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

Number *Page*

ACKNOWLEDGEMENTS

TBD

Omar’s Thesis

Abstract

by

OMAR LOUDGHIRI

TBD

Chapter 1

INTRODUCTION

1.1 Load Balancing

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 also provides redundancy; if a server fails, traffic is redirected to maintain continuous access.

Load balancing enhances security by minimizing attack surfaces and rerouting traffic if a server is compromised. Additionally, it 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.

A significant application of load balancing is the ability to utilize more capacity by aggregating multiple smaller links. For example, combining two 1 Gbps links instead of a single 10 Gbps link may be more cost-effective or easier to implement. This approach can improve network reliability, as multiple servers or router interfaces can mitigate some failure points. However, it is important to note that load balancing itself can become a single point of failure if the load balancer goes down.

1.2 Project Motivation and Goals

The primary motivation for this project was 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 most used paths in the internet, this research aims to enhance our understanding of their impact on network performance and security. 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.

1.3 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 was 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.

1.4 Areas of Study

Chapter 2

RELATED WORKS

2.1 Paris Traceroute

2.1.1 General Introduction

The traditional model of the Internet assumes a single path between a pair of end-hosts. However, modern commercial routers often include load balancing capabilities, creating multiple active paths between hosts. This shift challenges the conventional single-path assumption used by many Internet applications, network simulation models, and measurement tools.

In their paper *Measuring Load-balanced Paths in the Internet*, Augustin et al. [1] from LIP6 at Université Pierre et Marie Curie present a comprehensive study on load-balanced paths. They highlight the significance of recognizing load balancing in contemporary networking by demonstrating how it affects traffic distribution and path diversity. Published in 2007, the paper's conclusions may be somewhat outdated given the advancements in networking technology since then.

The authors enhance 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 underscores the necessity for the research community to reconsider the concept of a single Internet path and highlights the importance of accurately measuring and understanding load-balanced paths. The insights gained from this work are critical for developing more realistic network models and improving the design and reliability of Internet applications.

2.1.2 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, it helps diagnose network congestion, detect points of failure, and understand the network's topology. Traditional traceroute, however, 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.

2.1.3 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.

2.1.4 Usage of Paris Traceroute

The paper by Augustin et al. [1] 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, extensive measurements will be conducted 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:

- Expanding the scope of measurement to include a more diverse set of source-destination pairs.

- Investigating the effects of load balancing on different types of network traffic and applications.

2.2 Scamper

2.2.1 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.

2.2.2 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.

2.2.3 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. The 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.

2.3 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.

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 minimal 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.

2.4 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 my 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. Detailed explanations of MDA's integration and performance in my study will be provided in a later section of this document.

Chapter 3

DATASETS

3.1 Alexa Top 1 Million Websites

The Alexa Top 1 Million Websites list was utilized as the primary dataset for this research. The latest update to this list was in February 2023, ensuring its relevance at the start of the study. The Alexa list is renowned for its accuracy in ranking the most visited websites globally, making it an ideal source for identifying widely used routes on the Internet.

Alexa rank employs a proprietary methodology that combines visitor engagement and estimated traffic to generate the ranking list. Data is analyzed over the past three months, considering factors such as unique daily visitors and page views. Unique visitors are defined as the number of distinct Alexa users visiting a site on a single day, while page views represent the total number of URL requests generated by Alexa users for a site. The Alexa Toolbar, available as a browser extension for Chrome, Internet Explorer, and Firefox, facilitates data collection by recording URLs and IP addresses, sending traffic data to Alexa's central server.

The Alexa ranking is particularly useful for this research as it includes a diverse range of popular websites. This diversity is crucial for finding load balancers on heavily trafficked routes, allowing for a comprehensive analysis of load balancing behavior across different types of web traffic and services. By focusing on these high-traffic websites, the study can better capture the impact of load balancing on Internet reliability and performance.

It should be noted that while Alexa provides reliable rankings for the top 100,000 websites, its accuracy diminishes for lower-ranked sites due to limited data. This limitation is not heavily impacting our data collection as we are taking 2000 random samples from those top 100,000 websites for different stages of the data collection.

The domain names from the Alexa list are then resolved into the most current IP addresses using the DNS lookup tool. In the context of this project, this ensures that the analysis is based on up-to-date routing information, allowing for accurate identification and characterization of load balancers on the most heavily trafficked Internet routes.

3.2 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 robust service dedicated to mapping IP numbers to BGP prefixes and ASNs, based on BGP feeds from over 50 peers, updated at four-hour intervals.

This service supports multiple query methods, including WHOIS (TCP 43), DNS (UDP 53), and HTTPS (TCP 443). 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

It is important to note that the IP to ASN mapping service is not a GeoIP service. The country code, registry, and allocation date are based on data from regional registries, such as ARIN, RIPE, AFRINIC, APNIC, and LACNIC. As such, the accuracy of this information depends on the data present in the RIR databases.

In the context of this project, the Team Cymru service was essential for obtaining accurate WHOIS lookup results and mapping IP addresses to their respective ASNs. My advisor, Professor Mark Allman, automated a script that pulls all their current IP to ASN mappings every month. This list was used to cross-reference the IPs identified as load balancers, their next hop, and their destination IPs. This comprehensive cross-referencing provided detailed insights into the load balancers discovered.

Chapter 4

DATA COLLECTION AND ANALYSIS

4.1 Discovering Load Balancers

The data collection process for this research involved using a Python script to automate the execution of the Paris traceroute tool and process its output. The script was designed to systematically gather information about network routes and load balancers.

The process begins by reading two distinct datasets from the Alexa Top 1 Million Websites list, stored in a CSV file. This ensures that the analysis focuses on high-traffic websites likely to traverse significant portions of the Internet's infrastructure. The two datasets used are:

- **Rand-2000 Dataset:** This dataset comprises 2000 random domains selected from the top 100,000 websites on the Alexa list. It aims to provide statistically significant data about the overall Alexa dataset and to examine the Internet's topology comprehensively.
- **Top-2000 Dataset:** This dataset 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 highly significant web traffic.

4.1.1 Paris-Traceroute on datasets

For each domain in these datasets, the script runs the Paris traceroute tool using the Multipath Detection Algorithm (MDA). The traceroute is executed with a timeout of 40 seconds to prevent any hanging processes. The output from the traceroute is captured and saved to a file in a designated output directory. If a timeout or error occurs, appropriate messages are logged to keep track of any issues during execution.

Once the traceroute output is obtained, the script parses the data using regular expressions to extract relevant IP addresses and paths. This parsed data is then organized into a JSON structure, mapping each source IP to its multiple destination IPs. This structure is essential for analyzing the network paths and identifying load balancers.

In addition to the traceroute data, the script utilizes the Team Cymru IP to ASN mapping service to resolve IP addresses to their corresponding Autonomous System Numbers (ASNs). This list is used to cross-reference the IPs identified as load balancers, as well as their next hops and destination IPs.

By combining these datasets and cross-referencing the information, the study aims to provide a comprehensive analysis of the Internet's load balancing behavior and network topology. This approach ensures that both a broad overview and a focused examination of the most significant web traffic are achieved.

4.1.2 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 all 2000 IPs in approximately 23 hours, making it possible to conduct measurements on a daily basis.

The measurements were run continuously from January 23rd, 2024, to April 16th, 2024, on the ICSI virtual machines based in California. 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.

4.2 Data Analysis

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

Initially, the collected JSON data, which contains information about network routes and load balancers, was loaded for processing. A dictionary was used to keep track of load balancers and their associated IP groups. For each domain in the JSON data, the load balancer information was processed, grouping the IPs by their AS information while filtering out entries without information about ASs.

4.2.1 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 JSON data. Additionally, AS information for IPs was loaded to provide context for the identified next hops. A dictionary was populated to store next hops for each load balancer, and another dictionary was used to store sets of load balancers sharing the same 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.

4.2.2 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. The process involved loading the JSON data and counting the occurrences of matches and mismatches. The results were then visualized using a bar chart to compare the frequency of matches and mismatches for the top 20 next hop ASs. This analysis provided a detailed view of the relationships between load balancers and their next hops.

4.2.3 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. This step involved loading the JSON data and counting the number of next hops for each load balancer. The mean, median, and mode of the next hop counts were calculated to summarize the distribution. The results were visualized using a histogram, showing the frequency distribution of next hop counts. This analysis helped characterize the diversity of paths taken by traffic after passing through load balancers.

4.2.4 Filtering Subnet Consistency

The analysis also included filtering entries to ensure that all IP addresses associated with a load balancer were within the same subnet. This step aimed to identify inconsistencies and refine the dataset. The process involved checking if IP addresses in each group belonged to the same subnet. Entries that did not meet this criterion were filtered out. The filtered dataset provided a more accurate representation of network routes and load balancers.

4.2.5 Mapping IPs to ASNs

To enrich the dataset with AS information, IP addresses were mapped to their corresponding ASNs. This process involved creating and managing an index of IP to ASN mappings. The index was used to resolve IP addresses found in the traceroute data. For each domain, the load balancer IPs and their next hop IPs were mapped to their respective ASNs.

4.2.6 Identifying Top Load Balancer Groups

Finally, the analysis identified the top load balancer groups based on the number of domains they served. The process involved loading the JSON data and grouping load balancers by their AS information. The groups were then flattened into a list of tuples, each containing the count of domains, load balancer IP, AS information, and the list of domains. The list was sorted by count to identify the most significant load balancer groups. The results were saved to a text file and visualized using a horizontal bar chart. This analysis highlighted the most prominent load balancers and their impact on network traffic.

Chapter 5

TOPOGRAPHY OF LOAD BALANCERS

5.1 Statistics on Load Balancers Found

Still running code over all my data

5.2 Analysis of Common Next Hops

The analysis over a dataset from 02-21-2024 **(I will change this to include over the whole time, just wanted to give an example, won't be included in final)** revealed a total of 262 common next hops shared by at least three load balancers across different domains. The distribution of these common next hops is illustrated in Figure 5.1. The bar chart shows the frequency of next hops shared by varying numbers of load balancers, providing insights into the prevalence of shared network paths.

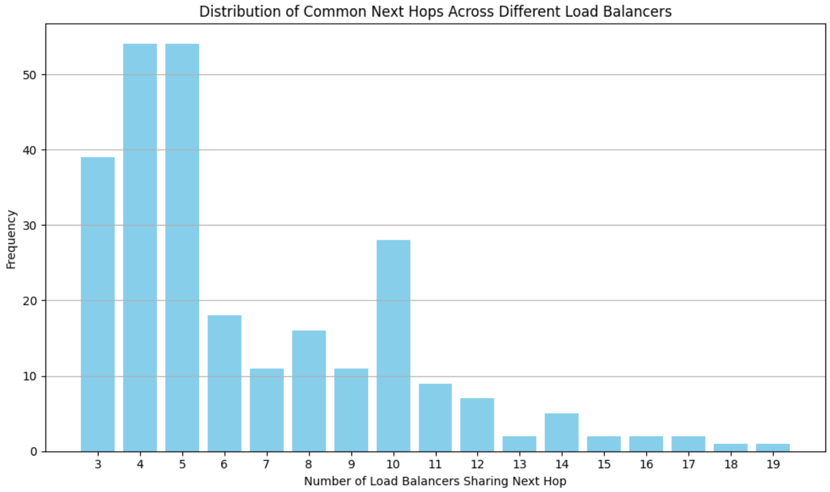

Figure 5.1: Distribution of Common Next Hops Across Different Load Balancers

With 262 occurrences of next hops shared by at least three load balancers, it suggests that network providers often deploy load balancers in a centralized manner.

This results in multiple load balancers directing traffic to a common set of next hop IP addresses within the same Autonomous System (AS). This strategy likely enhances network performance and redundancy.

Another possibility is that load balancers may lead to other load balancers, creating a layered load balancing architecture. This could result in shared next hops across different domains as traffic is further distributed. These hypotheses were further tested by examining the AS information for the load balancers and their next hops, which will be detailed in the next section. Understanding these patterns is crucial for comprehending the deployment and operation of load balancers in the Internet's topology.

Chapter 6

THIS IS THE SIXTH CHAPTER

Chapter 7

THIS IS THE SEVENTH CHAPTER

Chapter 8

THIS IS THE EIGHTH CHAPTER

Appendix A

QUESTIONNAIRE