performance of my Internet service?

The measurement node will produce traffic in and out of your home network, but
the total amount of traffic is a very small percentage of the total available bandwidth. We estimate the traffic volume to be between couple of Kbps and in the worst, be less than 100 Kbps. Given most ISPs provide links at 5-50 Mbps, the measurement node will consume between 0.002% and 0.02% of your total available bandwidth when active. Only in rare instances of bandwidth estimation during the node initialization, we could generate noticeable traffic. We will also analyze the traffic volume out of the node to ensure it consumes less than 1% of your network bandwidth.

8. WILL HOSTING A NODE COMPROMISE THE SECURITY OF MY NETWORK?
The measurement node has a very singular purpose; to send measurement probes into the Internet, and return the results to UNR server. Both server and IM node security is maintained by UNR IT. We monitor the nodes to assure it is secure from outside attacks as well to prevent anyone from hacking it or using it to compromise your network. The node has a minimal Linux operating system and is configured to obtain software upgrades and security patches automatically.

9. WILL HOSTING A NODE COMPROMISE MY PRIVACY?
The measurement node does not interfere with your own network traffic and has limited functionality. Only connection to it will be from the measurement server at the UNR. We will only monitor the traffic generated by the measurement node itself to ensure proper operation and detect security incidents. The node does not have any audio-video capability and only interacts with dedicated measurement servers at UNR, pre-configured software repositories to get software upgrades and security patches, and Internet devices it measures.

9. DOES THE MEASUREMENT NODE REQUIRE ANY MAINTENANCE FROM MY END?
The measurement node is completely self-contained. Once connected to power and your home network, it will establish a connection to server at UNR and will carry out its mission.

10. **HOW LONG WILL I NEED TO KEEP THE NODE?**

Our goal is to collect measurements over the next 5 years or so. We are very grateful for your willingness to help us in our study, so we will take whatever you are willing to give! If you can host the node for an extended amount of time, that would be very helpful. The node will not produce any noticeable effect on your network, so hopefully you can plug it in and forget about it. If you can no longer host it, simply return it at our expense. If you can no longer host the measurement node, you can return it to us.

11. **What should I do if I can no longer host the measurement node?**

If you can no longer host the measurement node, you can return it to us at our expense.
7.1 Summary

In this study, we discussed some of the fundamental concepts behind the structure of today’s Internet. We defined Autonomous systems, and explored the interconnection of these systems as building blocks of the larger network. We discussed the principle of Internet topology measurement, and some of the tools that can be utilized to perform measurements to discover network structure. We analyzed the public measurement data sets provided by ISI/Ant census, UCSD Caida Ark, UCLA CIDR, UW iPlane, IRL, M-Lab, and Ripe Atlas, and provide insights on their coverage of the Internet topology.

We first analyzed the BGP data sets and compared the three resources, analyzing their prefix announcements and IP address coverage. We observed that some data sets contain multiple AS announcements for the same IP address/prefix. Next, we analyzed the probe data sets from five data sources and compared their topological coverage. We observed that while there were variations in their approaches to data generation with varying degrees of network coverage, each data set provided a unique topological perspective. For instance, the edges discovered by different platforms are often unique as the vantage points of these platforms are often unique and not in the same subnetwork. We also observed persistent routing anomalies producing various loop configurations, as well as unresponsive routers in the trace data which require further processing. We also discussed the deployment of our own measurement platform that we have placed on the Internet to map the Internet backbone.
In general, we have discovered that regardless of the particular data collection strategy, the results indicate that the use of multiple data sets is important for building a comprehensive picture of the Internet topology as they each make a unique contribution. We also observe that with a well distributed set of vantage points, a more detailed picture of the Internet topology can be gained with a smaller number of measurements taken.

7.2 Future Work

We want to extend this analysis to see the effect of ingresses on the distributions as well as the network analysis of the underlying graphs of the data sources. We would also like to complete IP alias resolution so that routing anomalies we discovered when looking at loops and repeats in traceroutes can be more thoroughly and accurately analyzed. Ultimately, we would like to create an Internet measurement platform that would periodically generate a dynamic map of the Internet backbone that we could share with the research community to facilitate a greater understanding of the Internet backbone.
BIBLIOGRAPHY

[1] “Ant censuses of the internet address space” [online.] Available:
https://ant.isi.edu/address/

https://www.caida.org/projects/ark/ark-raspi-monitor-20130102.png

https://www.mdsec.co.uk/2015/09/an-introduction-to-hardware-hacking-the-ripe-atlas-probe/


http://alpha.mini.pw.edu.pl/ marcinp/images/Slice_vis_Silver.gif

http://alpha.mini.pw.edu.pl/ marcinp/images/Slice_vis_Silver.gif

http://alpha.mini.pw.edu.pl/ marcinp/images/Slice_vis_Silver.gif

http://www.cidr-report.org/as2.0/

https://www.measurementlab.net/about/

https://atlas.ripe.net/about/anchors/

https://atlas.ripe.net/about/probes/


[32] Hyun, Y.: Ark topology query system (February 2016)


