OMAR’S THESIS

by

OMAR LOUDGHIRI

Submitted in partial fulfillment of the requirements for the degree of
Master’s of Science in Computer Science

Department of Computer Science and Data Science

CASE WESTERN RESERVE UNIVERSITY

August, 2024

CASE WESTERN RESERVE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

We hereby approve the thesis/dissertation of

Omar Loudghiri

candidate for the degree of **Master's of Science in Computer Science**[1].

Committee Chair

An Wang

Committee Member

Mark Allman

Committee Member

Vincenzo Liberatore

Committee Member

Mehmet Koyuturk

Date of Defense

July 10th, 2024

[1] We also certify that written approval has been obtained for any proprietary material contained therein.

DEDICATION

TBD

TABLE OF CONTENTS

LIST OF TABLES

Number *Page*

LIST OF FIGURES

ACKNOWLEDGEMENTS

TBD

Omar’s Thesis

Abstract

by

OMAR LOUDGHIRI

TBD

Chapter 1

INTRODUCTION

Networks are fundamental to modern society, enabling communication, commerce, and data exchange. These networks are built from an array of user devices, routers, and servers. Understanding the architecture and dynamics of these networks is crucial for enhancing their efficiency and reliability. Research has long focused on mapping and analyzing network structures to improve their performance and resilience (Paxson, 1996). A key aspect of network optimization is load balancing, a practice employed by network operators to manage traffic distribution and scale network capacity. Keeping a current understanding of both network structures and load balancing techniques is essential for maintaining robust and efficient network operations.

Load balancing is a critical network management technique that distributes traffic across multiple servers or network paths. A load balancer is a piece of network equipment that can direct packets to one of several available routes, anticipating that each route will yield a similar result. This distribution helps prevent any single server from becoming overwhelmed and ensures efficient use of network resources.

Originally developed as network-based hardware, load balancing now often functions within routers. It plays a crucial role in modern infrastructure by ensuring high availability, scalability, security, and performance. Applications today must handle millions of simultaneous sessions, and load balancers dynamically distribute this traffic across servers with duplicate data, ensuring reliable and fast data delivery. This process can also provide redundancy; if a server fails, traffic may be redirected to maintain continuous access.

Load balancing has several benefits, including enhancing security by potentially minimizing attack surfaces and rerouting traffic if a server is compromised. It also optimizes performance by managing resource use and traffic spikes. Various algorithms, such as round-robin and least connections, help distribute traffic based on real-time conditions. However, it is important to note that load balancing itself can become a single point of failure if the load balancer goes down.

1.1 Project Motivation and Goals

The primary motivation for this project is to quantify the prevalence and characteristics of load balancing in the Internet. By measuring load balancing behavior and mapping the presence of load balancers across commonly used paths in the Internet, this research aims to enhance our understanding of their impact. The project focuses on identifying and categorizing load balancers, analyzing their deployment, and understanding the resulting network paths and their implications for the broader Internet infrastructure.

1.2 Impact of Load Balancing on Internet Reliability

Load balancing significantly enhances the reliability of the Internet by distributing traffic across multiple servers or paths. This helps prevent any single point of failure, manage high traffic volumes, and ensure continuous availability of services. Effective load balancing reduces latency, avoids downtime, and ensures smooth data delivery, contributing to a better user experience.

The primary motivation for this project is to explore how load balancing impacts Internet reliability. Understanding this impact is crucial, as load balancing directly affects the network's ability to handle failures, traffic spikes, and varying load conditions. Despite its importance, the extent of load balancing's contribution to Internet reliability has not been extensively quantified.

This research aims to measure and analyze the role of load balancing in maintaining Internet reliability. By mapping the global presence of load balancers and characterizing their behavior, we seek to provide a clearer picture of their impact on network resilience. Using tools like the Multipath Detection Algorithm (MDA), we will quantify how load balancing affects network performance.

In summary, load balancing is key to Internet reliability, and this project aims to quantify its significance.

1.3 Areas of Study

Chapter 2

RELATED WORKS

In [**Augustin2007**], Augustin et al. present a comprehensive study on load-balanced paths, highlighting the significance of recognizing load balancing in contemporary networking by demonstrating how it affects traffic distribution and path diversity. Our goal is to update the community's understanding of load balancing in the Internet 17 years after [**Augustin2007**] was published.

The authors enhanced a traceroute-like tool called Paris traceroute, designed to find all paths between a pair of hosts. Their methodology involves identifying load-balancing routers and characterizing the load-balanced paths. By conducting measurements from 15 sources to over 68,000 destinations, their study reveals that the traditional single-path concept no longer holds. They found that 39% of source-destination pairs traverse a load balancer, and this percentage rises to 70% when considering paths between a source and a destination network.

This study was significant in showing the prevalence of load balancers. The insights gained from this work are critical for developing more realistic network models and improving the design and reliability of Internet applications.

2.1 Traceroute

Traceroute is a network diagnostic tool used to track the path packets take from one IP address to another. It works by sending packets with gradually increasing time-to-live (TTL) values. Each router along the path decreases the TTL of the packet by one. When the TTL reaches zero, the router sends back an error message to the sender, revealing its IP address. This process is repeated with incrementing TTL values, allowing traceroute to map out the entire route to the destination.

Traceroute provides insights into the structure and behavior of the network by identifying each hop along the route. However, traditional traceroute may not handle load-balanced paths well, as it can be misled by the varying paths packets may take. To address this, Paris traceroute and MDA are used to obtain more accurate measurements by maintaining consistent flow identifiers, thus avoiding misinterpretation caused by load balancers.

2.2 Multipath Detection Algorithm (MDA)

The Multipath Detection Algorithm (MDA) is a key component of Paris traceroute, designed to identify and trace multiple load-balanced paths between a source and a destination. Traditional traceroute tools often fail to detect load balancing because they assume a single path. In contrast, MDA systematically discovers all paths by varying flow identifiers in probe packets.

The MDA operates hop-by-hop, sending probes to identify all interfaces at each hop. For a given interface r at hop $h-1$, MDA generates several flow identifiers to ensure probes reach r. It then sends these probes one hop further to discover the next-hop interfaces $s_1, s_2, \ldots, s_n$.

To determine the number of probes k needed to discover all paths with a high degree of confidence, MDA assumes r is part of a load balancer that splits traffic evenly across n paths. If fewer than n interfaces are found, MDA stops. Otherwise, it increases n and sends additional probes to test the hypothesis.

To identify whether a load balancer uses per-packet or per-flow balancing, MDA sends probes with a constant flow identifier. If responses come from multiple interfaces, it indicates per-packet balancing. If all responses come from the same interface, it suggests per-flow balancing. MDA uses statistical methods to ensure a high level of confidence (typically 95%) in its classification.

For instance, to reject the hypothesis of $n = 2$ with 95% confidence, MDA sends $k = 6$ probes. If load balancing across up to 16 interfaces is suspected, MDA may send up to $k = 96$ probes to ensure all paths are discovered. This process allows

MDA to effectively enumerate all paths and classify the type of load balancing in use.

2.3 Usage of Paris Traceroute

The paper by Augustin et al. [**Augustin2007**] is one of the few studies that actively measures the presence and behavior of load balancers in the Internet. Although their work provides a strong foundation, further investigation is needed to account for the evolving nature of Internet infrastructure and load balancing techniques.

Building on the foundational work of Augustin et al., this research aims to further explore the presence and behavior of load balancers in the Internet. By leveraging Paris traceroute and its MDA, we conduct extensive measurements to map the global distribution of load balancers and analyze their impact on network performance and reliability.

This study will add to the existing knowledge by:

- Including a diverse set of source-destination pairs.
- Investigating the effects of load balancing on different types of network traffic and applications.

2.4 Scamper

2.4.1 Introduction

Packet probing experiments capture simple measurements—typically delay, loss, reordering, and topology—that provide valuable insights into the structure and behavior of the Internet. For instance, early studies on packet size and delay led to improvements in the TCP RTO algorithm. As the Internet has grown, measuring it has become more complex, both technically and methodologically. Over the past decade, researchers have developed and operated many large-scale Internet measurement platforms, each involving significant software development.

In 2005, facing funding challenges, the Internet measurement community organized a workshop to plan a collaborative, community-oriented network measurement infrastructure. The workshop report highlighted the need for better organization of large-scale measurements.

This paper addresses a small part of this problem by focusing on building a packet-prober for large-scale measurements and well-defined data archiving. A packet-prober should simplify the coordination of measurements, provide APIs for operating system differences, offer accurate timing information, and produce detailed, easy-to-process output. This approach helps researchers avoid system programming and administrative challenges, allowing them to implement new measurement techniques and focus on analysis and validation.

2.4.2 Scamper Features

Scamper is a powerful packet-prober designed to support large-scale Internet measurement. It includes feature-rich implementations of traceroute, ping, MDA traceroute, four alias resolution techniques, Sting, and parts of TBIT.

2.4.3 Multipath Detection Algorithm (MDA) in Scamper

Scamper implements the Multipath Detection Algorithm (MDA) described by Augustin et al. to infer all interfaces visited between a source and destination in a per-flow load-balanced Internet path. MDA achieves this by deliberately varying the flow identifier that a router may compute when load balancing. Probes with different flow identifiers may take different paths, thereby revealing different parts of the forward IP path.

In addition to the ICMP and UDP methods originally implemented by Augustin et al., which vary the ICMP checksum and UDP destination port values, Scamper implements a UDP method that varies the source port instead of the destination port. This prevents the probes from appearing as a port scan and enables probing past firewalls that block UDP probes to ports above the usual range used by traceroute. Scamper also implements TCP methods that vary the flow identifier by changing either the source or destination port, depending on the user's choice.

Scamper's MDA traceroute functionality was used to conduct scheduled data collection throughout this project.

2.5 Classification of Load Balancing in the Internet and Multipath Classification Algorithm (MCA)

Recent advances in programmable data planes, software-defined networking, and the adoption of IPv6 have enabled more complex load balancing strategies. Almeida et al. introduced the Multipath Classification Algorithm (MCA), which enhances the existing Multipath Detection Algorithm (MDA). While MDA systematically varies probes' flow identifiers to identify load-balanced paths, MCA extends this by considering arbitrary combinations of bits in the packet header for load balancing.

The key contributions of MCA include:

- **Enhanced Classification:** MCA identifies the specific bits in the packet header used by load balancers, providing a more detailed and accurate classification than MDA, which primarily varies transport port numbers and the ICMP checksum.

- **Efficiency Optimizations:** The researchers developed optimizations to reduce the probing cost. MCA achieves this with a modest increase in the number of probes, only 34% higher than MDA, while maintaining accuracy.

- **Comprehensive Measurements:** Through large-scale measurement campaigns, MCA characterizes load balancing on both IPv4 and IPv6 Internet

> paths. The results show that load balancing is more prevalent and sophisticated than previously reported.

The study revealed that 74% of IPv4 and 56% of IPv6 routes traverse at least one load balancer. Additionally, 23% of IPv4 and 18% of IPv6 load balancers have three or more next hops, indicating complex load balancing configurations.

Building on this work, part of our research uses MCA to map the global presence of load balancers and analyze their impact on network performance and reliability. In the next section, we will discuss how MDA and MCA were integrated into our research and the rationale behind choosing these methodologies.

2.6 MDA vs MCA

While MCA offers significant improvements in identifying and classifying load balancers by considering a broader range of packet header bits, it is less accessible for fine-tuning and practical use. For our research, we opted to use MDA due to its better integrability with existing tools and faster runtime performance. MDA's established methodologies and ease of implementation make it a more practical choice for large-scale measurements. The decision to use MDA is further justified by its compatibility with current infrastructure and the ability to conduct measurements efficiently.

Chapter 3

METHODOLOGY

This chapter details the methods used to collect and analyze data for detecting and characterizing load balancers in network paths. We employed two lists derived from the Alexa Top 1 Million Websites list and performed Paris traceroute measurements to these hostnames. The collected data was then processed to identify load balancers and analyze their behavior.

3.1 Background Information

The Alexa Top 1 Million Websites list was used to obtain hostnames for our study. Additionally, the Team Cymru IP to ASN mapping service was utilized to resolve IP addresses to their corresponding Autonomous System Numbers (ASNs). These lists and tools provided the necessary foundation for our analysis.

3.1.1 Alexa Top 1 Million Websites

The Alexa Top 1 Million Websites list was utilized to obtain hostnames for this research. A current version of the Alexa list was obtained when we started our data collection in February 2023. The Alexa list is widely used in network measurement studies due to its popularity, even though it is not known for high accuracy, particularly for lower-ranked sites [**Alexa**].

For our study, we took two different subsets of this list:

- The top 2000 websites.
- A random selection of 2000 websites from the top 100,000 sites.

These subsets allowed us to collect a diverse set of data through Paris traceroute runs to the selected hostnames.

3.1.2 Team Cymru IP to ASN List

For this project, the Team Cymru IP to ASN mapping service was utilized to resolve IP addresses to their corresponding Autonomous System Numbers (ASNs). Team Cymru provides a service dedicated to mapping IP numbers to BGP prefixes and ASNs, based on BGP feeds from over 50 peers, updated at four-hour intervals [**Cymru**].

In this project, the Team Cymru service was essential for mapping IP addresses to ASNs. We collected ASN-to-IPv4 address information from Team Cymru every month, with their permission. This list was used to cross-reference the IPs identified as load balancers, their next hops, and their destination IPs, providing detailed insights into the load balancers discovered.

Here is an example of an entry with definitions of the fields:

Field	**Example**
BGP Origin ASN	23489
BGP Peer ASN	199.88.100.1
BGP Prefix	199.88.100.0/24
Prefix Country Code (assigned)	US
Prefix Registry (assigned)	arin
Prefix Allocation Date	1994-03-28
ASN Country Code (assigned)	US
ASN Registry (assigned)	arin
ASN Allocation Date	1994-03-28
ASN Description	MARINK12, US

Table 3.1: BGP and ASN Information

3.2 Discovering Load Balancers

The primary objective of this research is to record the paths between our vantage point and a set of popular hosts, and to detect and characterize load balancers along these paths. To achieve this, we used two distinct lists obtained from the Alexa Top 1 Million Websites list.

3.2.1 Measurement Frequency and Timeline

To ensure the feasibility of daily measurements, 2000 IP addresses were chosen for the Paris traceroute process. Each IP takes an average of 40 seconds to return a complete trace with load balancer information. This duration allows the script to run through 2000 IPs in approximately 23 hours, making it possible to conduct measurements on a daily basis.

The goal of daily measurements is to assess trends and variations in load balancing behavior over time. To maintain feasibility, we alternated between the Top-2000 list and the Rand-2000 list each day. This rotation allowed us to gather comprehensive data while managing the logistical constraints of daily measurements.

The measurements were run continuously from January 23, 2024, to April 16, 2024, on a Linux machine at the International Computer Science Institute (ICSI) in Berkeley, CA. This timeline ensured the collection of extensive data over several months, capturing potential variations and trends in load balancing behavior and network topology over time.

3.2.2 IP Lists

Two distinct lists were used for this study:

- **Top-2000 List:** This list includes the top 2000 domains from the Alexa list. It is designed to cover an important chunk of the most used websites around the Internet, ensuring that the analysis captures the behavior and infrastructure of significant routes.

- **Rand-2000 List:** This list comprises 2000 random domains selected from the top 100,000 websites on the Alexa list, with a new random selection made each time we ran the measurement. It aims to provide a well-rounded analysis of the Internet's topology by including popular but not exclusively top-ranked sites.

Using the top 2000 websites allows us to measure load balancers on sites that are heavily accessed, providing insights into the infrastructure of widely used services. The random selection of 2000 sites from the top 100,000 ensures a broader view of the Internet's topology, capturing data from a diverse set of sites.

3.3 Data Analysis

The data analysis process involved systematically processing and analyzing our collected traceroute data. This analysis aimed to identify and categorize load balancers, as well as to visualize the results effectively.

3.3.1 Paris Traceroute on IP Lists

For each domain in the Rand-2000 and Top-2000 lists, the Paris traceroute tool was used with the Multipath Detection Algorithm (MDA) to conduct traceroute measurements. The IP to ASN mapping service was then used to resolve IP addresses to their corresponding Autonomous System Numbers (ASNs). This information was used to cross-reference the IPs identified as load balancers, their next hops, and their destination IPs.

3.3.2 Counting Matches and Mismatches of Next Hop AS

The next part of the analysis involved counting how often the AS of a next hop matched or mismatched the AS of the load balancer. This step aimed to understand the distribution of next hops in relation to their associated load balancers. This analysis provided a detailed view of the relationships between load balancers and their next hops.

3.3.3 Characterizing Next Hop Counts

To gain further insights into the behavior of load balancers, the number of next hops for each load balancer was analyzed. The mean, median, and mode of the next hop counts were calculated to summarize the distribution. This analysis helped characterize the diversity of paths taken by traffic after passing through load balancers.

3.3.4 Identifying Top Load Balancer Groups

Finally, the analysis identified the top load balancer groups based on the number of domains they served. By examining the number of domains associated with each load balancer, we were able to determine the most significant load balancer groups. The results highlighted the most prominent load balancers, showing which load balancers serve the highest number of domains.

3.3.5 Finding Common Next Hops Across Domains

The analysis started with identifying common next hops shared by multiple load balancers across different domains. The first step was loading the mapping of destinations to load balancers and their next hops from the data. Additionally, AS information for IPs was loaded to provide context for the identified next hops.

The analysis focused on finding next hops that were common to at least three load balancers across different domains. The results provided insights into common paths used by different domains, highlighting shared infrastructure.

Chapter 4

TOPOGRAPHY OF LOAD BALANCERS

In this chapter, we examine the distribution and prevalence of load balancers across two different datasets resulting from two distinct lists: the Top-2000 and the Rand-2000. By analyzing these datasets, we aim to understand how load balancers are utilized in the infrastructure of popular and randomly selected websites.

4.1 Analysis of Load Balancer Distribution

4.1.1 Top-2000 Dataset

The Top-2000 dataset consists of 117 days of data collected between November 9, 2023, and April 16, 2024. During this period, the number of domains with at least one load balancer ranged from a minimum of 279 to a maximum of 1680. The average number of domains with load balancers was 1644.08, indicating that 82.2% of the paths to these top domains include load balancers. The median value of 1668 further supports the observation that the majority of these domains consistently use load balancers, highlighting the importance of load balancing in maintaining the reliability and performance of high-traffic websites.

The minimum of 279 is about one-sixth of the median, suggesting significant variability. This could be due to measurement errors or temporary changes in the network paths. Investigating these anomalies could provide further insights into the consistency and reliability of load balancer detection.

4.1.2 Rand-2000 Dataset

The Rand-2000 dataset includes 112 days of data collected over the same period, from November 9, 2023, to April 16, 2024. The number of domains with at least one load balancer in this dataset varied between 1205 and 1297. The average number of domains with load balancers was 1251.29, with a median of 1252. These statistics show that 62.5% of the paths to these randomly selected domains include load balancers, indicating a substantial use of load balancers across a diverse set of websites.

4.1.3 Comparison and Insights

The scatter plot in Figure 4.1 illustrates the number of domains with at least one load balancer over time for both the Top-2000 and Rand-2000 datasets. This visual representation shows that the number of detected load balancers is relatively constant over the measurement period.

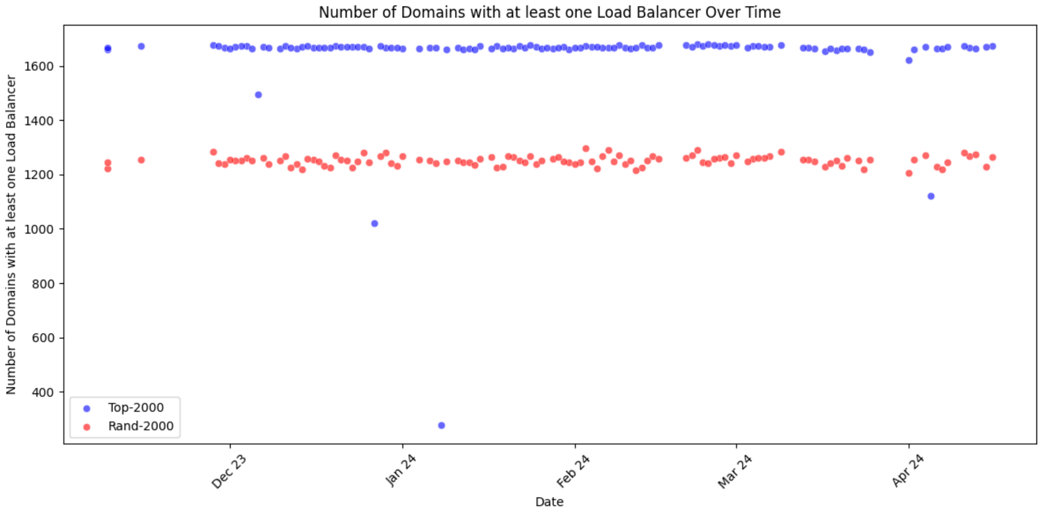

Figure 4.1: Number of Domains with at least one Load Balancer Over Time

The higher average and median values in the Top-2000 dataset underscore the higher usage of load balancers in managing traffic for the most visited websites. These sites likely experience higher and more variable traffic, necessitating robust load balancing solutions to ensure uptime and performance. Meanwhile, the Rand-2000 dataset demonstrates that load balancing is also prevalent for a broad spectrum of domains.

4.2 Analysis of Next Hops After Load Balancers

We focused on understanding the behavior of next hops after load balancers. This analysis helps us gain insights into how load distribution is managed.

4.2.1 Top-2000 Dataset

The Top-2000 dataset consists of 192,357 load balancers. The analysis reveals that the number of next hops after a load balancer ranges from a minimum of 2 to a maximum of 102. The average number of next hops is approximately 3.90, indicating a moderate level of load distribution across multiple paths. The median value stands at 3.0, suggesting that half of the load balancers have three or fewer next hops. The standard deviation is 5.75, reflecting significant variability in the number of next hops. Furthermore, the 75th percentile is at 5.0, while the 95th percentile reaches 13.0, showing that a small number of load balancers have a high number of next hops.

4.2.2 Rand-2000 Dataset

In comparison, the Rand-2000 dataset includes 140,144 load balancers. Here, the number of next hops ranges from a minimum of 1 to a maximum of 25. The average number of next hops is 1.58, indicating a lower level of load distribution compared to the Top-2000 dataset. The median number of next hops is 1, suggesting that more than half of the load balancers have only one next hop. The standard deviation is 1.34, indicating less variability. The 75th percentile is 2.0, and the 95th percentile is 3.0, showing a more concentrated distribution of next hops.

Figure 4.2 illustrates the cumulative distribution function (CDF) of the number of next hops after load balancers for both the Top-2000 and Rand-2000 datasets. The CDF provides a visual representation of the distribution and helps in comparing the two datasets. As shown, the Top-2000 dataset demonstrates a wider spread with

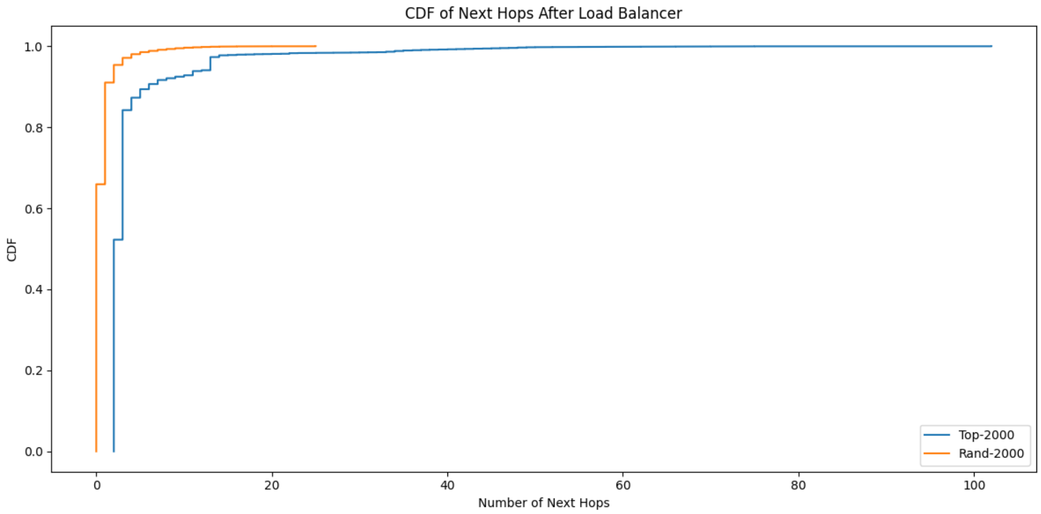

Figure 4.2: CDF of Next Hops After Load Balancers for Top-2000 and Rand-2000 Datasets

more load balancers having a higher number of next hops, whereas the Rand-2000 dataset shows a more concentrated distribution with most load balancers having fewer next hops.

The differences in the number of next hops between the two datasets highlight the varying network configurations and load distribution strategies. The higher variability in the Top-2000 dataset suggests a more complex and distributed network structure. In contrast, the Rand-2000 dataset's lower variability and fewer next hops suggest a simpler network configuration for less commonly used paths.

4.3 Analysis of ASes with Most Next Hops

The analysis of Autonomous Systems (ASes) with the highest average number of next hops reveals significant insights into the infrastructure and load balancing requirements of these networks. The ASes with the most next hops typically indicate a robust infrastructure with a greater need for load balancing. This could be due to a high volume of traffic, requiring efficient distribution across multiple servers to avoid bottlenecks, or due to specific requirements such as traffic filtering based on the source.

The comparison between the Top-2000 and Rand-2000 datasets shows that the differences are not very significant, with similar ASes appearing across both lists. This suggests that the predominant load balancing providers are optimizing for scalability, redundancy, geographic distribution, application segmentation, resource optimization, and compliance. These factors are crucial for maintaining high performance, fault tolerance, and legal compliance in diverse geographical locations.

Top 10 ASes by Average Number of Next Hops (Top-2000)	
AS	**Average Next Hops**
ORACLE-BMC-31898, US	90.69
FACEBOOK, US	24.97
ADJUST-, DE	24.16
CHINANET-SH-AP China Telecom Group, CN	20.59
CHINA169-BJ China Unicom Beijing , CN	19.82
CHINANET-BJ-AP, China Telecommunications, CN	15.44
CLOUDFLARENET, US	13.95
ALIBABA-CN-NET Alibaba Advertising , Ltd., CN	13.89
CHINANET-SCIDC-AS-AP CHINANET SiChuan, CN	13.82
CHINANET-BACKBONE No.31, Jin-rong Street, CN	13.74
Top 10 ASes by Average Number of Next Hops (Rand-2000)	
AS	**Average Next Hops**
ORACLE-BMC-31898, US	66.92
FACEBOOK, US	17.88
ADJUST-, DE	16.78
CHINANET-SH-AP China Telecom Group, CN	15.20
CHINA169-BJ China Unicom Beijing, CN	14.36
CHINANET-SCIDC-AS-AP CHINANET SiChuan, CN	11.33
CLOUDFLARENET, US	9.95
CHINANET-BACKBONE No.31, Jin-rong Street, CN	9.78
CT-HANGZHOU-IDC No.288, Fu-chun Road, CN	9.28
CHINANET-BJ-AP, China Telecommunications, CN	9.20

Table 4.1: Top 10 ASes by Average Number of Next Hops for Top-2000 and Rand-2000

The presence of ASes such as ORACLE-BMC-31898, FACEBOOK, and various Chinese telecommunications providers across both datasets underscores the need for efficient load balancing in these networks. High average next hops could suggest various needs such as:

- Optimization for scalability to handle large traffic volumes.
- Redundancy for fault tolerance.
- Geographic distribution for low-latency access.
- Compliance with data residency laws

While this list is not exhaustive, these factors are critical in ensuring that no single server becomes a bottleneck, maintaining service continuity, and providing optimal performance to users in various regions.

The similarities between the Top-2000 and Rand-2000 datasets indicate that these load balancing strategies are consistent across both high-traffic and more general websites, reflecting a broad application of these optimization techniques.

4.4 Overall Analysis of Next Hop ASes Matches and Mismatches

In our comprehensive analysis across all load balancers, we identified significant patterns in the matching and mismatching of next hops.

Our results revealed that out of a total of 192,357 load balancers analyzed in the Top-2000 dataset, and 140,144 load balancers in the Rand-2000 dataset, we observed a total of 104,214 unique load balancers. Among these, 80,979 were fully matching, where every next hop AS matched the load balancer AS, representing 77.7% of the total unique load balancers. Fully matching next hops indicate a high degree of routing consistency within the AS.

Partially matching next hops, where at least one but not all next hops matched the load balancer AS, accounted for 402 instances, or 0.4% of the total. These partial matches suggest a mixed routing strategy where some paths are optimized within the AS, while others diverge.

Next hops with no matching AS comprised 22,833 instances, making up 21.9% of the total load balancers. This scenario points to significant routing divergence, potentially due to dynamic routing policies or external influences.

Figure 4.3 illustrates the overall distribution of fully matching, partially matching, and no matching next hops. The pie chart provides a clear visual representation of the dominant patterns in next hop routing behavior.

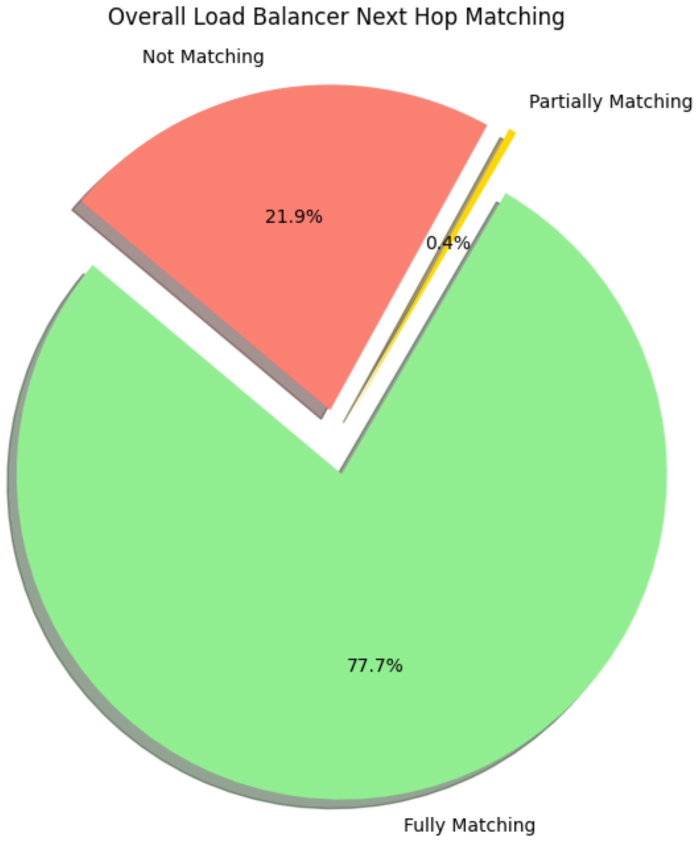

Figure 4.3: Overall Load Balancer Next Hop Matching

Our analysis also highlighted the top Autonomous Systems (ASes) based on the number of matches and mismatches. Table 4.2 presents the top ASes in each category.

Top 10 ASes with Fully Matching Next Hops	
AS	**Count**
UCB, US	30,890
Google, US	13,516
ChinaNet Backbone No.31, Jin-Rong Street, CN	6,420
Comcast-7922, US	5,460
China169-BJ China Unicom Beijing Province Network, CN	4,028
KDDI Corporation, JP	3,204
Facebook, US	2,321
Microsoft-Corp-MSN-AS-Block, US	2,260
Cogent-174, US	1,769
KIXS-AS-KR Korea Telecom, KR	1,744
Top 10 ASes with Partially Matching Next Hops	
AS	**Count**
ChinaNet-IDC-BJ-AP IDC, China Telecommunications Corporation, CN	158
ChinaNet Backbone No.31, Jin-Rong Street, CN	125
RelianceJio-IN Reliance Jio Infocomm Limited, IN	30
Yandex, RU	18
GlobalDC, FI	17
Level3, US	15
NL-Gigapop, US	12
HiNetUSA HiNet Service Center in U.S.A, TW	12
Alibaba-CN-Net Hangzhou Alibaba Advertising Co., Ltd., CN	4
Alibaba-CN-Net Alibaba US Technology Co., Ltd., CN	4
Top 10 ASes with No Matching Next Hops	
AS	**Count**
CONE, US	6,908
Domain_AS	5,580
UCB, US	2,525
CSUNET-NW, US	2,440
ChinaNet Backbone No.31, Jin-Rong Street, CN	854
Level3, US	849
Yahoo-1, US	608
CSUNET-NE, US	312
Google, US	216
BTN-ASN, US	207

Table 4.2: Top 10 ASes with Fully Matching, Partially Matching, and No Matching Next Hops

Overall, the findings suggest that while a majority of next hops maintain consistency within their ASes, there are notable instances of partial and no matches. This indicates that ASes mostly perform load balancing for their own nodes rather than for external nodes. This behavior makes sense because load balancing for external nodes would be costly without significant benefits to internal networks. However, the 22.3% of non-matching load balancers is quite surprising. This could imply that load balancing from network providers or CDNs like Cloudflare is offered as a service, or it could suggest that there might be some inaccuracies in AS matching in certain cases.

UCB, US, is the local Autonomous System and therefore it is normal to find disproportionate amounts of load balancers belonging to it. The findings suggest that load balancing is predominantly done within the same AS, with very minute instances of overlap indicating that inter-AS load balancing is rare. This is likely due to the high costs associated with implementing such infrastructure and its primary use for managing load within the autonomous system itself rather than for external nodes.

4.5 Change Over Time

This section presents the statistics for the Top-2000 and Rand-2000 datasets, including the total number of days observed, the average daily changes, and the number of reappearances. Additionally, the top 5 most consistent and least consistent load balancers for both datasets are listed.

4.5.1 Top-2000 Dataset

Over the observation period of 116 days, an average of approximately 253.47 load balancers were added per day in the Top-2000 dataset. Conversely, an average of 251.21 load balancers were lost each day. The dataset also recorded a total of 82,318 reappearances, indicating the number of times load balancers reappeared after having been previously lost. The average duration of presence for a load balancer in this dataset was 31.10 days.

Most Consistent Load Balancers	**Duration (days)**
169.229.0.140 : UCB, US	116
137.164.11.94 : CSUNET-NW, US	116
157.240.81.224 : FACEBOOK, US	116
157.240.112.88 : FACEBOOK, US	116
129.134.118.175 : FACEBOOK, US	116

Table 4.3: Top 5 Most Consistent Load Balancers in Top-2000 Dataset

In contrast, the least consistent load balancers, each appearing for only a single day, are shown in the following table:

Least Consistent Load Balancers	**Duration (days)**
202.97.27.181 : CHINANET-BACKBONE, CN	1
203.208.151.181 : SINGTEL-AS-AP, SG	1
203.208.178.185 : SINGTEL-AS-AP, SG	1
203.208.154.45 : SINGTEL-AS-AP, SG	1
203.208.171.9 : SINGTEL-AS-AP, SG	1

Table 4.4: Top 5 Least Consistent Load Balancers in Top-2000 Dataset

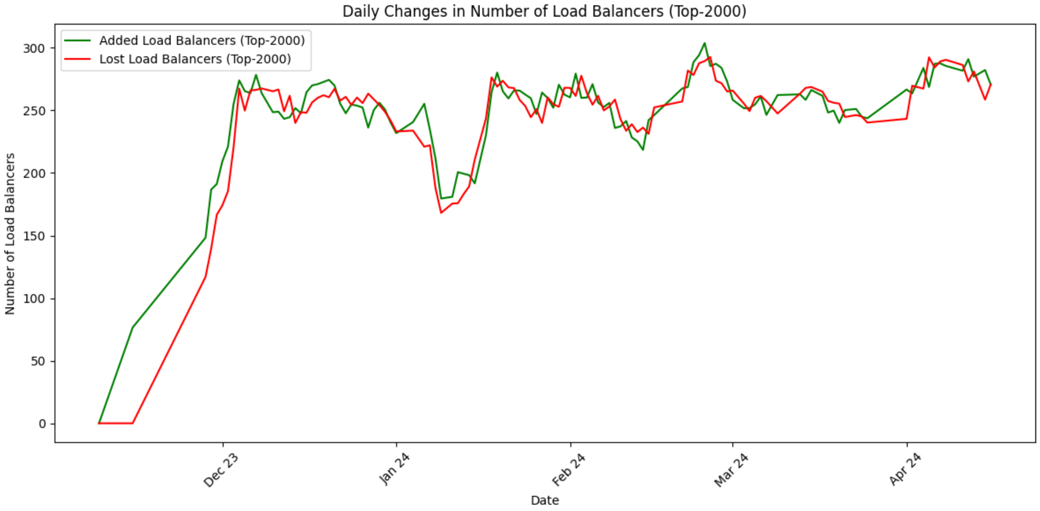

Figure 4.4: Daily Changes in Number of Load Balancers for Top-2000

4.5.2 Rand-2000 Dataset

The Rand-2000 dataset, observed over 111 days, provided the following insights: On average, approximately 161.02 load balancers were added per day, while an average of 162.22 load balancers were lost each day. The dataset also recorded a total of 12,139 reappearances, highlighting the instances where load balancers reappeared after being lost. The average duration of presence for a load balancer in this dataset was 12.49 days.

Most Consistent Load Balancers	**Duration (days)**
169.229.0.140 : UCB, US	111
137.164.11.94 : CSUNET-NW, US	41
142.251.231.97 : GOOGLE, US	40
142.251.231.99 : GOOGLE, US	32
74.125.50.18 : GOOGLE, US	30

Table 4.5: Top 5 Most Consistent Load Balancers in Rand-2000 Dataset

The least consistent load balancers in the Rand-2000 dataset, each appearing for a single day, are shown below:

Least Consistent Load Balancers	**Duration (days)**
104.44.19.140 : MICROSOFT-CORP-MSN-AS-BLOCK, US	1
104.44.18.166 : MICROSOFT-CORP-MSN-AS-BLOCK, US	1
157.240.51.143 : FACEBOOK, US	1
202.97.53.13 : CHINANET-BACKBONE, CN	1
163.253.2.16 : INTERNET2-RESEARCH-EDU, US	1

Table 4.6: Top 5 Least Consistent Load Balancers in Rand-2000 Dataset

The analysis shows that load balancers are mostly dynamic and not static, changing over time. Even though the graph for the number of load balancers (fig 4.1) shows that the number remains relatively stable, the individual load balancers themselves change frequently. This is more evident in the random data because it doesn't consistently target the same paths, leading to higher variability. The average duration of presence further supports this, with load balancers in the Top-2000 dataset averaging 31.10 days, while those in the Rand-2000 dataset average only 12.49

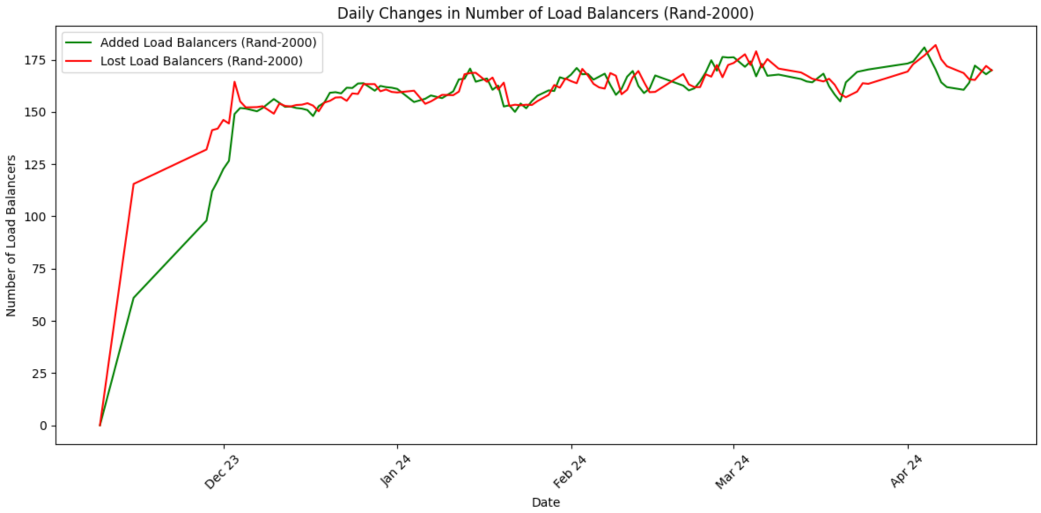

Figure 4.5: Daily Changes in Number of Load Balancers for Rand-2000

days. This indicates that load balancers adapt to varying network conditions and demands, demonstrating the dynamic nature of network infrastructure.

4.5.3 Autonomous Systems

This subsection presents the AS-specific statistics for both the Top-2000 and Rand-2000 datasets, focusing on daily averages and key metrics related to ASes.

The AS with the most additions per day indicates which AS consistently introduces the most new load balancers. This can reflect the AS's dynamic nature and high level of activity in adjusting its load balancing infrastructure.

The AS with the most removals per day shows which AS frequently removes load balancers. High removal rates can suggest either significant changes in traffic patterns when the load balancing functions are turned off, frequent updates to the network infrastructure when next hops are down, or issues requiring frequent load balancer replacement. It could also be caused by a different route being taken when accessing that website.

The AS with the longest average duration of presence has load balancers that remain active the longest on average. This indicates a more stable and persistent load balancing infrastructure.

The most consistent AS is present for the most number of times, indicating the AS's load balancers are active across the majority of the observation period, reflecting stability.

The most inconsistent AS is the one that is present for the fewest number of times, suggesting either sporadic use of load balancers or frequent changes in infrastructure. It could also mean that the measurment was a false positive and that it was not truly a load balancer.

The most fluctuating AS has the highest number of load balancer changes, indicating a high level of dynamism in their load balancing strategy, possibly reflecting a need to adapt rapidly to changing traffic conditions or network requirements.

Note: The UCB, US AS was excluded from these statistics as it would consistently be the most reliable AS, given that our tests are conducted from within it.

Top-2000 Dataset

For the Top-2000 dataset, observed over 116 days:

- **AS with most additions per day:** CHINANET-BACKBONE No.31, Jinrong Street, CN, Added: 50.34 per day
- **AS with most removals per day:** CHINANET-BACKBONE No.31, Jinrong Street, CN, Removed: 50.17 per day
- **AS with longest average duration:** CHINANET-BACKBONE No.31, Jinrong Street, CN, Average Duration: 159.62 days
- **Most consistent AS:** GOOGLE, US, Present: 75863 times
- **Most inconsistent AS:** AS-NETIA Warszawa 02-822, PL, Present: 1 time

- **Most fluctuating AS:** CHINANET-BACKBONE No.31, Jin-rong Street, CN, Changes: 11660 over 116 days

Rand-2000 Dataset

For the Rand-2000 dataset, observed over 111 days:

- **AS with most additions per day:** CHINANET-BACKBONE No.31, Jin-rong Street, CN, Added: 29.12 per day
- **AS with most removals per day:** CHINANET-BACKBONE No.31, Jin-rong Street, CN, Removed: 29.40 per day
- **AS with longest average duration:** CHINANET-BACKBONE No.31, Jin-rong Street, CN, Average Duration: 44.09 days
- **Most consistent AS:** GOOGLE, US, Present: 11153 times
- **Most inconsistent AS:** MTS, RU, Present: 1 time
- **Most fluctuating AS:** CHINANET-BACKBONE No.31, Jin-rong Street, CN, Changes: 6495 over 111 days

4.6 Shared Next Hops Analysis

This section analyzes the shared next hops between load balancers in both the Top-2000 and Rand-2000 datasets. The objective is to understand the extent to which next hops are shared among multiple load balancers, indicating the presence of common infrastructure and potential load balancing strategies.

4.6.1 Top-2000 Dataset

For the Top-2000 dataset, a total of 9,492 unique next hops were identified. The number of load balancers sharing a next hop ranged from a minimum of 1 to a maximum of 52. On average, each next hop was shared by 5.31 load balancers, with a median value of 2.0. The standard deviation of 6.70 indicates considerable variability in the number of load balancers sharing a next hop. The 75th percentile value was 7.0, and the 95th percentile value was 18.0, showing that a small number of next hops were highly shared among load balancers.

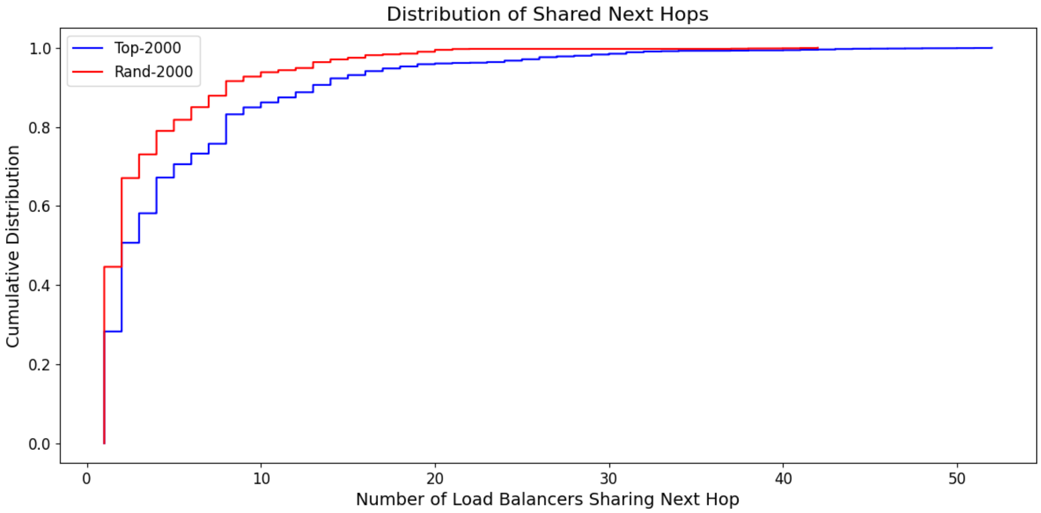

Figure 4.6: Distribution of Shared Next Hops for Top-2000 and Rand-2000

4.6.2 Rand-2000 Dataset

The Rand-2000 dataset revealed 15,797 unique next hops. The number of load balancers sharing a next hop ranged from a minimum of 1 to a maximum of 42. The average number of load balancers sharing a next hop was 3.35, with a median of 2.0. The standard deviation was 4.17, indicating a moderate level of variability. The 75th percentile value was 4.0, and the 95th percentile value was 13.0, suggesting that most next hops were shared by a relatively small number of load balancers.

The analysis of shared next hops between load balancers in both datasets highlights key differences and similarities. The Top-2000 dataset shows a higher average number of load balancers sharing a next hop compared to the Rand-2000 dataset. This suggests that the infrastructure supporting the most popular websites is more interconnected and utilizes shared pathways more frequently than the broader, randomly selected websites.

The distribution of shared next hops, as depicted in Figure 4.6, indicates that while most next hops are shared by a small number of load balancers, there are a few next hops that are highly shared, particularly in the Top-2000 dataset. This can be attributed to the reliance on major network infrastructure providers and common routing paths for high-traffic websites.

Overall, the presence of shared next hops signifies the use of common infrastructure, which can enhance efficiency but also poses risks in terms of single points of failure. The variability in the number of shared next hops across both datasets underscores the complexity and dynamic nature of load balancing in different segments of the internet.

Chapter 5

LAYER 3 LOAD BALANCING

In my research, I found specific load balancers within the Internet2 AS. The Internet2 Network was established to support data-intensive research and advanced campus and cloud computing needs. Internet2's GlobalNOC-developed Router Proxy tool allows users to run commands against network devices, enabling limited testing and viewing active configuration and state information.

These load balancers were accessible via the Internet2 looking glass, allowing deeper analysis of their load balancing techniques. Specifically, we observed that many of the next hops for these load balancers were defined in Cisco Express Forwarding (CEF) tables. The following sections elaborate on the workings of CEF and its role in load balancing.

5.1 Network Layer Load Balancing

Layer 3, known as the network layer, plays a crucial role in load balancing by routing packets based on their IP addresses. This approach focuses on the distribution of traffic across multiple servers without inspecting the packet contents. Unlike higher layers that can make decisions based on the data within the packets, Layer 3 load balancers rely solely on IP addresses and routing tables. This method is efficient and fast but offers less granularity in traffic management, as it does not consider the type or state of the application data. They cannot make decisions based on the content of the traffic, user sessions, or specific application states, which are critical for more advanced load balancing strategies. This means that while Layer 3 load balancers can efficiently manage large volumes of traffic, they lack the detailed traffic management capabilities provided by higher-layer solutions. Nevertheless, technologies like Cisco Express Forwarding (CEF) enhance the efficiency of Layer

3 load balancing by pre-computing forwarding information, ensuring rapid and reliable packet forwarding across the network.

5.2 Cisco Express Forwarding (CEF)

Cisco Express Forwarding (CEF) is a Layer 3 switching technology used to optimize network performance. CEF employs a forwarding information base (FIB) and an adjacency table to expedite the packet forwarding process. The following outlines the critical components and operations of CEF:

5.2.1 CEF Components

Forwarding Information Base (FIB): The FIB is used by CEF to make IP destination prefix-based switching decisions. It is conceptually similar to a routing table, maintaining a mirror image of the forwarding information in the IP routing table. When routing or topology changes occur, the IP routing table updates, and these changes are reflected in the FIB. The FIB ensures all known routes are covered, eliminating the need for route cache maintenance.

Adjacency Table: The adjacency table complements the FIB by storing Layer 2 addressing information necessary for packet forwarding. Nodes in the network that can reach each other with a single hop across a link layer have their Layer 2 next-hop addresses stored in this table.

5.2.2 CEF Operation Modes

Central CEF Mode: In this mode, both the FIB and adjacency tables reside on the route processor, which performs all express forwarding. This mode is useful when line cards are not available for CEF switching or when features incompatible with distributed CEF switching are needed.

Distributed CEF (dCEF) Mode: In dCEF mode, line cards maintain identical copies of the FIB and adjacency tables, enabling them to perform express forwarding independently. This offloads the route processor from being involved in the switching operation, making it particularly efficient for high-performance environments like the Cisco 12000 Series Router.

5.2.3 Packet Forwarding Process

The line cards play a crucial role in forwarding. When a packet arrives, it is placed into input buffers on the receiving line card. The Layer 2/Layer 3 forwarding engine accesses the packet's information and determines its route based on the FIB and adjacency table. The appropriate Layer 2 information is then appended to the packet using data from the adjacency table, and the packet is forwarded to its next-hop destination.

5.2.4 Synchronization and Updates

An Inter-Process Communication (IPC) mechanism ensures synchronization of FIBs and adjacency tables between the route processor and line cards. The GRP (route processor) updates the network routing table and sends forwarding update messages to the FIB tables on the line cards. It also updates the adjacency tables whenever new Layer 2 information is received.

5.3 Summary and Findings

CEF's efficiency in packet forwarding is achieved through its use of optimized data structures (FIB and adjacency tables) and its ability to distribute the forwarding process across different components of the router, particularly in environments where high-speed processing is required. This structured approach allows for rapid and reliable routing of packets across a network, enhancing overall network performance and scalability.

The insights gained from analyzing the Internet2 load balancers revealed that most of the next hops for these load balancers are defined within CEF. This ensures high efficiency and reliability in handling the vast amount of data traffic traversing the Internet2 network. It also means that CEF load balancing is less resource-intensive and does not require any deeper look at the packets.

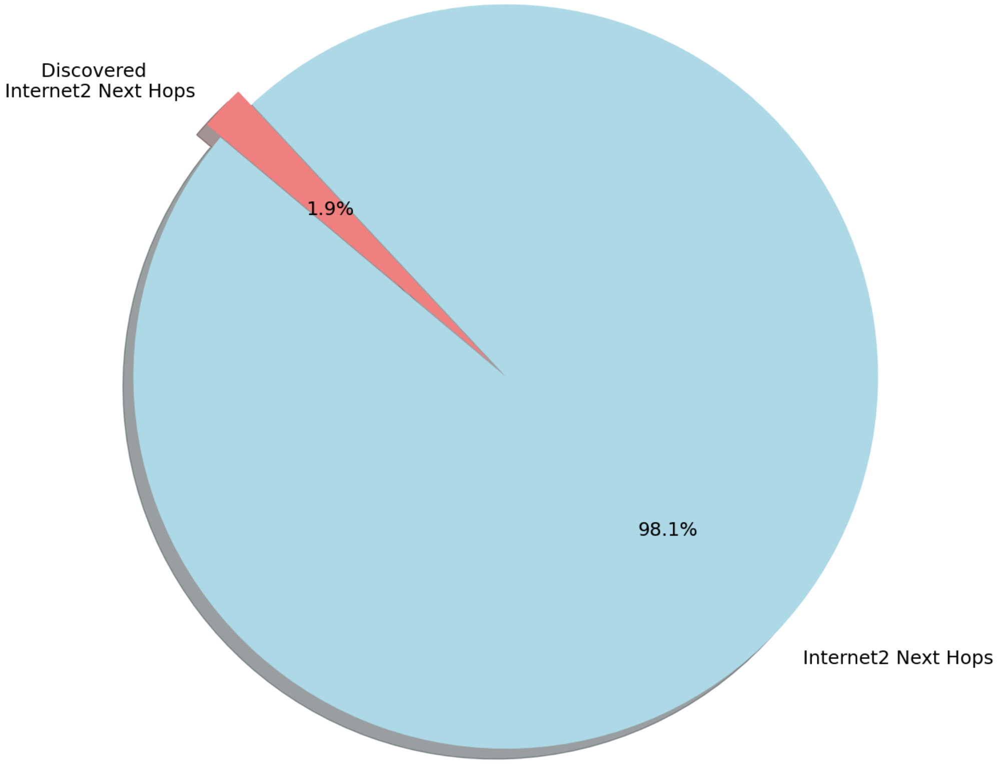

Figure 5.1: Pie Chart of discovered Internet2 Load Balancers

The implementation of load balancing at Layer 3 using CEF highlights the techniques employed to manage network traffic. This leads us to think that a lot of load balancers can be hidden behind the CEF forwarding, which is not usually visible using tools like Paris traceroute. The pie chart 5.1 showing the percentage of next hops found by MDA and the looking glass. Only 1.9% of the next hops were found using MDA, while the looking glass revealed that 97.9% were not found by our measurements, either because not enough probes were sent out by MDA, next hops were hidden, or they were not accessible on the specific route taken. This could mean that a great amount of load balancers are not identifiable using network probing tools and that they are a hidden part of

Chapter 6

Chapter 7

THIS IS THE SEVENTH CHAPTER

Chapter 8

THIS IS THE EIGHTH CHAPTER

Appendix A

QUESTIONNAIRE