

# A TOPOGRAPHICAL EVALUATION OF LOAD BALANCERS

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<sup>1</sup>**.

Committee Chair

**An Wang**

Committee Member

**Mark Allman**

Committee Member

**Vincenzo Liberatore**

Committee Member

**Mehmet Koyuturk**

Date of Defense

**July 10th, 2024**

---

<sup>1</sup>We also certify that written approval has been obtained for any proprietary material contained therein.

## DEDICATION

TBD

## TABLE OF CONTENTS

List of Tables . . . . .	vi
List of Figures . . . . .	vii
Acknowledgements . . . . .	viii
Abstract . . . . .	ix
Chapter I: Introduction . . . . .	1
1.1 Introduction to Load Balancing . . . . .	1
1.2 Project Motivation and Goals . . . . .	2
1.3 Areas of Study . . . . .	2
Chapter II: Related Works . . . . .	4
2.1 Paris Traceroute . . . . .	4
2.2 Multipath Detection Algorithm (MDA) . . . . .	5
2.3 Scamper . . . . .	6
2.3.1 Multipath Detection Algorithm (MDA) in Scamper . . . . .	7
2.4 Multipath Classification Algorithm (MCA) . . . . .	7
Chapter III: Methodology . . . . .	9
3.1 Data Sources and Selection Criteria . . . . .	9
3.2 Discovering Load Balancers . . . . .	10
3.2.1 Measurement Frequency and Timeline . . . . .	10
3.2.2 Sample Measurement . . . . .	11
3.3 Team Cymru IP to ASN List . . . . .	12
3.4 Ethical Considerations . . . . .	13
Chapter IV: Data Analysis . . . . .	14
4.1 Load Balancer Distribution . . . . .	14
4.2 Analysis of Next Hops After Load Balancers . . . . .	16
4.3 Analysis of ASes with Most Next Hops . . . . .	18
4.4 Overall Analysis of Next Hop ASes Matches and Mismatches . . . . .	20

4.5	Change Over Time . . . . .	23
4.6	Change Over Time for Autonomous Systems . . . . .	26
4.7	Change Over Time for Autonomous Systems . . . . .	27
4.8	Shared Next Hops Analysis . . . . .	29
Chapter V: Layer 3 Load Balancing . . . . .		31
5.1	Network Layer Load Balancing . . . . .	31
5.2	Cisco Express Forwarding (CEF) . . . . .	32
5.3	Summary and Findings . . . . .	32
5.4	Modifications to the MDA Algorithm . . . . .	33
Chapter VI: Summary . . . . .		35
Chapter VII: Future Work . . . . .		38

## LIST OF TABLES

<i>Number</i>	<i>Page</i>
3.1 BGP and ASN Information from team Cymru data . . . . .	13
4.1 Statistical Overview of Load Balancer Usage . . . . .	14
4.2 Statistical Overview of Next Hop Distribution After Load Balancers .	16
4.3 Top 10 ASes by Average Number of Next Hops for Top-2000 . . . .	18
4.4 Top 10 ASes by Average Number of Next Hops for Rand-2000 . . .	19
4.5 Summary of Load Balancer Matching Statistics . . . . .	20
4.6 Top 10 ASes with Fully Matching Next Hops . . . . .	21
4.7 Top 10 ASes with Partially Matching Next Hops . . . . .	22
4.8 Top 10 ASes with No Matching Next Hops . . . . .	22
4.9 Top-2000 Dataset Statistics . . . . .	27
4.10 Top-2000 Dataset Highest Values . . . . .	27
4.11 Rand-2000 Dataset Statistics . . . . .	28
4.12 Rand-2000 Dataset Highest Values . . . . .	28
4.13 Summary of Shared Next Hops Statistics in the Top-2000 and Rand- 2000 Datasets . . . . .	29

## LIST OF FIGURES

<i>Number</i>	<i>Page</i>
4.1 Number of Domains with at least one Load Balancer Over Time . . .	15
4.2 CDF of Next Hops After Load Balancers for Top-2000 and Rand- 2000 Datasets . . . . .	17
4.3 Overall Load Balancer Next Hop Matching . . . . .	20
4.4 Daily Changes in Number of Load Balancers for Top-2000 . . . . .	24
4.5 Daily Changes in Number of Load Balancers for Rand-2000 . . . . .	24
4.6 Cumulative Distribution Function of Load Balancer Durations . . . .	25
4.7 Distribution of Shared Next Hops for Top-2000 and Rand-2000 . . .	30

## ACKNOWLEDGEMENTS

TBD



# A Topographical Evaluation of Load Balancers

## Abstract

by

OMAR LOUDGHIRI

This thesis investigates the deployment and operation of load balancers in internet routing, emphasizing their prevalence on various internet pathways. Data collection spanned from November 2023 to April 2024, employing Paris Traceroute equipped with the Multipath Detection Algorithm (MDA) to analyze path measurements across the internet. Our analysis reveals that load balancers are present on 71.9% of paths to popular websites and on 52.3% to a broader, randomly selected set of sites, highlighting their critical role in managing network traffic. The study observed load balancers exhibiting frequent changes, maintaining an average presence of approximately one month on popular site paths and about two weeks on paths to random sites. Although Layer 3 load balancing techniques such as Cisco Express Forwarding (CEF) were noted, specific impacts and efficiencies are part of ongoing investigations. This work lays the foundation for understanding load balancing dynamics and identifies aspects for future research to enhance detection methods and improve the accuracy of network performance analyses.

# *Chapter 1*

## INTRODUCTION

### 1.1 Introduction to Load Balancing

Networks are essential to the Internet, facilitating communication, commerce, and entertainment on a global scale. These networks consist of an interconnected framework of user devices, routers, and servers. Understanding the architecture and dynamics of these networks has been pivotal in enhancing their efficiency and reliability over the last 25 years [paxson\_endtoend96].

Load balancing, a crucial network optimization strategy, manages traffic distribution and scales network capacity to prevent any single server from becoming overwhelmed, thus promoting efficiency and reliability [10371400]. Initially, load balancers were standalone network devices designed to direct packets across multiple routes to balance server load [bourke2001server]. This function is now often integrated within routers, playing a vital role in ensuring high availability, scalability, security, and performance of modern infrastructure [f52023loadbalancing].

Modern applications require the ability to handle millions of simultaneous sessions, and load balancers distribute this traffic across servers with duplicate data to ensure reliable and rapid data delivery. Although load balancers enhance system redundancy—redirecting traffic to maintain access if a server fails—they are not fail-proof and can become single points of failure if not part of a redundant setup [chang2005complex].

Furthermore, load balancing enhances security by minimizing attack surfaces. It reduces the risk of individual servers being overwhelmed by distributing traffic, thus decreasing the overall attack surface and enhancing system resilience against attacks. However, the security benefits depend heavily on the secure configuration

and robustness of the load balancer itself [10.1145/3098593.3098595].

Performance optimization is another benefit of load balancing, which utilizes algorithms like round-robin and least connections to distribute traffic based on real-time conditions. This prevents any server from becoming a bottleneck, ensuring the network operates smoothly and efficiently [8316818].

## 1.2 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, this research aims to enhance our understanding of their impact. This includes identifying and categorizing load balancers, analyzing their deployment, and understanding the resulting network paths and their implications for the broader Internet infrastructure.

Understanding how load balancing affects network performance and reliability is crucial for developing more efficient and resilient network systems. Effective load balancing can prevent single points of failure, manage high traffic volumes, and ensure continuous service availability. By studying current load balancing practices and their outcomes, we can identify areas for improvement and develop strategies to enhance network stability and efficiency. This research will contribute to a better understanding of the critical role load balancing plays in maintaining the robustness and efficiency of the Internet.

## 1.3 Areas of Study

In the following chapters, we explore the presence of load balancers across the Internet. We begin by discussing related work in Chapter 2, which covers essential tools we use, such as Traceroute, the Multipath Detection Algorithm (MDA),

Paris Traceroute, and Scamper. We will delve into their features and applications in detecting and classifying load balancers on the Internet.

In Chapter 3, we provide an overview of our research methodology. This includes background information on resources like the Alexa Top 1 Million Websites and the Team Cymru IP to ASN list. Our approach to discovering load balancers is detailed, including measurement frequency and IP list compilation.

Chapter 4 examines the topography of load balancers. We analyze the distribution of load balancers using our datasets, compare insights from these datasets, and delve into the analysis of next hops after load balancers. We identify Autonomous Systems with the most next hops and explore changes over time.

In Chapter 5, we focus on Layer 3 load balancing, particularly the modifications to the MDA algorithm for better network layer load balancing efficacy. We discuss the challenges and future work associated with enhanced probing trials and the integration of Cisco Express Forwarding (CEF) into our analysis. This chapter aims to provide concrete insights into the application of advanced network routing technologies to optimize traffic distribution.

## *Chapter 2*

### RELATED WORKS

Traceroute is a network diagnostic tool used to track the path packets take from one IP address to another. The tool 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.

#### **2.1 Paris Traceroute**

In [4261334], the authors present an enhanced version of Traceroute to identify load balancers along with 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.

In their second paper [augustin2010measuring], which aimed to measure the presence of load balancers in the Internet 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 the routes to 39% of the destinations traversed a load balancer. Some of their results suggest up to 72% when considering

different types of load balancing. While the specifics of these different types of load balancing are out of scope for this project, our goal is to update the community's understanding of the topology of load balancers almost 10 years after their paper.

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.

## 2.2 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  $r$ . Flow identifiers are unique markers within packet headers, such as combinations of source and destination IP addresses, port numbers, and protocol types. It then sends these probes one hop further to discover the next-hop interfaces  $s_1, s_2, \dots, s_n$ .

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, the number of probes 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.

Per-flow balancing means that all packets within the same flow (i.e., packets sharing the same source and destination IP addresses, port numbers, and protocol) follow the same path through the network. This ensures that packets arrive in order, which is crucial for the correct reassembly and processing of data streams.

Per-packet balancing, on the other hand, distributes individual packets across multiple paths. While this can maximize the use of available network resources, this type of balancing can lead to packet reordering since packets from the same flow might take different paths and arrive out of order. This can complicate the reassembly process and potentially impact the performance of applications sensitive to packet order.

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.

The paper by [4261334] 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.

By leveraging Paris Traceroute and MDA, we conduct extensive measurements to map the global distribution of load balancers and analyze their impact on network performance and reliability.

## 2.3 Scamper

Scamper, presented in [luckie2010scamper], is a versatile tool used for conducting large-scale Internet measurements. Scamper’s code was easily modified to support more fine-tuned measurements using the Multipath Detection Algorithm (MDA), enhancing its capability to identify and analyze load-balanced paths.

### 2.3.1 *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.

## 2.4 **Multipath Classification Algorithm (MCA)**

Recent advances in network technology and the adoption of IPv6 have enabled more complex load balancing strategies. [9155387] 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 and comprehensive measurements. MCA identifies the specific bits in the packet header used by load balancers, providing a more detailed and accurate classification than MDA.



Additionally, MCA characterizes load balancing on both IPv4 and IPv6 Internet paths, showing that load balancing is more prevalent and sophisticated than previously reported.

Despite these advancements, using MCA was not feasible for our research due to its higher complexity and longer runtime. MCA's improvements come at the cost of increased probing time and complexity, making it less practical for large-scale measurements.

While MCA offers improvements in identifying and classifying load balancers, 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 (scamper) and faster runtime performance in order to conduct daily measurements. MDA's established methodologies and ease of implementation which makes it a more practical choice for large-scale measurements.

## *Chapter 3*

### METHODOLOGY

This chapter details the methods used to collect 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 Data Sources and Selection Criteria

To ensure the feasibility of daily measurements, pilot measurements were conducted, which indicated that approximately 2000 hostnames could be processed per day. This constraint informed our selection of two distinct subsets from the Alexa list, enabling daily measurements while managing logistical constraints.

The Alexa Top 1 Million Websites list was used to obtain hostnames for this research. A current version of the Alexa list was obtained when we started our data collection in November 2023. The Alexa list is widely used in network measurement studies due to its popularity, since the list is not known for high accuracy for ranks below 100,000 sites [**alexa2023top1m**], only the top 100,000 is in consideration.

For our study, we selected two distinct subsets from the Alexa list:

- **Top-2000 List:** This list includes the top 2000 domains from the Alexa list, designed to cover the most used websites on 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. This aims to provide a well-rounded analysis of the Internet’s topology by including popular but not exclusively top-ranked sites. This random list excludes the top-2000 hostnames from the previous list.

## 3.2 Discovering Load Balancers

We recorded the paths between our vantage point and a set of popular hosts to detect and characterize load balancers along these paths.

### 3.2.1 *Measurement Frequency and Timeline*

To ensure the feasibility of daily measurements, 2000 hostnames were chosen for the Paris Traceroute process. Each hostname takes an average of 40 seconds to return a complete trace with load balancer information. This duration allows the script to run through 2000 hostnames 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 ran measurements in parallel for the Top-2000 list and the Rand-2000 list each day. Each measurement started at 5 am EST and ran in Alexa rank order for Top-2000 and in the random order the list was created in Rand-2000. After completing the measurements, there was a one-hour buffer before the next run began at 5 am the following day.

The measurements were run continuously from November 9, 2023, to April 16, 2024, on a Linux machine at the International Computer Science Institute (ICSI) in Berkeley, CA. Some pilot measurements were also run beforehand from both machines at ICSI and at CWRU. 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.

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.

### 3.2.2 *Sample Measurement*

Below is the annotated output of a Paris traceroute measurement, detailing the path from a local network to various network nodes. Annotations are provided to highlight the starting point and the presence of a load balancer.

This is the start:

```
192.150.187.1 -> [ 169.229.0.140 ]
```

This is what a load balancer looks like, it creates two branches:

```
169.229.0.140 -> [ 128.32.255.6, 128.32.255.8 ]
```

First Branch

```
128.32.255.6 -> [ 128.32.0.38 ]
```

```
128.32.0.38 -> [ 137.164.3.26 ]
```

```
137.164.3.26 -> [ 137.164.11.94 ]
```

```
137.164.11.94 -> [ 4.15.122.45 ]
```

A hidden next hop, the IP of the next hop was hidden (None)

```
4.15.122.45 -> [ None, 200.189.213.6 ]
```

The IP of the load balancer was hidden

```
None -> [ 8.243.153.10, 8.243.152.12 ]
```

```
200.189.213.6 -> [ 8.243.153.10 ]
```

```
8.243.153.10
```

Second Branch

```
128.32.255.8 -> [ 137.164.11.94 ]
```

```
137.164.11.94 -> [ 4.15.122.40 ]
```

A load balancer with three next hops, all three seem to belong to the same /24

```
4.15.122.40 -> [ 200.189.213.42, 200.189.213.6, 200.189.213.38 ]
.
.
8.243.153.10
```

### 3.3 Team Cymru IP to ASN List

An Autonomous System (AS) is a collection of IP networks and routers under the control of a single organization that presents a common routing policy to the internet. Each AS is assigned a unique identifier known as an Autonomous System Number (ASN), which facilitates the routing of data between different ASes. ASes play a critical role in the overall structure of the internet, as they help manage the flow of data, ensuring efficient and reliable connectivity across various networks. They are often managed by internet service providers (ISPs), large enterprises, or educational and government institutions.

To map routers to organizations in our analysis, we used the Team Cymru IP to ASN mapping service to resolve IP addresses to their corresponding Autonomous System Numbers (ASNs). Team Cymru maps IP numbers to BGP prefixes and ASNs using data from over 50 BGP feeds, updated every four hours [teamcymru2023ipasn].

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. Table 3.1 shows the fields we obtain for each IP address, detailing data about the Autonomous System it belongs to.

For each domain in the Rand-2000 and Top-2000 lists, the Paris Traceroute

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 from team Cymru data

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.

### 3.4 Ethical Considerations

This research adhered to strict ethical standards to ensure no harm was caused during data collection. We performed active measurements with care, ensuring they did not overflow the network. We ran only the two necessary probes in parallel to prevent any network disruptions.

No personal information was collected in our data. We ensured that our data collection methods did not cause any disruptions. Ethical considerations were carefully followed based on the recommendations in [partridge2016ethical]

## *Chapter 4*

### DATA ANALYSIS

In this chapter, we examine the distribution and prevalence of load balancers across two distinct datasets derived from previously described hostname lists: Top-2000 and Rand-2000. Our aim is to understand the deployment of load balancers in the infrastructures of both popular and randomly selected websites.

#### 4.1 Load Balancer Distribution

The datasets analyzed cover data collected over 117 days from November 9, 2023, to April 16, 2024. We exclude data from UCB as it reflects local network conditions rather than the global internet trends we aim to analyze. This exclusion provides a clearer picture of how load balancers are utilized globally.

The Top-2000 dataset initially revealed that the number of domains with at least one load balancers ranged from 81 to 1482 domains, with an average of 1439.03 and a median of 1463, revealing that 71.9% of paths include load balancers.

Conversely, the Rand-2000 dataset, spanning 112 days within the same period, showed that the numbers of load balancers range from 994 to 1094 domains, with an average of 1046.27 and a median of 1046, indicating a 52.3% usage of load balancers among these randomly selected domains.

The following table summarizes the key statistics for both datasets:

Dataset	Min	Max	Average	Median	% of Total Routes
<b>Top-2000 (without UCB)</b>	81	1482	1439.03	1463	<b>71.9</b>
<b>Rand-2000 (without UCB)</b>	994	1094	1046.27	1046	<b>52.3</b>

Table 4.1: Statistical Overview of Load Balancer Usage

Figure 4.1 illustrates the number of domains with at least one load balancer over time for both datasets, both including and excluding UCB influences.

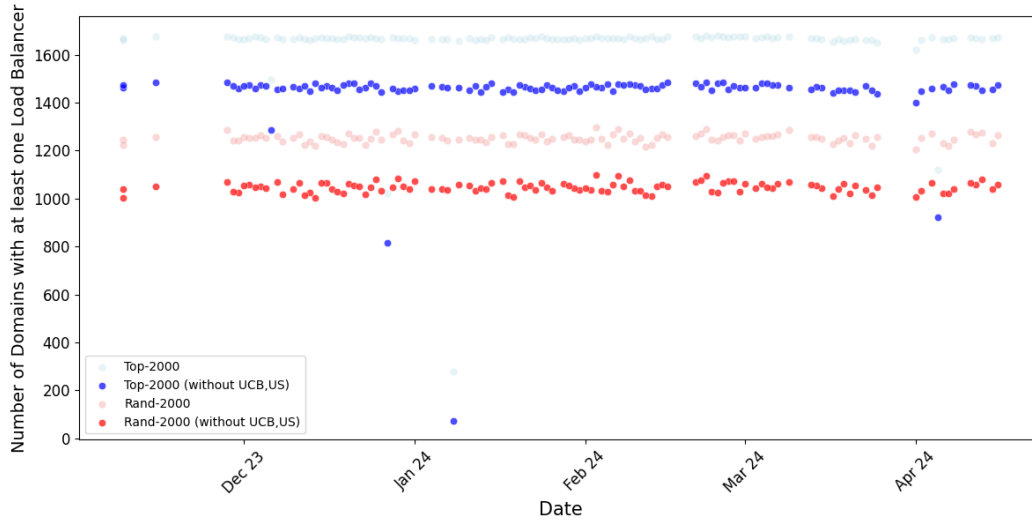


Figure 4.1: Number of Domains with at least one Load Balancer Over Time

The higher average and median values in the Top-2000 dataset reflect the more frequent use of load balancers in managing traffic for the most visited websites. Such sites often experience higher and more variable traffic, necessitating robust load balancing solutions to maintain uptime and performance. Conversely, the Rand-2000 dataset indicates that load balancing is prevalent across a wide variety of domains. By excluding UCB data, we obtain a clearer representation of load balancer usage that is unaffected by local network biases.

The plot distinctly illustrates two trends: one influenced by the local network and one without. Light colors represent the original data, while dark colors depict the adjusted data, excluding UCB. This approach helps contextualize our data, ensuring that local network characteristics do not skew our understanding of the global network.



## 4.2 Analysis of Next Hops After Load Balancers

We focus on understanding the behavior of next hops after load balancers to gain insights into how load distribution is managed across various domains.

The Top-2000 dataset comprises 192,357 load balancers. The analysis shows that the number of next hops after a load balancer ranges from 2 to 102, with an average of approximately 3.90. This indicates a moderate level of load distribution across multiple paths. The median value is 3.0, suggesting that half of the load balancers distribute to either three or fewer next hops. The standard deviation, at 5.75, points to significant variability in the number of next hops. The 75th percentile stands at 5.0, while the 95th percentile reaches 13.0.

In comparison, the Rand-2000 dataset includes 140,144 load balancers, where the number of next hops ranges from 2 to 25. The average number of next hops is 2.58, suggesting that more than half of the load balancers distribute to only two next hops. This dataset shows lower variability with a standard deviation of 1.34. The 75th percentile is 2.0, and the 95th percentile is 3.0, indicating that most load balancers in this dataset are simpler, primarily distributing to 2 or 3 next hops, with some outliers having up to 25.

Dataset	Min	Max	Average	Median	Standard Deviation
Top-2000	2	102	3.90	3.0	5.75
Rand-2000	2	25	2.58	2.0	1.34

Table 4.2: Statistical Overview of Next Hop Distribution After Load Balancers

This analysis highlights the differences in load balancing complexity between the datasets, reflecting the broader range of load distribution strategies and setups in environments with varying traffic loads and requirements.

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

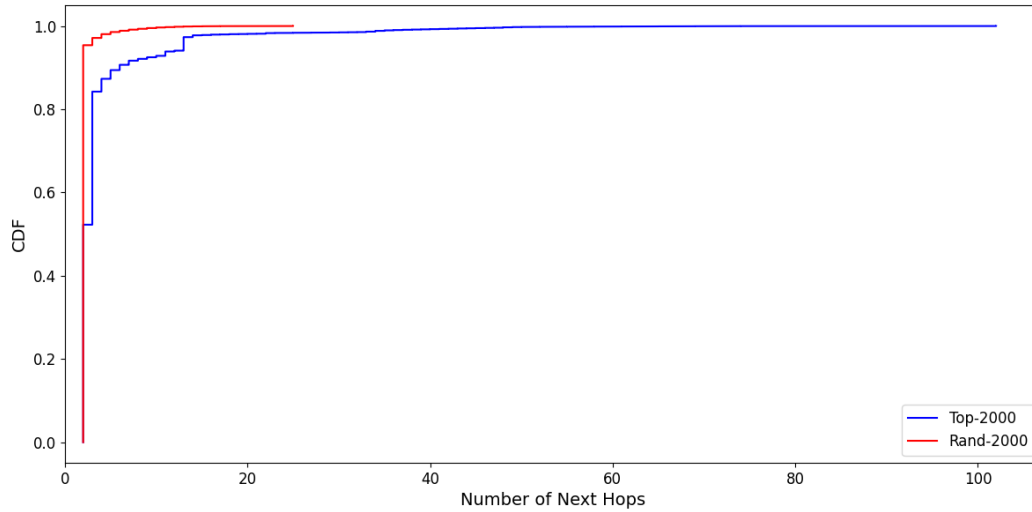


Figure 4.2: CDF of Next Hops After Load Balancers for Top-2000 and Rand-2000 Datasets

the two datasets. As shown, the Top-2000 dataset demonstrates a wider spread with 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, indicating the need for more expansive infrastructure due to the higher traffic these sites receive. In contrast, the Rand-2000 dataset's lower variability and fewer next hops suggest a simpler network configuration for less commonly used paths. These findings align with our hypothesis that the most visited websites require more extensive load balancing to manage their significant traffic demands.

### 4.3 Analysis of ASes with Most Next Hops

We next aim to understand the Autonomous Systems (ASes) with the highest average number of next hops. This reveals significant insights into the infrastructure and load balancing at an organizational level. 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. Table 4.3 shows the ASes that own the load balancers with the most next hops accross top-2000. While table 4.4 shows the data for rand-2000. The AS names come directly from the Cymru list.

<b>Top 10 ASes by Average Number of Next Hops (Top-2000)</b>		
<b>AS</b>	<b>AS Number</b>	<b>Average Next Hops</b>
ORACLE-BMC-31898, US	31898	90.69
FACEBOOK, US	32934	24.97
GOOGLE, US	15169	24.16
ADJUST-, DE	205184	20.59
CHINA169-BJ Unicom Beijing, CN	4808	19.82
CHINANET-BJ-AP, China Telecom, CN	23724	15.44
CLOUDFLARENET, US	13335	13.95
ALIBABA-CN-NET Alibaba Ads, , CN	37963	13.89
CHINANET-SCIDC-AS-AP, CN	38283	13.82
CHINANET-BACKBONE No.31, CN	4134	13.74

Table 4.3: Top 10 ASes by Average Number of Next Hops for Top-2000

The order of ASes is has many similarities across both datasets, suggesting that larger ASes with more budget tend to develop their infrastructure and add many next hops to their load balancers. This means that even if an AS already has a lot of infrastructure, it remains prevalent even in the less popular paths.

We see a significant presence of Chinese load balancers. According to Bhaskar et al. [bhaskar2021], some Chinese providers use load balancers to enforce censorship, which may explain their prevalence. Their study found that packet headers,

<b>Top 10 ASes by Average Number of Next Hops (Rand-2000)</b>		
<b>AS</b>	<b>AS Number</b>	<b>Average Next Hops</b>
ORACLE-BMC-31898, US	31898	23.03
FACEBOOK, US	32934	7.88
CHINA169-BJ Unicom Beijing, CN	4808	6.78
CHINANET-SH China Telecom, CN	134768	5.20
ADJUST-, DE	205184	4.36
CHINANET-SCIDC-AS-AP, CN	38283	3.25
CT-IDC No.287, Jin-rong Street, CN	24353	2.95
CHINANET-BACKBONE No.31, CN	4134	2.78
CT-HANGZHOU-IDC No.288, CN	58461	2.28
CHINANET-BJ-AP, China Telecom, CN	23724	2.20

Table 4.4: Top 10 ASes by Average Number of Next Hops for Rand-2000

such as source IP address and source port, can influence DNS censorship. They discovered that 37% of IPs across 56% of ASes showed changes in censorship behavior based on these parameters. This means that Chinese load balancers are used not only for load distribution but also to control access to information, demonstrating their dual role in managing traffic and enforcing censorship.

In the Rand-2000 dataset, the top 7 to 10 ASes have an average number of next hops below three. Since the minimum is two, these numbers aren't as significant, indicating that most load balancers in this range are of similar rank and complexity.

Despite the similarities in the order of ASes, the Top-2000 dataset shows higher numbers of next hops due to the higher average usage and need for more extensive load balancing. This underscores the importance of expansive infrastructure for the most visited websites, which require robust load balancing solutions to manage their significant traffic demands.

#### 4.4 Overall Analysis of Next Hop ASes Matches and Mismatches

The next section examines how often load balancers and their next hops belong to the same Autonomous System (AS) and the implications of matching and non-matching pairs.

Figure 4.3 shows the distribution of fully matching, partially matching, and no matching next hops, indicating how ASes manage their network traffic.

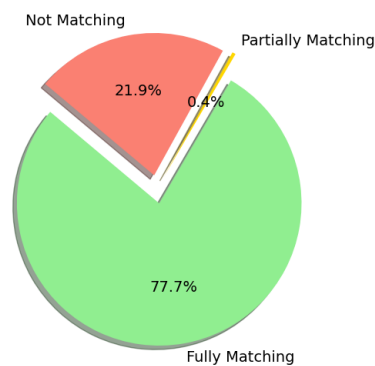


Figure 4.3: Overall Load Balancer Next Hop Matching

Following the distribution illustration, Table 4.5 presents a detailed summary of the data, showing the total counts for load balancers across different categories of match-mismatch scenarios.

Category	Number	%
Fully Matching	50,089	70.7
Partially Matching	402	0.6
No Matching	20,308	28.7
Total Unique Load Balancers	70,799	100

Table 4.5: Summary of Load Balancer Matching Statistics

Fully matching next hops, where every next hop AS matched the load balancer AS, accounted for a significant portion of all unique load balancers. This demon-

strates that large organizations often manage their network traffic internally. Internal load balancing is not only a coherent strategy but also cost-effective, especially for organizations expecting substantial traffic. Table 4.6 presents the top ASes with fully matching next hops.

<b>Top 10 ASes with Fully Matching Next Hops</b>	
<b>AS</b>	<b>Count</b>
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
CLOUDFLARENET, US	1,290
<b>Total</b>	<b>42,012</b>

Table 4.6: Top 10 ASes with Fully Matching Next Hops

Partially matching next hops, where at least one but not all next hops matched the load balancer AS, suggest a mixed routing strategy. This approach might optimize some paths within the AS while others diverge to external networks. Table 4.7 details the top ASes with partially matching next hops. However, the amount of load balancers with partially matching next hops is negligible.

<b>Top 10 ASes with Partially Matching Next Hops</b>	
<b>AS</b>	<b>Count</b>
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
<b>Total</b>	<b>402</b>

Table 4.7: Top 10 ASes with Partially Matching Next Hops

Next hops with no matching AS comprised a smaller percentage, where the load balancer AS did not match any next hop AS, suggesting external load balancing services. Table 4.8 lists the top ASes with no matching next hops.

Overall, while a majority of load balancers manage traffic within their own AS, the presence of partial and no matches suggests a diverse range of load balancing strategies.

<b>Top 10 ASes with No Matching Next Hops</b>	
<b>AS</b>	<b>Count</b>
CONE, US	6,908
Cogent-174, US	2,580
ChinaNet Backbone No.31, Jin-Rong Street, CN	2,440
ChinaNet-IDC-BJ-AP IDC, China Telecommunications Corporation, CN	854
Level3, US	849
Yahoo-1, US	608
CSUNET-NE, US	312
Google, US	216
BTN-ASN, US	207
CT-HANGZHOU-IDC No.288, CN	205
<b>Total</b>	<b>15,179</b>

Table 4.8: Top 10 ASes with No Matching Next Hops

In the case of no matching, it is possible that entities like CyrusOne (CONE) and Cogent, which are major players in the data center and ISP sectors respectively,

might have connections to multiple other ISPs. This setup allows them to balance loads across these connections to enhance network robustness, no matching could also potentially mean that ISPs are offering load balancing services to smaller data centers or their clients. While ChinaNet load balancers do show up in the no matching category, they are not the dominant presence there. Instead, ChinaNet is more frequently observed in the matching load balancers category. This suggests that despite potential involvement in censorship, ChinaNet predominantly employs its load balancing capabilities to manage internal network traffic.

#### 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. In terms of daily changes, "added" load balancers refer to those that were not present in the dataset from the previous day but appeared in the current day's list. Conversely, "lost" load balancers were present in the dataset on the previous day but did not appear in the current day's list. This dynamic illustrates the fluctuation and turnover within the dataset, showing how frequently load balancers enter and exit the observation scope.

**Top-2000 Dataset:** Over the observation period of 116 days, an average of approximately 253 load balancers were added per day in the Top-2000 dataset, and an average of about 251 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 roughly 31 days.

**Rand-2000 Dataset:** Similarly, the Rand-2000 dataset, observed over 111 days, showed that on average, approximately 161 load balancers were added per day, while an average of about 162 load balancers were lost each day. The dataset also recorded a total of 12,139 reappearances, highlighting the instances where load



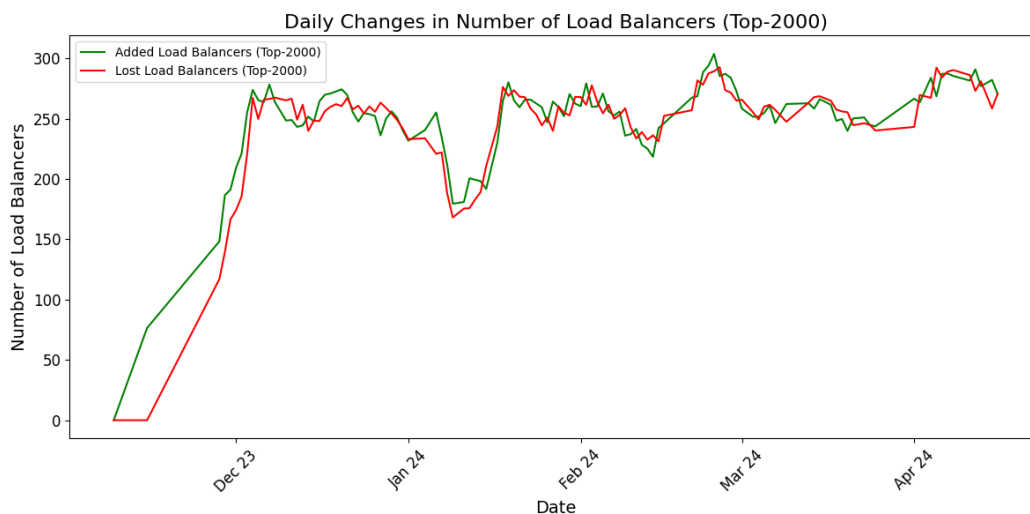


Figure 4.4: Daily Changes in Number of Load Balancers for Top-2000

balancers reappeared after being lost. The average duration of presence for a load balancer in this dataset was roughly 12 days.

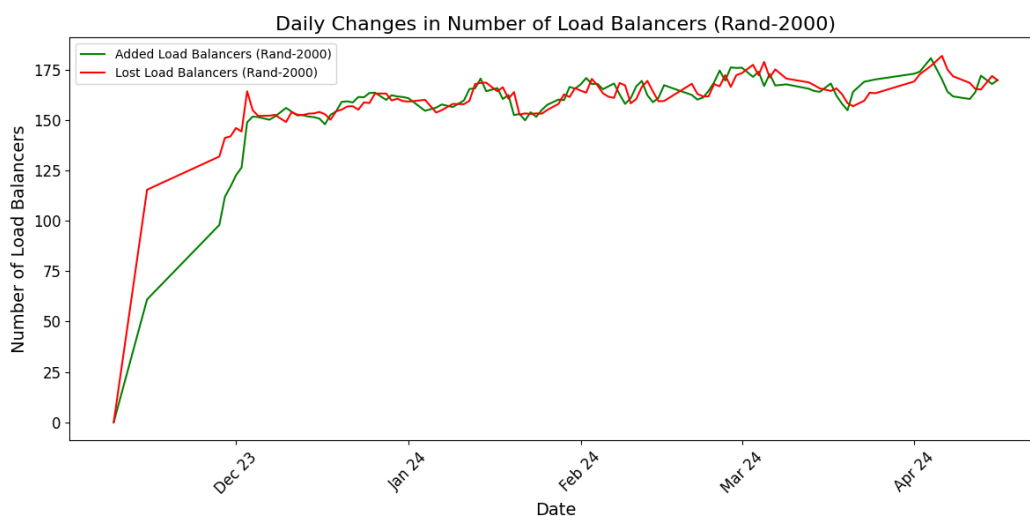


Figure 4.5: Daily Changes in Number of Load Balancers for Rand-2000

Both datasets illustrate the dynamic nature of load balancer presence, with frequent entries and exits from the datasets. To further analyze the consistency of load balancer presence, Figure 4.6 which the cumulative distribution function (CDF) for the duration of load balancers across both datasets.

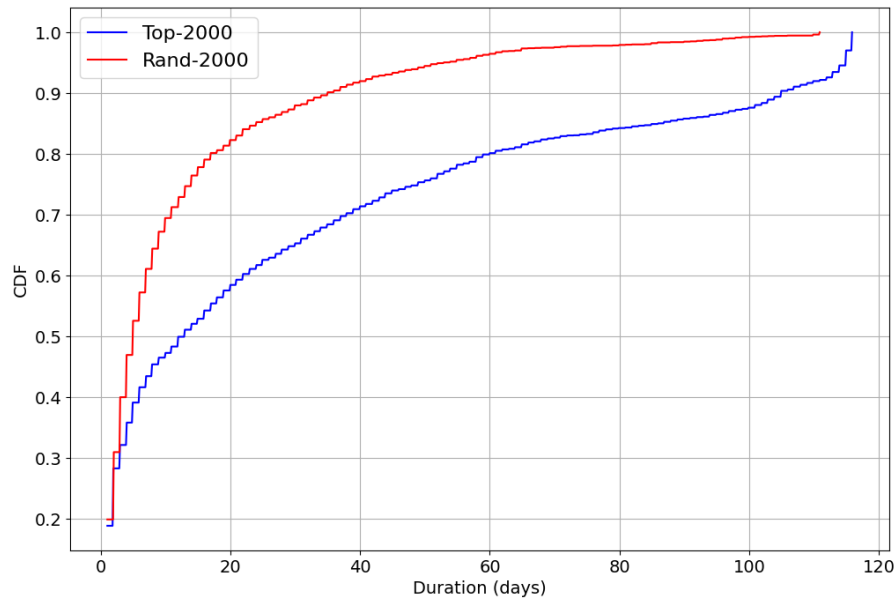


Figure 4.6: Cumulative Distribution Function of Load Balancer Durations

This CDF plot provides insights into the duration that load balancers typically remain active within each dataset. While the operational durations for the bottom 30% of the sets are similar, the datasets start to diverge around the 40% mark. Notably, load balancers in the Top-2000 dataset tend to stay active for longer periods compared to those in the Rand-2000 dataset. This divergence suggests that the most visited sites, typically represented in the Top-2000, maintain their load balancers for longer durations once a path is identified to have high traffic loads. This indicates that there is a longer turnover value for load balancers at highly visited sites. The constant for which a load balancer remains active when introduced to the network seems higher on the most visited sites.

Future work on the presence of load balancers on a more granular hourly basis might reveal whether this dynamic aspect holds true. Such studies could investigate if there is a higher presence of load balancers during peak traffic hours, suggesting that load balancers are actively managed in response to real-time traffic demands.

## 4.6 Change Over Time for Autonomous Systems

This section presents the Autonomous System (AS)-specific statistics for both the Top-2000 and Rand-2000 datasets, focusing on key metrics that illustrate how each AS manages its load balancing infrastructure over time. These metrics include the daily rate of load balancer additions and removals, indicating the dynamism and adaptability of the infrastructure; the average duration of load balancer presence, reflecting stability; and the consistency of presence, highlighting the reliability of each AS's load management strategy.

**Additions per day** indicate which AS actively scales up its load balancing infrastructure, reflecting a dynamic response to changing network demands. **Removals per day** highlight those ASes that frequently decrease their load balancer counts, possibly due to varying traffic patterns, network updates, or temporary route changes. The **average duration of presence** typically suggests stability within the network. **Consistency** is characterized by load balancers that are active throughout most of the observation period, denoting reliability and operational stability. **Inconsistency**, on the other hand, indicates sporadic or irregular use of load balancers, possibly due to infrastructural changes or misidentifications. Lastly, **fluctuation** in the number of load balancers shows a high level of dynamism, potentially adapting quickly to network requirements or changing traffic conditions.

To further clarify the metrics analyzed, each AS can have multiple unique load balancers listed in our dataset. When any of these load balancers is added or removed, it contributes to the total count for that AS in the respective category. For example, **fluctuation** is calculated based on the number of times any unique load balancer from the same AS was either added or removed across all observed days. Conversely, **consistency** measures only the presence of load balancers, counting how many separate times any load balancer from the AS is active throughout the entire duration of the study, without considering the changes per day.

## 4.7 Change Over Time for Autonomous Systems

### Top-2000 Dataset

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

Metric	Average	25th Percentile	75th Percentile	99th Percentile
# of Additions per AS per day	6.18	3.00	15.00	50.25
# of Removals per AS per day	6.15	2.00	12.47	50.10
Duration of presence per AS	7.44 days	0.29 days	13.63 days	39.01 days
# of Times an AS is present	4292.99	34.25	1820.50	65867.57
# of Fluctuation per AS	450.33	4.25	712.00	6615.72

Table 4.9: Top-2000 Dataset Statistics

The AS with the most additions per day in the Top-2000 dataset is **FACEBOOK, US**, with 50.34 additions per day, and the AS with the most removals per day is **CSUNET-NW, US**, with 50.17 removals per day. These two ASes each account for roughly 33% of the average additions and removals from the Top-2000 dataset.

Metric	AS with Highest Value	Value
Most additions per day	FACEBOOK, US	50.34 per day
Most removals per day	CSUNET-NW, US	50.17 per day
Longest average duration	CHINANET-BACKBONE No.31, CN	59.62 days
Most consistent AS	GOOGLE, US	75,863 times
Most inconsistent AS	AS-NETIA Warszawa 02-822, PL	1 time
Most fluctuating AS	CHINANET-BACKBONE No.31, CN	11,660 changes

Table 4.10: Top-2000 Dataset Highest Values

### Rand-2000 Dataset

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

In the Rand-2000 dataset, the AS with the most additions per day is **CHINANET-BACKBONE No.31, CN**, with 29.12 additions per day, and the AS with the most removals per day is **FACEBOOK, US**, with 29.40 removals per day, each accounting for 20% of the total additions and removals.

<b>Metric</b>	<b>Average</b>	<b>25th Percentile</b>	<b>75th Percentile</b>	<b>99th Percentile</b>
# of Additions per AS per day	14.08	0.75	16.25	21.60
# of Removals per AS per day	15.22	0.80	22.50	21.42
Duration of presence per AS	2.33 days	0.08 days	4.13 days	30.68 days
# of Days an AS is present	702.16	9.00	1920.10	10375.10
# of Fluctuation per AS	309.30	15.75	473.75	4290.20

Table 4.11: Rand-2000 Dataset Statistics

<b>Metric</b>	<b>AS with Highest Value</b>	<b>Value</b>
Most additions per day	CHINANET-BACKBONE No.31, CN	29.12 per day
Most removals per day	FACEBOOK, US	29.40 per day
Longest average duration	CSUNET-NW, US	44.09 days
Most consistent AS	GOOGLE, US	11,153 times
Most inconsistent AS	MTS, RU	1 time
Most fluctuating AS	CHINANET-BACKBONE No.31, CN	6,495 changes

Table 4.12: Rand-2000 Dataset Highest Values

The most consistent AS, **GOOGLE, US**, likely expects a high volume of traffic consistently, necessitating a stable and continuous load balancing infrastructure. In contrast, less consistent ASes, like **AS-NETIA**, may need to adapt their usage due to the high costs associated with maintaining load balancer infrastructure. This can lead to more sporadic use and frequent changes in infrastructure.

Although such figures highlight ASes as outliers, these systems are amongst the most popular across our dataset and unusual behavior becomes noteworthy. Interestingly, there was only one AS that had just one load balancer, notable enough to be mentioned as the least consistent in both datasets. Furthermore, the most fluctuating AS stands out significantly, with a difference of 1000 from the second most fluctuating AS, making it a focal point of analysis. The fluctuation of **CHINANET-BACKBONE No.31**, despite having a close total number of load balancers, raises intriguing questions about the factors driving such high variability in its load balancing activities.

## 4.8 Shared Next Hops Analysis

This section analyzes the shared next hops between load balancers in both the Top-2000 and Rand-2000 datasets 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.

In the **Top-2000** dataset, a total of 90,492 unique next hops were identified, with each next hop shared by an average of 5.31 load balancers and a median of 2.0. The sharing ranged from a minimum of 1 to a maximum of 52 load balancers per next hop, with a standard deviation of 6.70, suggesting considerable variability. The 75th percentile for shared next hops was 7.0, and the 95th percentile reached 18.0.

Comparatively, the **Rand-2000** dataset included 85,797 unique next hops. Here, the average number of load balancers sharing a next hop was lower at 3.35, with the same median of 2.0. The range of sharing extended from 1 to 42, and the standard deviation was 4.17, indicating a moderate level of variability. The 75th percentile in this dataset was 4.0, while the 95th percentile was 13.0.

Metric	Top-2000	Rand-2000
Total Unique Next Hops	90,492	85,797
Minimum Load Balancers per Next Hop	1	1
Maximum Load Balancers per Next Hop	52	42
Average Load Balancers per Next Hop	5.31	3.35
Median Load Balancers per Next Hop	2.0	2.0
Standard Deviation	6.70	4.17
75th Percentile	7.0	4.0
95th Percentile	18.0	13.0

Table 4.13: Summary of Shared Next Hops Statistics in the Top-2000 and Rand-2000 Datasets

Table 4.13 shows the shared next hops statistics across the Top-2000 and Rand-2000 datasets, highlighting the distribution and variability of load balancers sharing next hops.

The distribution of shared next hops, as depicted in Figure 4.7, illustrates that most next hops are shared by a relatively small number of load balancers. However, a small subset of next hops in the Top-2000 dataset are highly shared, reflecting more interconnected infrastructure and the utilization of common pathways more frequently than in the broader, randomly selected websites of the Rand-2000 dataset.

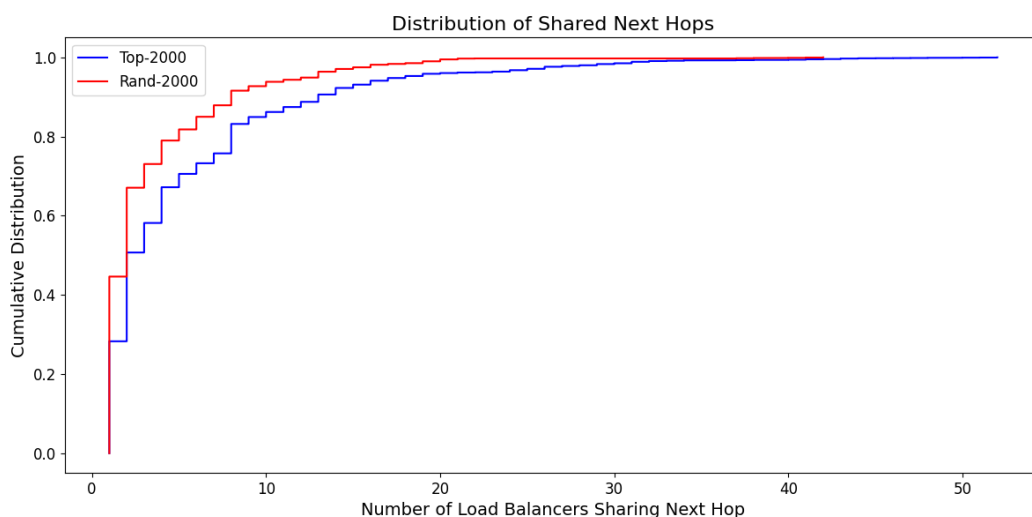


Figure 4.7: Distribution of Shared Next Hops for Top-2000 and Rand-2000

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 our research, we identified many load balancers within the Internet2 Autonomous System. The Internet2 Network, established to support data-intensive research and academic computing needs, provides tools such as the Internet2 Looking Glass [[internet2\\_looking\\_glass](#)], which allow users to run commands against network devices, enabling testing and viewing active configuration and state information.

The load balancers discovered via Paris Traceroute were also validated through the Internet2 Looking Glass, affirming the effectiveness of our detection methodology. 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, the network layer, performs load balancing by routing packets based on their IP addresses, focusing on traffic distribution across multiple servers without inspecting packet contents. This method is efficient and fast but offers less granularity in traffic management compared to higher-layer solutions, which can make decisions based on the data within the packets. Layer 3 load balancers cannot manage traffic based on content, user sessions, or specific application states, which are typically used in more common load balancing strategies. Although Layer 3 load balancers can efficiently handle large volumes of traffic, they lack the detailed traffic management capabilities provided by higher-layer solutions [[zhang](#)].



## 5.2 Cisco Express Forwarding (CEF)

Cisco Express Forwarding (CEF) enhances the efficiency of Layer 3 load balancing by pre-computing forwarding information based on historical network usage data. CEF uses two primary data structures: the Forwarding Information Base (FIB) and the adjacency table [cisco2017cef].

The **Forwarding Information Base (FIB)** is used by CEF to make IP destination prefix-based switching decisions. It is 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, ensuring all known routes are covered without the need for route cache maintenance.

The **Adjacency Table** stores 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.

When a packet arrives, it is placed into input buffers on the receiver hardware component. 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. CEF also maintains a log of forwarding history to inform future strategies.

## 5.3 Summary and Findings

Analyzing the Internet2 load balancers revealed that most next hops are defined within CEF. Internet2's load balancing is less resource-intensive, as it does not require deep inspection of packets. The adjacency table is what we can see on the looking glass, not the FIB. When something is moved to the FIB, it becomes an active next hop. However, many next hops found were active in the adjacency table,

were not discoverable using our tools.

Our findings suggest that a significant number of load balancers could be hidden in the network because Cisco routers did not see the need to update the FIB with a second next hop. The prevalence of CEF load balancing in the Internet2 network implies that while load balancing may appear dynamic, it is not dynamic in real-time. Instead, CEF learns from previous data for efficiency and does not adapt to specific packet details.

The validation process using the Internet2 Looking Glass confirmed that the load balancers we detected via Paris Traceroute indeed existed in the ground truth as seen through the Looking Glass. However, we weren't able to reach more than 97% the defined next hops in the adjacency table. We attempt some techniques discover a higher number of next hops in the next section.

#### 5.4 Modifications to the MDA Algorithm

The number of probes sent out by the Multipath Detection Algorithm (MDA) might have caused the lower than expected number of load balancers on the Internet2 network. These values were theoretically derived in the MDA paper. We modified the way it decides how many probes to send. The probe table used by MDA, which outlines the number of probes needed for varying hop counts, is shown below:

```
static const int k[][2] = {
    { 0, 0 }, { 0, 0 }, { 6, 8 }, { 11, 15 },
    { 21, 28 }, { 27, 36 }, { 33, 43 }, { 38, 51 },
    ...
    { 712, 866 }, { 720, 876 }, { 729, 886 }, { 737, 896 },
};
```

The values represent the number of probes to send for different hop counts, with the first column indicating probes for a 95% confidence level and the second column for a 99% confidence level.

To discover more load balancers, we modified the MDA algorithm by multiplying the values in the probe table by 10. This brute-force approach aimed to increase the likelihood of uncovering hidden paths and load balancers. The results show that in some instances, two or three additional load balancers were discovered. However, the increase in discoverable load balancers was inconsistent, with only about a 2% average increase. This minor increase does not justify the added cost of such measurements.

Moreover, the increased number of probes significantly extended the measurement duration, taking approximately 20 times longer than standard MDA measurements. This makes the brute-force approach impractical for extended data collection. Interestingly, this approach validates the accuracy of the original MDA probe table, confirming its effectiveness in typical scenarios.

The challenges faced during the enhanced probing trials highlight the limitations of brute-force approaches in discovering hidden load balancers. While increased probes reveal on average 2% more load balancers, the overall impact was minimal compared to the substantial increase in measurement time. This suggests that simply increasing the number of probes is not the most effective way to uncover hidden load balancers.

Future work could focus on optimizing the probe sending strategy, exploring adaptive probing techniques, and integrating additional contextual information to improve the efficiency of the MDA algorithm. Targeted testing over a longer period could reveal rotations of next hops, and using different types of packets might help uncover more load balancers. These and other potential improvements are discussed in the future works chapter.

## *Chapter 6*

### SUMMARY

In this study, we investigated the prevalence and characteristics of load balancing on the Internet using comprehensive data collection and analysis. Our research aimed to quantify load balancing behavior and its impact on network performance, providing insights into the structure and dynamics of modern web traffic.

#### **Data Collection and Methodology**

In Chapters 3 and 4, we detailed our methodology for data collection, which involved daily measurements from November 2023 to April 2024. We utilized the Alexa Top 1 Million Websites list to select two subsets: the Top-2000 and the Rand-2000 lists. Our measurements were conducted using Paris Traceroute with the Multipath Detection Algorithm (MDA) to identify and classify load balancers.

#### **Load Balancer Distribution and Trends**

We observed significant trends in load balancer distribution:

- The Top-2000 dataset showed a high prevalence of load balancers, with an average of 82.2% of paths containing load balancers. The Rand-2000 dataset had 62.5%, indicating widespread use across diverse websites. Previous data from [augustin2010measuring] showed that 39% of paths traversed load balancers, with up to 72% when considering different types of load balancing. Our data focuses on the most common paths, providing a different perspective. Our overall average is about the same as their highest value, which could suggest that the overall number has remained similar.

- Analysis of next hops revealed that popular websites often utilize more complex load balancing structures, while randomly selected sites showed simpler configurations.
- Our next hop analysis suggested that the infrastructure supporting the most popular websites is more interconnected and utilizes shared pathways more frequently than the broader, randomly selected websites.
- The consistent presence of ASes like GOOGLE, US and CONE,US , in the Top-2000 dataset but their inconsistent presence in the Rand-2000 dataset suggests deliberate adjustments based on traffic expectations. The Top-2000 list likely experiences more predictable high traffic, requiring continuous load balancing, whereas the Rand-2000 list may see more variable traffic patterns, leading to less consistency.
- We observed significant presence of Chinese load balancers, which not only manage traffic but also enforce censorship.

### **Dynamic and Static Properties**

Our study identified both dynamic and static properties of load balancing:

- Load balancers showed dynamic behavior, with daily additions and removals reflecting adaptive traffic management. The average presence duration was 31.10 days for the Top-2000 dataset and 12.49 days for the Rand-2000 dataset.
- Static properties included the overall number of load balancers discovered for each list. While the load balancers themselves are volatile, load balancing remains constant over time.

## **Layer 3 Load Balancing and Cisco Express Forwarding**

We explored Layer 3 load balancing, focusing on Cisco Express Forwarding (CEF):

- CEF optimizes packet forwarding by utilizing pre-computed forwarding information, but does not react to the network condition in real time.
- Our analysis of Internet2 load balancers revealed a high reliance on CEF, with many next hops predefined in CEF tables.

### **Challenges**

- It was a challenge to identify load balancers we knew were present on Internet2 routes.
- Enhanced probing trials using a modified MDA algorithm provided limited success, suggesting the need for more sophisticated techniques.
- Future work could focus on adaptive probing methods and integrating contextual information to improve load balancer detection.

Overall, our research contributes to a deeper understanding of load balancing on the Internet, highlighting both its complexities and areas for further exploration.

## *Chapter 7*

### FUTURE WORK

There are several areas related to this research that we could not explore because of time or scope constraints.

#### **Increasing the Number of Destinations**

One potential for future work is to increase the number of destinations significantly. [augustin2010measuring] conducted measurements from 15 sources to over 68,000 destinations, revealing that many routes pass through load balancers. Given enough time and resources, our goal would be to match or exceed these numbers to get a better understanding of load balancing across the Internet. While our different lists attempted to show an as complete as possible view, higher numbers would ultimately help provide a more complete picture of global traffic patterns.

#### **Adapting MCA to Scamper**

Another area for future research is to adapt the Multipath Classification Algorithm (MCA) to work with Scamper or a new traceroute tool. MCA improves the identification of load balancers by considering different bits in the packet header. Integrating MCA with a tool like Scamper would allow for efficient data collection, similar to the Multipath Detection Algorithm (MDA) and Paris Traceroute. This could help us identify and classify load balancers more accurately and efficiently.

#### **Per-Flow and Per-Destination Load Balancers**

[4261334] highlighted the importance of distinguishing between per-flow and per-destination load balancers. Per-flow load balancing ensures that all packets in

the same flow follow the same path, while per-destination load balancing distributes traffic based on destination IP addresses. Future work could explore these two types of load balancing in more detail, examining how common they are and their impact on network performance. Understanding these differences could help improve network efficiency and reliability.

### **Developing Tools for Identifying CEF Load Balancers**

We spent considerable time trying to modify MDA to discover more Cisco Express Forwarding (CEF) load balancers. A future goal could be to develop a tool that can better identify CEF load balancers or detect if some load balancers are hidden. This would help in providing a more accurate picture of load balancing techniques used in modern networks.

Overall, expanding this research in these directions could provide deeper insights into load balancing on the Internet, helping improve network performance and reliability.