2014-06

Comparing internet probing methodologies through an analysis of large dynamic graphs

Landry, Britton

Monterey, California: Naval Postgraduate School

https://hdl.handle.net/10945/42669

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

COMPARING INTERNET PROBING METHODOLOGIES THROUGH AN ANALYSIS OF LARGE DYNAMIC GRAPHS

by

Britton Landry

June 2014

Thesis Advisor:	Ralucca Gera
Second Reader:	Robert Beverly

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)* 12–6–2014	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* 2012-01-01—2013-09-20

4. TITLE AND SUBTITLE

COMPARING INTERNET PROBING METHODOLOGIES THROUGH AN ANALYSIS OF LARGE DYNAMIC GRAPHS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Britton Landry

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

Naval Postgraduate School
Monterey, CA 93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

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

14. ABSTRACT

The Internet is an evolving, robust system with built in redundancy to ensure the flow of information regardless of any act of nature or man-made event. This makes mapping the Internet a daunting task, but important because understanding its structure helps identifying vulnerabilities and possibly optimizing traffic through the network. We explore CAIDA's and NPS's probing methodologies to verify the assentation that NPS's probing methodology discovers comparable Internet topologies in less time. We compare these by modeling union of traceroute outputs as graphs, and using standard graph theoretical measurements as well as a recently introduced measurement. Ultimately, the researchers verified the NPS's probing methodology was comparable to the CAIDA's probing methodology. We also propose additional avenues for further exploration from our initial discoveries. We also introduced a technique that can possibility identify stable core existence among the whole Internet and explore case studies of two country sub-graphs.

15. SUBJECT TERMS

Distance, dissimilarity between graphs, symmetric difference, Internet topology

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
Unclassified	Unclassified	Unclassified	UU	83	

THIS PAGE INTENTIONALLY LEFT BLANK

COMPARING INTERNET PROBING METHODOLOGIES THROUGH AN ANALYSIS OF LARGE DYNAMIC GRAPHS

Britton Landry
Major, United States Army,
B.S, United States Military Academy, 2004

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN APPLIED MATHEMATICS

from the

NAVAL POSTGRADUATE SCHOOL
June 2014

Author: Britton Landry

Approved by: Ralucca Gera
Thesis Advisor

Robert Beverly
Second Reader

Carlos Borges
Chair, Department of Applied Mathematics

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The Internet is an evolving, robust system with built in redundancy to ensure the flow of information regardless of any act of nature or man-made event. This makes mapping the Internet a daunting task, but important because understanding its structure helps identifying vulnerabilities and possibly optimizing traffic through the network. We explore CAIDA's and NPS's probing methodologies to verify the assentation that NPS's probing methodology discovers comparable Internet topologies in less time. We compare these by modeling union of traceroute outputs as graphs, and using standard graph theoretical measurements as well as a recently introduced measurement. Ultimately, the researchers verified the NPS's probing methodology was comparable to the CAIDA's probing methodology. We also propose additional avenues for further exploration from our initial discoveries. We also introduced a technique that can possibility identify stable core existence among the whole Internet and explore case studies of two country sub-graphs.

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

Appendices

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Acronyms and Abbreviations

Ark	Archipelago
ARPANet	Advanced Research Projects Agency Network
AS	Autonomous System
ASN	Autonomous System Number
CAIDA	Cooperative Association of Internet Data Analysis
GB	gigabyte
IETF	Internet Engineering Task Force
IPv4	Internet Protocol version 4
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISP	Internet service provider
NPS	Naval Postgraduate School
NTC	Network Topology Capture
RFC	request for comment
RTT	round-trip time
TTL	time to live
ICMP	Internet control message protocol
NPS	Naval Postgraduate School
ESD	edge symmetric difference
VSD	vertex symmetric difference
TCP	transmission control protocol
UDP	user datagram protocol
DoS	denial of service
DDoS	distributed denial of service
IPS	ingress point spreading

THIS PAGE INTENTIONALLY LEFT BLANK

Executive Summary

The Internet is an evolving, robust system with built in redundancy to ensure the flow of information regardless of any act of nature or man-made event. This makes mapping the Internet a daunting task, but important because understanding its structure helps identifying vulnerabilities and possibly optimizing traffic through the network. We explore CAIDA's and NPS' probing methodologies to verify the assentation that NPS' probing methodology discovers comparable Internet topologies in less time. We compare these by modeling union of traceroute outputs as graphs, and study the graphs by using vertex and edge count, average vertex degree, clustering coefficient and the Pearson coefficient. The results from these measures show CAIDA's and NPS's probing methdologies are compromable. However, using a recently introduced measurement, the probing methodologies actual discover up to 40 percent different sets of vertices and edges captured during almost simulanteous probing. Ultimately, the researchers verified that the NPS's probing methodology was comparable to the CAIDA's probing methodology. We also proposed additional avenues for further exploration from our initial discoveries. We introduced a technique that can possibily identify stable core existence. We conducted preliminary analysis on the interesection of six inferred topologies with promising results. We believe additional probing samples might display the stable core of the Internet. Additionally, we identified South Korea and China as skewed for the NPS probing methodology and conducted a case study of each. We analyzed using the standard graph comparision measures and the intersection to identify a possible stable core. We observed only five percent of stable vertices in China but 40 percent in South Korea.

THIS PAGE INTENTIONALLY LEFT BLANK

Acknowledgements

There are numerous people that made this thesis possible. First I would like to express my thanks and gratitude to my advisor Dr. Ralucca Gera whose patience and motivation were indispensable. Her work ethic and knowledge of the subject was greatly appreciated. I am also indebted to Dr. Robert Beverly, whose ability to make difficult tasks seem easy was greatly appreciated. Without his patience and understanding in teaching me programming, this thesis would not have been possible.

I would also like to thank my family and loved ones for their understanding and encouragement along the way. To my wife, Amanda, thank you for your help and motivation, and to my children Britton Jr., Ashton, Colton and Charleston, thank you for your constant love and support.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 1:
Introduction

1.1 Communications Evolution

Throughout time societies have constantly explored ways to communicate, despite vast distances, either for control, correspondence, commerce or spread of knowledge. We build vast networks to efficiently and rapidly broadcast the information with the newest technology at the time, (e.g., smoke signals, pony express, and telegraph systems). Recently, the computer age has increased the available amount of information which has consumed the inferior methods of distribution before it. This led the United States government to commission a more robust and fault tolerant communications system during the 1960s making way for the Advanced Research Projects Agency Network (ARPANet) which eventually became the Internet.

Unfortunately, with the decommission of the ARPANet, the Internet structure became decentralized, thus harder to map. The physical structure of the Internet became proprietary technology to different business organizations making it difficult to understand how truly information is shared. This has made it increasingly difficult for people to effectively and successfully map the topology of the Internet. Thus a common interest in understanding exactly how links/edges are assigned among routers/nodes has emerged, with the main goal of developing algorithms to track the topology. The current thesis will address this with the goal of measuring and comparing outputs of two such algorithms.

1.2 Why Measure the Internet?

Developing an approximate Internet map is important because having an understanding about how routers connect and interact with each other can lead to better security, or increased efficiency of traffic flow. The algorithms developed to discover the topology of the Internet are useful in understanding how an adversary can limit the exchange of information or completely disable the accessibility of a local network. This is just one type of attack that is commonly referred to as a denial of service (DoS) or similarly a distributed denial of service (DDoS) which targets a system by typically overloading it either through bandwidth or memory. This is a very effective method of disruption which countries and companies spend billions of dollars to prevent. With an understanding of the Internet topology, we can help to mitigate some of the bandwidth bottlenecking that DoS attacks target. Additionally, it could help us understand where to create better redundancy in the network that will prevent accidental DoS by an organization similar to DigitalGlobe's recent request for users help to scour through vast amounts of satellite imagery to find the missing Malaysian Air flight 370 [1].

1.3 Research Question

We began our research wanting to know the following questions.

- What are the substantiative differences between the NPS probing methodology and the CAIDA probing methodology?
- Is there any bias in the algorithms?
- Does one algorithm geographically, by country, discover more of a network?

We were limited on our research by the number of probing cycles available for analysis. We investigated the available probing cycles and made some educated inferences to how NPS and CAIDA algorithms compare to each other.

1.4 Thesis Contribution

We use existing graphical analysis to compare large graphs at a very course granularity. We did this to check similarity of the multiple probing cycles using the existing analysis tools, (i.e., average degree, vertice count, edge count and clustering coefficient). We then used a recently introduced concept of VSD and ESD to compare vertex to vertex and edge to edge how similar two probing cycles are to discovering the same topology. Next we found the intersection between cycles to identify the amount of the Internet that we call the Stable Core found by each probing methodology. We then compared CAIDA and NPS probing methodology's Stable Core to discover similarities. We then compare sub-graphs (data divided by country) to find which probing methodology more accurately represents the Internet. We compared the number of vertices within each country and check if it represents the reported Internet users in [2] and country allocated IP space.

1.5 Organization of Thesis

In investigating the research question, this thesis is organized as follows:

- Chapter 1 discusses the motivation of the research.
- Chapter 2 discusses prior and related work in the fields of measuring the Internet.

- Chapter 3 introduces the machinery used in the analysis conducted in the course of the research.
- Chapter 4 details the data used and the methodology that was developed.
- Chapter 5 contains the results of case studies conducted using our measures.
- Chapter 6 contains the summary and discusses possible areas for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 2:
Background

In this chapter, we establish a base knowledge of the Internet and some key terminology. This will allow a common language for understanding how the Internet is formed, controlled and how it evolved in time.

2.1 Overview of the Internet

The Internet is a global system of connected routers which follow an agreed-upon standard of protocol suites. These protocols are established through the Internet Engineering Task Forces (IETFs) and published in Request for Comments (RFCs)[1] that serve billions of users worldwide. The protocols are known as IPs, which are similar to a mailing address. Information is first packeted with the destination addresses and are routed by routers within Autonomous Systems (ASes) where information is then transmitted. The ASes are a group of IP prefixes, under central control of one or more network operators that presents a common, clearly defined routing policy to the Internet [4]. The routes are nothing more then a path connecting vertices (users) through ASes to each other.

We view the Internet as a group of many symbiotic communities that operate as collective ASes. These ASes work together by connecting to each other forming larger networks. The AS is typically an Internet service provider (ISP) or other large organization with connections to multiple other ASes (e.g., Comcast, Verizon and universities). Each AS has an officially registered Autonomous System Number (ASN) and adheres to the RFCs to properly route the information. When multiple routers establish connections/edges with other routers, they build a routing table and share their table among all the connected routers. The routers distribute these tables in order to compute the most efficient paths that adhere to the businesses constraints of the providers. This occurs because the routers identify preferred routes between themselves to other routers/vertices. However, there exists a delineation between internal and external routers within an AS. Internal routers will only handle the traffic within their ASes and will refer to all ingress facing or transit routers for any connection outside their organic ASes. The internal AS routers can save IP space by allowing large companies to subdivide the company's network to enable more users without affecting the ingress router, an expression called "piggybacking" or

[1] Document series containing technical and organizational notes about the Internet [3].

"subnetting." Likewise, all ingress/external routers will ignore any subnetting groups of large networks, namely their own AS and outside AS. The design of internal and ingress/external routers help to reduce the number of entries in the ingress/external transit routing tables. The router accomplishes this by delegating some roles to the internal routers within the ASes and other responsibilities to the ingress/external routers.

In Figure 2.1 [5], the boundary routers (e.g., R11, R12, R21 and R31 in Figure 2.1) are ingress routers that bridge the ASes. These routers are commonly referred to as ingress, while internal routers such as R13, and R14 in Figure 2.1 handle the traffic inside their AS. Physically, the routers are connected, via their interfaces, and wired or wireless connections between interfaces on other routers/devices. A trivialized representation of the Internet is shown in Figure 2.1 where the ASes are identified by their respective ASNs. In this illustration, the transit [2] edges connect ASN 1 to ASN 2 and ASN 1 to ASN3.

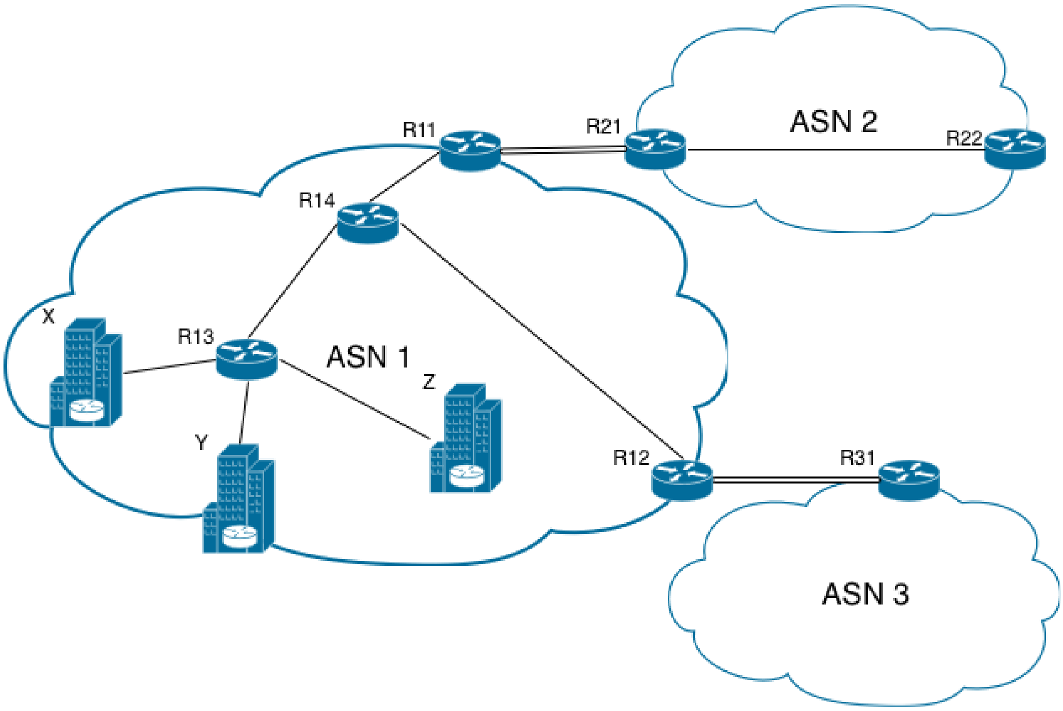


Figure 2.1: The Internet simplified from [5].

[2] Edge between two external routers.

2.2 Internet Topology

The true Internet topology is a difficult problem to truly represent due to the ever changing structure of the Internet, sheer size and evolution. When we map the Internet we are looking at a "snap shot" in time of what the Internet looked like. The research community hopes that taking multiple "snap shots" throughout time will allow the researcher to map and gain some insight to the Internet's structure. For our research, what is more arduous is what these "snap shots" mean and how we compare them, before being able to predict how the Internet will act.

2.2.1 Levels of Internet Topology

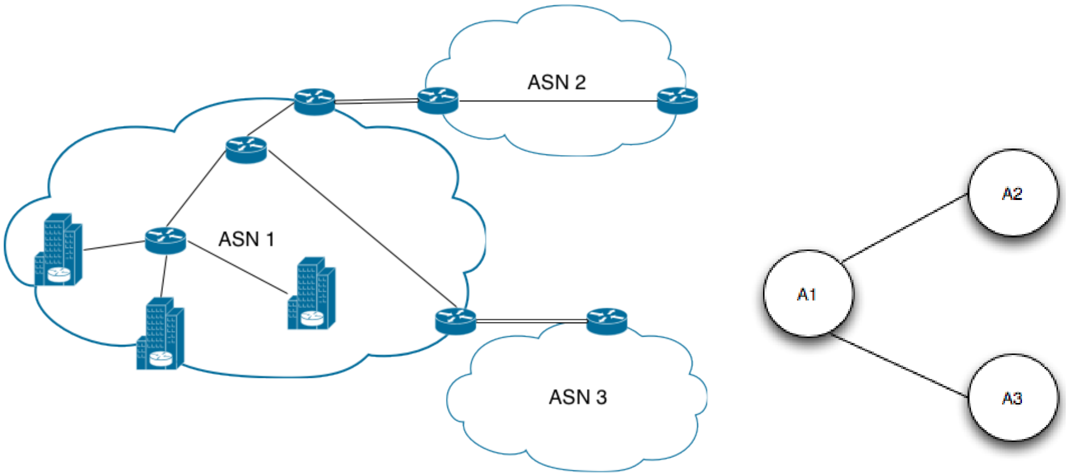
There are numerous granularity levels we can study within the Internet, (e.g., fiber, IP address, router, and ASes). For each level, we represent a fictitious network with a corresponding graphical representation(s). We are particularly interested in interface-level mapping because it offers the most clarity of connectivity and if we have the IP level graph, then the others could be inferred. We limit our research to this interface-level and the connectivity of the segments of the Internet. Below we discuss the different levels for completeness.

AS-level topology. At the AS level, the ASes are vertices and communication paths between them form edges between the vertices. This is seen more prominently as provider-customer and peer relations [6]. With this system, a customer pays a provider to connect to the Internet outside of its administrative domain. Then these providers contractually agree to exchange information traffic between each other's customers mostly free of charge. Thus, the Intranets and their connections that the ISPs provide to its customers form the Internet between the administrative domains. This means that the AS level captures more of an economic relationship between ISPs rather than the physical connections between routers of the Internet.

The AS-level graph representation of Figure 2.2a is shown in Figure 2.2b from [5].

Subnet-level topology. Subnet-level mapping [7] involves discovering the IP addresses that are hosted on the same subnet. These subnets are defined by the interfaces that they connect to. The subnet level topology has the subnets as the vertices and the routers as the edges connecting the various subnets. This methodology is shown in Figure 2.3 from [5].

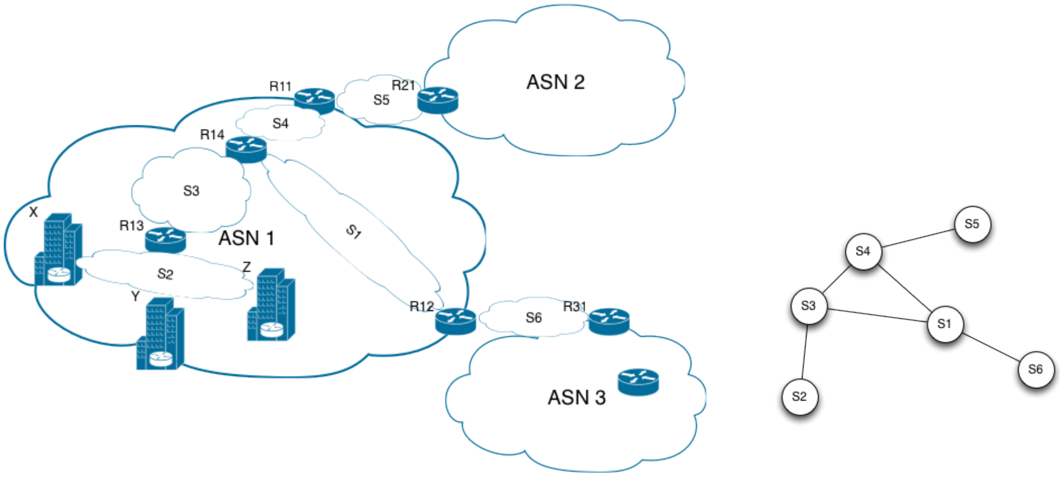
Interface-level topology. Interface-level or IP level routing works similarly as subnet-level mapping because router interfaces and end hosts are captured. The connections between two router interfaces form an edge, while the interfaces themselves are the vertices. It is important to note how routers can have multiple interfaces, which could require multiple graphs to ensure the



(a) Network map.

(b) Graph.

Figure 2.2: Autonomous System-level representations from [5].



(a) Network map.

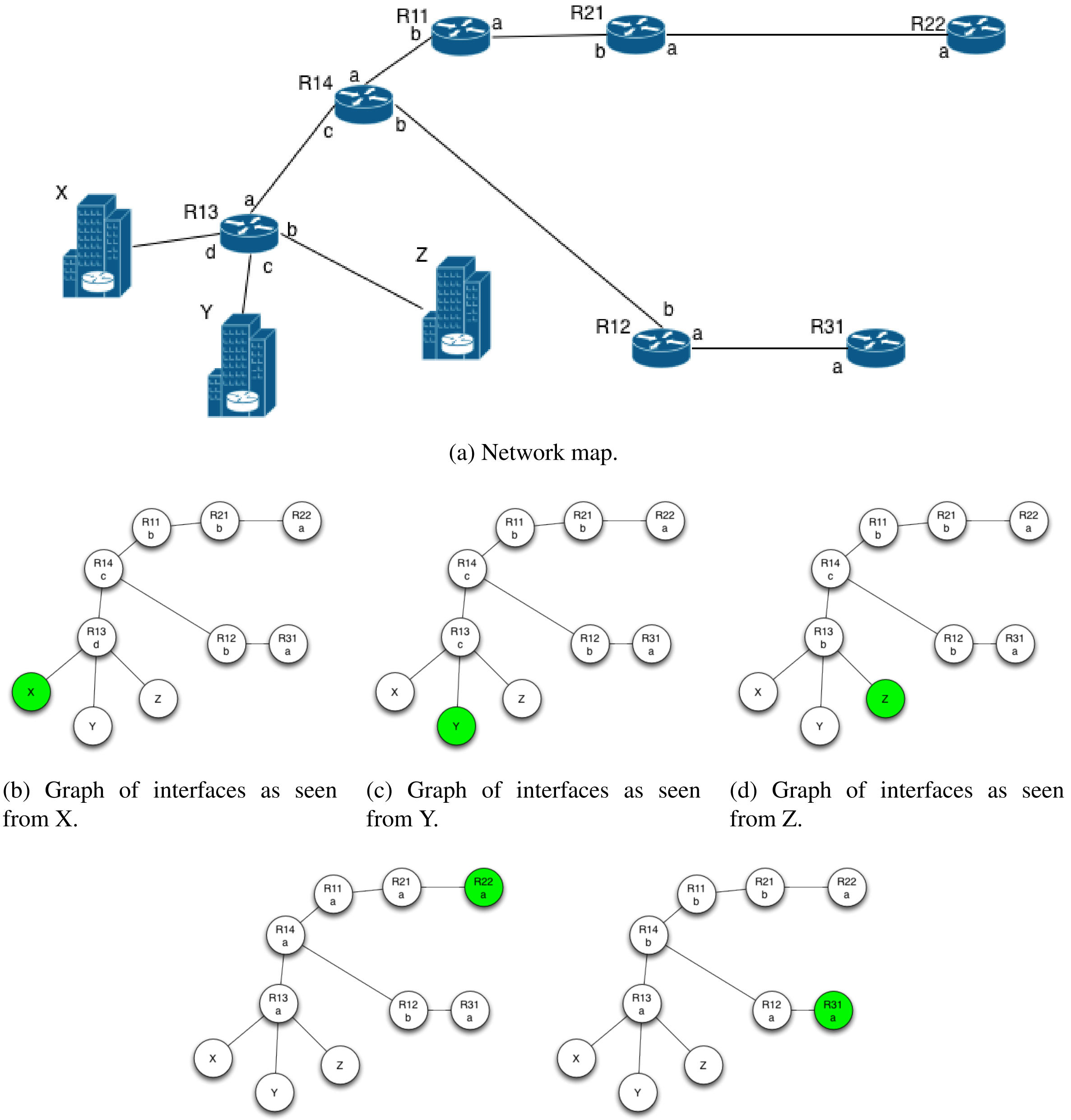
(b) Graph.

Figure 2.3: Subnet-level representations from [5].

discovery of all connections/edges between the routers. Figure 2.4 [5] displays some variations of graphs depicting various perspectives to different end points.

Router-level topology. Routers have multiple interfaces, each with a different IP address, and by IP Alias Resolution [3] they are combined to represent one vertex in the Router-level graph. These vertices are then connected by edges, which represent a possible physical link between the routers. These vertices and edges form the graph used to represent the Internet.

[3] A process to identify IPs which belong to the same router.



(a) Network map.

(b) Graph of interfaces as seen from X.

(c) Graph of interfaces as seen from Y.

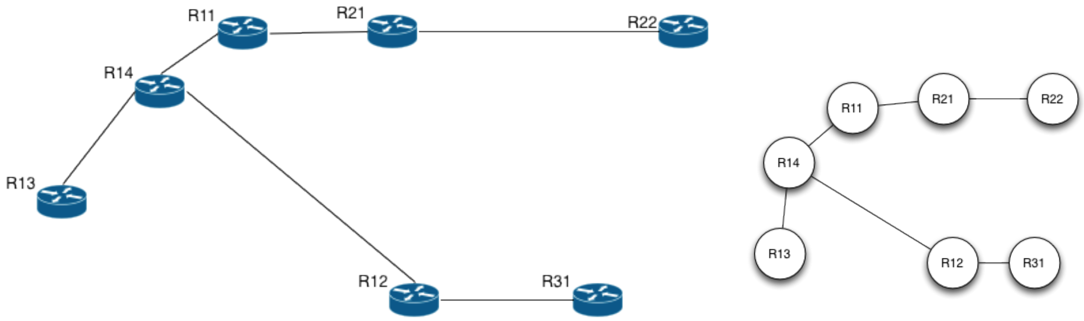
(d) Graph of interfaces as seen from Z.

(e) Graph of interfaces as seen from R22.

(f) Graph of interfaces as seen from R31.

Figure 2.4: Interface-level representations from [5].

The topology of these graphs formed from this method are more useful because the vertices represent the physical routers and the edges the physical links/edges of the actual physical network layout. However, IP Alias Resolution is not precise, although current research shows promise in solving this problem [8–10]. Thus, having an understanding at the IP level will help generate an understanding of the Internet at the router level.



(a) Network map.

(b) Graph.

Figure 2.5: Router-level representations from [5].

2.2.2 Acquiring Active Network Topology

There are numerous Network Topology Capture (NTC) algorithms to acquire network topologies including, DIMES, IPlane, Ark IPv4 All Prefix /24 and recently NPS probing methodology. NPS probing methodology is different from the others because it includes adaptive probing techniques which leverage ingress knowledge and data from previous cycles(s) to choose the best probe destination and assignment of vantage point to destination [11]. The idea is by exploiting previous knowledge of the data, one can reduce the cost, mainly time and additional traffic load on the Internet. NTC's goal is to reduce the discovery costs (i.e., probes sent and time discovery time), while ensuring maximum coverage of the network [12]. However, there are limitations of NTCs from offline systems, firewalls or overloaded edges during the time of probing. In [13], the authors investigated if the time of day for a probing cycle matters. They did not find evidence that the time of the day matters, but they caution that their sampling size was small and additional research should be done.

Ideally, we want to use ground-truth [4] to compare the effectiveness of NTC algorithms. Unfortunately, it is impossible to obtain ground-truth because many organizations will not share the information for security reasons. An organization's network graph can expose security flaws and allow the possibility of sabotage. Organizations like CAIDA provide datasets [14] from both active and passive measurement of the Internet that are readily available to the research community. We will use some of CAIDA's datasets throughout our research, which of course is not ground-truth, but it provides us data to compare NPS's probing methodology.

[4] Actual existing graph of a real network.

2.2.3 Traceroute

Traceroute is a network diagnostic tool that allows a user to identify how a network sends information between devices/vertices. The use of traceroute allows a user to assess a network to help identify and fix connection issues. Typically most traceroutes use Internet control message protocol (ICMP) and a Time to live (TTL) that increment at every point of routing. This is useful because after the completion of each trace, a history of the forward interface-level path and time to send and acknowledge are available to analyze. However, traceroute may not return all router path information because some routers are formated to reply anomalously.

Paris traceroute improved traceroute by accommodating load balancing routers [15, 16]. The improvement provides more accurate information to paths along a network because the load balancing routers have the ability to direct information or probes along different paths and hide the reality of the path. An example of Paris traceroute is provided in Figure 2.6 from [5].

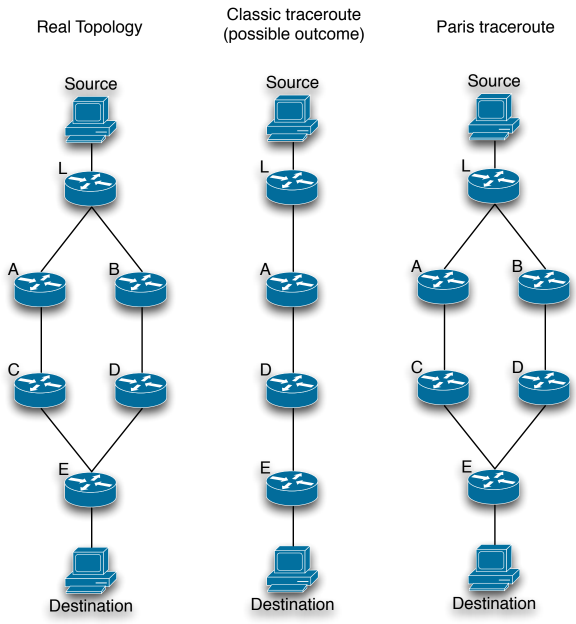


Figure 2.6: Classic versus Paris traceroute adapted from [5, 15].

Figure 2.6 illustrates how the Paris traceroute is preferred over the classic traceroute. The left diagram in Figure 2.6 provides the ground-truth of the network where router L provides load balancing across two paths. The middle diagram shows the representation of traceroute result, while the one of the right is the Paris traceroute. The Paris traceroute captures a better representation of the real topology of the network.

2.3 Existing Internet Topology Views

There are many different views to the exact structure of the Internet and how to graphically represent the Internet. In Figure 2.7, the Internet has a core of fiber-optic cables connecting ISPs to the end users, (i.e., home and business users).

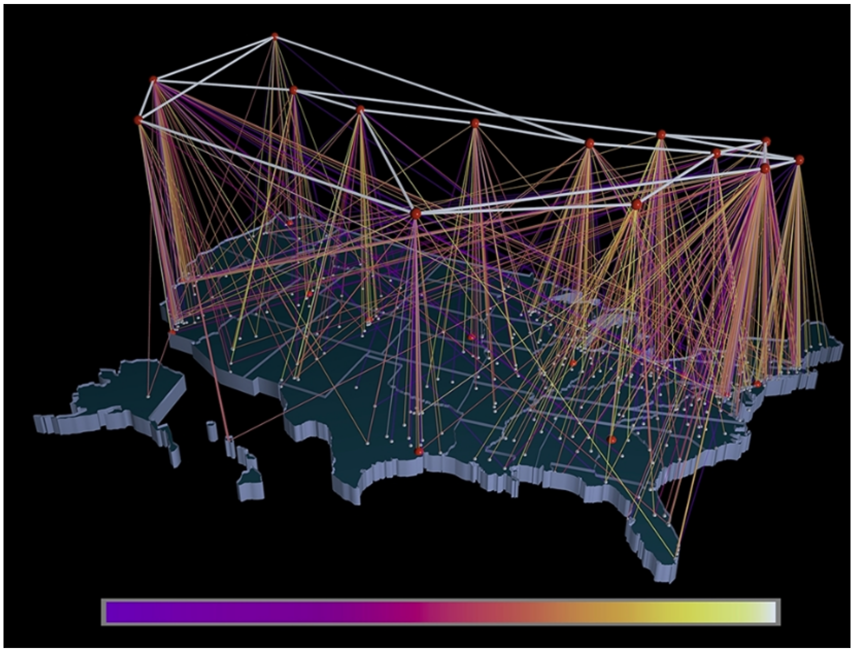
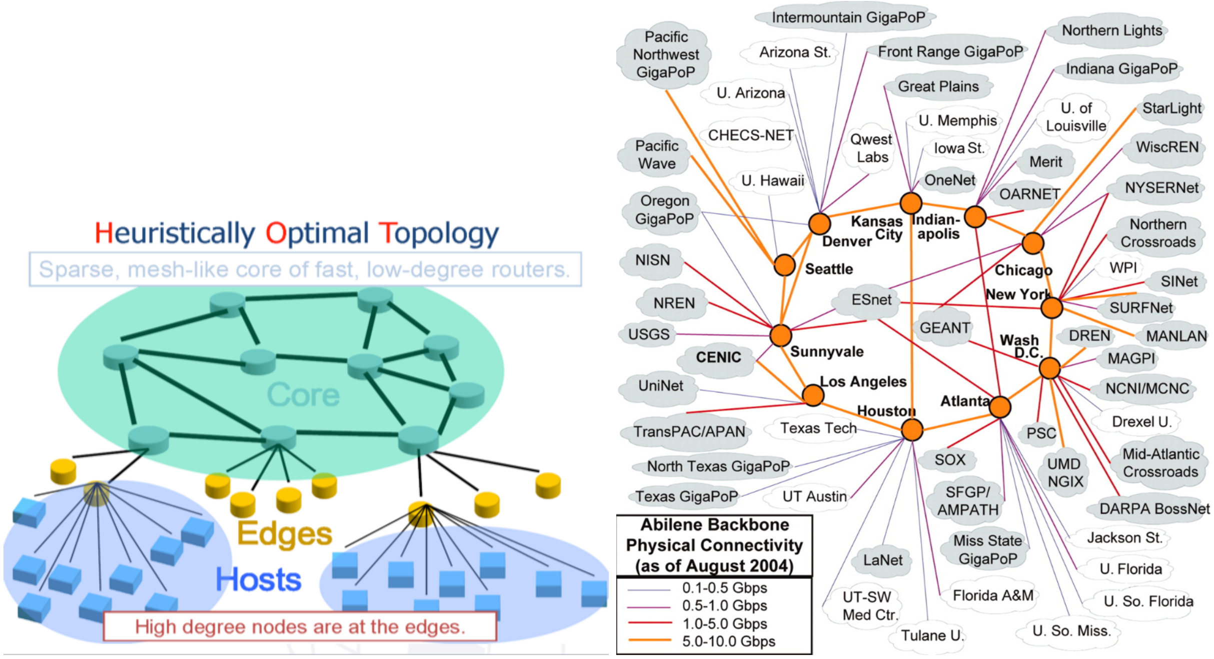


Figure 2.7: Visualization of USA Internet from [17].

Another view many researchers share is that the Internet is a heuristically optimal topology where the core is sparse with low degree routers which connect to high degree edge providers then to hosts/end users. An example of this structure is found in Figure 2.8a.

Both of the views on the Internet structure are currently accepted by researchers. We also wanted to show some previous graphical representation of the Internet from actual traceroutes. Cheswick has done considerable research in developing an algorithm to display traceroute information obtained primarily from CAIDA. An example of some of his visualizations of the Internet are found in Figure 2.9. Cheswick has named the Internet images as a "Peacock on a Windshield." Another Internet visualization example is found in Figure 2.10. This visual only



(a) Network map.

(b) Router-level topology of Abilene.

Figure 2.8: Router-level representations from [18].

displays the backbone of the Internet, but shows that even smaller subsets of the Internet are overwhelming to visually analyze.

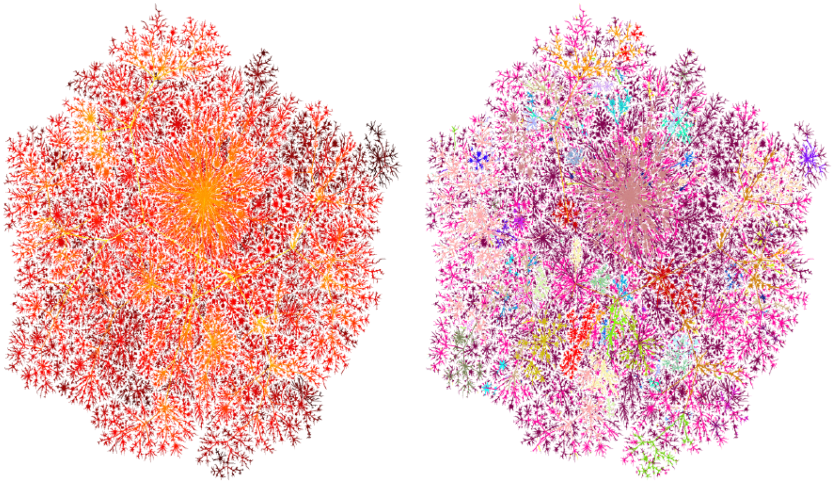


Figure 2.9: Cheswick map of Internet from [19].

We will model snapshots of the Internet by graphs to facilitate its measurement. We will use

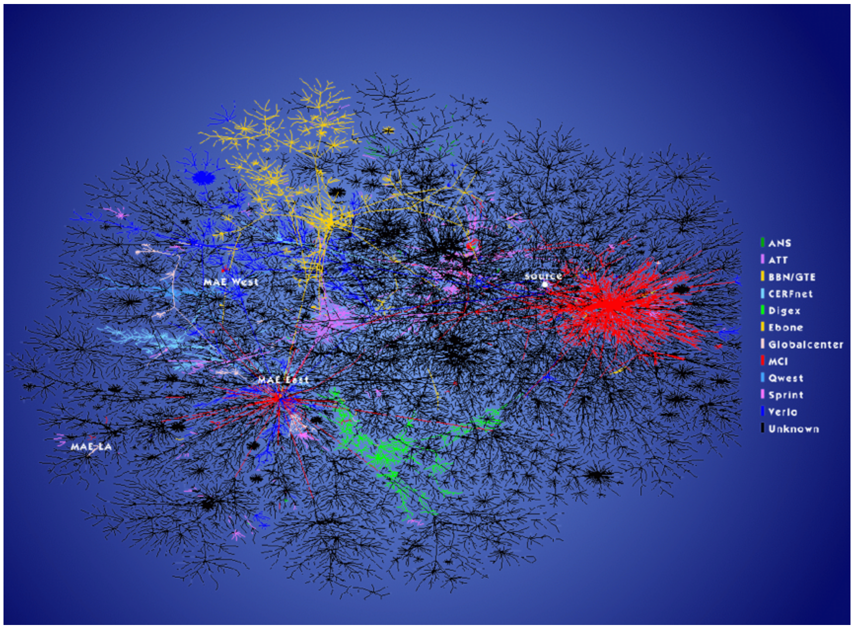


Figure 2.10: CAIDA IP map of Internet from [20].

python coding (particularly NetworkX) to analyze existing data comparison algorithms. Additionally, we will refer to CAIDA's measuring tools to compare large networks and we append additional tools. As previously mentioned, we will model the Internet at the interface-level using multiple "snap shots," with the a vertex representing an interface or IP and an edge representing a connection between two interfaces obtained from a traceroute.

CHAPTER 3: Mathematical Background

In this chapter, we introduce some mathematical theory that will create a common language as we refer to the Internet as a graph. We focus on set theory, graph theory and established complex graph comparison tools. These tools are helpful to compare large graphs because of ease of calculation and mathematically proven characteristics. Additionally, we will introduce a technique we believe will lead into understanding the stable core of the Internet.

3.1 Set Theory Terminology and Terms

A basic understanding in set theory will allow the reader to follow the results obtained from the measures of similarities and differences between graphs. We also use graph theory to understand the characteristics that large data sets exhibit, allowing further insight into the information's meaning. We accomplish this by taking the large data sets and turning them into graphs. We then run existing measures in NetworkX[5] to aid in understanding the data behind the graphs.

The following definitions for Set Theory are found in [21]. The terminology and theory are a base knowledge and reference to understand our methodology and results.

A **set** G is an unordered collection of objects, called elements or members of the set. A set is said to contain its elements. We write $a \in A$ to denote that a is an element of the set A. The notation $a \notin A$ denotes that a is not an element of the set A.

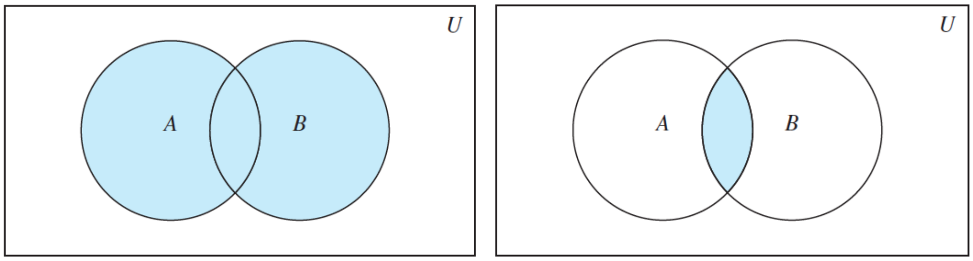
The **union**, $A \cup B$, of the sets A and B, denoted by $A \cup B$, is the set that contains those elements that are either in A or in B, or in both, see Figure 3.1.

The **intersection**, $A \cap B$, of the sets A and B, denoted by $A \cap B$, is the set containing those elements in both A and B, see Figure 3.1.

The generalized unions of a collection of sets is the set that contains those elements that are members of at least one set in the collection (e.g., Figure 3.2).

While the generalized intersections of a collection of sets is the set that contains those elements that are members of all the sets in the collection also as seen in Figure 3.2.

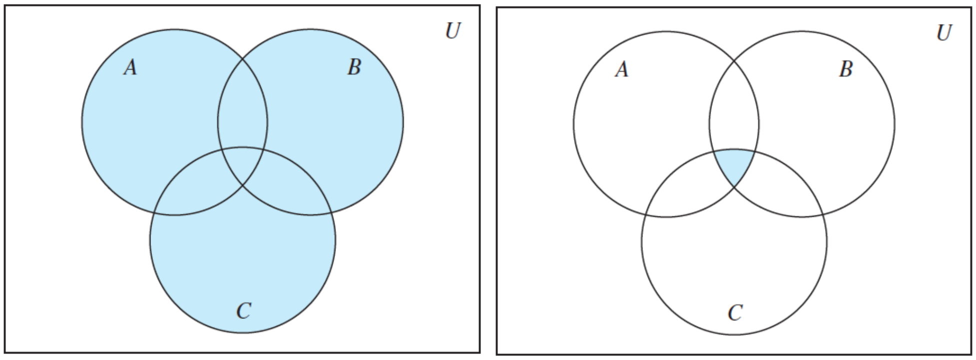
[5] Python based software package for creation, manipulation, and study of the structure, dynamics and functions of complex networks.



(a) Union of A and B.

(b) Intersection of A and B.

Figure 3.1: Union and intersection from [21].



(a) Union of A, B, and C.

(b) Intersection of A, B and C.

Figure 3.2: Generalized Union and intersection from [21].

The **Symmetric Difference of A and B**, $A \oplus B$, is the set containing those elements in exactly one of A and B, see Figure 3.3.

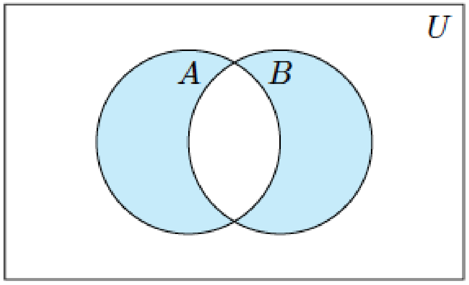


Figure 3.3: Symmetric Difference $A \oplus B$.

3.2 Graph Theory Terminology and Concepts

The below definitions and concepts are found in [22]. Any additional terminology we will individually reference.

A **graph** G consists of a finite nonempty set V of objects called **vertices** and a set E of 2-element subsets of V called **edges**, see Figure 3.4. The sets V and E are the **vertex set** and the **edge set** of G, respectively. Vertices are also called **points** or **nodes** and edges are sometimes called **lines** or **arcs**. For Figure 3.4, the vertex set is

$$V(G) = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$$

and the edge set is

$$\begin{aligned} E(G) = \Big\{ &\{c_1,c_2\},\{c_1,c_3\},\{c_1,c_5\},\{c_1,c_7\},\{c_2,c_3\},\{c_2,c_4\},\{c_2,c_7\}, \\ &\{c_3,c_4\},\{c_3,c_5\},\{c_4,c_5\},\{c_4,c_6\},\{c_4,c_7\},\{c_6,c_7\}\Big\}. \end{aligned}$$

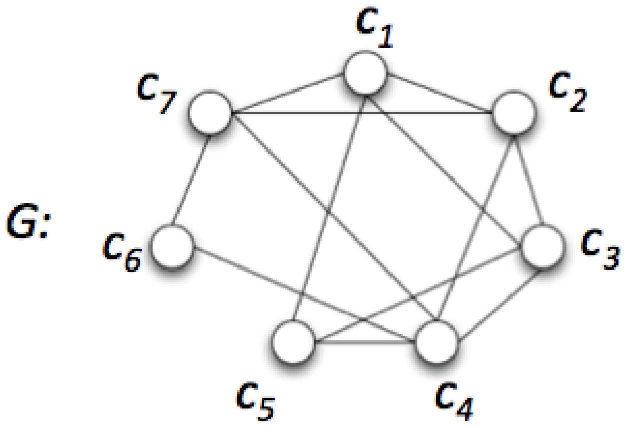


Figure 3.4: Example graph.

Node or vertex Count The vertex count of a graph G, commonly denoted $|V(G)|$ or $|G|$, is the number of vertices in G. In other words, it is the cardinality of the vertex set.

The **edge Count** of a graph G, commonly denoted $M(G)$ or $E(G)$ and sometimes also called the edge number, is the number of edges in G. In other words, it is the cardinality of the edge set.

A **path** is used to describe both a manner of traversing certain vertices and edges of G and a subgraph consisting of the sequence of those vertices and edges. A **path** is a $u-v$ walk in a graph in which no vertices are repeated. An example of a path in Figure 3.4, from c_1 to c_4 is

$$P = (c_1, c_3, c_4)$$

and it is not unique since there also exist another path.

$$P = (c_1, c_2, c_3, c_4)$$

There are actually multiple paths from c_1 to c_4.

A graph G is **connected** if every two vertices of G are connected, that is, if G contains a path $u-v$ for every pair u, v of vertices of G. A graph G that is not **connected** is called **disconnected**.

A **trail** is terminology borrowed from the Old West and defined as a $u-v$ trail in a graph G to be a $u-v$ walk in which no edge is traversed more than once. An example of a trail in Figure 3.4, from c_5 to c_2 is

$$T = (c_5, c_4, c_6, c_7, c_4, c_2)$$

and can repeat vertices as we did with c_4. A **circuit** in a graph G is a closed trail of length 3 or more. Our previous example of a trail can be a circuit, $c_5, ..., c_5$, but another example is a circuit from c_1 to c_6 is

$$C_1 = (c_1, c_7, c_4, c_2, c_7, c_6, c_1)$$

but also

$$C_2 = (c_1, c_2, c_3, c_4, c_7, c_6, c_1)$$

this shows a circuit is not unique between two vertices. A **cycle** is a circuit that repeats no vertex, except for the first and last. A $k-cycle$ is a cycle of length k, (i.e., k vertices), for example a $3-cycle$ is commonly referred to as a triangle. From Figure 3.4 an example of a $3-cycle$ is

$$C_3 = (c_1, c_5, c_3, c_1)$$

while an example of a $5-cycle$ is

$$C_4 = (c_1, c_2, c_3, c_4, c_7, c_1)$$

The **degree of a vertex** v in a graph G is the number of edges incident with v. For example in

Figure 3.4 the degree of c_7.

$$\deg\ c_7 = 4$$

For two vertices u and v in a graph G, the **distance** $d(u,v)$ from u to v is the length of a shortest $u-v$ path in G. A $u-v$ path of length $d(u,v)$ is called a $u-v$ **geodesic**. The path c_1, c_5, c_4 is a geodesic from c_1 to c_4.

For a vertex v in a connected graph G, the **eccentricity** $e(v)$ of v is the distance between v and a vertex farthest from v in G. The minimum eccentricity among the vertices of G is its **radius** and the maximum eccentricity is its **diameter**, with are denoted by $rad(G)$ and $diam(G)$, respectively. For example, $e(c_1) = 2$, $rad(G) = 2$ and $diam(G) = 2$.

Symmetric difference [23] Conventionally, if G and H are graphs with vertex set V, then the symmetric difference $G \triangle H$ is the graph with vertex set V whose edges are all those edges appearing in exactly one of G and H. Note that the $G \triangle H$ the set operation of symmetric difference is done on the edges of the graph, and it is not what we use in this research.

In this paper, we model Internet connections as an undirected graph. We do this because the nature of bi-directional information exchanged on the Internet between endpoints. Therefore, we will not use directed graphs, pseudo graphs or multi-graphs to represent the Internet from our data.

3.3 Complex Network Measures

In our analysis we use some measures that CAIDA has used in previous research papers [24,25] and have provided the definitions below. We later augment with other measures.

Average vertex degree (k), is the ratio of edges to vertices, where n is the **number of vertices** in the graph and m is the **number of edges** in the graph:

$$k = \frac{2m}{n}.$$

This is the average degree of all the degrees in the graph, and is derived from the first theorem of graph theory.[6] This is considered one of the coarsest measurements for graph comparison, but serves as an easy reference when comparing two large graphs. Additionally, a topology whose graph has a larger average vertex degree is likely to be more efficient and robust than those of

[6] The sum of degrees is twice the number of edges in the graph.

lower average vertex degree. We use v for vertices and k for the degree of the vertex.

Average clustering coefficient for a graph G is the average,

$$C = \frac{1}{n} \sum_{v \in G} c_v$$

where n is the number of vertices in G [26].

Additionally, we will examine a few other measures: the **transitivity** [27], **degree Pearson correlation coefficient** [28], **VSD** and **ESD** [5].

Similar to clustering, **transitivity** [27] computes the fraction of all possible triangles present in G. This is accomplished by identifying the total number of triangles out of all possible triads.

$$T = 3 \frac{\#\, triangles}{\#\, triads}$$

A triad is a path (see Section 3.2) on three vertices that has the possibility of being a triangle or a path of length three.

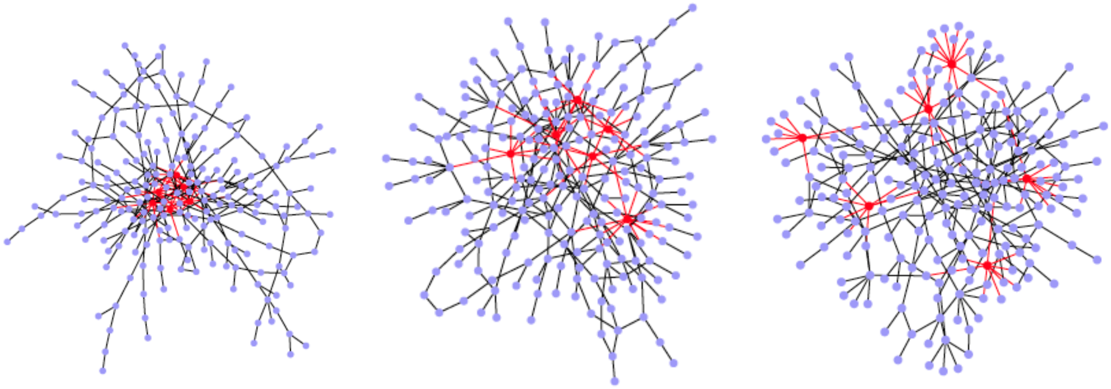
We use the **degree Pearson correlation coefficient** to compute the degree assortativity of a graph. the pearson correlation coefficient is defined as

$$r = \frac{M^{-1} \sum_i j_i k_i - [M^{-1} \sum_i \frac{1}{2}(j_i + k_i^2)]}{M^{-1} \sum_i \frac{1}{2}(j_i^2 + k_i^2) - [M^{-1} \sum_i (j_i + k_i)]^2}$$

where j_i, k_i are the degrees of the vertices at the ends of the ith edge, with $i = 1, ..., M$ [29]. This is used to measure the similarity of connections in the graph with respect to the vertex degree of the hubs. An example of each is displayed in Figure 3.5. The assortative graphs, 3.5a, show a preference of hubs to link to each other. While a disassortative graph, 3.5c, the hubs seem to avoid each other.

A recently introduced comparison [5], is the **VSD** and **ESD**, which measure the percentage of change between two graphs either in terms of the vertices or the edges. The below definitions and examples for **VSD** and **ESD** were taken from [5].

Definition 3.3.1. *For two graphs G and H, the* vertex symmetric difference $vsd(G,H)$ *is defined*


(a) Assortative

(b) Neutral

(c) Disassortative

Figure 3.5: Example graphs for degree Pearson correlation coefficient from [30].

to be:

$$vsd(G,H) = \frac{|V(G) \setminus V(H)| + |V(H) \setminus V(G)|}{|V(G)| + |V(H)|}.$$

As example, consider the two graphs in Figure 3.6, for which we have:

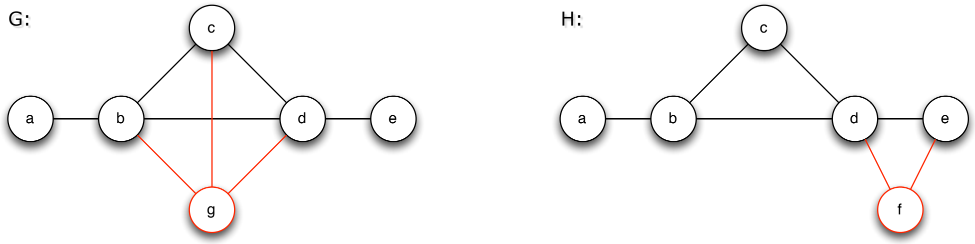


Figure 3.6: Example to illustrate *vsd* and *esd* between two graphs from [5].

$$vsd(G,H) = \frac{|V(G) \setminus V(H)| + |V(H) \setminus V(G)|}{|V(G)| + |V(H)|} = \frac{1+1}{6+6} = \frac{2}{12} = 16.7\%$$

Definition 3.3.2. *For two graphs G and H, the* edge symmetric difference *$esd(G,H)$ is defined as*

$$esd(G,H) = \frac{|E(G) \setminus E(H)| + |E(H) \setminus E(G)|}{|E(G)| + |E(H)|}.$$

Informally, we first count the edges present in one graph and not the other and then reversed.

This is then normalized over the total number of edges in both the graphs, so that it is relative to the size of the graphs.

In the case where graphs G and H have exactly the same edges, $esd(G,H) = 0$. If graphs G and H are disjoint, or have totally different edges, $esd(G,H) = 1$.

Thus, we see that the *esd* of any two graphs, G and H, lies in the closed interval $[0,1]$ (i.e., $0 \leq esd(G,H) \leq 1$).
On this scale, we are able to say, intuitively, if the difference between the two graphs (edge-wise) is significant.

For example, consider the two graphs in Figure 3.6 [5]

We have

$$esd(G,H) = \frac{|E(G) \setminus E(H)| + |E(H) \setminus E(G)|}{|E(G)| + |E(H)|} = \frac{3+2}{8+7} = \frac{5}{15} = 33.3\%$$

In this thesis, we introduce the idea of a possible **stable core**. We hypothesis that the stable core can be found by taking the generalized intersections of multiple "snapshots" of the Internet in time. We notice the stable core measurement of the network is the part that has limited change per many cycles, in our case six cycles. This part might represent more of the backbone of the ever changing Internet.

CHAPTER 4:
Data and Methodology

In this chapter, we outline our data collection procedures and explain some known limitations of our data usage. The application to the mathematical concepts will follow in Chapter 5.

4.1 Source of Data

We used data from two topology collection methodologies, NPS and CAIDA, to accrue our two sets of topology probing cycles and ran the NPS algorithm two additional probing cycles to identify trends we observed in the previous probing cycles. We ran each collection technique simultaneously[7] to negate any interference with time or network load and also used the same set of vantage points. The first set of probing cycles was run on 4 September 2013 followed by two additional pair of probing cycles on 13 December 2013. After initial results from the two sets of NPS and CAIDA probing cycles, we ran two additional NPS probing probing cycles to identify if we continued to observe certain trends that we will expand upon in Chapter 5. The dates for the probing cycles are shown in Table 4.1.

Probing Cycle	Date
NPS 1	8/31/13
CAIDA 1	9/4/13
CAIDA 2	12/13/13
NPS 2	12/15/13
NPS 3	4/4/14
NPS 4	4/9/14

Table 4.1: The dates of the corresponding probing cycles.

4.1.1 CAIDA data

CAIDA uses an active probing methodology called Archipelago (Ark), which we will also call "CAIDA probing methodology" throughout the paper. CAIDA developed Ark to:

- reduce the effort needed to develop and deploy sophisticated large-scale measurements

[7] Runs were conducted back to back switching which probing method was conducted first.

- provide a step toward a community-oriented measurement infrastructure by allowing collaborators to run their vetted measurement tasks on a security-hardened distributed platform [31].

Ark uses coordinated large-scale traceroute-based topology measurements from a process called team probing to gather measurement to all routed /24's[8] [31]. The team probing dynamically divides the IP space among three teams to provide a parallelization of all /24's destinations, and collects data for in about two to three days per team of 17-18 monitors. The teams operate independently to prove the entire IP address space. Currently, CAIDA has 86 active monitors, seen in Figure 4.1, to provide data collected in parallel, with at least one in every continent except Antarctica [32]. The Ark measurement uses *scamper*, a powerful and flexible active measurement tool which supports Internet protocol version 4 (IPv4), Internet protocol version 6 (IPv6), traceroute and ping [31]. CAIDA uses scamper because it supports transmission control protocol (TCP), user datagram protocol (UDP), and ICMP based measurements and Paris Traceroute variations [31].

Ark collects the data by sending probes continuously from random monitor vantage points, within a team, to destination IP addresses. The destinations IP address prefixes are randomly selected among the /24 space to ensure the data has representation across /24 space. Using a tool, like sc_analysis_dump tool[9] included in the scamper distribution package [33] we can extract the information needed for this research from the probing cycles. This will be analyzed in Chapter 5.

4.1.2 NPS Data

The NPS probing methodology is a Python script program with the goal of minimizing the time required to gather the network information while maximizing the number of vertices and edges discovered. The authors of the program call the technique of adaptive network mapping ingress point spreading (IPS). IPS aims to increase probing efficiency by first inferring the number of ingress points for a given network, then for each new probe, selecting the vantage point with the highest likelihood to traverse an ingress point that has not been covered before [34]. The probing methodology uses data from a prior probing cycle to infer potential ingress points at different network boundaries for each target prefix. The process is designed to

- Discover the degree of subnetting within edge networks through and iterative interroga-

[8] Approx 9.5 million addresses.

[9] Tool that produces output in textual format of each summary trace. It is included in Appendix A.

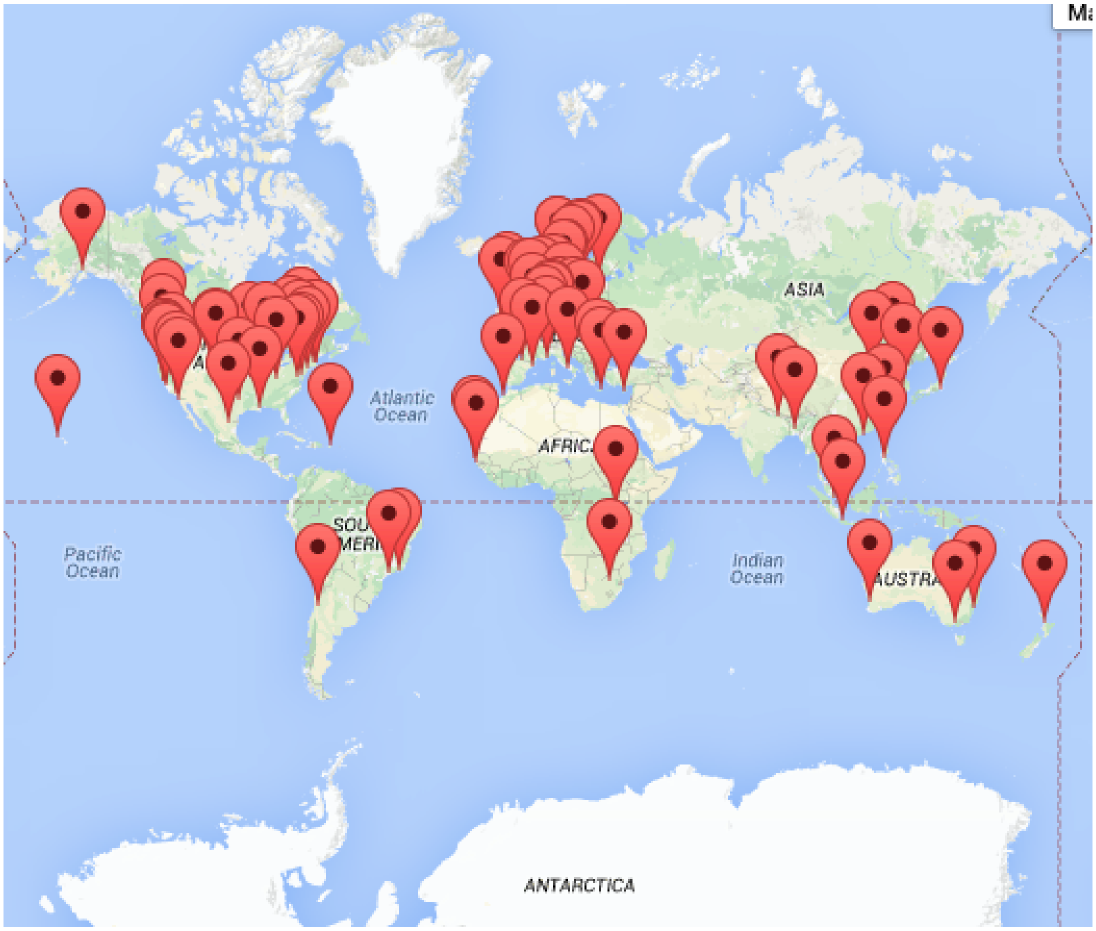


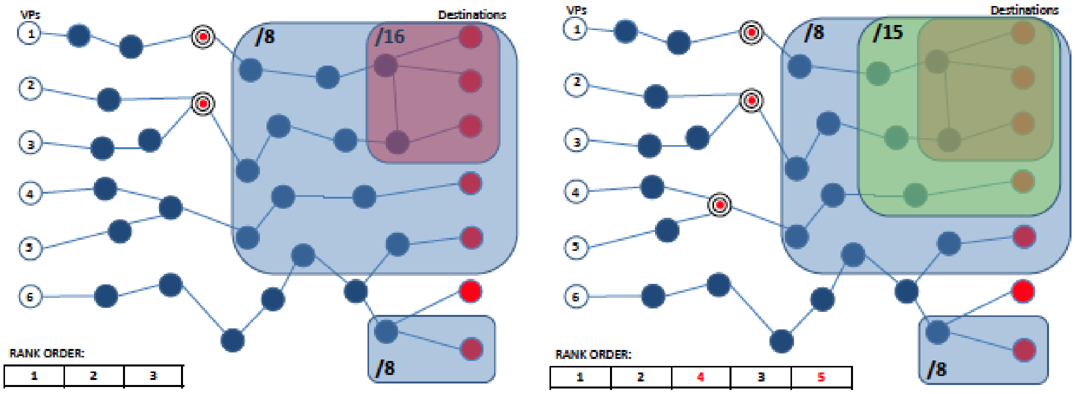
Figure 4.1: The map of all CAIDA monitor locations from [32].

tion process [34].

- Discover sources of path diversity into networks by finding and exploiting the target's ingress points [34].

In [34], the authors state that by spreading probing across ingresses it will prevent early termination and therefore discover more diverse paths. An example of IPS is illustrated in Figure 4.2. In Figure 4.2a, six previous vantage points are displayed to various destinations, (in red). The desired /16 is shaded in red in Figure 4.2a and has three destination IPs and encompasses two ingresses into the /8 that lead to paths into the /16 prefix. In Figure 4.2b, vantage points 1 and 2 are chosen as first priority in rank order list. Vantage points 1 and 2 are ranked first because each traversed a diverse ingress in Figure 4.2a.

However, IPS wants to discover a total rank order over all vantage points, typically larger than these six example vantage points to discover more. Therefore, IPS expands from /16 to /15 prefixes (green shaded box in Figure 4.2b), which includes a new ingress point from point 4. This new ingress point rank prioritizes point 4 third because vantage point 3 shared the same /8 ingress point as point 2 in Figure 4.2a. The IPS probing methodology continues this ranking through /14, /13, etc. until all vantage points are rank ordered.



(a) Target /16 prefix with two ingresses.

(b) Expansion to find notational ingresses.

Figure 4.2: IPS example of six vantage points from [34].

4.2 Data Selection and Preparation

In Section 4.1, we described the two different probing methodologies and how each discovers the Internet. We explained the algorithms to help the reader understand the difference in the probing methodologies. We now explain to the reader the preparation of each probing cycle, the approximate collection time and the time it takes to allow a computer to analyze the information via our measures.

4.2.1 Preparation

The output from sc_analysis_dump tool[10] is a list of traceroute like data. Specifically, the data provides the information that resembles a classic IP trace as well as a round-trip time (RTT) of packets sent and received between routers along the way. We take the data from the sc_analysis_dump, strip the RTT information and keep the interface data. This provides a list of interfaces and the links between them (given by the sequence of IPs as in Figure 4.3).

[10] A tool used to provide a list of traceroute data in a readable script. Each line contains information about the each single trace to include interfaces visited and time between transmission. Details of the output are seen in Appendix A.

We then use the router interface as the vertex and identify two consecutive IPs which will be represented by an edge. This is easily done allowing us to string the interfaces together in order, from the first vertex to the second via an edge and the second to third via another edge in sequence to represent the path taken for each trace. We continue this process through each trace in the probing cycle to obtain several paths forming a union and discarding edge and vertex multiplicity. From the union of the paths we build a graph representing a "snapshot" of the Internet as discovered by each probing methodology. This technique allows us to compare the graphs by the various methods described in Chapter 3. It is important to note that all interfaces respond with their address, due to security settings, and we discard all "q" responses. We have suggested additional future work to identify the amount discarded and possible way of assigning the unknown routers to a geographical set, in Section 6.2

4.2.2 Resources "Time"

We previously discussed the three or more days needed to obtain a complete CAIDA cycle that we call a "snapshot" of the Internet. We also briefly explained how one of the main goals of the NPS probing methodology was to discover more of the Internet's topology in less time. In order for us to best compare CAIDA's and NPS's probing methodologies, we first compared two probing cycles of each methodology collected as near simultaneously as possible using the same monitors, (in order to negate any difference due to the time collected or traffic load on the Internet). This allowed us to better compare the two probing cycles to each other. Each set of probing cycles took a day to two days to collect, then another day to two to analyze. We were able to accomplish this by using a dedicated server to run the compiling, parsing and test continuously over the four to five day period. We also ran a few additional tests, specifically diameter and radius testing per country, which took nearly two weeks of dedicated time on the server per cycle.

4.2.3 Probing Data Challenges

The Internet is ever changing, making it difficult to predict and accurately map. Additionally, not all the routers probed respond, making it difficult to know exactly which routers are captured. During both CAIDA and NPS probing methodologies, we would receive a "q" for each unresponsive router, as seen in Figure 4.3. This meant our probe would identify a router's existence but not its IP address. Unresponsive routers occur because many companies and users increase the security settings on their routers to block ICMP traffic. This makes obtaining accurate information increasingly difficult since in our research, we discard the routers with a "q."

Since we discard the unresponsive routers and therefore their interfaces, some of the Internets subsets appear disconnected, when they really are connected.

```
203.181.248.60        203.181.248.60        203.181.248.60
203.181.249.21        203.181.249.21        q
203.181.102.129       203.181.102.129       203.181.102.129
118.155.197.1         118.155.197.1         118.155.197.1
203.181.100.126       203.181.100.126       203.181.100.126
59.128.2.210          59.128.2.210          59.128.2.210
65.19.143.9           65.19.143.9           65.19.143.9
72.52.92.37           72.52.92.37           72.52.92.37
72.52.92.233          72.52.92.233          72.52.92.233
216.66.77.102         216.66.77.102         216.66.77.102
209.152.158.18        209.152.158.18        209.152.158.18
q                     209.152.158.18
q
q
209.152.158.18
```

Figure 4.3: Comparison of traceroutes with same source and destination addresses with "q" response from [5].

Additionally, we used MaxMind geoIP lite database[11] for geographic reference data base for IP addresses. This is problematic because there is not a method for us to confirm the exact location assigned to each IP address.

4.3 Methodology

In the current thesis, data from two probing methodologies (CAIDA vs. NPS) was obtained with the goal of using graph comparisons to differentiate their outcomes. We ran general standard statistics on multiple cycles of each of the two probing methodologies, as well as the two recently introduced metrics of VSD and ESD.

Additionally, we partitioned the data (from the probing cycles) into geographical countries to identify the locations of the additional vertices and edges discovered by NPS probing methodology and not discovered by CAIDA probing methodology. We also analyzed the possibility of

[11] The exact method of how the database is populated is proprietary, but is known to do worse on router interfaces than client addresses.

any bias between the probing cycles to identify if monitor location caused the increase in found vertices and edges in some countries.

This data was then analyzed by reapplying the same statistics per country, as well as a graphical analysis in Gephi[12], VSD and ESD, and stable core intersection since these graphs were much smaller.

[12] A visual interactive graph modeler and analyzer.

THIS PAGE INTENTIONALLY LEFT BLANK

CHAPTER 5:
Results

In this chapter, we explain our findings of the two topology probing methods using the predefined comparison methods from [24]. We first examine each probing method's data as a comprehensive graph, then sub-divide the graph into geographic countries to identify if there is bias towards a country. We also explore some possible pitfalls when using certain methods to ensure awareness when comparing the graphs.

5.1 General Topological Comparison Between NPS Probing Method and CAIDA Probing Method

We analyzed our data holistically using the existing measures we explained in 3.2 and 3.3 whose validity was addressed in [25]. In Table 5.1, we identify that the NPS probing method discovers more vertices and edges compared to the CAIDA probing method. We also observe that the average vertex degree, for the first two probing cycle sets, are slightly less for the NPS probing method, but not significantly. We believe this was attributed to the discovery of 12 percent more vertices and only 10 percent more edges, which will decrease the average degree. However, after two additional NPS probing cycles, the average vertex degree increases and is greater than the first two CAIDA probing cycles meaning additional probing cycles are needed. We then compared the average clustering coefficient to identify if it implies difference in the structure of the data obtained from the two probing methods. Our data, in Table 5.2, does not support a significant difference, meaning that the structure discovered by the two probing methods are comparable to each other. We explored the Pearson coefficient to identify any difference between how the two probing methods relate degree correlations, because the clustering coefficients were similar. We discovered that the Pearson coefficient for NPS's probing method was negative, but small, which means the graph is slightly disassortative. CAIDA's probing method had a positive Pearson coefficient, but also very small, which would mean a slightly assortative graph. However, we cannot draw any statistically significant conclusions from our data to the meaning of these results, rather conjecture that NPS probing methodology is slightly disassortative while CAIDA probing method remains slightly assortative. As future work, we recommend the study of additional probing cycles and their comparisons in order to be able to draw significant conclusions.

	Number Vertices	Number Edges	Average Node Degree
NPS 1	524,366	1,360,855	5.190
CAIDA 1	466,072	1,236,530	5.306
NPS 2	520,906	1,333,079	5.118
CAIDA 2	464,553	1,202,778	5.178
NPS 3	543,073	1,441,963	5.310
NPS 4	554,753	1,470,550	5.302

Table 5.1: The basic statistics for each probing cycle.

	Average Cluster Coefficient	Pearson Coefficient
NPS 1	0.017	-0.034
CAIDA 1	0.017	0.021
NPS 2	0.016	-0.033
CAIDA 2	0.017	0.039
NPS 3	0.020	-0.032
NPS 4	0.019	-0.035

Table 5.2: The connectivity table for each probing cycle.

5.1.1 Dissimilarity Measures

We compared VSD and ESD to evaluate each edge and vertex per graph to determine how similar the graphs are per probing cycle between NPS and CAIDA probing methods. In Table 5.3, we present the VSD between all probing cycles of NPS and CAIDA probing methods. We identify around 40 percent different interfaces/vertices between the first two NPS and CAIDA probing cycles, which were collected nearly simultaneously, see Table 4.1. This confirms the two probing methods discover different sets of IP of the Internet, independent of time collected. Table 5.3 also displays the difference between NPS 3 and NPS 4 probing cycles, which were taken nearly four months apart, see Table 4.1. This seems to confirm the ever changing Internet. Interestingly, the VSD between NPS 1 versus NPS 2 is nearly seven percent less then CAIDA 1 versus CAIDA 2, (recall NPS 1 and CAIDA 1 at the same time, as was NPS 2 and CAIDA 2). We attribute the small difference to the randomness in target selection within /24's for CAIDA versus the NPS directed strategy. Furthermore, the difference between all pairwise NPS' probing cycles are smaller than they are to CAIDA's probing cycles. This could mean that NPS' probing method discovers more stable interfaces/vertices than CAIDA's probing method. We will address additional possible stable core results in Section 5.1.2.

The ESD between the probing cycles displays the edges/paths that the interfaces/vertices used to transmit the packet information between routers. Interestingly, there is nearly a 50 percent

	NPS1	NPS 2	NPS 3	NPS 4	CAIDA 1	CAIDA2
NPS 1	0.00	0.25	0.34	0.34	0.37	0.43
NPS 2	0.25	0.00	0.26	0.26	0.43	0.37
NPS 3	0.34	0.26	0.00	0.10	0.48	0.43
NPS 4	0.34	0.26	0.10	0.00	0.48	0.44
CAIDA 1	0.37	0.43	0.48	0.48	0.00	0.32
CAIDA 2	0.43	0.37	0.43	0.44	0.32	0.00

Table 5.3: The values of VSD per probing cycle.

difference in the edges the packets took between the probing cycles comparing the two probing methodologies, shown in Table 5.4. The difference appears to hold regardless of the probing method. There is a slight outlier between NPS 3 and NPS 4 probing cycles which only differ by 24 percent. We attribute this to the small time frame between the same probing methodology (four days between NPS 3 and NPS 4 probing methods versus the nearly four months between NPS 1, NPS 2, NPS 3/4, CAIDA 1 and CAIDA 2 probing cycles).

	NPS 1	NPS 2	NPS 3	NPS 4	CAIDA 1	CAIDA 2
NPS 1	0.00	0.43	0.54	0.54	0.41	0.54
NPS 2	0.43	0.00	0.44	0.44	0.54	0.42
NPS 3	0.54	0.44	0.00	0.24	0.62	0.54
NPS 4	0.54	0.44	0.24	0.00	0.62	0.55
CAIDA 1	0.41	0.54	0.62	0.62	0.00	0.46
CAIDA 2	0.54	0.42	0.54	0.55	0.46	0.00

Table 5.4: The values of ESD per probing cycle.

5.1.2 Commonality Measures

This subsection introduces the idea of measuring a stable core, by which we mean the part of the Internet that is discovered by the majority of the probing cycles, see Section 3.3. We used intersections of probing cycles to discover the graph (vertices and edges) that all probing cycles discovered. The basic statistics of the possible stable core are found in Table 5.5. We predicted that the number of vertices and edges would decrease to a smaller subset during every iteration, which our analysis supports. Interestingly, the vertex and edge counts still remain relatively large, about 40 percent of the original graphs. This shows promising evidence that there exist a relatively stable core, but we need to compare additional probing cycles before we can confirm. We believe that with this knowledge we could possibly identify the preferred crucial Internet infrastructure that is important to the speed of the Internet we currently enjoy and identify those paths that are critical for vulnerability assessments.

	Node Count	Edge Count	Avg Degree	Clustering	Pearson
Both CAIDA Graphs	316,422	659,788	4.170	0.011	0.084
First Two NPS	392,791	761,599	3.878	0.008	-0.025
Intersection of NPS 1-2 and CAIDA 1-2	233,607	432,412	3.702	0.007	0.083
Intersection NPS 1-2, CAIDA 1-2 with NPS 3	204,860	334,689	3.267	0.005	0.092
Intersection NPS 1-2, CAIDA 1-2 withNPS 4	204,508	333,928	3.266	0.005	0.094

Table 5.5: The graph statistics for the intersection of graphs.

We tested our conjecture of the existance of a stable core using the intersection of the first four graphs, (NPS 1, NPS 2 and CAIDA 1 and CAIDA 2) and tested them on the last two probing cycles, (NPS 3 and NPS 4). In Table 5.5, the test on both NPS 3 and NPS 4 both have a vertex count difference of only 352 vertices/IPs with only an edge count difference of only 761. This is significant since there are over 500,000 vertices and over 1,200,000 edges for each cycle. This shows that the intersections of the graphs over time might lead to the discovery of a stable core. The researchers recommend additional future probing cycles to test to identify the rate of change and determine the number of probing cycles to have statistical data to ensure validity to our findings. We will also test the stable core conjecture on smaller country graphs in Section 5.2.3.

5.2 Per Country Topological Comparison Between NPS and CAIDA Probing Methods

From our findings in Section 5, we parsed the cycles to identify the countries each traversed to determine if bias existed. In Figure 5.1[13], both NPS and CAIDA probing methods discover the same top 10 countries for vertex and edge count, but NPS' probing method found significantly more IPs in the United States, China and South Korea than CAIDA's probing method. In Figure 5.2, NPS probing method also did better at discovering more edges in the United States, China and significantly better in South Korea. It makes sense for a probing method to probabilistically find more vertices and edges in China and the United States because both have more users than any other country [2]; see Table 5.8 and discussion in Section 5.2.1 we will further discuss. It is worth mentioning that there are not any monitors in China. That is, we analyzed the vantage point monitors, seen in Figure 5.3, to see if there was any discovery bias

[13] The x-axis is by country aligned per probing iteration.

from monitor vantage point. Our data, from the monitor vantage point, does not support any monitor bias because they used nearly the same monitor vantage points with little effect on IP discovery per probing cycle.

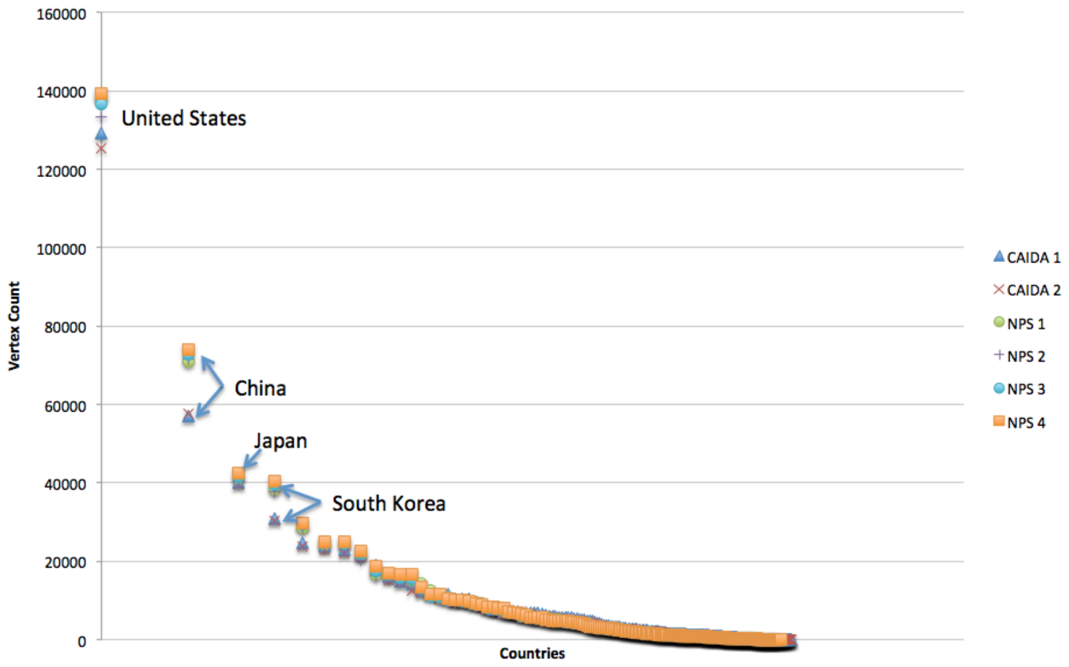


Figure 5.1: The vertex count per country.

We compared the clustering coefficient to test connectivity of the different cycles at the country level. In Figure 5.4, each probing method follows generally the same slope. The graph is skewed a little for NPS probing method because it does not find as many countries as CAIDA probing. Notice, however that NPS probing method still finds the same amount of connected countries. In fact, NPS found 23 percent of countries with a clustering coefficient of zero, while CAIDA finds 27 percent of countries with a clustering coefficient of zero. This means NPS probing method may actually be better at finding connectivity than CAIDA probing method.

Next, we compared highest and lowest country clustering coefficient to determine which countries were more strongly connected. We conjectured the clustering coefficient might identify if there was any trend to security or censorship. We first hypothesized that a country with a high clustering coefficient would show less censored countries, while a country with a low clustering coefficient would mean more censorship. We conjectured this because a country with a

	CAIDA 1	CAIDA 2	NPS 1	NPS 2	NPS 3	NPS 4
United States	129,161	125,379	138,127	133,330	136,636	139,258
China	57,099	57,528	70,892	71,687	72,889	74,013
Japan	40,211	39,653	41,177	40,273	41,514	42,364
South Korea	30,889	30,377	38,283	37,704	39,547	40,371
Germany	24,571	23,679	28,423	28,437	29,320	29,843
Great Britain	23,781	22,776	24,475	23,416	24,962	24,950
Brazil	22,904	21,997	24,377	22,927	24,491	25,032
Canada	22,044	21,156	21,815	20,298	22,097	22,672
Russian Federation	19,103	18,023	16,772	15,142	16,756	17,031
Italy	16,108	15,293	16,621	15,267	15,769	16,710
Netherlands	12,336	11,644	16,350	16,216	17,678	18,591

Table 5.6: Sample of the top vertex count by country.

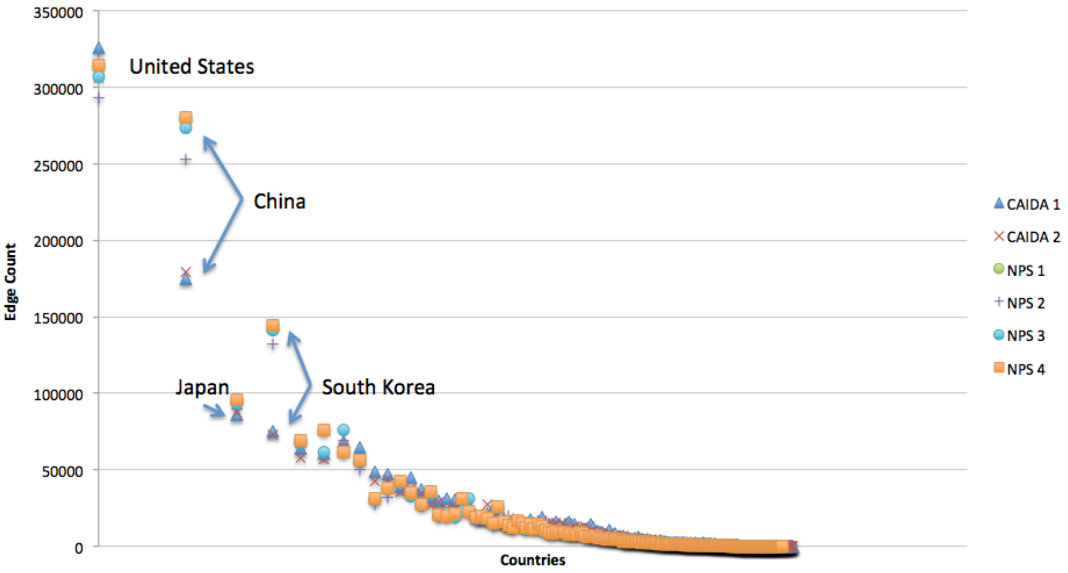


Figure 5.2: The edge count per country.

low clustering coefficient will show less of the observed Internet or only the external routers leading to paths. Unfortunately, our data did not support our hypothesis because our data did not recognize known heavily censored countries. Additionally, we observed often low clustering coefficient countries were small island countries, that happen to have few vertices and few edges, hence the presence of a few triangles (3-cycles) made the clustering coefficient large.

	CAIDA 1	CAIDA 2	NPS 1	NPS 2	NPS 3	NPS 4
United States	325,872	306,171	314,434	292,848	306,988	314,289
China	174,641	179,043	239,681	253,089	273,374	280,173
South Korea	75,106	73,005	133,237	131,953	141,314	144,373
Japan	86,226	87,284	87,437	89,684	93,395	95,700
Brazil	68,526	64,707	74,115	69,335	76,316	76,224
Germany	63,569	58,029	68,667	64,492	67,361	69,180
Great Britain	60,580	56,948	64,253	59,939	61,600	61,430
Canada	64,805	56,432	59,141	50,198	55,732	56,580
Italy	47,068	43,011	42,475	38,837	40,879	42,502
Spain	44,646	37,571	39,064	32,272	34,504	35,738
Russian Federation	48,455	42,525	38,778	32,849	37,705	38,176
Sweden	30,414	25,742	33,919	32,295	33,119	35,144
France	38,623	35,852	31,458	26,864	26,643	27,285
Netherlands	31,746	26,974	31,308	27,757	30,494	31,418

Table 5.7: Sample of the top edge count by country.

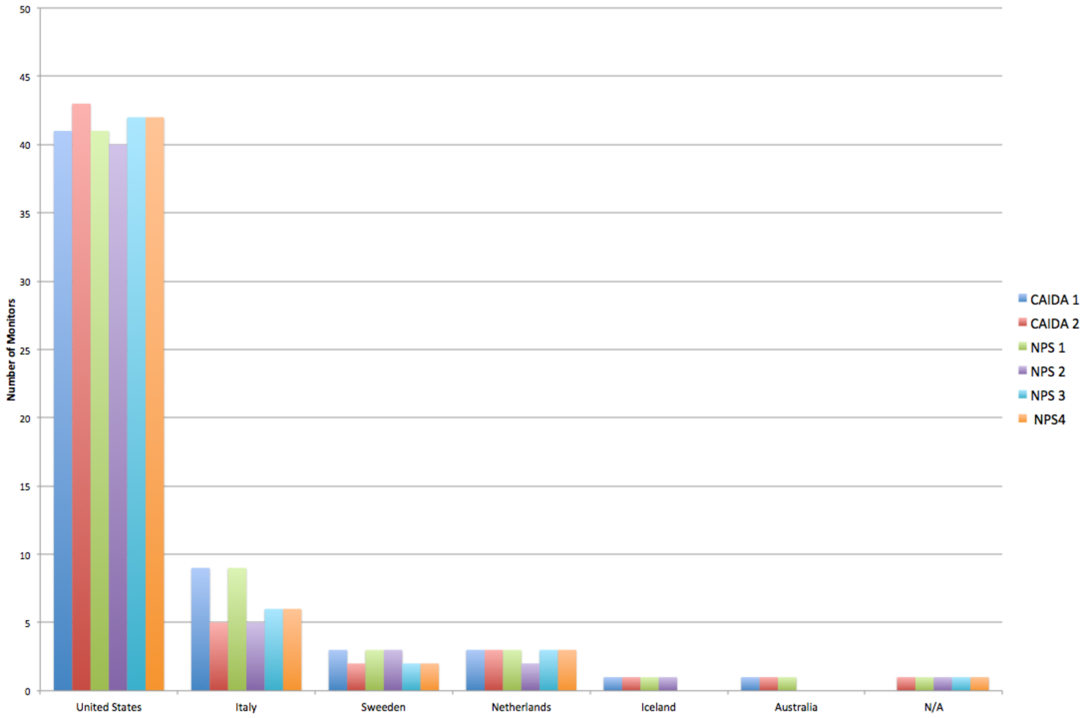


Figure 5.3: The unique monitor location per probing cycle.

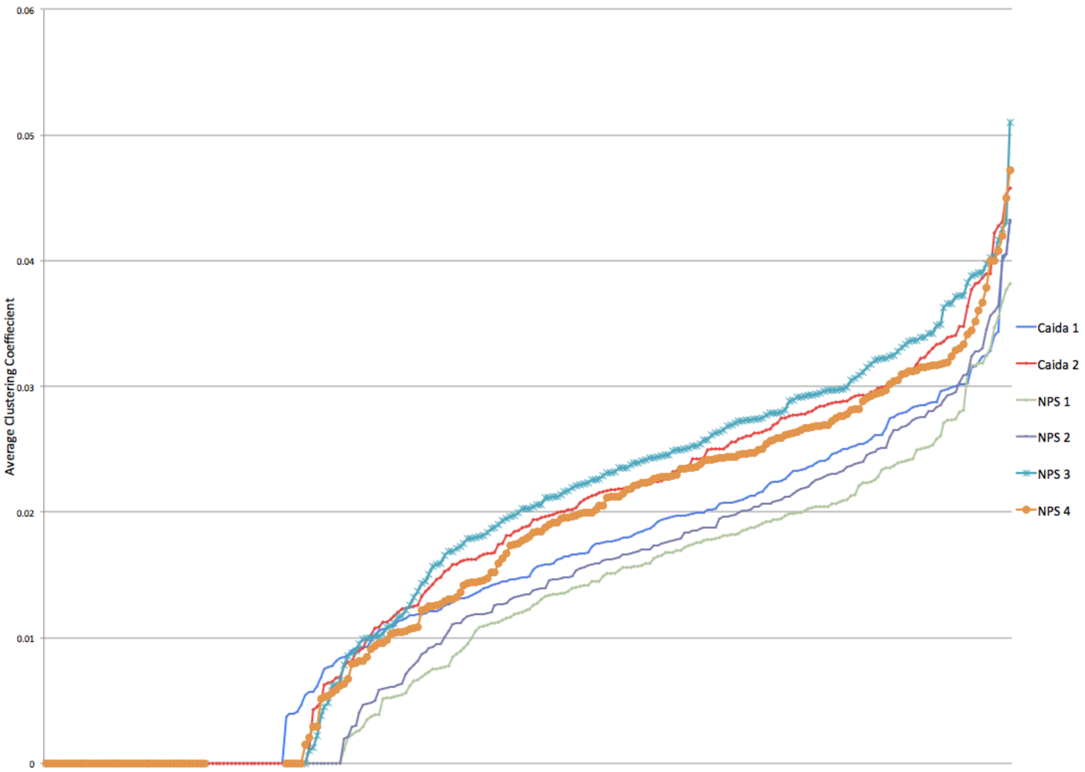


Figure 5.4: The clustering coefficient distributions by country.

5.2.1 Country IP Allocation per Graph

We previously noted the significantly larger number of vertex and edge counts of the NPS probing method for discovering IP addresses within China and South Korea, but still have not identified the reason. We already tested the monitor location in Section 5.2 without any results. This led us to analyze known allocated IP space to identify any trends. Table 5.8 displays the assigned sorted IP space percentage per country to what we discovered during each probing cycle.[14] We normalized each country's probing cycle discovery by the number of total vertices to have a method of comparison. We did this because the assigned IP space percentages are done in a similar fashion by dividing the total assigned IP space for each country by the total possible IP space. In Table 5.8, we see the NPS and CAIDA probing methodologies match the known percentages of a country's allocated IP space. Interestingly, none of the probing cycles discover the 35 percent that the United States has assigned. However, in all seven cycles the United States had the largest discovered IP space. A possible reason for this is the number of unresponsive routers, "q" as discussed in Section 4.2.3, that were possibly returned within

[14] List of all countries found Appendix B.

the United States. We would suggest additional research in this area to identify probabilistic IP/vertex location from the returned "q" for better representation.

	Assigned IP Space	CAIDA 1	CAIDA 2	NPS 1	NPS 2	NPS 3	NPS 4
United States	35.90%	14.41%	14.72%	16.18%	16.58%	16.07%	16.00%
China	7.70%	6.37%	6.75%	8.30%	8.91%	8.57%	8.50%
Japan	4.70%	4.49%	4.65%	4.82%	5.01%	4.88%	4.87%
Great Britain	2.90%	2.65%	2.67%	2.87%	2.91%	2.94%	2.87%
Germany	2.80%	2.74%	2.78%	3.33%	3.54%	3.45%	3.43%
South Korea	2.60%	3.45%	3.57%	4.48%	4.69%	4.65%	4.64%
France	2.20%	1.66%	1.71%	1.69%	1.68%	1.55%	1.55%
Canada	1.90%	2.46%	2.48%	2.56%	2.52%	2.60%	2.60%
Italy	1.20%	1.80%	1.79%	1.95%	1.90%	1.85%	1.92%
Brazil	1.10%	2.56%	2.58%	2.86%	2.85%	2.88%	2.88%
Australia	1.10%	1.28%	1.33%	1.18%	1.17%	1.17%	1.14%
Netherlands	1.10%	1.38%	1.37%	1.91%	2.02%	2.08%	2.14%
Russian Federation	1.00%	2.13%	2.12%	1.96%	1.88%	1.97%	1.96%
Taiwan	0.80%	1.39%	1.40%	1.12%	1.11%	1.16%	1.15%
India	0.80%	1.15%	1.10%	1.10%	1.01%	1.03%	1.02%

Table 5.8: The top 15 countries of IP space allocation.

5.2.2 Some results on the diameter and radius for the country graphs

We analyzed the data to identify if one probing method was better at discovering depth versus breadth, previously discussed in Section 3.2. We did this by subtracting the radius from the diameter for each graph. Recall from 3.2, the diameter was the longest shortest path, while the radius was the smallest eccentricity. Therefore, the higher difference between the diameter and the radius, would indicate better breadth into a network, while a smaller number would indicate better depth. Unfortunately, to compute the radius and diameter each graph must be connected. This was a problem when we parsed our data to the country level, because nearly a third of the countries were not connected due to unresponsive routers or other reasons. In Figure 5.5, we can see that there is a fairly even distribution with the mean diameter minus radius, $\mu = .37$

and most of the countries falling within two standard deviations, (the dashed lines). We also annotated the third standard deviation with a solid line and identified the outliers by name. Interestingly, the outliers were all island nations. However, we were still able to compare the remaining data and we observed that NPS discovered eight percent more breadth than CAIDA.

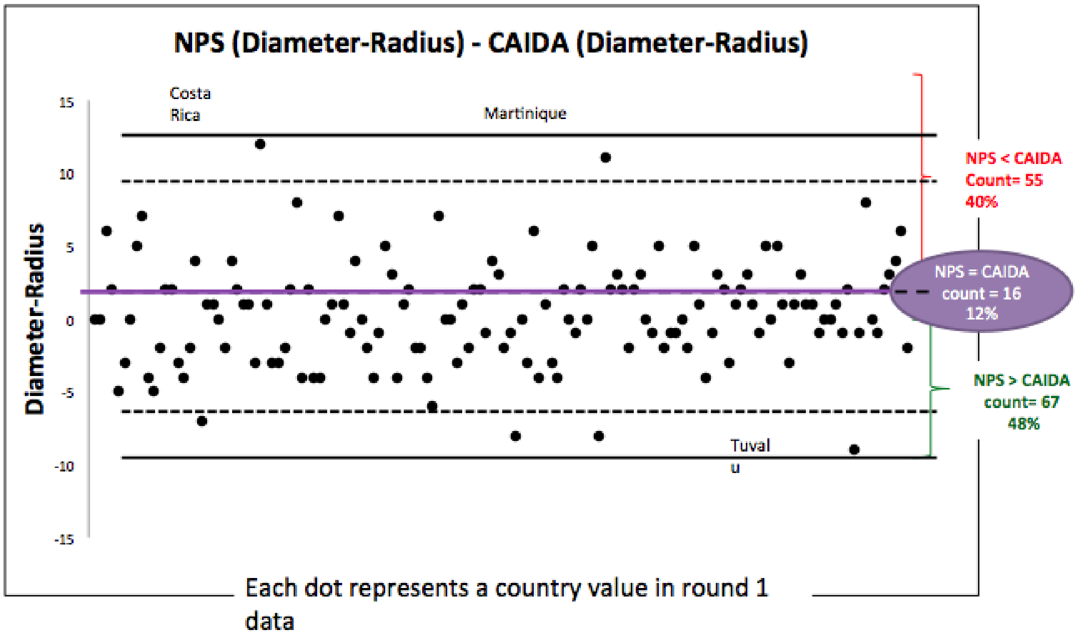


Figure 5.5: The diameter minus the radius (depth versus breadth).

5.2.3 Case Studies for some Countries

From our results in Section 5.2, we chose to further analyze China, and South Korea. We also will display an issue we discovered when comparing the clustering coefficients per country, specifically with Wallis and Futuna.

Case Study: China

The NPS probing methodology discovered significantly more vertices and edges than the CAIDA probing methodology, as seen in Table 5.6 and Table 5.7. We further explored the data to determine if the monitor vantage point impacted the results. In Figure 5.6, we notice that the probing algorithms used nearly the same monitor vantage points for each probing cycle. We determine from this data that the monitor vantage point has little effect on the discovery since nearly the same monitor vantage points were used. It is important to note that generally vantage point does have a major impact on discovered topology, but our data did not.

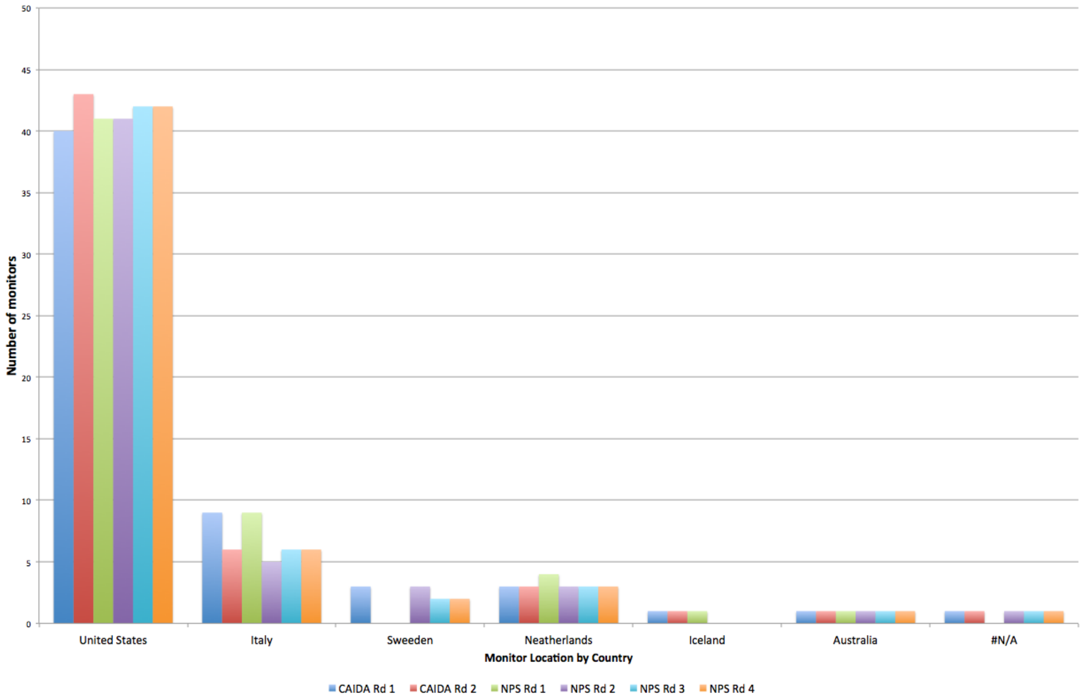


Figure 5.6: The probing monitors for China.

Table 5.9 displays the results of the VSD. We identified an average change in vertices of 30-50 percent appear between CAIDA 1 and the NPS probing methods. Interestingly the difference between NPS 1 and CAIDA 1 are 30-52 percent different even though they were probed nearly simultaneously. This further shows how different the probing methods are at discovering the Internet. In Table 5.10, the ESD is highest between CAIDA 1 and the last two NPS probing methods. These results show how the different probing methodologies discover different sets of IPs even in smaller subgraphs. It also confirms the ever changing presence of the Internet.

	NPS 1	NPS 2	NPS 3	NPS 4	CAIDA 1	CAIDA 2
NPS 1	0.00	0.25	0.42	0.42	0.30	0.52
NPS 2	0.25	0.00	0.34	0.34	0.36	0.29
NPS 3	0.42	0.34	0.00	0.10	0.57	0.50
NPS 4	0.42	0.34	0.10	0.00	0.57	0.50
CAIDA 1	0.30	0.36	0.57	0.57	0.00	0.30
CAIDA 2	0.52	0.29	0.50	0.50	0.30	0.00

Table 5.9: The VSD per graph for China.

We also tried to discover the existence of a stable core within China. In Table 5.11, we conducted analysis on the first four graphs (first two NPS and first two CAIDA) as learning and

	NPS 1	NPS 2	NPS 3	NPS 4	CAIDA 1	CAIDA 2
NPS 1	0.00	0.51	0.61	0.61	0.41	0.63
NPS 2	0.51	0.00	0.50	0.50	0.53	0.42
NPS 3	0.61	0.50	0.00	0.20	0.69	0.59
NPS 4	0.61	0.50	0.20	0.00	0.70	0.60
CAIDA 1	0.41	0.53	0.69	0.70	0.00	0.49
CAIDA 2	0.63	0.42	0.59	0.60	0.49	0.00

Table 5.10: The ESD per graph for China.

comparing on the remaining two NPS probing cycles (NPS 3 and NPS 4). We identified that the intersection of the first four graphs might have identified a possible stable core because when we compared to NPS 3 and NPS 4 graphs, the difference was only ten vertices and less than a thousand edges. An additional interesting insight is how quickly China went from an average vertex count of approx 60,000 (in Table 5.6) to only 5,000 then 2,500 (in Table 5.11). This could show how volatile China's Internet topology is, especially when one compares the results to those of South Korea in Section 5.2.3. We hypothesis this may be caused by Chinas restrictive Internet, but further work is needed to confirm.

Graphs	Node Count	Edge Count	Avg Degree	Clustering	Pearson
Both CAIDA Graphs	5,083	7,704	3.031	0.011	0.027
First Two NPS	4,722	6,091	2.580	0.015	-0.086
2 CAIDA vs 2 NPS	3,158	4,175	2.644	0.012	0.020
Intersection NPS 1-2, CAIDA 1-2 with NPS 3	2,564	3,013	2.350	0.011	0.044
Intersection NPS 1-2, CAIDA 1-2 with NPS 4	2,554	2,987	2.339	0.012	0.053

Table 5.11: The basic stats of China's stable core.

South Korea

As well, South Korea was interesting because of the difference in discovered vertices and edges from the NPS and CAIDA probing methodologies. Table 5.12 displays the VSD results from the probing cycles. The trend continues in the difference between the greatest change between the CAIDA and the NPS probing methodologies. In Table 5.13, the ESD shows a large difference in those edges that are detected per probing cycle, but not as much as China. Again, the large VSD and ESD between cycles confirms the ever changing inferred topology. When we conducted the stable core testing against the probing cycles for South Korea, the vertex and edge counts of the intersection graphs only decrease by approx 40 percent, from average of 37,000 vertices (in Table 5.6) to approximately 15,000 vertices (in Table 5.14). This is intriguing because we observed a large decrease in the graph to stable core graphs (compared that found in Table 5.6

and Table 5.7) of China but not in South Korea. The results might show that South Korea has a more stable Internet.

	NPS 1	NPS 2	NPS 3	NPS 4	CAIDA 1	CAIDA 2
NPS 1	0.00	0.22	0.30	0.31	0.31	0.38
NPS 2	0.22	0.00	0.24	0.24	0.36	0.30
NPS 3	0.30	0.24	0.00	0.10	0.41	0.37
NPS 4	0.31	0.24	0.10	0.00	0.41	0.37
CAIDA 1	0.31	0.36	0.41	0.41	0.00	0.31
CAIDA 2	0.38	0.30	0.37	0.37	0.31	0.00

Table 5.12: The VSD per graph for South Korea.

	NPS 1	NPS 2	NPS 3	NPS 4	CAIDA 1	CAIDA 2
NPS 1	0.00	0.37	0.46	0.47	0.47	0.57
NPS 2	0.37	0.00	0.39	0.39	0.56	0.48
NPS 3	0.46	0.39	0.00	0.22	0.62	0.58
NPS 4	0.47	0.39	0.22	0.00	0.63	0.58
CAIDA 1	0.47	0.56	0.62	0.63	0.00	0.50
CAIDA 2	0.57	0.48	0.58	0.58	0.50	0.00

Table 5.13: The ESD per graph for South Korea.

Graphs	Node Count	Edge Count	Avg Degree	Clustering	Pearson
Both CAIDA Graphs	29,706	83,838	5.645	0.006	0.151
First Two NPS	21,080	37,087	3.519	0.009	0.083
2 CAIDA vs 2 NPS	17,164	29,556	3.444	0.006	0.085
Intersection NPS 1-2, CAIDA 1-2 with NPS 3	15,436	24,387	3.160	0.005	0.126
Intersection NPS 1-2, CAIDA 1-2 with NPS 4	15,443	24,296	3.147	0.005	0.126

Table 5.14: The basic stats of South Korea's stable core.

Wallis and Futuna

In Section 5.2 we discussed how the clustering coefficient could easily skew results if considered in isolation. We identified Wallis and Futuna, after sorting the data to identify those countries with the highest clustering coefficients. We originally conjectured the possibility of high clustering coefficient with countries that restrict control of the Internet, however our data did not support this conjecture. We found countries, (e.g., Wallis and Futuna in Figure 5.7), that

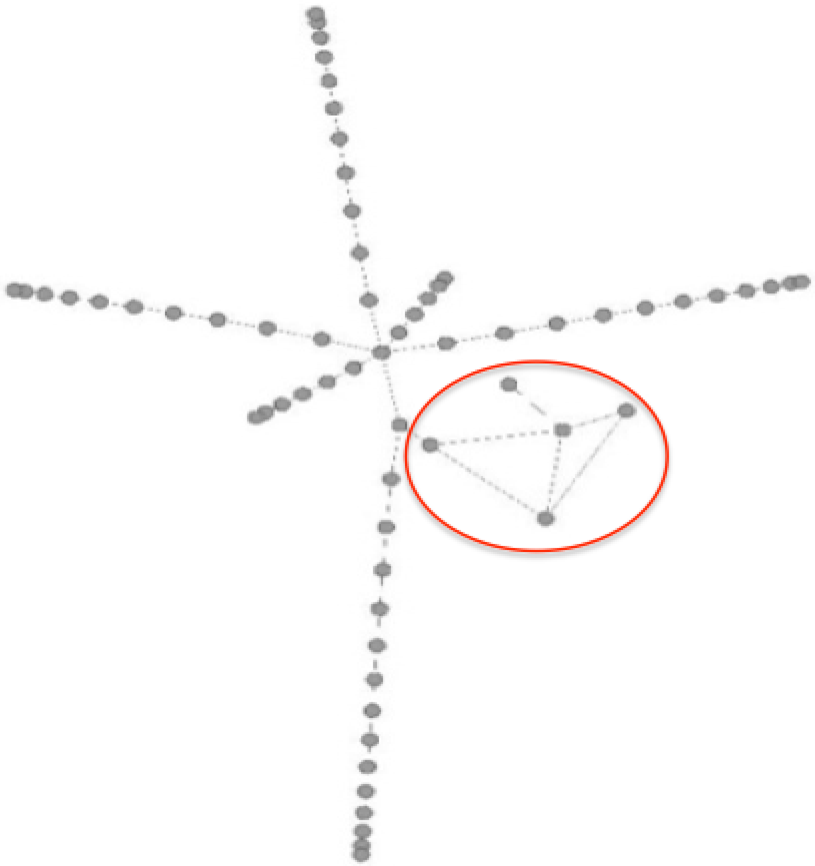


Figure 5.7: Graph of Wallis and Futuna with high clustering coefficient.

had high clustering coefficient, but were not on any lists of known Internet restricted countries. Wallis and Futuna is a relatively small nation but it had one of the highest clustering coefficients in one of our cycles because of the relatively small number of vertices connected to each other, almost a tree, so a few triangles (3-cycles) made a big impact, circled in Figure 5.7. This outlier was easy to visualize due to the small number of overall vertices, but larger graphs may not easily show this result. We caution the use of clustering coefficient when comparing graphs unless you can easily visualize your graph or other measures are used in combination with the clustering coefficient (such as density of edges in a graph).

CHAPTER 6: Future Work and Conclusion

In this chapter, we present our findings and provide insights into areas that might require further research.

6.1 Summary

Our goal was to provide an in-depth comparison of the NPS probing methodology and CAIDA probing methodology using existing graphical measurements. We also applied the same analysis to country specific sub-sets to see how the measurements behave with some surprising results. There is still an abundance of research that is needed to understand how the Internet is connected and its properties.

6.2 Future Work

We have barely scraped the surface for using these measurements, and there remains much to be explored. The following are some areas for possible future research.

(1) Perform more statistical analysis on additional runs

We were limited on the amount of probing cycles we could accomplish during the duration of our research. Making specific claims, such as the Pearson coefficient, about how the Internet or algorithms behaves as a result of this study is not statistically supported. We also only used a limited number of monitor vantage points and suggest a more methodical study of how the probing cycles discover topology differ by vantage point. Specifically, do certain monitors discover more depending on geographic region, or does the geographic distance even affect the return of trace information per destination IP.

(2) Does the stable core of the Internet exist or does a stable core exist in a sub-set of the Internet? Does the existence of a stable core give us any additional information, specifically threat vulnerability and assessment?

In Chapter 5, we conjectured that the intersections of intersections would produce a stable core of vertices that we could use to conduct threat analysis and other network security studies. We introduced the intersection of intersections to discover the stable core, but did not prove how many graph intersections it would take to confirm the existence of a stable core. Additionally, it would be interesting to know if the number of probing cycles needed to discover a stable core differ per country or other subset. Specifically, does a small set of intersection vertices compared to the total vertex found indicate a censored or heavily restricted Internet region, similar to our results for China in Section 5.2.3.

(3) How many unions of graphs does it take to represent the known IP allocation per country?

We studied the distribution of vertices discovered per country by the known alloted IP space assigned in Section 5.2.1 and noted the disparity of those discovered versus the amount allocated. It would be interesting to identify how many probing cycles it would take to gather enough data to provide an accurate distribution.

(4) Identify the number of "q" returns per country and group them with the number of discovered IP vertices.

We recommend further study of this to identify if the percentage of discovered IPs more closely represents that of the known IP allocation by country and further work from [35]. The actual location of the "q" is difficult if not impossible to know, but you can infer the location of the router that returns the "q" by the preceding and following router locations. The researchers propose a possible use of bins for those "q" returned. For example, if a trace returned US, US, "q", US, CH, then it would fall into a confident bin for US. Alternatively, if a trace returned (US, US, "q", "q", CH), then the "q"s could be apart of the United States or China, therefore a second bin per country for possible "q"s that belong to the country. Obviously there would be a lot of inferring to the location in this process, and we only suggest this method, while other methods could also work.

6.3 Conclusion

We performed numerous known standard graph comparisons in Section 5. Some of the results led to additional questions instead of answering our original questions, but we have made some headway. The CAIDA and NPS probing methodologies both accomplish the same task of giving an inferred map of the Internet topology, though NPS seems to discover more IPs in less time, as suggested in [36], from one probing cycle. In our research, we provided a concise study for a better understanding about the inferred representation of each of the probing methodologies. The researchers also offer additional directions to provide better insight into the inferred topology. Specifically, we hope our conjecture of a stable core, (see Section 5.2.3 and Section 5.2.3) could prove impactful.

One of the greatest challenges to this study was (1) the sheer size of the data sets[15] that we had to first strip into usable forms, and (2) needing a dedicated server to run the analytical analysis. Internet topology is not a topic that will have a simple solution from a single paper, but continued work to understand how components link and share the information on the Internet can prove insightful.

We also believe some of the metrics we used could be useful in other complex networks, such as biological and social networks. It could prove insightful to understand the most logical next step or underlying structure or the existence of a stable core or backbone for information distribution.

[15] Typical size of graph comparisons used six to ten Gigabyte (GB) of RAM per test for upwards of 14 days.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A:
SC Analysis Dump Format

```
# =======================================================================
# This file contains an ASCII representation of the IPv4 paths stored in
# the binary skitter arts++ and scamper warts file formats.
#
# This ASCII file format is in the sk_analysis_dump text output
# format: imdc.datcat.org/format/1-003W-7
#
# =======================================================================
# There is one trace per line, with the following tab-separated fields:
#
#
# 1. Key -- Indicates the type of line and determines the meaning of the
#           remaining fields.  This will always be ’T’ for an IP trace.
#
# -------------------- Header Fields ------------------
#
# 2. Source -- Source IP of skitter/scamper monitor performing the trace.
#
# 3. Destination -- Destination IP being traced.
#
# 4. ListId -- ID of the destination list containing this destination
#              address.
#
#      This value will be zero if no list ID was provided.  (uint32_t)
#
# 5. CycleId -- ID of current probing cycle (a cycle is a single run
#               through a given list).  For skitter traces, cycle IDs
#               will be equal to or slightly earlier than the timestamp
#               of the first trace in each cycle. There is no standard
#               interpretation for scamper cycle IDs.
```

```
#
#      This value will be zero if no cycle ID was provided.  (uint32_t)
#
# 6. Timestamp -- Timestamp when trace began to this destination.
#
# -------------------- Reply Fields ------------------
#
# 7. DestReplied -- Whether a response from the destination was received.
#
#      R - Replied, reply was received
#      N - Not-replied, no reply was received;
#          Since skitter sends a packet with a TTL of 255 when it halts
#          probing, it is still possible for the final destination to
#          send a reply and for the HaltReasonData (see below) to not
#          equal no_halt.  Note: scamper does not perform this last-ditch
#          probing at TTL 255.
#
# 8. DestRTT -- RTT (ms) of first response packet from destination.
#      0 if DestReplied is N.
#
# 9. RequestTTL -- TTL set in request packet which elicited a response
#      (echo reply) from the destination.
#      0 if DestReplied is N.
#
# 10. ReplyTTL -- TTL found in reply packet from destination;
#      0 if DestReplied is N.
#
# -------------------- Halt Fields ------------------
#
# 11. HaltReason -- The reason, if any, why incremental probing stopped.
#
# 12. HaltReasonData -- Extra data about why probing halted.
#
#        HaltReason            HaltReasonData
#        ------------------------------------
#        S (success/no_halt)    0
#        U (icmp_unreachable)   icmp_code
#        L (loop_detected)      loop_length
#        G (gap_detected)       gap_limit
#
# -------------------- Path Fields ------------------
#
# 13. PathComplete -- Whether all hops to destination were found.
#
#        C - Complete, all hops found
#        I - Incomplete, at least one hop is missing (i.e., did not
#            respond)
#
# 14. PerHopData -- Response data for the first hop.
#
#       If multiple IP addresses respond at the same hop, response data
#       for each IP address are separated by semicolons:
#
#       IP,RTT,numTries                    (for only one responding IP)
#       IP,RTT,numTries;IP,RTT,numTries;... (for multiple responding IPs)
#
#         where
#
#       IP -- IP address which sent a TTL expired packet
#       RTT -- RTT of the TTL expired packet
#       num_tries -- num tries before response received from TTL.
#
#       This field will have the value 'q' if there was no response at
#       this hop.
#
# 15. PerHopData -- Response data for the second hop in the same format
#       as field 14.
#
```

```
# ...
#
# N. PerHopData -- Response data for the destination
#       (if destination replied).
#
```

APPENDIX B:
IP Space by Country

COUNTRY	CAIDA 1	CAIDA 2	NPS1	NPS2	NPS 3	NPS 4
Afghanistan	0.074%	0.081%	0.042%	0.026%	0.028%	0.026%
Albania	0.082%	0.083%	0.037%	0.042%	0.053%	0.053%
Algeria	0.292%	0.262%	0.147%	0.067%	0.080%	0.083%
American Samoa	0.009%	0.010%	0.013%	0.006%	0.016%	0.013%
Andorra	0.022%	0.024%	0.003%	0.007%	0.006%	0.006%
Angola	0.129%	0.116%	0.074%	0.090%	0.095%	0.095%
Anguilla	0.006%	0.008%	0.002%	#N/A	0.007%	0.003%
Antarctica	0.030%	0.035%	0.002%	0.003%	0.004%	0.004%
Antigua and Barbuda	0.024%	0.024%	0.005%	0.012%	0.008%	0.007%
Argentina	0.669%	0.660%	0.707%	0.661%	0.650%	0.640%
Armenia	0.157%	0.159%	0.103%	0.098%	0.129%	0.122%
Aruba	0.030%	0.028%	0.004%	0.005%	0.002%	0.002%
Australia	1.282%	1.331%	1.180%	1.168%	1.170%	1.135%
Austria	0.756%	0.769%	0.697%	0.701%	0.699%	0.722%
Azerbaidjan	0.132%	0.123%	0.058%	0.054%	0.069%	0.069%
Bahamas	0.129%	0.133%	0.056%	0.055%	0.064%	0.062%
Bahrain	0.171%	0.181%	0.146%	0.140%	0.154%	0.150%
Bangladesh	0.232%	0.231%	0.219%	0.193%	0.213%	0.209%
Barbados	0.066%	0.065%	0.080%	0.062%	0.077%	0.076%
Belarus	0.141%	0.170%	0.126%	0.118%	0.135%	0.132%
Belgium	0.643%	0.637%	0.500%	0.461%	0.533%	0.536%
Belize	0.169%	0.172%	0.069%	0.033%	0.040%	0.032%
Benin	0.011%	0.011%	0.005%	0.007%	0.004%	0.004%
Bermuda	0.124%	0.130%	0.108%	0.118%	0.120%	0.113%
Bhutan	0.047%	0.050%	0.094%	0.080%	0.090%	0.087%
Bolivia	0.191%	0.180%	0.261%	0.230%	0.212%	0.211%
Bosnia-Herzegovina	0.264%	0.256%	0.309%	0.314%	0.303%	0.305%
Botswana	0.053%	0.051%	0.042%	0.048%	0.039%	0.038%

Brazil	2.555%	2.582%	2.855%	2.851%	2.880%	2.876%
British Indian Ocean Territory	0.004%	0.004%	#N/A	#N/A	#N/A	#N/A
Brunei Darussalam	0.074%	0.075%	0.037%	0.035%	0.035%	0.036%
Bulgaria	0.547%	0.530%	0.566%	0.511%	0.521%	0.515%
Burkina Faso	0.042%	0.051%	0.097%	0.075%	0.090%	0.089%
Burundi	0.008%	0.006%	#N/A	#N/A	0.001%	0.001%
Cambodia	0.084%	0.086%	0.061%	0.076%	0.109%	0.112%
Cameroon	0.075%	0.075%	0.080%	0.054%	0.060%	0.064%
Canada	2.459%	2.483%	2.555%	2.524%	2.598%	2.605%
Cape Verde	0.010%	0.011%	0.013%	0.010%	0.020%	0.018%
Cayman Islands	0.043%	0.043%	0.029%	0.029%	0.034%	0.030%
Central African Republic	0.003%	0.003%	#N/A	#N/A	#N/A	#N/A
Chad	0.004%	0.002%	0.002%	#N/A	0.001%	0.002%
Chile	0.539%	0.540%	0.399%	0.382%	0.383%	0.390%
China	6.370%	6.752%	8.303%	8.915%	8.571%	8.504%
Christmas Island	0.002%	0.002%	#N/A	#N/A	#N/A	#N/A
Cocos (Keeling) Islands	0.003%	0.004%	#N/A	#N/A	#N/A	#N/A
Colombia	0.607%	0.575%	0.600%	0.587%	0.552%	0.557%
Comoros	0.016%	0.013%	0.005%	0.007%	0.008%	0.018%
Congo	0.023%	0.020%	0.024%	0.017%	0.011%	0.016%
Cook Islands	0.026%	0.025%	0.028%	0.033%	0.049%	0.047%
Costa Rica	0.376%	0.321%	0.204%	0.193%	0.165%	0.161%
Croatia	0.400%	0.340%	0.335%	0.294%	0.283%	0.284%
Cuba	0.099%	0.108%	0.063%	0.035%	0.037%	0.036%
Cyprus	0.212%	0.196%	0.233%	0.191%	0.167%	0.168%
Czech Republic	1.046%	1.033%	1.197%	1.177%	1.178%	1.211%
Denmark	0.804%	0.773%	0.899%	0.908%	0.940%	0.941%
Djibouti	0.041%	0.075%	0.013%	0.013%	0.013%	0.013%
Dominica	0.023%	0.029%	0.017%	0.010%	0.015%	0.015%
Dominican Republic	0.263%	0.252%	0.121%	0.104%	0.115%	0.109%
Ecuador	0.294%	0.301%	0.354%	0.335%	0.324%	0.312%
Egypt	0.773%	0.709%	0.372%	0.379%	0.349%	0.345%

El Salvador	0.231%	0.220%	0.105%	0.121%	0.119%	0.114%
Equatorial Guinea	0.030%	0.028%	0.010%	0.026%	0.025%	0.029%
Eritrea	0.003%	0.005%	#N/A	#N/A	#N/A	#N/A
Estonia	0.332%	0.309%	0.509%	0.484%	0.486%	0.486%
Ethiopia	0.089%	0.087%	0.050%	0.032%	0.044%	0.043%
Falkland Islands	0.006%	0.009%	0.005%	0.006%	0.031%	0.029%
Faroe Islands	0.071%	0.069%	0.053%	0.048%	0.044%	0.044%
Fiji	0.129%	0.122%	0.049%	0.051%	0.053%	0.046%
Finland	0.722%	0.704%	0.574%	0.547%	0.546%	0.543%
France	1.662%	1.713%	1.687%	1.680%	1.548%	1.546%
France (European Territory)	0.003%	0.002%	0.002%	#N/A	0.001%	0.001%
French Guyana	0.077%	0.076%	0.019%	0.020%	0.022%	0.021%
French Southern Territories	0.004%	0.002%	#N/A	#N/A	#N/A	#N/A
Gabon	0.152%	0.158%	0.050%	0.062%	0.070%	0.069%
Gambia	0.002%	0.004%	#N/A	#N/A	#N/A	#N/A
Georgia	0.249%	0.252%	0.339%	0.308%	0.318%	0.331%
Germany	2.741%	2.779%	3.329%	3.536%	3.448%	3.429%
Ghana	0.142%	0.166%	0.065%	0.047%	0.116%	0.115%
Gibraltar	0.017%	0.014%	0.013%	0.016%	0.014%	0.013%
Great Britain	2.653%	2.673%	2.867%	2.912%	2.935%	2.867%
Greece	0.623%	0.556%	0.517%	0.504%	0.488%	0.495%
Greenland	0.035%	0.035%	0.026%	0.060%	0.027%	0.043%
Grenada	0.028%	0.030%	0.031%	0.017%	0.030%	0.025%
Guadeloupe (French)	0.132%	0.143%	0.089%	0.092%	0.089%	0.088%
Guam (USA)	0.144%	0.133%	0.104%	0.099%	0.100%	0.093%
Guatemala	0.155%	0.152%	0.168%	0.095%	0.172%	0.182%
Guinea	0.045%	0.040%	0.023%	0.024%	0.032%	0.020%
Guinea Bissau	0.003%	0.002%	#N/A	#N/A	0.001%	0.001%
Guyana	0.017%	0.019%	0.048%	0.022%	0.023%	0.023%
Haiti	0.106%	0.079%	0.041%	0.048%	0.043%	0.044%
Honduras	0.176%	0.170%	0.095%	0.103%	0.102%	0.100%
Hong Kong	0.854%	0.872%	1.071%	1.056%	1.040%	1.043%

Hungary	0.706%	0.680%	0.643%	0.679%	0.652%	0.656%
Iceland	0.294%	0.260%	0.194%	0.195%	0.189%	0.189%
India	1.147%	1.096%	1.101%	1.011%	1.031%	1.022%
Indonesia	0.515%	0.489%	0.508%	0.493%	0.532%	0.541%
Iran	0.679%	0.684%	0.498%	0.500%	0.480%	0.503%
Iraq	0.226%	0.243%	0.123%	0.104%	0.131%	0.129%
Ireland	0.606%	0.609%	0.594%	0.559%	0.583%	0.563%
Israel	0.631%	0.608%	0.613%	0.580%	0.582%	0.591%

REFERENCES

[1] People overload website, hoping to help search for missing jet, March 2014. http://wnmufm.org/post/people-overload-website-hoping-help-search-missing-jet.

[2] Country comparison: Internet users, Febuary 2014. https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html.

[3] Request for comments (RFC), 2014. https://www.ietf.org/rfc.html.

[4] John Hawkinson and Tony Bates. Guidelines for creation, selection, and registration of an autonomous system (AS). 1996. http://tools.ietf.org/html/rfc1930.

[5] Daryl Lee. Toward large-graph comparison measures to understand internet topology dynamics, September 2013. http://calhoun.nps.edu/public/bitstream/handle/10945/37658/13Sep_Lee_Daryl.pdf?sequence=1.

[6] Lixin Gao. On inferring autonomous system relationships in the internet. In *Global Telecommunications Conference, 2000. GLOBECOM'00. IEEE*, volume 1, pp. 387–396. IEEE, 2000.

[7] Mehmet H Gunes and Kamil Sarac. Inferring subnets in router-level topology collection studies. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, pp. 203–208. ACM, 2007.

[8] Adam Bender, Rob Sherwood, and Neil Spring. Fixing ally's growing pains with velocity modeling. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, pp. 337–342. ACM, 2008.

[9] Ken Keys. Internet-scale ip alias resolution techniques. *ACM SIGCOMM Computer Communication Review*, 40(1):50–55, 2010.

[10] Mehmet H Gunes and Kamil Sarac. Resolving ip aliases in building traceroute-based internet maps. *IEEE/ACM Transactions on Networking (ToN)*, 17(6):1738–1751, 2009.

[11] Robert Beverly, Arthur Berger, and Geoffrey G. Xie. Primitives for active internet topology mapping: Toward high-frequency characterization. In *Proceedings of the Tenth ACM SIGCOMM/USENIX Internet Measurement Conference (IMC)*, November 2010.

[12] Benoit Donnet, Philippe Raoult, Timur Friedman, and Mark Crovella. Efficient algorithms for large-scale topology discovery. In *ACM SIGMETRICS Performance Evaluation Review*, volume 33, pp. 327–338. ACM, 2005.

[13] Jamar Wright. Temporal comparisons of the internet topology measures, June 2014.

[14] Cooperative association for internet data analysis data, January 2014.

[15] Paris traceroute, 2013. http://www.paris-traceroute.net/.

[16] Brice Augustin, Xavier Cuvellier, Benjamin Orgogozo, Fabien Viger, Timur Friedman, Matthieu Latapy, Clémence Magnien, and Renata Teixeira. Avoiding traceroute anomalies with paris traceroute. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, pp. 153–158. ACM, 2006.

[17] NFSFNET T1 Backbone and Regional Networks, 1991. http://avl.ncsa.illinois.edu/project-archive/visualizing-the-early-internet.

[18] John C. Doyle, David L. Alderson, Lun Li, Steven Low, Matthew Roughan, Stanislav Shalunov, Reiko Tanaka, and Walter Willinger. The “robust yet fragile” nature of the internet. *Proceedings of the National Academy of Sciences of the United States of America*, 102(41): 14497–14502, 2005. http://www.pnas.org/content/102/41/14497.abstract.

[19] Cheswick map visual, March 2014. http://research.lumeta.com/ches/map/.

[20] Caida annual report 1998. http://www.caida.org/home/about/annualreports/1998/.

[21] Kenneth Rosen. *Discrete Mathematics and Its Applications 7th edition*. McGraw-Hill Science, New York, NY, USA, 2011.

[22] Gary Chartrand and Ping Zhang. *A First Course in Graph Theory*. Courier Dover Publications, Boston, MA, USA, 2012.

[23] D.B. West. *Introduction to graph theory*. Prentice Hall, Englewood Cliffs, NJ, USA, 2001. http://books.google.com/books?id=TuvuAAAAMAAJ.

[24] B. Huffaker, M. Fomenkov, and K. Claffy. Internet Topology Data Comparison. May 2012. http://www.caida.org/publications/papers/2012/topocompare-tr/.

[25] Priya Mahadevan, Dmitri Krioukov, Kevin Fall, and Amin Vahdat. Systematic topology analysis and generation using degree correlations. In *ACM SIGCOMM Computer Communication Review*, volume 36, pp. 135–146. ACM, 2006.

[26] Jari Saramäki, Mikko Kivelä, Jukka-Pekka Onnela, Kimmo Kaski, and Janos Kertesz. Generalizations of the clustering coefficient to weighted complex networks. *Physical Review E*, 75(2):027105, 2007.

[27] Transitivity. 2014. http://networkx.github.io/documentation/latest/reference/generated/networkx.algorithms.cluster.transitivity.html.

[28] M. E. J. Newman. Mixing patterns in networks. *Physical Review E*, 67(2):026126, 2003.

[29] M. E. J. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89:208701, Oct 2002. http://link.aps.org/doi/10.1103/PhysRevLett.89.208701.

[30] Albert-László Barabási, Baruch Barzel, and Mauro Martino. Evolution of the social network of scientific collaborations. *Physica A: Statistical Mechanics and its Applications*, 311(3): 590–614, 2002.

[31] Young Hyun. Archipelago measurement infrastructure. In *Proceedings of the 7th CAIDA-WIDE Workshop*, 2006.

[32] Young Hyun. Caida monitors: The archipelago measurement infrastructure, 2009. http://www.caida.org/data/monitors/monitor-map-ark.xml.

[33] Matthew Luckie. Scamper: a scalable and extensible packet prober for active measurement of the internet. In *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, pp. 239–245. ACM, 2010.

[34] Michalis Faloutsos and Aleksandar Kuzmanovic, editors. *Ingress Point Spreading: A New Primitive for Adaptive Active Network Mapping*, volume 8362 of *Lecture Notes in Computer Science*. Springer International Publishing, AG Gewerbestrasse 11, CH, 2014. http://dx.doi.org/10.1007/978-3-319-04918-2_6.

[35] B. Yao, Ramesh Viswanathan, F. Chang, and D. Waddington. Topology inference in the presence of anonymous routers. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pp. 353–363 vol.1, March 2003.

[36] Guillermo Baltra. Efficient strategies for active interface-level network topology discovery, September 2013. http://calhoun.nps.edu/public/bitstream/handle/10945/37583/13Sep_Baltra_Guillermo.pdf?sequence=1.

THIS PAGE INTENTIONALLY LEFT BLANK

Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California