EFFICIENT LARGE SCALE NETWORK TOPOLOGY MEASUREMENT

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

by
Ibrahim Ethem Coskun
Dr. Mehmet Hadi Gunes / Thesis Advisor
December 2015
THE GRADUATE SCHOOL

We recommend that the thesis prepared under our supervision by

IBRAHIM ETHEM COSKUN

Entitled

Efficient Large Scale Network Topology Measurement

be accepted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Mehmet H. Gunes, Ph.D., Advisor

Murat Yuksel, Ph.D., Committee Member

Gokhan Pekcan, Ph.D., Graduate School Representative

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

December, 2015
ABSTRACT

Studies related to the Internet topology measurement are challenging because of the large-scale topology of the network with limited number of vantage points. As topology mapping studies rely on the few vantage points to probe a large number of destinations, there is considerable redundancy in the collected measurements. Large number of redundant probes during a measurement study typically contain significant overlaps, which increases measurement traffic and duration. To decrease the probing overhead, in this thesis, we propose an efficient subnet-work probing technique by identifying ingress points of the autonomous systems towards the subnetwork. To achieve this, we first probe announced prefixes of a given autonomous system until the ingress points of the target prefixes are identified. After identifying the ingress points of autonomous system with respect to the subnetworks, the probing is conducted based on the vantage point reachability of the ingresses. As shown in experimental study, this approach can considerably reduce the probing overhead while minimizing the loss of information.
ACKNOWLEDGEMENTS

I would like to dedicate the accomplishment of this study to my beloved parents Huriye Coskun, MD and Hasan Coskun, MD, and my siblings that I overlove, Kubra, Emre, Esra, and Emir.

I would like to acknowledge the support from National Science Foundation that enabled the work in this thesis via NSF Computer and Network Systems award # 1321164.

I would also like to say that, this study could not have been completed without the sincere support, help and encouragement of far too numerous people. Firstly, I would like to specially express my gratitude to following people that always stand by me:

Dr. Mehmet Gunes, for all his support, guidance, and all the help until the very last minute of the completion of this study.

My committee members, Dr. Murat Yuksel and Dr. Gokhan Pekcan, for their kind support and encouragement.

Finally, I want to thank to all my friends that stand by me in every step of my life. And, special thanks to Hilmi Ekim, Kadir Bayrak, Naci Ozyildirim, Adilet Kachkeev, Ergun Akturk, Emrah Demirbilek, Ali Coban, Suleyman Demirhan, Murat Turhan, Ahmet Soran, Burkay Sucu, Sevki Cesmeci, Hicabi Bozkaya, Mehmet Dogan, Esra Erdin, Prasun Dey, Vahid Behzadan, Hemel Khan, Paulo Regis, Suman Bhunia, Jay Thom, Prabath Jose, Deepak Tosh, Manoj Popuri, Moinul Rifat, Sandeep Mathew, Mustafa Ak, Kamran Bekirov, Konur Cabuk, Umar Abdullayev, Kamran Shakhpazov, Kamran Nuratdinov, Anil Yildir, Selcan Aynal, Levent Colak, Hasan Ufaktepe, Gurhan Gunduz, Davut Ucar, Burak Akgun, Omer Kilavuz, Tarik Karaoglu, Burak Coskun, Lisa Cody, Ozkan Ozer, Zachary Newell, Halim-Nil Ates, Secil Yuzuk, Sina Yapanal, and Cevdet.
# Table of Contents

Abstract ................................................................. i
Acknowledgements ...................................................... ii
Table of Contents ......................................................... iii
List of Tables ............................................................ v
List of Figures ............................................................ vi

1 Introduction ......................................................... 1

2 Background .......................................................... 4
  2.1 Internet Topology ................................................. 4
    2.1.1 Autonomous System (AS) .................................. 5
    2.1.2 Border Gateway Protocol (BGP) ......................... 6
    2.1.3 Regional Internet Registry (RIR) ....................... 7
    2.1.4 WHOIS Protocol ........................................... 7
    2.1.5 Network Measurement ..................................... 8
    2.1.6 Traceroute Tool .......................................... 9
  2.2 Internet Measurement Platforms ............................. 11
    2.2.1 RIPE Atlas ................................................ 12
    2.2.2 PlanetLab ............................................... 13

3 Related Work ..................................................... 15
  3.1 Rocketfuel ..................................................... 15
  3.2 Scriptroute ...................................................... 16
  3.3 Doubletree ...................................................... 16
  3.4 Ingress Point Spreading ...................................... 17
  3.5 Cheleby ........................................................ 17

4 InSub: Ingress to Subnet relation based Probe Reduction 19
  4.1 Subnet Specific Ingress Point Identification (SSIPI) .... 20
    4.1.1 Ingress Point Identification .......................... 20
    4.1.2 Ingress Point Identification of a Subnet ............. 22
  4.2 Ingress to Subnet based Vantage Point Probing (ISVPP) .. 25
  4.3 Subnetwork Probing .......................................... 28
  4.4 Tree based Tracing through an Ingress ..................... 29
5 Experimental Results
   5.1 Subnet Specific Ingress Point Identification (SSIPI) Phase 33
   5.2 Ingress to Subnet based Vantage Point Probing (ISVPP) Phase 37
   5.3 Ingress reachability of Subnets from Vantage Points 43
   5.4 Network Characteristics 59

6 Summary and Future Work

Bibliography
List of Tables

4.1 Potential vantage points for ingress-subnet pairs . . . . . . . . . . . . 27
4.2 Ingress-subnet pair assignment to vantage points . . . . . . . . . . . 28
5.1 Measured Autonomous Systems . . . . . . . . . . . . . . . . . . . . . . 33
## List of Figures

2.1 Traceroute tool implementation ............................................. 10
2.2 RIPE Atlas vantage points per destination AS (log-log scale) ........ 12
2.3 PlanetLab vantage points per destination AS (log-log scale) ........ 13

4.1 Ingress point identification and hop distance determination .......... 21
4.2 SSIPI work schema, when the first and last IP addresses of a subnet enter the target AS through the same ingress point. ...................... 23
4.3 SSIPI work schema, when the first and last IP addresses of a subnet enter the target AS through different ingress points. ...................... 24
4.4 Vantage point assignment for a sample topology ....................... 27
4.5 Sample topology for tree based tracing ................................... 30

5.1 Probed subnets per AS (log-scale). Blue bars indicate the number of subnets for which ingress point was successfully identified while red bars indicate subnets whose ingress points were not discovered. 34
5.2 SSIPI destination IPs per AS (log scale). Blue bars indicate the number of probed IPs while green bars indicate the number of reached IPs. ................................................................. 35
5.3 SSIPI probes per AS (log scale). Blue bars indicate the number of probes in traces while green bars indicate the number of probes if stopped at ingress. ...................................................... 36
5.4 SSIPI observed IPs per AS (log scale). Blue bars indicate the number of observed IPs in traces while green bars indicate the number of IPs if stopped at ingress. .............................................. 37
5.5 SSIPI observed edges per AS (log scale). Blue bars indicate the number of observed edges in traces while green bars indicate the number of edges if stopped at ingress. ............................... 37
5.6 Destination IPs per AS. ............................................................ 38
5.7 Vantage points per AS. Blue bars indicate the number of vantage points utilized in measurement of each AS. Additional red bars indicate vantage points that were selected but did not complete probing. ................................................................. 39
5.8 Ingresses per AS (log scale) ..................................................... 39
5.9 Number of probes per AS for each method (log scale) ............... 40
5.10 Observed IPs per AS for each method (log scale) .................... 41
5.11 Ratio of IPs discovered by each method .................................. 41
5.12 Observed edges per AS for each method (log scale) ................. 42
5.13 Ratio of edges discovered by each method .............................. 42
5.14 Sample vantage point to ingress point reachability ................. 43
5.15 Vantage points per Ingress (log-log scale) .............................. 45
5.16 Ingresses per vantage point ................................................ 46
5.17 Topology discovery ranking of ingresses - AS 174 (log scale) ...... 48
5.18 Topology discovery ranking of ingresses - AS 1273 (log scale) ... 48
5.19 Topology discovery ranking of ingresses - AS 6762 (log scale) ... 49
5.20 Topology discovery ranking of ingresses - AS 2828 (log scale) ... 49
5.21 Topology discovery ranking of ingresses - AS 1239 (log scale) .. 50
5.22 Topology discovery ranking of ingresses - AS 3356 (log scale) ... 50
5.23 Topology discovery ranking of ingresses - AS 6939 (log scale) ... 51
5.24 Topology discovery ranking of ingresses - AS 6461 (log scale) ... 51
5.25 Topology discovery ranking of ingresses - AS 3491 (log scale) ... 52
5.26 Topology discovery ranking of ingresses - AS 6453 (log scale) ... 52
5.27 Topology discovery ranking of vantage points - AS 174 (log scale) 54
5.28 Topology discovery ranking of vantage points - AS 1273 (log scale) 54
5.29 Topology discovery ranking of vantage points - AS 6762 (log scale) 55
5.30 Topology discovery ranking of vantage points - AS 2828 (log scale) 55
5.31 Topology discovery ranking of vantage points - AS 1239 (log scale) 56
5.32 Topology discovery ranking of vantage points - AS 3356 (log scale) 56
5.33 Topology discovery ranking of vantage points - AS 6939 (log scale) 57
5.34 Topology discovery ranking of vantage points - AS 6461 (log scale) 57
5.35 Topology discovery ranking of vantage points - AS 3491 (log scale) 58
5.36 Topology discovery ranking of vantage points - AS 6453 (log scale) 58
5.37 Average Degree of collected topologies ................................. 59
5.38 Degree distribution of resulting graphs - AS 174 (log-log scale) ... 60
5.39 Degree distribution of resulting graphs - AS 1273 (log-log scale) ... 60
5.40 Degree distribution of resulting graphs - AS 6762 (log-log scale) ... 60
5.41 Degree distribution of resulting graphs - AS 2828 (log-log scale) ... 61
5.42 Degree distribution of resulting graphs - AS 1239 (log-log scale) ... 61
5.43 Degree distribution of resulting graphs - AS 3356 (log-log scale) ... 61
5.44 Degree distribution of resulting graphs - AS 6939 (log-log scale) ... 62
5.45 Degree distribution of resulting graphs - AS 6461 (log-log scale) ... 62
5.46 Degree distribution of resulting graphs - AS 3491 (log-log scale) ... 62
5.47 Degree distribution of resulting graphs - AS 6453 (log-log scale) ... 63
5.48 Average Degree of collected topologies ................................. 63
5.49 Clustering coefficient of collected topologies (log-scale) .......... 64
5.50 Clustering coefficient distribution (log-log scale) ................. 65
Chapter 1

Introduction

Today, it is an incontestable reality that the Internet is the greatest data providing environment because it provides access to the information quickly and easily. Many industries and sectors are able to provide prompt interaction compared to those days when the Internet was not ubiquitous. Internet enabled a digital age where any person in the world can connect to others with minimal effort.

Internet measurement helps us better understand the characteristics of the Internet and thus letting us to be able to analyze the changes occurring on the Internet and improve of the Internet based services [23]. Network measurement also helps in network management of large scale ISPs [8, 9]. Internet measurement is concerned with traffic [37, 36] or topology [33].

Network practitioners obtain sample Internet topologies or generate synthetic topologies to evaluate performance of new network protocols and devices. Sample topologies are obtained by sampling networks using measurement probes [22] and processing obtained results to determine unresponsive routers [28, 35], alias
IP addresses of routers [25, 26, 24, 30], and subnets of link-layer connectivity [27]. Often these topologies are large scale and require efficient methods for processing [32]. Synthetic topologies are often based on measured Internet characteristics [14, 6, 7, 5].

The complicated structure of the Internet protocols and devices makes it challenging to measure the Internet. One of the principal challenges faced when mapping a network is the significantly redundant traffic while probing destinations from vantage points [18]. Minimizing the redundant probing is also important in terms of the time it takes to complete a measurement campaign. Collecting network topology data in a timely manner is important as the networks are constantly growing. High volumes of probing traffic towards a destination autonomous system might be interpreted as port scanning or denial of service attack and blocked by the autonomous system, preventing us obtaining a map of the destination network [15]. Thereby, reducing the probe redundancy is substantial not only in the sense of getting efficient results to help mapping a network more properly but to abstain from involuntarily disruption on a target network.

In this thesis, we present our subnet probing technique that focuses on identifying the ingress points of autonomous systems towards destination subnets. Researches have shown that the ingress point utilization has an considerable effect on probe overhead minimization [13, 12]. Through this identification, we set up a probing mechanism utilizing the vantage point to ingress relations to highly diminish the unnecessary probing to perform an efficient measurement and to not to disrupt a network. We implemented and tested different approaches for probe overhead minimization and discussed their advantages and disadvantages.

In Chapter 2, we provide brief background on Internet topology measurement
studies. In Chapter 3, we present related studies. In Chapter 4, we present our probe reduction system based on identification of autonomous system ingresses towards a destination subnet. In Chapter 5, we discuss the results of our experiments and compare the outcomes of different approaches. Lastly, in Chapter 6, we summarize our findings and discuss potential improvement in the future.
Chapter 2

Background

In this chapter, we provide a brief description of the Internet topology mapping concepts. Since we deal with the link level Internet topology, it is important to get acquainted with how the current topology is designed. Additionally, we touch on the tools and platforms that we utilize in mapping Internet topology.

2.1 Internet Topology

Internet is not an integral system in which one entity can control and monitor every transaction, communication attempt, or traffic from a stationary point. It has been designed as the result of distinct Autonomous Systems (AS) connecting to each other through direct peering links or Internet Exchange Points (IXPs) and exchange traffic. Internet is described as a network of networks which consists of thousands of distinct ASes being connected to each other.
2.1.1 Autonomous System (AS)

An Autonomous Systems (AS) manages either a single network or a number of interconnected networks. Each AS has its own routing policy that can differ from the routing policies of the other ASes policies. ASes are categorized in two roles as a Transit AS, which is connected to multiple ASes and allows the data traffic of the other ASes to go through itself or a Stub AS, which has a connection with only a Transit AS. A Stub AS might be multi-homed by having multiple provider ASes as Transits but does not let any transit data traffic to pass through itself. To be able to provide the connectivity between ASes, each AS needs to know how to reach the rest of the Internet.

An Autonomous System Number (ASN) is a unique number that is assigned to an AS by Regional Internet Registry (RIR). With use of AS number, a single network or a group of networks that is administrated by a unitary authority can be represented as a single entity. The ASN is used when a router advertises a new IP prefix to other ASes. ASes that receive the advertisement add their ASN’s to the AS path before propagating the advertisement to their neighbors.

BGP protocol is designed for implementing the interconnection between ASes in the Internet by providing mechanisms for exchange of routing information, i.e., AS paths, between ASes [4]. BGP also allows ASes to enforce peering agreements with neighboring ASes and thereby implements the aggregate structure of the Internet.
2.1.2 Border Gateway Protocol (BGP)

Border Gateway Protocol (RFC 4271) provides mechanisms to exchange routing information between ASes through their border routers. While IP is used to provide distinct identification of devices across the Internet, BGP is essential for communication in the Internet as it is the only protocol that provides reachability across the whole Internet.

Internal BGP (iBGP) protocol exchanges routing information inside an AS and external BGP (eBGP) protocol exchanges routing information between neighboring ASes. The routers that run eBGP are called the border routers and each of these border routers have a direct link to another border router in neighboring AS. While the border routers advertise a new internal or external routing information that they received to neighboring border routers of other ASes with eBGP, they advertise external routing information to internal routers with iBGP.

Border routers which run eBGP can simply be defined as the connection points between ASes. Hence, our main focus will be on eBGP running routers, which are also called ingress point of an AS. Since eBGP routers of an AS are the entrance point of that AS, we can eliminate redundant probing by starting probing of a target AS from its eBGP routers. This could save a considerable amount of redundant probes are generated from vantage points towards the target AS. However, since the routing table information of these eBGP running routers is not publicly available, we do not have a certain information about the interfaces of these routers. We can only predict the relationships between ASes based on the current AS relationship inference techniques [17, 45, 20, 16].
2.1.3 Regional Internet Registry (RIR)

Regional Internet Registries (RIR) are organizations that allocate IPs and ASes in distinct geographical areas worldwide. There are five RIRs as; ARIN, which is taking care of the United States, Antarctica, many parts of the Caribbean, North Atlantic Islands, and Canada; AfriNIC, which is responsible for managing IPs and ASes in Africa; APNIC is dominating the Asia Pacific region; LACNIC providing registration services for some parts of the Caribbean and Latin America, and lastly, RIPE NCC is responsible for Europe, the Middle East, and central Asia.

2.1.4 WHOIS Protocol

WHOIS is a query/response protocol that provides information about domain names, IP address blocks, and ASes by querying the databases of Regional and National Internet Registries such as APNIC, AFRINIC, RIPE NCC, ARIN, LACNIC, APJII, CNNIC, JPNIC, KRKNIC/NIDA, TWNIC, VNNIC, IRINN [3].

By querying an IP address using WHOIS tool, we can get information about the CIDR, ASN, Internet Registry name and geographical address of the organization’s headquarters of that IP address. However, instead of generating extra traffic on a target network using WHOIS tool for each IP address, we perform bulk download of the announced AS prefix data [2].
2.1.5 Network Measurement

Internet measurement involves active measurement and passive measurement. Passive measurement is simply measuring a network without generating any additional traffic on the measured network. Namely, passive measurement techniques do not generate probing packets on the measured network that can alter the data traffic. As an example, Routeviews Project [38] gathers BGP routing table data of ASes which contains information about how ASes are connected. It is possible to make inferences about the characteristics of the target network by using passive measurement techniques, as well as to discover the topology of a network [19]. Packet capturing and network sniffing tools are used to perform passive measurements.

On the other hand, active measurement methods are generally used to gather information of a network by injecting probing traffic on the target network. It is possible to get information of a network such as; number of edges and nodes it contains, hop distance between different nodes, how nodes are connected, etc. It is also possible to have information about the characteristics of targeted networks such as the most central nodes, the most traversed edges, the most visited nodes, edges with the most intense data traffic, etc.

Two basic active measurement tools that are widely used in measurement studies are Ping and Traceroute. Ping is used to check if a target node is reachable. It simply utilizes the Internet Control Message Protocol (ICMP) as it sends ICMP ECHO request packets to a target IP address and waits for an ICMP ECHO reply message from the target. If the target IP responds, then it measures the Round Trip Time (RTT) of each packet. RTT is the amount of time between a data packet is
sent to a target point, and the reply message of the target unit is received. RTT is used to make inferences about the physical distances between Internet devices.

### 2.1.6 Traceroute Tool

*Traceroute* has more capabilities as it provides information about how nodes are connected in the network. That is, network links between nodes and their IP addresses can be identified by Traceroute tool. Data packets of a traceroute probe can be selected as either User Datagram Protocol (UDP), Transmission Control Protocol (TCP), or ICMP. Traceroute relies on the Time To Live (TTL) values of an IP packet. TTL is simply defined as the maximum hop distance that a data packet can traverse from the sender. When the TTL value expires, the router typically sends an ICMP Time Exceeded message to source IP address.

Traceroute sends a probe packet to a destination by limiting the TTL value starting from 1 up to a maximum value, a default of 30. Each router on the path, that is to be traced towards a target, decrements the TTL value of incoming packets by 1. If at some point, a router observes that the TTL becomes zero, than the router drops the packet and sends an ICMP Time Exceeded message to the sender. IP addresses of the routers is obtained from these ICMP messages. Finally, if a probe packet reaches the given target IP address, ICMP ECHO or ICMP port unreachable reply message is sent by the destination to the sender based on the protocol used in tracing. Using this mechanism, traceroute infers the links between routers on the path toward a given destination.

Figure 2.1 presents the probing mechanism of Traceroute tool. In the figure, source host initiates traces toward destination $D$ with a TTL of 1. Once the packet
reaches router with A with $IP_A$ TTL is decremented to 0, which triggers an ICMP Time Exceeded message back to the host. This message reveals the IP address of router A as it has source IP of $IP_A$. Then, source host sends probes toward destination $IP_D$ with a TTL of 2. These packets are discarded at router B as TTL becomes 0 and a warning is sent to the host. In a similar manner all routers between source host and destination D are discovered.

We utilized an more improved version of Traceroute tool, i.e., Paris-traceroute, because of its ability to trace the actual paths in presence of load balancer routers. *Paris traceroute* tries to fix traced paths so that load balancing routers do not yield erroneous paths while performing a Traceroute [11].

We set the probing algorithm to 'hopbyhop' that enables all packets sent to a
destination point to hold the same flow-identifier so that it reveals accurate paths when there is a load balancing router in the target network. We also modified the source code of Paris-traceroute and set the return flow-identifier of the packets to force them to follow the same path to the source. We also set the return flow-id of the probes using (-r) flag that comes from the original Paris-traceroute tool. However, as indicated in tokyo-ping, it is not always possible to accurately control the return flow identifier of the probes [41]. Another flag of Paris-traceroute that we used is ‘-w’, which enables to set the waiting time until getting a response from a destination point.

We set the ‘-w’ flag to 500 milliseconds as it was shown that response time of over 99.9% of the responsive nodes is 500 milliseconds [34]. We also chose ICMP as our probing protocol as it provides the highest responsiveness by compared to TCP and UDP based probes [21].

2.2 Internet Measurement Platforms

Studies has shown that vantage point quantity and geographic location has an important effect on internet measurement studies [42, 12, 34]. An Internet measurement platforms is a collection of distributed vantage points provided by an organization that paves the way for measurement experiments on the Internet. In order to perform large scale measurements on the Internet, we need to utilize one of such Internet measurement platforms.

On the occasion of this foresight, we examined two well known publicly accessible Internet measurement platforms, RIPE Atlas and PlanetLab.
2.2.1 RIPE Atlas

RIPE (Rseaux IP Europens) NCC provides a measurement platform, named RIPE Atlas, via small measurement hardware [10]. As RIPE Atlas nodes are small probing devices that are distributed to interested users for free, there are over 7381 of them deployed worldwide. We collected ASN of 7,378 of these nodes which were scattered over 2,829 different ASes. We observed that, there are 1,997 ASes with 1 measurement node, 744 ASes with 2 to 10 measurement nodes, and 5 ASes with more than 100 measurement nodes (see Fig. 2.2). ASes with more than 100 measurement nodes were very large ASes.
Figure 2.3: PlanetLab vantage points per destination AS (log-log scale)

### 2.2.2 PlanetLab

PlanetLab is a global research network that allows network experimentation through worldwide vantage points [31]. There are currently 1,353 nodes deployed on 717 sites. However, only approximately about 450 of these nodes are in boot condition, namely less than half of these nodes are ready to be used for experimentation. Additionally, when we pinged the hostnames of 1,030 nodes which were announced on PlanetLab, only 865 of these nodes replied. Hence, we could get ASN information of 865 of PlanetLab nodes. In Figure 2.3, we see that just one AS includes 41 nodes and there are 236 ASes which have 2-10 nodes. The red nodes in the Figure 2.3 were the ones available during our measurement campaigns in Chapter 5.

As RIPE Atlas provides greater number of vantage points, it seems to be a better option to perform Internet topology measurements. However, as RIPE Atlas vantage points are small hardware, there are limitations on the time and number
of node and probe allowance per user. Hence, we utilize PlanetLab as it also has pretty good number of VPs that are distributed over the world. PlanetLab enables its users to have much more authority on the available node and provides a sufficient number of easily monitorable nodes with enough capacity to run the required tools that we used in our experiments.
Chapter 3

Related Work

Reducing the probing redundancy is an important challenge, especially for large-scale network measurements. In this chapter, we provide a brief description of current approaches to network probe overhead reduction.

3.1 Rocketfuel

Rocketfuel presented one the leading works in implementing a probe reduction tool which uses a system that they called Directed Probing [43]. Directed Probing is based on recognizing transit traceroutes that just pass through a network. To identify transit traceroutes, they utilize RouteViews [38] to detect AS peering. They introduced three techniques, namely, Ingress Reduction, Egress Reduction, and Next-Hop AS Reduction, to predict the future movement of traces by identifying the previous entry and exit points of the traces. However, in Rocketfuel’s system, only the overhead occurring on the boundaries of ASes are targeted. Hence,
any redundancy that inside the AS because of merging paths is ignored. Another weakness of Rocketfuel is that, their system depends on previously estimated BGP table information. As they indicated, lacking BGP views and dynamic changes on the target network results in errors.

3.2 Scriptroute

Reverse Path Tree (RPT) tool introduced by Scriptroute [44] brings a solution to handle inter-monitor redundancy by considering the structure of a destination node that is probed by multiple vantage points as a destination-rooted tree. When a trace towards a destination is completed by one of the vantage points, the information about the visited IP addresses in the trace is embedded in the measurement script. If the script recognizes that it reached part of an already traversed path, then it stops. However, this system does not deal with intra-monitor redundancy.

3.3 Doubletree

A common probe redundancy reduction system is by Doubletree [18]. Doubletree takes advantage of the tree-like structure of Internet routes. First, a tuning parameter is decided to specify a mid point for a path, and a vantage point starts probing both forwards to the destination node and backwards to the vantage point itself starting from the mid point. Each vantage point adds the traversed paths to local and global stop sets. This solution considerably reduces probing redundancy as it deals with both intra- and inter-monitor redundancies. However, because the vantage points need to share information about the visited paths (local and global
stop sets), this causes additional control traffic between the vantage points which reduces the efficiency and imposes an extra burden on the system.

### 3.4 Ingress Point Spreading

More recently three strategies have been introduced to reduce the redundant probing, namely, Subnet Centric Probing, Interface Set Cover, and Vantage Point Spreading [13]. Subnet Centric Probing utilizes subnetting structure of networks and expands the Doubletree method by eliminating the mid point parameter. Using the Levenshtein algorithm, they first calculate the edit distance between the paths that are visited. A high value of edit distance indicates that paths are distant while a low edit distance value shows that two paths are nearly identical. Hence, they simply distribute the probes to distinct IP prefixes by avoiding wasted probing. Ingress Point Spreading utilizes the ingress points of target ASes and apply a vantage point selection algorithm based on the inferred ingress points [12]. Their results show that there is a significant increase in the number of edges and vertices discovered compared to Ark, while the amount of probes stood nearly %50 of Ark’s. Finally, appointing VPs to destination nodes in a systematic manner gives highly efficient results both for the redundancy reduction and additional topology identification.

### 3.5 Cheleby

In Cheleby [40, 34], they apply partial traces to destination nodes and save a considerable amount of redundant probes. They also distribute the VPs in teams based
on the geographical location of VPs with the expectation of physically close VPs to reveal different edges of a network is small. So, they probe a target network from 1 VP of a team where a team contains geographically close VPs. In our work, we improved their partial trace method by including the contribution of ingress points of target subnets as they suggest. We also implemented a different approach for work load distribution to VPs.

The general deficiency of Rocketfuel is that its efficiency depends on prior rounds of probing results. Most of the studies presented to date that aspired to reduce the redundant probing that derives from tracing a same path excessively in a centralized manner. Even though these approaches could reduce probing overhead by adjusting how traceroute acts, they have trade-offs that cannot be ignored when the general aim of the Internet measurement study is to map a network as complete and efficient as possible. Doubletree’s method generates extra traffic burden on the network arising from the communication process of the vantage points. Scriptroute provides a solution to inter-monitor redundancy, but vantage points to share information with each other. Ingress Point Spreading also requires communication of ingress points among vantage points. In this thesis, we extend Cheleby [40, 34] approach in probe overhead reduction without control traffic between vantage points.
Chapter 4

InSub: Ingress to Subnet relation based Probe Reduction

In this chapter, we present ingress to subnet relation based probe reduction approach for large scale Internet topology measurement studies. InSub probe reduction approach focuses on two essential points. One is to discover the ingress points of subnets that the target AS announces, and the other is to efficiently distribute the burden of probing the subnets to all available vantage points based on the identified ingress points. As in any Internet topology measurement campaign, vantage point distribution is crucial for obtaining a complete unbiased graph [18, 13, 12]. Based on this motivation, our approach in the first phase consists of two steps:

1. Subnet Specific Ingress Point Identification (SSIPI)
2. Ingress to Subnet based Vantage Point Probing (ISVPP)
4.1 Subnet Specific Ingress Point Identification (SSIPI)

We first gathered AS-prefix data from RIPE-NCC [2] which contains the announced prefix information of each AS. We filtered the data by removing the overlapping smaller sized subnets (with the larger CIDR notation) from each AS. Namely, if there are multiple subnets announced by an AS and they overlap, we only keep the one which contains the others as the smaller subnets overlap with the larger one. This eliminates redundant probing of the subnets during Subnet Specific Ingress Point Identification (SSIPI).

4.1.1 Ingress Point Identification

When Internet topology mapping is focused on a target AS, probes to preceding ASes are typically redundant. Hence, we try to avoid tracing of routers outside the target AS to eliminate such redundant probes [34]. We achieve this by arranging the initial TTL of traces for a target subnet. We initialize the first trace to a target subnet starting from $TTL_{min}$, which is typically 1 but can be set to a higher value to bypass internal routers. We then set the initial TTL value of the rest of the traces toward the subnet based on the hop distance $Hi$ of the ingress point that the first trace revealed.

If any of the subsequent traces to the subnet have an ingress hop distance $Hi$ greater than the previously set TTL, then we do not change the initial TTL value. However, if any of the ingress hop distance $Hi$ of subsequent traces is smaller than the previously set TTL value, then we probe backwards until the new ingress point is identified, and update the initial TTL value.
For example, assume we have a network topology as shown in Figure 4.1. We send the first trace to Target IP1, and identify the ingress point at the 3rd hop from the vantage point VP. Hence, we set the initial TTL value to 3 for subsequent traces. Then, we send the second trace to Target IP2 with the initial TTL value of 3. As shown in the Figure 4.1, the second trace starts from hop ‘A’ as the ingress point for Target IP2 is actually on the 4th hop from the VP. In this case, we keep the TTL value as 3. For the 3rd trace, when we start the trace from the 3rd hop, i.e., ‘B’, we cannot identify the ingress point by probing only forward as the trace is already inside the target AS. Then, we probe backwards toward the VP until the ingress point is observed at hop distance 2 from the VP to Target IP3. Hence, we update the initial TTL value for the rest of the traces to 2. Finally, the 4th trace starts from 2nd hop ‘C’ for Target IP4.

During tracing, we gather the AS numbers of each IP address by looking up the hash table of AS-Prefix list to identify ingress points of ASes. We avoid using whois AS look-up tools to prevent the system from generating additional traffic for such resolution. Having the information of target AS number and target IP
address in the target prefix, we consider three different situations:

- **Probe reaches the target AS**: When the tracing probe reaches the target AS, we say that the last seen IP address on the path that does not belong to the target AS is an ingress point to the target AS prefix. However, there are some cases that, even the probe enters the target AS, there may be some unresponsive routers before the target AS is revealed. If there are any number of unresponsive hops between the target IP (which belongs to target AS) and the last seen IP (does not belong to target AS), then we consider the last seen IP as a predicted ingress point.

- **Probe does not enter the target AS**: In this case, we consider that the target AS is blocking the ICMP packets for the target subnet, so we log these cases as ‘BLOCKED’ probes. If all probes that we performed to a target subnet are ‘BLOCKED’, then we end up having no ingress point information toward that target subnet.

- **First seen IP address on the path belongs to the target AS**: A situation like this indicates that the vantage point is inside the target AS. Hence, there is no point of looking for an ingress point for the target AS.

### 4.1.2 Ingress Point Identification of a Subnet

In order to determine ingress points of a subnet, we probe the first and last IP addresses in each subnet within the AS-Prefix list of the given target AS. We then apply Ingress Point Identification method to traces toward both the first and last IP addresses of the target subnet. Then, we compare the identified ingress points of the first and last IP addresses of the subnet.
Figure 4.2: SSIP1 work schema, when the first and last IP addresses of a subnet enter the target AS through the same ingress point.

If the results show that these two probes enter the target AS through the same ingress point, then it is highly likely that all IP addresses in the selected subnet will reach the target AS through the same ingress point. For example, consider a subnet address of 193.255.106.0/24 as shown in Figure 4.2. As the first IP, i.e., 193.255.106.1, and the last IP, i.e., 193.255.106.254, of the subnet enter the target AS from the same ingress point A, we mark the ingress point of the whole subnet as A.

However, if the ingress point of the two traces differ, we then divide the target subnet into two subnets and probe each one separately. As the first IP address of the first subnet is already probed, we probe the last IP address of the first subnet to determine its ingress point. Similarly, as the last IP address of the second subnet is already probed, we probe the first IP address of the second subnet to determine its ingress point. We recursively continue to divide and probe the subnets until identified ingress points of each subnet match or the end points of the subnet are
Figure 4.3: SSIP work schema, when the first and last IP addresses of a subnet enter the target AS through different ingress points.

unreachable.

For example, consider a subnet address of 193.255.106.0/24 as shown in Figure 4.3. As the first IP, i.e., 193.255.106.1, and the last IP, i.e., 193.255.106.254, of the subnet enter the target AS from different ingress points, i.e., A and B, respectively, we split the subnet into two as 193.255.106.0/25 and 193.255.106.128/25. Then, we recursively try to determine ingress points of both subnets. In the presented example, the last IP of the first subnet, i.e., 193.255.106.126, also enters from the ingress point A. Hence, 193.255.106.0/25 subnet is marked to have an ingress point of A. As the first IP of the second subnet, i.e., 193.255.106.129, enters the target AS from ingress point C, 193.255.106.128/25 subnet is further divided into two subnets as 193.255.106.128/26 and 193.255.106.192/26.
Note that due to various reasons traces toward an IP address might end up without reaching any IP address of the targeted AS. In such instances, we can not determine the ingress point of the AS toward and discard such subnets.

### 4.2 Ingress to Subnet based Vantage Point Probing (ISVPP)

In this section, we discuss the mapping phase of a chosen AS via determined subnet IP addresses. In order to efficiently map a destination AS, we distribute the vantage points to target subnets based on the identified ingress points and update the previously generated subnet-ingress table while tracing toward IP addresses of subnets.

In our system, a central server coordinates vantage points in mapping a destination AS. After identifying ingress points, the server determines optimal vantage point set to trace a destination subnet of the chosen AS. Vantage points do not communicate with each other but simply obtain probing tasks from the server and report the results back to the server. The server also monitors vantage point status as vantage points such as PlanetLab are known to be unsustainable.

After identification of ingress points of subnets with respect to vantage points with SSIP, we determine vantage points that will trace through each ingress point. Ideally, we would like to trace every subnet via each of the ingresses. Different edges of a network can be identified if it is traced from different ingress points [12]. However, that is not possible due to the peering relation between ASes. In particular, ASes announce subnets through certain ingress points and this makes the subnet unreachable through other ingress points of the AS.
Additionally, we want to evenly distribute the probing workload of subnet tracing to all vantage points that can trace through each ingress point. Note that there are two optimization problems, (i) to evenly distribute workload among all vantage points that can reach a particular ingress point and (ii) to evenly distribute workload among all vantage points. The optimization of vantage point load distribution becomes additionally challenging due to unstable vantage points and varying probing times of destinations.

Our system generates a table of each subnet-ingress pair and their corresponding vantage points based on the previous subnet ingress identification mechanism. In assigning vantage point to probe a subnet, we consider the following factors.

- Status of vantage point
- Potential vantage point work load
- Vantage point hop distance to ingress

Assignment process starts with ingress-subnet pairs that are least covered. If there are multiple ready vantage points for the selected ingress-subnet pair, we select the one that is closest to the ingress point. This choice effectively reduces probing time as vantage points closer to the ingress will have a smaller round trip time (RTT). If there are multiple vantage points at the same distance to the ingress, we select the one that has the smallest potential work load where potential workload is determined by the number of IP addresses it can trace for remaining ingress-subnet pairs.

Figure 4.4 presents a situation of vantage point connections to ingress-subnet pairs previously gathered from the ingress identification phase. In this example
there are seven subnets (i.e., S1, S2, S3, S4, S5, S6, and S7) and three vantage points (i.e., VP1, VP2, and VP3).

First of all, the system generates an ingress-subnet table as shown in Table 4.1. Subnets S1, S2, S3, S4, S6, and S7 can be reached by ingress points, A, B, C, D, \{D, E\}, E and F respectfully. Vantage point VP1 reaches subnet S1 through ingress point A in 4 hops, subnet S2 through ingress point B in 3 hops, and subnets S4 and S5 through ingress point D in 4 hops.

After the Table 4.1 is filled with possible assignment of vantage points to ingress-subnet pair, we assign vantage points to ingress-subnet pair from the least covered ingress-subnet pair. In the example, the least covered ingress-subnet pairs are A-S1, C-S3, D-S4, and D-S5 having only 1 VP in their coverage set. Hence, these pairs are first assigned to their only vantage point, resulting with VP1 with three tasks (i.e., A-S1, D-S4, D-S5) and VP3 with one task (i.e., C-S3). As only VP2 is left with-
Table 4.2: Ingress-subnet pair assignment to vantage points

<table>
<thead>
<tr>
<th>VP1</th>
<th>VP2</th>
<th>VP3</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-S1</td>
<td>F-S7</td>
<td>C-S3</td>
</tr>
<tr>
<td>D-S4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D-S5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

out a task, we pick least covered ingress-subnet pair among the ones that can be performed by VP2. In this case, E-S5, E-S6, and F-S7 ingress-subnet pairs have two potential vantage points and V2 is closest to F-S7 among them. Hence, V2 is assigned F-S7 as shown in Table 4.2. The system will then wait for one of the vantage points to become free to assign remaining ingress-subnet pairs.

4.3 Subnetwork Probing

After we assign vantage points to the ingress-subnet pairs of a target AS, we trace IP addresses of the subnet starting from the corresponding ingress point. If a trace completes by revealing the same ingress point at the hop number of \( h \), then it is considered as a successful trace. However, there are cases where the previously identified ingress point of the trace toward destination IP address is seen at a hop number of \( h+x \) (where \( h_{max} - h \geq x \geq 1 \)) or a hop number of \( h-x \) (where \( 1 \leq x \leq (h-1) \)). If the ingress point is seen at a hop number of \( h+x \), we have redundant probes sent to preceding AS of the target AS. However, if the ingress point is seen at a hop number of \( h-x \), then we probe backwards to identify the ingress point of the destination IP address.

The backwards probing mechanism works by setting the initial hop of the trace to \( h-1 \) and max hop number of \( h_{max} \) to \( h \). Backwards probing continues until the ingress point is identified by recursively decrementing \( h \) and \( h_{max} \) values.
Moreover, the ingress point of an IP address might differ from previously identified ingress point of the subnet due to topological or traffic changes. In such cases, we update the ingress point of the subnet to newly identified one.

### 4.4 Tree based Tracing through an Ingress

As routing algorithms are based on (weighted) shortest paths, path traces from an ingress lead to tree like graphs. In order to reduce redundant probing due to traces that overlap, we modify the Doubletree [18] approach. Different from Doubletree, we do not coordinate between vantage points to stop traces. Instead each vantage point observes tree-like graphs from each ingress point that it is assigned for subnet tracing.

For each subnet, we probe the first IP address starting from the previously identified ingress point. If the target IP address is reached at a hop distance of $h$, then we perform backwards tracing for the rest of the IP addresses in the subnet starting from the hop distance of $h$. As subnet IP addresses are one hop away of each other, we expect that all of the IP addresses of a subnet to be at the same distance $h$ except only one of them being closer to the vantage point at a distance of $h - 1$.

Assume we have a trace from ingress point at distance $h_I$ to an IP address $IP_D$ of a subnet $S$, and that the trace concluded at a hop distance of $h_D$. For subsequent IP addresses $IP_S$ of the subnet $S$, we start the backward traces with a hop distance of $h_S = h_D$ until we see an IP address from any of the previous traces. Whenever we see an IP address $IP_x$ that was previously observed, we stop the trace.

Additionally, if we do not observe the target IP address $IP_S$, then we probe
forward to reach the target IP address $IP_S$. If the trace reaches the target IP address $IP_S$ or return 5 consecutive unresponsive routers, we stop the trace. If the $IP_S$ is observed at a greater hop distance, then $h_S$ is updated to the new distance. Note that likelihood of a responsive router after 5 unresponsive routers is very low [29].

Figure 4.5 presents a sample graph to clarify tree bases tracing of destination subnets through an ingress point. In the sample graph there are two subnets that will be traced. Assume we start with a trace toward D. In this case, we will start from hop distance of 5 from the ingress and increment TTL values till we reach destination D and obtain a trace of A-B-C-D. Then, trace to the next destination E of the subnet will start from the hop distance of 8 (i.e., distance of D) and backtrack till it merges with the tree. In this case, we will simply have E-C trace as backward
tracing will stop at C, an IP address that has already been observed. Trace to the destination C of the subnet will simply be ignored as C has already been observed. Additionally, trace to destination F of second subnet, will start from ingress point at hop distance of 5 and reveal a path of A-B-F. As F was at hop distance of 7, the next destination of subnet (i.e., G), will be traced starting from hop distance of 7. As 7th hop would reveal F, which is part of tree backward tracing will stop. However, as G has not been reached forward tracing will reveal F-G path. As G was at 8th hop, any other IP addresses of the subnet would have been traced backward starting from TTL of 8.
Chapter 5

Experimental Results

We performed our experiments utilizing Planetlab monitors around the world. We chose Planetlab as it provides a platform to run dynamic code that allow for different measurements. Because Planetlab nodes are not stable and some of the nodes did not support the required code, we could only utilize around 200 of PlanetLab nodes to perform our experiments.

We also relied on ANT Census’s Internet Address Censuses Data [39] for determining destination IP addresses of ASes. ANT Census data is gathered by pinging every allocated IP addresses once in two months. We utilized data from June - August 2015 probing period.

Our system, at first, fills the subnetworks, whose ingress points has been previously identified, with IP addresses in target IP data list. Then it gets the ingress point information (including the hop number $h$ of the ingress point) of each subnetwork from the previous run (SSIPI) results. The next step is to probe the selected target IP address starting from the hop number $h$ of the ingress point.
Table 5.1: Measured Autonomous Systems

<table>
<thead>
<tr>
<th>ASN</th>
<th>AS rank</th>
<th>AS peers</th>
<th>IPv4 prefixes</th>
<th>IPv4 addresses</th>
<th>Probed IPs</th>
</tr>
</thead>
<tbody>
<tr>
<td>174</td>
<td>2</td>
<td>4,768</td>
<td>134,631</td>
<td>648,411,904</td>
<td>110,411</td>
</tr>
<tr>
<td>1273</td>
<td>11</td>
<td>367</td>
<td>34,978</td>
<td>147,194,624</td>
<td>97,017</td>
</tr>
<tr>
<td>6762</td>
<td>7</td>
<td>370</td>
<td>82,712</td>
<td>238,719,232</td>
<td>85,065</td>
</tr>
<tr>
<td>2828</td>
<td>9</td>
<td>1,121</td>
<td>49,541</td>
<td>249,949,784</td>
<td>61,900</td>
</tr>
<tr>
<td>1239</td>
<td>20</td>
<td>650</td>
<td>35,281</td>
<td>329,687,040</td>
<td>61,059</td>
</tr>
<tr>
<td>3356</td>
<td>1</td>
<td>4,239</td>
<td>190,138</td>
<td>715,498,496</td>
<td>57,827</td>
</tr>
<tr>
<td>6939</td>
<td>8</td>
<td>3,980</td>
<td>74,594</td>
<td>305,295,872</td>
<td>39,847</td>
</tr>
<tr>
<td>6461</td>
<td>13</td>
<td>1,439</td>
<td>26,306</td>
<td>106,985,984</td>
<td>30,553</td>
</tr>
<tr>
<td>3491</td>
<td>12</td>
<td>569</td>
<td>50,952</td>
<td>184,550,656</td>
<td>26,054</td>
</tr>
<tr>
<td>6453</td>
<td>6</td>
<td>674</td>
<td>108,546</td>
<td>447,639,040</td>
<td>22,553</td>
</tr>
</tbody>
</table>

Some of the traces do not reach the target AS as they end up with unresponsive hops before revealing the target AS. Hence, we could not have any ingress information of some of the subnetworks.

We selected 10 medium sized ASes with high customer cone based on the CAIDA rankings by customer cone size [1]. Table 5.1 shows the target ASes selected for mapping. Note that AS rank is based on customer cone size as determined by CAIDA. AS peers indicate number of neighboring ASes as identified from BGP announcements and traceroutes; IPv4 prefixes and IPv4 addresses are based on BGP announcements of ASes as collected by CAIDA. Additionally, probed IP addresses are obtained from ANT Census data along with IP addresses observed during ingress identification of subnets. Note that presented tables and figures will be sorted based on the probed IP addresses of each AS.

5.1 Subnet Specific Ingress Point Identification (SSIPI) Phase

In this section, we perform Subnet Specific Ingress Point Identification (SSIPI) phase on the selected ASes during Nov 2015.
Figure 5.1 presents the subnet that were identified based on the announced prefixes of ASes during SSIPPI phase. The identified subnets are more than the announced IP prefixes as we split a prefix if it is observed via different ingresses from any of the vantage point. We obtained 2.5 times more subnets on average of all ASes with only AS6939 having fewer subnets than announced prefixes. Note that there were overlaps in announced prefixes and we ignored smaller ones when determining initial subnets, which were then split based on ingress reachability.

We observe a large number of prefixes that we could not identify an ingress for (indicated with red bars in the Figure 5.1). As prefixes are split recursively to determine ingress points sub-prefixes that do not reveal ingress points for edge IPs are marked as non-discovered. Additionally, the identified subnet prefixes might not correspond to a single physical link but it is rather the sub-prefix that we observe to enter through the same ingress with from any vantage point that can reach it.

Figure 5.2 presents destinations selected from end-points of subnets during SSIPPI. Blue bars indicate the total number of destinations traced while green bars indicate the traces that reached the targeted IP address. As destination IP ad-
Figure 5.2: SSIPI destination IPs per AS (log scale). Blue bars indicate the number of probed IPs while green bars indicate the number of reached IPs.

dresses are simply chosen from the end points of subnets, they might not correspond to a real system. Hence, there is a much larger number of IP addresses that could not be reached. On average only 26.5 % of traced IPs of ASes were reached while in total only 15.7 % of 2 447 249 traced IPs were reached.

Five ASes with largest number of non-discovered subnets (red-bars in Figure 5.1) had the lowest reachability rates as well. Note that, however, even though an IP address might not be reached, the trace towards that IP address might still reveal the ingress point of the subnet.

Figure 5.3 presents the number of probes that were sent to each AS during Ingress to Subnet discovery phase. Note that different from commonly used 3 probes per hop, we sent 2 probes per hop during ingress identification. Also, we eliminate redundant probing by starting traces from hop distance of closest ingress identified so far. While the blue bars indicate the total number of probe packets generated, the green bars indicate the total number of probes if the traces were to be stopped right after determining the ingress point of the AS. As we are looking for the same ingress from both ends of a subnet, observing a different IP address at the ingress hop would indicate the mismatch. On average 58.2 % of probes to
Figure 5.3: SSIPI probes per AS (log scale). Blue bars indicate the number of probes in traces while green bars indicate the number of probes if stopped at ingress.

ASes would have been eliminated if traces were stopped at the ingress point of destination ASes while overall 60.4% of all of the 238 745 538 probes would have been eliminated.

Figure 5.4 presents the number of IP addresses that were discovered during the SSIPI phase. While the blue bars indicate the total number of observed IPs, the green bars indicate the total number of observed IPs if the traces were to be stopped right after determining the ingress point of the AS. On average 12.7% of IPs would have been observed if traces were stopped at the ingress point of destination ASes while overall 8.7% of all of the 462 973 IPs would have been observed.

Figure 5.5 presents the number of edges that were discovered during the SSIPI phase. While the blue bars indicate the total number of observed edges, the green bars indicate the total number of observed edges if the traces were to be stopped right after determining the ingress point of the AS. On average 30.4% of edges would have been observed if traces were stopped at the ingress point of destination ASes while overall 24.6% of all of the 742 440 edges would have been observed.
5.2 Ingress to Subnet based Vantage Point Probing (ISVPP) Phase

In this section, we perform Ingress to Subnet based Vantage Point Probing (ISVPP) phase on the selected ASes during Nov 2015. In the following, we compared our InSub approach to commonly utilized aggressive approach of tracing all destinations from all vantage points and tree based approaches.

Figure 5.6 presents the destination IP addresses that were probed during ISVPP
phase of AS topology discovery based on ingress to subnet relation. The total of 592,286 IP addresses are combination of the 258,955 from ANT Census’s Internet Address Censuses Data [39] and 384,645 from IP addresses observed during the SSIPI phase of ingress to subnet discovery of vantage points. Blue part indicates IP addresses unique to ANT census, green part indicates IP addresses unique to SSIPI discovery phase, and red part indicates common intersection of both. Note that ANT Census data contain IP addresses that replied to a ICMP ping requests between June - August 2015 whereas SSIPI data contains IPs that generated ICMP Time Exceeded replies.

Figure 5.7 presents the number of vantage points used with each of the AS. Blue bars indicate the number of vantage points utilized in measurement of each AS while additional red bars indicate the vantage points that were selected but did not complete probing. The red vantage points are due to PlanetLab monitors that either failed or became unreachable during the measurement. While AS6453 is the only one that was probed by all of the assigned vantage points, AS3356 had the largest number of lost vantage points with 21 monitor failures.

Figure 5.8 presents the number of ingress points identified per AS. Higher num-
Figure 5.7: Vantage points per AS. Blue bars indicate the number of vantage points utilized in measurement of each AS. Additional red bars indicate vantage points that were selected but did not complete probing.

Figure 5.8: Ingresses per AS (log scale)

The number of ingresses typically allow for better mapping of an AS as the ingresses are the real vantage into the AS.

Figure 5.9 presents the number of probes generated by each approach of All (where all IP addresses are traced from all available VPs), InSub (where only one vantage point traced subnets with respect to ingress points), Tree (where each vantage point traced all IP addresses in a tree-like fashion), and InSub & Tree (where Tree approach is applied on top of InSub). Note that, all approaches started tracing of destination IP addresses from the identified hop distance of ingress to subnet with respect to vantage point from the SSIP phase.
We observe that in total, \textit{All}, \textit{Tree}, \textit{InSub} and \textit{InSub & Tree} generates 1 244 190 006, 579 337 950, 555 686 688 and 437 499 165 probes, respectively. Compared to extensive \textit{All} approach, on average of all AS\text{s}, \textit{Tree}, \textit{InSub} and \textit{InSub & Tree} approaches generate 50.9 \%, 54.5 \% and 64.4 \% fewer probes, respectively. Overall, \textit{InSub} approach saves a little more probes (i.e., 3.6 \%) than \textit{Tree} approach while they can together further reduce probes. Considering sum of all probes, \textit{Tree}, \textit{InSub} and \textit{InSub & Tree} approaches generate 53.4 \%, 55.3 \% and 64.8 \% fewer probes, respectively compared to \textit{All}.

Figure 5.10 presents the number of observed IP addresses with each approach and collectively. Combined indicates the union of all approaches. In total, \textit{All}, \textit{Tree}, \textit{InSub} and \textit{InSub & Tree} approach discovered, 581 706, 578 145, 568 930 and 562 513, respectively, of all of the 608 667 combined IP addresses.

Figure 5.11 presents the ratio of IP addresses discovered by each method compared to the union of all approaches. On average, \textit{All}, \textit{Tree}, \textit{InSub} and \textit{InSub & Tree} approach discovered 96.0 \%, 95.2 \%, 94.4 \% and 93.3 \% of each AS\text{'}es combined IP addresses, respectively. Considering sum of all IP addresses of each method, \textit{All}, \textit{Tree}, \textit{InSub} and \textit{InSub & Tree} approach discovered 95.6 \%, 95.0 \%, 93.5 \% and 92.4 \% of each AS\text{'}es combined IP addresses, respectively.
Figure 5.10: Observed IPs per AS for each method (log scale)

Figure 5.11: Ratio of IPs discovered by each method

% of combined IP addresses, respectively.

Figure 5.12 presents the number of observed edges with each approach and collectively. Combined indicates the union of all approaches. In total, All, Tree, InSub and InSub & Tree approach discovered 2 842 622, 2 698 174, 2 581 213, and 2 476 526, respectively, of all of the 3 170 984 combined edges.

Figure 5.13 presents the ratio of edges discovered by each method compared to the union of all approaches. On average, All, Tree, InSub and InSub & Tree approach
discovered 89.2 %, 83.2 %, 83.9 % and 79.9 % of each AS’es edges, respectively. Considering sum of all IP addresses of each method, All, Tree, InSub and InSub & Tree approach discovered 89.6 %, 85.1 %, 81.4 % and 78.1 % of combined IP addresses, respectively.

Note that, while InSub misses 0.7 % IP addresses compared to Tree approach, it observes 0.5 % more edges among them.
5.3 Ingress reachability of Subnets from Vantage Points

In this section, we analyze the reachability of ingresses from vantage points. Considering an AS with 8 ingresses that can be probed via 6 vantage points in Figure 5.14, we can analyze both ingresses per vantage point and vantage points per ingress.

First, we plot the distribution of ingresses reachable from a particular number of vantage points. Considering sample graph in Figure 5.14, there are four ingresses reachable from just one vantage point, three ingresses that are reachable from exactly two vantage point, and one ingress that is reachable from exactly three vantage points.

In Figure 5.15 each point indicates the number of ingresses reachable from a particular number of vantage points. That is, in AS 174, there are 145 ingresses
that can be reached only from one vantage point, 100 ingresses that can be reached from two vantage points, and so on.

Note that the ingresses reachability from a particular number of vantage points seems to be power-law-like distribution where there are large numbers of ingresses that can be reached from very few vantage points while there are very few ingress points that can be reached from majority of vantage points. This indicates that we should prioritize probing from ingresses with limited reachability. An issue with the vantage points that can reach them indicates we will loose visibility through that ingress.

Next, we plot the distribution of vantage points reaching a particular number of ingresses. Considering sample graph in Figure 5.14, there are two vantage points reaching just one ingress, two vantage points reaching exactly two ingresses, one vantage point reaching exactly three ingresses, and one vantage point reaching exactly four ingresses.

In Figure 5.16, each point indicates the number of vantage points that reach a given number of ingresses. That is, in AS 174, there are 2 vantage points that can reach only two ingresses, 1 vantage point that can reach four ingresses, and so on.

Different from ingress reachability from vantage points, vantage points do not show such a skewed distribution in terms of the number of ingresses they can reach. However, the distributions are not uniform either as we do not observe a consistent pattern. This indicates that vantage points’ ingress reachability distribution depends on the particular destination AS.

Finally, we analyzed the number of IP addresses and edges that were discovered through each ingress with each of the methods.
Figure 5.15: Vantage points per Ingress (log-log scale)
Figure 5.16: Ingresses per vantage point
Figures 5.17, 5.18, 5.19, 5.20, 5.21, 5.22, 5.23, 5.24, 5.25 and 5.26 show the number of IP addresses and edges identified per ingresses for four different approaches for AS174, AS1273, AS6762, AS2828, AS1239, AS3356, AS6939, AS6461, AS3491, and AS6453, respectively. Note that, ingresses are ranked based on the number of IP addresses in each of the figures.

Overall, figures indicate that the topological view from each ingress is not same. Especially in ASes with many ingress points, only few ingresses provide large number of IP addresses and edges. For instance, in AS174, while about 50 ingresses identify more than 20 000 IP addresses more than 150 ingresses could identify less than 100 IP addresses. We believe this is mainly due to hot potato routing as ASes prefer to dump traffic to neighboring ASes through closest ingress. This can also been seen in vantage point reachability of ingresses in Figure 5.16 as there are only very few vantage points that can reach more than 50 ingresses among all mapped ASes.

One interesting thing here is that there seem to be some ingresses through which a large number of IP addresses are seen but with a much smaller number of edges among them. That is because of the unresponsive routers that did not reveal links between IP addresses.
Figure 5.17: Topology discovery ranking of ingresses - AS 174 (log scale)

Figure 5.18: Topology discovery ranking of ingresses - AS 1273 (log scale)
Figure 5.19: Topology discovery ranking of ingresses - AS 6762 (log scale)

Figure 5.20: Topology discovery ranking of ingresses - AS 2828 (log scale)
Figure 5.21: Topology discovery ranking of ingresses - AS 1239 (log scale)

Figure 5.22: Topology discovery ranking of ingresses - AS 3356 (log scale)
Figure 5.23: Topology discovery ranking of ingresses - AS 6939 (log scale)

Figure 5.24: Topology discovery ranking of ingresses - AS 6461 (log scale)
Figure 5.25: Topology discovery ranking of ingresses - AS 3491 (log scale)

Figure 5.26: Topology discovery ranking of ingresses - AS 6453 (log scale)
Figures 5.27, 5.28, 5.29, 5.30, 5.31, 5.32, 5.33, 5.34, 5.35 and 5.36 show the number of IP addresses and edges identified per vantage point for four different approaches for AS174, AS1273, AS6762, AS2828, AS1239, AS3356, AS6939, AS6461, AS3491, and AS6453, respectively. Note that, vantage points are ranked based on the number of IP addresses in each of the figures.

In the figures, we observe that vantage points have a similar number of observed IP addresses and edges in \textit{All} and \textit{Tree} approaches. However, as \textit{InSub} approach pairs ingress to subnet pairs to vantage points, there is a skewed observation through vantage points for \textit{InSub} and \textit{InSub \& Tree} approaches. Hence, with \textit{InSub} approach one should distribute probing dynamically to vantage points so that they collect similar amount of network measurement.

Some of the vantage points return very few IP addresses and edges compared to others. These were mainly due to an issue at vantage point where it could not complete assigned probing tasks. Additionally, we observe some of the vantage points discover considerably less edges compared to IP addresses. This might be due to firewalls, blocking a source IP address and replying with random IP addresses. Hence, during processing of traces, one should be careful of such short replies.
Figure 5.27: Topology discovery ranking of vantage points - AS 174 (log scale)

Figure 5.28: Topology discovery ranking of vantage points - AS 1273 (log scale)
Figure 5.29: Topology discovery ranking of vantage points - AS 6762 (log scale)

Figure 5.30: Topology discovery ranking of vantage points - AS 2828 (log scale)
Figure 5.31: Topology discovery ranking of vantage points - AS 1239 (log scale)

Figure 5.32: Topology discovery ranking of vantage points - AS 3356 (log scale)
Figure 5.33: Topology discovery ranking of vantage points - AS 6939 (log scale)

Figure 5.34: Topology discovery ranking of vantage points - AS 6461 (log scale)
Figure 5.35: Topology discovery ranking of vantage points - AS 3491 (log scale)

Figure 5.36: Topology discovery ranking of vantage points - AS 6453 (log scale)
5.4 Network Characteristics

In this section, we present the network characteristics of collected topologies by each approach.

Figure 5.37 presents the average degree of each graph. We observe that most of graphs have a similar average degree. Only in AS6762 and AS3491 there is an average degree difference of more than 2 degrees.

![Average Degree of collected topologies.](image)

Figures 5.38, 5.39, 5.40, 5.42, 5.43, 5.44, 5.45, 5.46 and 5.47 show the degree distribution for four different approaches for AS174, AS1273, AS6762, AS2828, AS1239, AS3356, AS6939, AS6461, AS3491, and AS6453, respectively. We observe that the degree distributions show similar patterns in all ASes.
Figure 5.38: Degree distribution of resulting graphs - AS 174 (log-log scale)

Figure 5.39: Degree distribution of resulting graphs - AS 1273 (log-log scale)

Figure 5.40: Degree distribution of resulting graphs - AS 6762 (log-log scale)
Figure 5.41: Degree distribution of resulting graphs - AS 2828 (log-log scale)

Figure 5.42: Degree distribution of resulting graphs - AS 1239 (log-log scale)

Figure 5.43: Degree distribution of resulting graphs - AS 3356 (log-log scale)
Figure 5.44: Degree distribution of resulting graphs - AS 6939 (log-log scale)

Figure 5.45: Degree distribution of resulting graphs - AS 6461 (log-log scale)

Figure 5.46: Degree distribution of resulting graphs - AS 3491 (log-log scale)
Figure 5.47: Degree distribution of resulting graphs - AS 6453 (log-log scale)

Figure 5.48: Average Degree of collected topologies.

Figure 5.48 presents the assortativity of each graph. Assortativity value is in the range of [-1, 1]. A value of 1 indicates assortative behavior, i.e., high degree nodes are linked to other high degree nodes, a value of -1 indicates disassortative behavior, i.e., high degree nodes are linked to low degree nodes, and a value of 0 indicates non-assortative behavior. We observe that most of graphs have a similar assortativity values. All networks are either slightly disassortative (e.g., AS174 and AS2828) or disassortative. Only in AS3356 there is difference of more than 0.1.

Figure 5.49 presents the clustering coefficient of each graph. Clustering value is in the range of [0, 1] and is calculated as the number of triangles divided by the number of triplets in the graph. A value of 0 indicates lack of triangles in the graph.
while a value of 1 indicates a clique topology. We observe that all graphs have a similar clustering coefficient.

Figure 5.50 presents the clustering coefficient distribution with respect to node degree of each AS. We observe that clustering coefficient distributions are also similar for each probing approach.
Figure 5.50: Clustering coefficient distribution (log-log scale)
Chapter 6

Summary and Future Work

This thesis presents a new approach for minimizing redundant probes while measuring a destination network topology. InSub approach is based on subnet ingress identification along with vantage point allocation to utilize ingress points of subnets. We first identify ingress points of announced subnet prefixes by tracing subnet end-points. During this process, we recursively split subnets that enter the destination AS through different ingress points. Then, we probe each subnet from one vantage point per identified ingress of the AS. This allows us to considerably reduce generated probes without significant loss in observed topological information. Overall, our method was able to reduce probing overhead with comparable results to the commonly used Tree approach.

Additionally, we analyzed the reachability of ingress points and vantage points. Our results indicate that while vantage points have similar discovery potential in general, ingress points are not uniformly reachable. The location of vantage points with respect to ingress points of ASes play an important role into whether one can probe an AS trough that ingress point.
As we did not account for loss of vantage points during measurement, we lost some of the information with InSub approach. As InSub traces each ingress to subnet from one vantage point, loss a vantage point typically means that visibility is lost. Hence, in the future a more dynamic system could better replace lost vantage points with respect to ingress points.

Additionally, we realized that our vantage points were able to capture only a small fraction of peering relations of ASes. This indicates, we were not able to probe through most of the ingress points of ASes. In the future, we could look for a larger number of vantage points to have a better visibility of a target AS.
Bibliography


[44] Neil T Spring, David Wetherall, and Thomas E Anderson. Scriptroute: A