A Practical study of the problems of current internet routing tables

Arnav Ghosh

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation
Accessed from
A PRACTICAL STUDY OF THE PROBLEMS OF CURRENT
INTERNET ROUTING TABLES

Arnav Ghosh

November 2012

Thesis submitted in partial fulfillment of the requirements
for the degree of
Master of Science in
Networking and System Administration

Department of Networking, Security and Systems
Administration

Rochester Institute of Technology
B.Thomas Golisano College
of
Computing and Information Sciences
Rochester Institute of Technology

B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in
Networking and System Administration

Thesis Approval Form

Arnav Ghosh

A practical study of the problems of current
Internet routing tables

Thesis Committee

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Bruce Hartpence</td>
<td></td>
<td>11/01/2012</td>
</tr>
</tbody>
</table>

Chair

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Nirmala Shenoy</td>
<td></td>
<td>11/01/2012</td>
</tr>
</tbody>
</table>

Committee Member

<table>
<thead>
<tr>
<th>Name</th>
<th>Signature</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof. Daryl Johnson</td>
<td></td>
<td>11/01/2012</td>
</tr>
</tbody>
</table>

Committee Member
Thesis Reproduction Permission Form

Rochester Institute of Technology
B. Thomas Golisano College
of
Computing and Information Sciences

Master of Science in
Networking and System Administration

A PRACTICAL STUDY OF THE PROBLEMS OF CURRENT
INTERNET ROUTING TABLES

I, Arnav Ghosh, hereby grant permission to the Wallace Library of the Rochester Institute of Technology to reproduce my thesis in whole or in part. Any reproduction must not be for commercial use or profit.

________________________  __________________________
Date                        Signature of Author
ABSTRACT

The phenomenal growth of the Internet amazes even the creators of this worldwide network. Apart from the constantly changing data protocols and services that the Internet has to adapt to, the sheer volume of users has been one of the biggest challenges with which the Internet is coping with. This thesis is directed towards the study of Internet scale routing tables in a lab environment to understand the dynamics of route processing by routers and the effect of increasing number of routing table entries on the overall performance of the network. This study is an effort to simulate an Internet scale network in the lab to shed light on some of the practical problems of the Internet routing table size and its performance and security implication.
ACKNOWLEDGEMENTS

I would like to thank Prof. Bruce Hartpence, Prof. Darryl Johnson and Prof. Nirmala Shenoy for their ideas which helped me complete this study.
CONTENTS

Abstract v
Acknowledgments vi
Contents vii
List of Figures x
List of Tables xi

1. Introduction 1

2. Problem Statements 5

3. The Proposal 7

4. Hypotheses 8

5. Related Work 9

6. Methodology and Approach 13

7. Experiments 14

7.1 Phase 1 Case Study 1 ...................................................... 14
    7.1.1 Equipment ...................................................... 15
    7.1.2 Network Diagram ............................................. 15
    7.1.3 Setup Explanation ........................................... 15
    7.1.4 Procedure .................................................... 16
    7.1.5 Data Collection .............................................. 18
    7.1.6 Data Sample .................................................. 20

7.2 Phase 1 Case Study 2 .................................................. 21
    7.2.1 Equipment .................................................... 21
    7.2.2 Network Diagram ............................................. 21
    7.2.3 Setup Explanation ........................................... 22
7.2.4 Procedure ................................................................. 22
7.2.5 Data Collection ............................................................ 23

7.3 Phase 2 Case Study 1 ....................................................... 24
7.3.1 Equipment ................................................................. 24
7.3.2 Network Diagram ......................................................... 24
7.3.3 Setup Explanation ......................................................... 25
7.3.4 Procedure ................................................................. 25
7.3.5 Data Collection ............................................................ 26

7.4 Phase 2 Case Study 2 ....................................................... 27
7.4.1 Equipment ................................................................. 27
7.4.2 Network Diagram ......................................................... 27
7.4.3 Setup Explanation ......................................................... 28
7.4.4 Procedure ................................................................. 28
7.4.5 Data Collection ............................................................ 29

8. Results .......................................................... 30

8.1 Phase 1 Case Study 1 ....................................................... 30
8.1.1 Observations & Inferences ............................................. 32
8.1.2 Observations & Inferences ............................................. 33
8.1.3 Observations & Inferences ............................................. 34

8.2 Phase 1 Case Study 2 ....................................................... 37
8.2.1 Observations & Inferences ............................................. 40
8.2.2 Observations & Inferences ............................................. 41
8.2.3 Observations & Inferences ............................................. 42

8.3 Phase 2 Case Study 1 ....................................................... 45
8.3.1 Observations & Inferences ............................................. 47
8.3.2 Observations & Inferences ......................................................... 49

8.4 Phase 2 Case Study 2 ................................................................. 50
  8.4.1 Observations & Inferences ..................................................... 52
  8.4.2 Observations & Inferences ..................................................... 55

9. Conclusions ........................................................................... 57

10. Future Work ............................................................................ 58

11. Bibliography ............................................................................. 59

12. Appendix ................................................................................. 60
LIST OF FIGURES

1. Figure 1a, 1b and 1c - Apple Inc Internet block snapshots 2
2. Figure 2 - Ph1 case study 1 network diagram 15
3. Figure 3 - Data sample example 20
4. Figure 4 - Ph1 case study 2 network diagram 21
5. Figure 5 - Ph2 case study 1 network diagram 24
6. Figure 6 - Ph2 case study 2 network diagram 27
7. Figure 7 - Ph1 case study 1 route count Vs Memory utilization graph 32
8. Figure 8 - Ph1 case study 1 route count Vs Pkt processing time graph 33
9. Figure 9 - Ph1 case study 1 route count Vs Jitter graph 34
10. Figure 10 - Ph1 case study 2 route count Vs Memory utilization graph 40
11. Figure 11 - Ph1 case study 2 route count Vs Pkt processing time graph 41
12. Figure 12 - Ph1 case study 1 route count Vs Jitter graph 42
13. Figure 13a, 13b and 13c – Ph1 result comparison graph 43
14. Figure 14 - Ph2 case study 1 route sample Vs Pkt processing time graph 47
15. Figure 15 - Ph2 case study 2 route sample Vs Pkt processing time graph 52
16. Figure 16 – Ph2 result comparison graph 56
LIST OF TABLES

1. Table 1 – Ph1 Case Study 1 Static Route Increment data  
   Page 30
2. Table 2 - Ph1 Case Study 2 BGP route Increment data  
   Page 36
3. Table 3 - Ph2 Case Study 1 Static route sample IP packet processing data  
   Page 45
4. Table 4 - Ph2 Case Study 2 BGP route sample IP packet processing data  
   Page 50
1. INTRODUCTION

In the 1970s an experimental network to connect a few research universities and government bodies, known as ARPANET, was started. It was an experiment that never ended and has evolved to become the world's largest network, the Internet. The Internet enables us to access information and data from around the globe within seconds, an incredible achievement, keeping in mind the fact that the kind of data we send over the Internet is nowhere close to what the researchers had in mind when they had started the experiment.

The Internet as we know it is a huge mesh, interconnecting network worldwide. The meshed Internet architecture makes it inherently resilient to most network failures. This meshed architecture is the strength of the Internet and makes it tolerant to link failures, node failures and sometimes even network segment failures.

The Internet is based on Internet Protocol or IP addressing that was first developed in 1980. IP addresses are logical addresses that make identification and communication possible across multiple networks. IPv4 addresses were initially divided into classful groups of A, B, C, D and E. Classful addressing and routing protocols meant that addresses in a particular class would be advertised to different networks across classful boundaries with the class specific subnet mask. This method was inefficient and resulted in wastage of IPv4 addresses. In 1993 researchers came up with Classless Inter-Domain Routing (CIDR) that allowed networks belonging to a specific class being advertised with any possible valid subnet mask. This development was a great step and allowed efficient use of the IPv4 address space. CIDR was being widely used all around the world and as more and more IPv4 addresses were allocated, a hidden problem of Internet routing table explosion emerged. Since CIDR allowed the propagation of IPv4 networks with any possible valid subnets masks, ISPs, enterprises and universities, etc started advertising their network block in multiple smaller chunks.
rather than one huge network.

For example, Apple Computer Inc. owns the entire 17/8 subnet and would advertise a single 17.0.0.0/255.0.0.0 block before the existence of CIDR. With the introduction of CIDR, Apple Inc divides this class A subnet into 115 subnets as shown below,
Figures 1a, 1b and 1c are routing table snapshots from a router running BGP with the entire Internet routing table on it. This information has been obtained by using a publicly accessible router server at http://www.routeviews.org [13]. The figures clearly show one class A route being represented as 115 variably subnetted routes. In this example we can see CIDR in action. This example highlights the root cause of the Internet routing table explosion.

The other important factor that has led to the huge increase in current Internet routing table size is the sheer volume of users and nodes that are now connected to the Internet. There are 1,966,514,816 Internet users as of June 30th, 2010 [15]. This means almost one-third of the population of this world’s population are now connected to the Internet. The Internet has
penetrated almost each and every country. This wide spread geography of the Internet meant that large chunks of IP addresses had to be broken down to cater to the needs of people around the globe. This need has led to the wide spread use of CIDR to break down classful addresses into smaller subnets to be distributed among various nations, ISPs, cities, businesses and homes. This exponential increase in the users and nodes connected to the Internet eventually led to a large number of advertised networks; meaning that the routers at the core of the Internet would have to keep all these advertised networks in their routing table and then make an appropriate routing decision. For each packet that a router has to route, it has to perform a route lookup and if the routing table is large, the route lookup can cause significant delay. This factor is a major problem when building high performance routers.

Border Gateway Protocol or BGP is the routing protocol of the Internet. BGP is a path vector routing protocol making complex routing decisions based on various negotiated policies. If there are multiple routes to a single network, BGP selects the best route and places that particular route on the routing table. Since the Internet is a huge mesh there could be many possible ways to reach a particular network. The BGP Routing Information Base or RIB, table contains all possible routes to all the networks in use and then selects the best routes. As of July, 2010 the number of RIB entries in Internet core routers has reached to more than 325000 [1]. The huge size of the RIB table is another concern that vendors are coping with. Processing such a large number of entries causes significant load on the routers and affects how they process routing.
2. PROBLEM STATEMENT

The unprecedented growth of the Internet has led to some fundamental routing problems. The routing table and the BGP RIB sizes have become enormous and continue to increase in size with each passing day. The global routing table size increases by almost 16 percent annually compared to the RAM speed which increases only 10 percent annually [7], even though Moore's law remains true today. Chip manufactures like Intel and AMD are able to double the density of transistors based on Moore's law, but these chips are prohibitively expensive and not used as a commodity that goes into every PC. Router manufacturers now use specialized products that do not make the routers as cost effective, thus Moore's law does not apply to building high-end routers [7].

Routers at the core of the Internet make routing decisions after going through large tables that affect their efficiency. If a router reaches a point where it does not have memory to store any more routes, it might simply crash or begin behaving abnormally [2]. The presence of these large tables also could lead to a new kind of Denial of Service or DoS attack.

The other issue with large routing tables and BGP tables is the time it takes these routers to converge. If a router crashes or routing information changes for a large network segment, these routers have to process the new routing information which takes a significant amount of time, even with enough memory and CPU cycles. Routers can only process a limited amount of information at a given time interval. CPU cycles are allocated in a way that the new routing information does not overwhelm the router. This can result in a significant amount of down time.

Vendors and ISPs have started to look beyond traditional routing architectures and many of them have started migrating to Multi Protocol Label Switching or MPLS which is a relatively new
switching technology. In order to run a network with BGP, all core routers as well as the edge routers, need to run BGP, whereas when using BGP in conjunction with MPLS, only the edge routers need to run BGP, reducing routing load. Moreover, traditional IP routing is connectionless, which effectively means that route lookup would be performed each and every time a packet flows through a network, even though it is between the same source and destination. MPLS packet flows, on the other hand, are connection-oriented where packets are routed on pre determined Label Switched Paths (LSPs). This significantly reduces routing load as the routes are pre determined for all packets for a particular connection.

The vendors are trying hard to keep up with this growth but since the rate at which the Internet is growing is much higher than the rate at which hardware technology is progressing, vendors have begun to realize that they could eventually be heading toward a dead end.
3. The Proposal

As mentioned and cited earlier in this document, the Internet currently is facing some major issues, and routing scalability is one of the most important ones. This research topic helps us better understand the practical implications of these issues and listed below are the areas of my study:-

1. Implementing a lab-based setup capable of handling routing information similar to the scale of the present Internet. This setup would provide a clear understanding of the dynamics of the current Internet in a lab environment and would allow for easy modifications of the setup as well as data collection. With the lab-based setup which emulates the Internet, statistical data for various performance parameters like packet processing time, jitter and router memory would be collected.

2. Understanding the effect of loading the routers beyond their resources and how it would affect the forwarding decisions made by the routers.

3. Understanding the way routers handle large routing tables. The lab setup would be very helpful in analyzing whether routers dynamically process frequently used prefixes faster than other routes. The other important thing to understand and study would be the way a router arranges different prefixes, whether it is similar on all routers or whether the router determines the sequences dynamically.

4. Understanding the security implications of the way routers handle the routing and RIB tables based on all the above. I would like to determine if it is possible to degrade network performance by attempting to access unused prefixes in the routing tables, similar to some DoS attacks.
4. HYPOTHESIS

*Large routing tables affect the performance of routers, and studying the route lookup process on routers with large routing tables could enable us to generate a new exploitation technique to slow down the routing process by forcing lookup of infrequently used routes.*

This research will aid in validating this hypothesis through in-depth study of routers with large routing information both in static routing and dynamic routing protocols. Analysis of empirical data collected during the experiment using Cisco 2811 series routers will assist this validation.
5. RELATED WORK

This section presents a few related research ideas and results that are significant to this study of current Internet routing tables. The literature review has been broadly classified into two sections. The first section focuses on previous research information about Internet routing instability and their causes. This section is important because my research will focus on overloading routers to ensure their performance parameters. The second section deals with how routers handle route information and how they process their huge tables. This section is important as this could have various security implications which are also part of my research.

The authors of “Internet Routing Instability” [3] and “Origins of Internet Routing Instability” [8] papers were among the first to research the various reasons of Internet routing instability. The authors pointed out that routing instability in the Internet could originate due to router configuration errors, link problems, and lack of router resources. Any one of these could cause routes to appear or disappear (route flaps), which could eventually lead to degradation of end to end performance of the Internet. My research would try to focus on the router resources as a factor of Internet routing instability. My research will determine the effect that router resources or a lack thereof has on the forwarding process.

"An empirical study of router response to large BGP routing table load" [4], was an Internet Measurement Workshop paper. The authors performed lab-based analyses of a few commonly used routers from different vendors. The main goal of this research was to study the effects on a router when it is overloaded with excessive routing information, such as large routing updates and large routing tables. An incorrectly configured router could sometimes introduce a large amount of routing information into the Internet routing system. This could result in crashing or incorrect operation of other routers in the chain and could eventually lead to cascading failures.
The authors tested a few routers from Cisco and Juniper Networks. They experimentally loaded the routers beyond the specified limit. The various routers running different firmware versions each responded differently, but the overall result proved that the routers would stop responding to the particular peer that caused the problem, or reset all its BGP peers, or the particular interface would stop responding. Any of these would result in network disruptions and at some point would require manual intervention. The researchers were able to conclude that it is possible for routers to fail if they were inadequately equipped to handle large amount of routing information injected into the network.

A part of my research would be similar to what the authors of the paper [4] did. I would also load the routers with a large amount of routing information to check if they crash or not.

My research would go a step further than the above mentioned paper to determine the effects that an over loaded router has on routing traffic.

The authors of “BGP routing stability of popular destinations” [5], workshop paper performed experiments in the AT&T backbone to understand the stability of BGP in the Internet. The authors wanted to establish a relationship between the popularity of prefixes and its BGP stability. Previous research had shown that BGP updates were very frequent and could cause traffic delivery problems. This research was the first of its kind to understand the dynamics of the BGP updates. With this research, the authors were able to establish the fact that a few popular destinations were responsible for the carrying majority of the traffic and had relatively stable routes. The majority of the BGP updates and instability are caused by unpopular destinations that do not carry a lot of traffic. My research would have a similar outlook but my experiment would focus not merely on BGP updates for popular destinations but also on the routing table sequencing itself.
This research would try to ascertain whether more frequently used routes are placed higher on the routing table for faster lookups.

“AS-level Traffic Characteristics and their Implications on Traffic Engineering” [6], was a presentation given at the Measuring and Modeling the Internet Symposium. This was a study to understand the Autonomous System or AS level dynamics of the internet traffic flow. This study also reinstated the fact that a small percentage of the AS level topology is used to send and receive the majority of the traffic on the Internet.

The author of the research paper “On the Impact of DoS Attacks on Internet Traffic Characteristics and QoS” [9], primarily focused on the importance of service guarantee in networks. The author then explores the various DoS attacks, which are difficult to fight, and could easily disrupt genuine network traffic. With real time traffic in the picture, Quality of Service or QoS is very important and the author concluded with the fact that the DoS can impact and reduce the QoS offered by networks.

Following the lines of [5] [6], my research would try to determine whether the popular destinations that carry out the majority of the traffic are routed faster than the other traffic and also determine whether a router arranges its routing table lookup in the order of most used prefixes.

My research also will look at the security implications of the way routers handle the routing table. As mentioned by paper [9], DoS attacks can degrade network performance and are usually hard to mitigate.

My research will try to establish whether it is possible degrade network performance by generating legitimate network traffic (similar to some DoS attacks) directed towards not so
frequently used prefixes of the routing table.
6. Methodology and Approach

This research is quantitative in nature. The research results and conclusions are based on data collected from the lab-based setup and often compared to real-world data. Mentioned below are the main steps that I followed during this research -

1. Reading and understanding related research work so as to reduce redundant data and improve efficiency of the research.

2. Implementing pseudo Internet in the lab with 2-3 routers, comparing the setup to popular Route Servers [10], and working toward making the setup comparable to core Internet routers.

3. Recording baselines of various network performance parameters once the setup was complete. Beyond this the routers were overloaded in incremental steps and network performance parameters recorded at each increment.

4. Once the basic of route processing were understood, traffic was generated (script generated) making a clear demarcation between frequently used prefixes and infrequently used prefixes.

5. Based on the above results, the feasibility of a possible exploitation will be assessed and carried out if possible.

6. Results from all the tests would be put in the report and presented.

Note: The entire research setup and data collection would involve numerous scripts as the configuration of routers could reach up to almost 400,000 lines. Traffic generation and data collection also would be done using the help of scripts due to the sheer volume of data and performance parameters. Tools similar to Iperf [12] which is used to generate traffic and also measure various traffic parameters also would be used to gather data.
7. **Experiments**

This research was completed in two phases. The first phase is directed toward studying the effects of loading routers with large amounts of routing information. The focus of the second phase is to understand how routers process large amounts of routing information and whether it is possible to exploit this behavior of routers. Each of the phases in turn has two distinct case studies based on the differences in setup. The first case study deals with experiments using only static routes to load the routers. The second case study deals with experiments using BGP as a routing protocol to load the routers.

<table>
<thead>
<tr>
<th>Phase1CaseStudy1</th>
<th>This experiment deals with studying the router performance while incrementally (20000 route increments) loading it with large amount of static routes (60K–400K).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase1CaseStudy2</td>
<td>This experiment deals with studying the router performance while incrementally (20000 route increments) loading it with large amount of BGP routes (60K–200K).</td>
</tr>
<tr>
<td>Phase2CaseStudy1</td>
<td>This experiment deals in determining how routers perform route lookup on largest static routing tables (400K routes).</td>
</tr>
<tr>
<td>Phase2CaseStudy2</td>
<td>This experiment deals in determining how routers perform route lookup on large BGP routing tables (200K routes).</td>
</tr>
</tbody>
</table>

7.1 **Phase 1 Case Study 1**

This case study deals with studying the effect of loading routers with large number of static routes.
7.1.1 Equipment

3 Cisco2811RoutersrunningIOS12.4(AdvancedIPservices)

1 Cisco2950 switch,

2 TestPCs

7.1.2 Network Diagram

Figure 2

7.1.3 Setup Explanation

As seen in Figure 2, the three routers are connected in a daisy chain. Router R1 has the test computer PC1 connected to it through VLAN1 1 on the switch S1; router R3 has the test
computer PC2 connected to it through VLAN1 3 on the switch S1. The router R2 is the central router in this experiment and as seen in the diagram both R1 and R3 are connected to R2 directly. R2 is the router that is incrementally loaded with static routes. Routers R1 and R3 have simple configurations with static default routes pointing toward R2 to keep the route processing on R1 and R3 as simple as possible. PC1 is the computer that originates all the tests. PC2 is the target for the IPerf based test.

7.1.4 Procedure

1. Once the basic network connectivity was complete, the respective devices were configured with appropriate IP addresses and connectivity from PC1 to PC2 was verified. Routers R1 and R3 had default routes pointing toward R2 to simplify route processing on them. This configuration is called the base configuration for the devices where all the experiments would start.

2. At this point a Perl script, Appendix – 1, was used to generate a configuration file for router R2, which at the first instance of the experiment contained 600000 static routes pointing toward router R3, as shown below:

```
ip route 193.196.189.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 193.196.190.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 193.196.191.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 193.196.192.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 193.196.193.0 255.255.255.0 FastEthernet0/0 192.168.1.10
```

Since the routers had only 256Kbytes of non-volatile configuration memory, the generated configurations file which had a size 3.78 Mbytes, had to be copied to the running memory, which was 256 Mbytes. TFTP was used to copy the configuration file to the running configuration memory using the following command,
copy tftprunning-config

3. Once the file is copied to the running memory, the routers take a significant amount of time to process this configuration file and actually use this. In this experiment I loaded the routers with static routes starting from 60000 through 400000, in 20000 route increments. The time to process the configuration file at 60000 routes was about 5 minutes. This value increased as the number of routes increased and while loading the router with 400000, the router required a whooping 18-1/2 hours to process the file that was copied into the running memory. During this period when the router slowly churns through the configuration file, the router is almost non responsive and even the console connection to the router becomes non interactive.

Note: The upper limit of this experiment was chosen as 400000 due to the time required for each iteration and since this number was significantly higher than the current internet routing table size.

4. Once router R2 has processed the entire configuration and reaches a stable state to pass traffic; router R3 is configured to have 5 loopback interfaces with IP addresses randomly selected from the static routes configured on router R2. These 5 IP addresses become the destinations for which data is collected. The computer PC2 is also used to collect data.

5. Once data is collected, router R2 is reset to base configuration and the above mentioned steps are repeated with 20000 additional static routes maxing out at 400000 routes.
7.1.5 DATA COLLECTION

The data collection approach used in this thesis is to collect useful numerical data from the experimental network that would be understandable and easily relatable to a router's routing table load. The following parameters were used to reach the conclusion of this research.

**ROUTER MEMORY**

This is a parameter used to measure the amount of Random Access Memory or Running configuration memory that a router uses to process and maintain routing tables. This information is direct indication about the increasing load on the router.

This data was collected each time the router R2, finished processing the configuration that was copied to the running configuration memory. The following command was used to measure the memory used by the routing table,

```
R2#shiproutesummary
IProutingtablenameisDefault-IP-Routing-Table[0]
IProutingtablemaximum-pathsis16
RouteSource Networks Subnets Overhead Memory[bytes]
connected 2 0 144 272
static 105199 235177 24507072 46291136
internal 120 138720
Total 105321 235177 24507216 46430128
R2#
```

**PACKET PROCESSING TIME**

This is a parameter used to measure the packet processing time at the router which is being loaded with the routing information. One of the most important parameters in this experiment as this routing load directly affected processing time at the routers. In this case, processing time at router R2 was measured.
The processing time was measured using TRACEROUTE as shown below.

```
silvers@silvers-desktop:~ $ sudo traceroute -l -n 10.0.0.5
[sudo] password for silvers:
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 40 byte packets
1  192.168.200.1  1.081 ms 0.972 ms 1.264 ms
2  192.168.100.1  166.722 ms 166.735 ms 166.832 ms
3  10.0.0.5  1.568 ms 1.699 ms 1.683 ms
```

**JITTER**

Jitter is an important network parameter that is often used to measure the variation of packet latency across networks. Increasing routing load on routers would directly impact the jitter and hence this was another important parameter for this research.

End to end jitter was measured between PC1 and PC2 using a tool called IPerf as shown below.

```
Command on PC2
iperf -s -u -i 1

Command on PC1
iperf -c A.B.C.D -u -b 10m

Actual report from PC1

silvers@silvers-desktop:~/Desktop$ iperf -c 10.0.99.2 -u -b 10m

Client connecting to 10.0.99.2, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 109 KByte (default)

[ 3] local 192.168.200.50 port 52955 connected with 10.0.99.2 port 5001  jitter value
[ 3] 0.0-10.0 sec  11.9 MBytes  10.0 Mbits/sec
[ 3] Sent 8505 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec  28.7 KBytes  23.5 Kbits/sec  31.432 ms 8485/8504 (1e+02%)
```
7.1.6 DATA SAMPLE

Figure 3

Table: Router Memory vs. 5 Random Destinations IP addresses with corresponding processing times (ms)

<table>
<thead>
<tr>
<th>Route Count</th>
<th>Route 16.0.0.5</th>
<th>Route 16.0.76.186</th>
<th>Route 16.0.141.25</th>
<th>Route 16.0.234.116</th>
<th>Route 16.0.235.73</th>
<th>Route 16.0.99.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>80000</td>
<td>0.832</td>
<td>0.7405</td>
<td>51.896</td>
<td>0.795</td>
<td>0.7965</td>
<td>2.164</td>
</tr>
<tr>
<td>7.68Mbytes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3 is an example of the raw data collected. 5 instances of packet processing is selected to rule out any discrepancies and are averaged when plotting graphs. All the times measured are in milliseconds and the routing memory is represented in Megabytes.
7.2 PHASE 1 CASE STUDY 2

This case study deals with studying the effect of loading routers with large number of BGP routes.

7.2.1 EQUIPMENT

3 Cisco 2811 Routers running IOS 12.4 (Advanced IP services)

1 Cisco 2950 switch

2 Test PCs

7.2.2 NETWORK DIAGRAM

Figure 4
7.2.3 Setup Explanation

As seen in Figure 4, the three routers are connected in a daisy chain. Router R1 has the test computer PC1 connected to it through VLAN1 1 on the switch S1; router R3 has the test computer PC2 connected to it through VLAN1 3 on the switch S1. Router R2 is the central router in this experiment and as seen in the diagram both R1 and R3 are connected to R2 directly. R3 is the router that is incrementally loaded with static routes. eBGP neighborship between R2 (AS 200) and R3 (AS 300) populates the routing table of R2 with BGP routes. PC1 is the computer that originates all the tests. PC2 is the target for the IPerf based test.

7.2.4 Procedure

1. Once the basic network connectivity is complete, the respective devices are configured with appropriate IP addresses and connectivity from PC1 to PC2 is verified. Routers R1 and R3 have default routes pointing toward R2 to simplify route processing on them. This configuration is called the base configuration for the devices where all the experiments would start.

2. At this point a Perl script Appendix 1 is used to generate a configuration file for router R3, which at the first instance of the experiment contained 60000 static routes. Since the routers have only 256Kbytes of non volatile configuration memory, the generated configurations file which has a size 3.78 Mbytes has to be copied to the running memory which was 256 Mbytes. TFTP is used to copy the configuration file to the running configuration memory using the following command.
copy tftp running-config

3. Once the router R3 has processed the entire configuration and reaches a stable state; eBGP is enabled on R2 and R3. This populates the routing table on R2 with large amount of BGP routes. Once router R2 has processed the entire configuration and reaches a stable state to pass traffic, router R3 is configured to have 5 loopback interfaces with IP addresses from randomly selected from the BGP routing table on router R2. These 5 IP addresses become the destinations for which data is collected. Computer PC2 is also used to collect data.

Note: The upper limit of this experiment is 200000 BGP routes due to lack of memory on the routers.

4. Once data is collected, routers R2 and R3 are reset to base configuration and the above mentioned steps are repeated with 200000 additional static routes.

7.2.5 DATA COLLECTION

Data collection in this case study is similar to the one used in case study 1. Data for the following three parameters was collected in exactly the same manner:

1. >Router Memory
2. >Processing Time
3. >Jitter
PHASE 2

Phase 2 of this research is focused on determining how the routers process large amount of routing information. This understanding determines if it is possible to exploit routers by making them lookup routes that are not used as frequently. Similar to Phase 1 of this experiment, there are 2 cases in this phase as well. Case 1 deals with static routing and Case 2 deals with BGP.

7.3 PHASE 2 CASE STUDY 1

7.3.1 Equipment

2 Cisco 2811 Routers running IOS 12.4 (Advanced IP services)

1 Test PC

7.3.2 Network Diagram

Figure 5
### 7.3.3 Setup Explanation

As seen in Figure 5, the two routers are connected back to back. Router R2 has the test computer PC1 connected to it directly. Router R2 is the central router in this experiment and is loaded with routing information. Router R3 has a simple configuration with static default routes pointing toward R2 to keep the route processing on R3 as simple as possible. PC1 is the computer that originates all the tests.

### 7.3.4 Procedure

1. Once the basic network connectivity is complete, the respective devices are configured with appropriate IP addresses, and basic connectivity from PC1 is verified. Router R3 has default routes pointing toward R2 to simplify route processing on it. This configuration is called the base configuration for the devices where all the experiments would start.

2. At this point the configuration file generated during Phase 1 containing 400000 static routes is copied to router R2’s running configuration memory.

3. Once router R2 has processed the entire configuration and reaches a stable state; the complete routing table from router R2 is captured on a text file as shown below,

```
R2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
   D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
   E1 - OSPF external type 1, E2 - OSPF external type 2
   i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
   ia - IS-IS inter area, * - candidate default, U - per-user static route
   o - ODR, P - periodic downloaded static route
Gateway of last resort is not set

S  200.102.174.0/24 [1/0] via 192.168.1.10, FastEthernet0/0
S  200.85.157.0/24 [1/0] via 192.168.1.10, FastEthernet0/0
```
With the entire routing table (400000 routes) on a text file, I selected 50 IP addresses, every 8000 (400000/50) routes from the routing table. These equally spaced IP addresses would form the test samples to determine the processing time of different routes positioned at different locations of the routing table.

4. Router R3 is configured to have 50 loopback interfaces with IP addresses previously selected from the routing table of R2.

5. Data for all 50 IP addresses is collected from PC1, which directly connects to R2.

7.3.5 DATA COLLECTION

Since Phase 2 of this research is focused on determining the route lookup processing of routers, the only parameter required is PACKET PROCESSING TIME. This is a parameter used to measure the packet processing time at the router that is being loaded with the routing information. In this case, processing time at router R2 was measured using TRACEROUTE as shown below:

```
silvers@silvers-desktop:~$ sudo traceroute -I -n 93.188.0.245
traceroute to 93.188.0.245 (93.188.0.245), 30 hops max, 40 byte packets
  1 192.168.100.1 8.548 ms 8.252 ms 8.152 ms
  2 93.188.0.245  8.067 ms  7.988 ms  7.908 ms
```

Hop at router R2
7.4 PHASE 2 CASE STUDY 2

This case is similar to previous case with the only difference being the routing protocol used to load the experimental router. BGP is used instead of static routes to load the routers.

7.4.1 EQUIPMENT

2 Cisco 2811 Routers running IOS 12.4 (Advanced IP services)

1 Test PC

7.4.2 NETWORK DIAGRAM

FIGURE 6

[Diagram showing the network setup for Phase 2 Case Study 2]
7.4.3 Setup Explanation

As seen in Figure 6, the two routers are connected back to back. Router R2 has the test computer PC1 connected to it directly. In order to load router R2 with BGP routes, R3 is first loaded with static routes. Once router R3 is stable, eBGP neighborship is established between R2 (AS 200) and R3 (AS 300). This eBGP neighborship populates the routing table of router R2. The number of BGP routes on R2 is proportional to the number of static routes on R3 and can be controlled by altering the number of static routes on R3.

7.4.4 Procedure

1. Once the basic network connectivity is complete, the respective devices are configured with appropriate IP addresses, and basic connectivity from PC1 is verified. Router R3 has default routes pointing toward R2 to simplify route processing. This configuration is called the base configuration for the devices where all the experiments would start.

2. At this point the configuration file generated during Phase 1 containing 200,000 static routes is copied to the router R3 running configuration memory.

3. Once router R3 has processed the entire configuration and reaches a stable state, eBGP is enabled on R2 and R3. This populates the routing table on R2 with a large amount of BGP. This routing table is captured to a text file using the command shown below,

```
R2#sh ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
    D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
    N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
    E1 - OSPF external type 1, E2 - OSPF external type 2
    i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
    ia - IS-IS inter area, * - candidate default, U - per-user static route
    o - ODR, P - periodic downloaded static route
```
4. With the entire routing table (200000 routes) on a text file, I selected 50 IP addresses, every 4000 (200000/50) routes from routing table. These equally spaced IP addresses would form the test samples to determine the processing time of different routes positioned at different locations of the routing table.

5. Router R3 is configured to have 50 loopback interfaces with IP addresses previously selected from the routing table of R2.

6. Data for all the 50 IP addresses is collected from PC1 which directly connected to R2.

### 7.4.5 DATA COLLECTION

Similar to Phase 2 Case Study 1, the only parameter which was recorded in the case study was PACKET PROCESSING TIME.
8. RESULTS

This section evaluates the results gathered during the experiments explained in the Experiment section. Each case study result is followed by a detailed observations and inferences section. The observations and inferences section after each section makes it easier to correlate to the results.

8.1 PHASE 1 CASE STUDY 1

The focus of this case study was to determine the effects of loading router with large amount of static routes. The overall trend of the data collected during this experiment partially validated the hypothesis.

The router memory, packet processing time and jitter values gradually increased with the increase in routing information load. The following table contains the raw data collected at each of the routing increments.

Table1: StaticRouteIncrementdata

<table>
<thead>
<tr>
<th>Route Count</th>
<th>Packet</th>
<th>Processing Time</th>
<th>Jitter</th>
</tr>
</thead>
<tbody>
<tr>
<td>60000</td>
<td>1.026</td>
<td>2.625</td>
<td>0.809</td>
</tr>
<tr>
<td>8.1GBytes</td>
<td>1.026</td>
<td>2.625</td>
<td>0.809</td>
</tr>
<tr>
<td>100000</td>
<td>0.678</td>
<td>0.735</td>
<td>0.714</td>
</tr>
<tr>
<td>10.8Mbytes</td>
<td>0.678</td>
<td>0.735</td>
<td>0.714</td>
</tr>
<tr>
<td>13.5GBytes</td>
<td>1.284</td>
<td>0.745</td>
<td>3.112</td>
</tr>
<tr>
<td>120000</td>
<td>1.284</td>
<td>0.745</td>
<td>3.112</td>
</tr>
<tr>
<td>16.3Mbytes</td>
<td>1.284</td>
<td>0.745</td>
<td>3.112</td>
</tr>
<tr>
<td>140000</td>
<td>0.67</td>
<td>0.649</td>
<td>0.598</td>
</tr>
<tr>
<td>19.06 MBytes</td>
<td>0.67</td>
<td>0.649</td>
<td>0.598</td>
</tr>
</tbody>
</table>
Table 1 shows the raw data collected during this experiment. This experiment was started at 60000 routes and ended at 400000. The upper limit of this experiment was selected due to the very high churn time of the routers to process this huge routing table into memory. This value also was selected as this was significantly above the current Internet routing table size.

The data from the above table has been transformed into graphs to provide better understanding.
8.1.1 OBSERVATIONS & INFERENCES

Figure 7 is a clear representation of the increasing memory usage with the increase in routing table size. This parameter displayed behavior as expected at the start of the experiment.
8.1.2 OBSERVATIONS AND INFERENCES

Figure 8 shows the increasing trend of packet processing time with the increase in routing table size. At the start of the experiment, I had expected a much more linear increase in the packet processing times. As seen in the graph, though the overall trend of packet processing time increases with the increase in routing table size, there are a few unexplained spikes in the packet processing time.
8.1.3 OBSERVATIONS AND INFERENCES

Figure 9 compares jitter to the increasing routing table size. The graph maintains a steady level until the routing table size reaches 340,000. At this point the router runs out of memory to incorporate the entire routing table into the Forwarding Information Base (FIB) table.
The FIB table is used by the Cisco Express Forwarding (CEF) [11] mechanism used by Cisco routers to speed up packet forwarding without having to perform traditional lookups and tax the CPU. The following is the message from the router clearly indicating CEF failure at about 340,000 routes,

```
R2#sh ip route summary
IP routing table name is Default-IP-Routing-Table(0)
IP routing table maximum-paths is 16
Route Source Networks Subnets Overhead Memory (bytes)
connected 2 0 144 272
static 105199 235177 24507072 46291136
internal 120 138720
Total 105321 235177 24507216 46430128
R2#
*May 13 01:20:55.661: %SYS-2-MALLOCFAIL: Memory allocation of 65536 bytes failed from 0x400B9194, alignment 16
Pool: Processor  Free: 5413320  Cause: Memory fragmentation
Alternate Pool: None  Free: 0  Cause: No Alternate pool

-Process= "IP RIB Update", ipl= 0, pid= 92
-Traceback= 0x40AA6570 0x4009AB98 0x4009FC00 0x400B919C 0x4010F9F8 0x40C6C564 0x40C72E18 0x40C7A7DC 0x40C91470 0x410FBBDC 0x410CC888 0x410CC968
*May 13 01:20:55.661: %FIB-3-NOMEM: Malloc Failure, disabling CEF
-Traceback= 0x40AA6570 0x40C6C5B0 0x40C72E18 0x40C7A7DC 0x40C91470 0x410FBBDC 0x410CC888 0x410CC968
```

The above error indicates Memory Allocation Failure (MALLOCFAIL) forcing the router to turn off CEF globally.

With the failure of CEF, the jitter value skyrocketed from about 2ms to around 35ms. This in turn meant that packet delay variation had increased significantly now that the router had to go through the actual routing table before forwarding packets.
8.2 PHASE 1 CASE STUDY 2

The focus of this case study was to determine the effects of loading the router with large amount of BGP routes. The result from this case study partially validates the hypothesis. The following table contains the raw data collected at each of the routing increments:

Table2

<table>
<thead>
<tr>
<th>Route Count</th>
<th>Packets</th>
<th>Processing Time</th>
<th>After</th>
</tr>
</thead>
<tbody>
<tr>
<td>60000</td>
<td>0.832</td>
<td>0.7405</td>
<td>51.656</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.795</td>
<td>0.7965</td>
</tr>
<tr>
<td>7.68Mbytes</td>
<td></td>
<td></td>
<td>2.164</td>
</tr>
<tr>
<td>80000</td>
<td>10.0.0.5</td>
<td>10.0.76.186</td>
<td>10.0.134.116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.0.354.116</td>
<td>10.0.354.116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.0.354.116</td>
<td>10.0.354.116</td>
</tr>
<tr>
<td></td>
<td>10.0.5</td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td>10.25Mbytes</td>
<td></td>
<td></td>
<td>3.256</td>
</tr>
<tr>
<td>100000</td>
<td>10.0.0.5</td>
<td>10.0.76.186</td>
<td>10.0.354.116</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>2.05</td>
</tr>
<tr>
<td>12.12Mbytes</td>
<td></td>
<td></td>
<td>1.86</td>
</tr>
<tr>
<td>150000</td>
<td>10.0.5</td>
<td>10.0.76.186</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td>17.95Mbytes</td>
<td></td>
<td></td>
<td>1.86</td>
</tr>
<tr>
<td>200000</td>
<td>10.0.5</td>
<td>10.0.76.186</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td>20.48Mbytes</td>
<td></td>
<td></td>
<td>1.86</td>
</tr>
<tr>
<td>200000</td>
<td>10.0.5</td>
<td>10.0.76.186</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td>23.04Mbytes</td>
<td></td>
<td></td>
<td>1.86</td>
</tr>
<tr>
<td>200000</td>
<td>10.0.5</td>
<td>10.0.76.186</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.1.137.35</td>
<td>10.0.99.2</td>
</tr>
</tbody>
</table>

Table2 shows the raw data collected during this experiment. This experiment was started at 60000 routes and ended at 200000. The upper limit of this experiment was selected due to lack of memory on the router to be able to handle such a huge BGP routing table. The router would run out of memory and shut down the BGP process at higher route counts and would cause the router to crash in a few occasions. The following outputs from the router shows the router resetting and
deleting BGP neighborship,

*Jan 31 21:17:02.275: %BGP-3-NOTIFICATION: received from neighbor 192.168.2.10 4/0 (hold time expired) 0 bytes
*Jan 31 21:17:35.583: %BGP-5-ADJCHANGE: neighbor 192.168.2.10 Up
*Jan 31 21:29:29.799: %SYS-2-MALLOCFAIL: Memory allocation of 2000 bytes failed from 0x400AA81C, alignment 0

Pool: Processor Free: 70616 Cause: Memory fragmentation
Alternate Pool: None Free: 0 Cause: No Alternate pool
-Process= "Exec", ipl= 0, pid= 3 -Traceback= 0x41182E94 0x4009A648 0x4009F254 0x4009F750 0x400AA824 0x400AB754 0x40E44044 0x41851ABC 0x4185D94C 0x41872BBC 0x41872DEC 0x41873A50 0x41874A1C 0x411C35C0 0x411DFBF4 0x411DF814
*Jan 31 21:29:50.939: %BGP-5-ADJCHANGE: neighbor 192.168.2.10 Down No memory
*Jan 31 21:34:43.991: %SYS-2-MALLOCFAIL: Memory allocation of 20000 bytes failed from 0x400AA81C, alignment 0

Pool: Processor Free: 68764 Cause: Memory fragmentation
Alternate Pool: None Free: 0 Cause: No Alternate pool
-Process= "Virtual Exec", ipl= 0, pid= 219 -Traceback= 0x41182E94 0x4009A648 0x4009F254 0x4009F750 0x400AA824 0x400AB754 0x418516C0 0x41859168 0x418553E4 0x418555C0 0x41855C78 0x418676BC 0x41867E30 0x418681CC 0x411C35C0 0x411DF6EC
*Jan 31 21:34:45.507: %BGP-5-ADJCHANGE: neighbor 192.168.2.10 Down Neighbor deleted
*Jan 31 21:34:58.255: %SYS-2-CHUNKSIBLINGS: Attempted to destroy chunk with siblings, chunk 461A4610. -Process= "Virtual Exec", ipl= 0, pid= 219 -Traceback= 0x41182E94 0x400AAFD0 0x400AB144 0x40E2898 0x40E294D4 0x418676BC 0x418681CC 0x411C35C0 0x411DF6EC 0x42196BE4 0x42196BC8

*Jan 31 21:35:17.927: %SYS-5-CONFIG_I: Configured from console by vty (192.168.2.10)
473090 bytes copied in 1480.084 secs (3196 bytes/sec)

The following output from the router shows a crash following memory allocations failure,

It also includes a generic message to open a case with Cisco TAC for this issue,
Possible software fault. Upon recurrence, please collect

crashinfo, "show tech" and contact Cisco Technical Support.

Writing crashinfo to flash:crashinfo_20100131-220059

Possible software fault. Upon recurrence, please collect

crashinfo, "show tech" and contact Cisco Technical Support.

-Traceback= 0x4007B928 0x40079FA4 0x411A03F0 0x411A3C80 0x41208B6C 0x411C35C0 0x411DCFB4 0x411DD184 0x411DD2FC 0x411DD3A8 0x4161D3EC 0x41617D10 0x41610A74 0x4161A144 0x4161A2C8 0x4161A30C
$0 : 00000000, AT : 44870000, v0 : 00000000, v1 : 00000000
a0 : 00000000, a1 : 0000FF00, a2 : 00000000, a3 : 42B40000
t0 : 4496E4F0, t1 : 44A60000, t2 : 40080EE8, t3 : FFFF00FF
t4 : 42B38934, t5 : FFFFFFFF, t6 : 00000000, t7 : 00000000
s0 : 00000000, s1 : 00000000, s2 : 44570000, s3 : 44310000
s4 : 00010020, s5 : 44BE0BC0, s6 : 00000149, s7 : 44BE0BC0
t8 : 44A60000, t9 : FFFFFFFF, k0 : 49450364, k1 : 421A5938
gp : 44873980, sp : 459AF530, s8 : 43B90000, ra : 40079FA4
EPC : 4007B928, ErrorEPC : 0478223A, SREG : 3400FF03
MDLO : 00000000, MDHI : 00000002, BadVaddr : C5BF7C9B
DATA_START : 0x42B2FCC0
Cause 00000024 (Code 0x9): Breakpoint exception
DATA_START : 0x42B2FCC0

Cause 00000024 (Code 0x9): Breakpoint exception

-Traceback= 0x4007B928 0x40079FA4 0x411A03F0 0x411A3C80 0x41208B6C 0x411C35C0 0x411DCFB4
  0x411DD184 0x411DD2FC 0x411DD3A8 0x4161D3EC 0x41617D10 0x41610A74 0x4161A144
  0x4161A2C8 0x4161A30C

System Bootstrap, Version 12.4(1r) [hqluong 1r], RELEASE SOFTWARE (fc1)

Copyright (c) 2005 by cisco Systems, Inc.

Initializing memory for ECC

c2811 processor with 262144 Kbytes of main memory

Main memory is configured to 64 bit mode with ECC enabled

Readonly ROMMON initialized

program load complete, entry point: 0x8000f000, size: 0xc940

program load complete, entry point: 0x8000f000, size: 0xc940

The router reloads from the crash with a core dump.

The data collected during this experiment has been represented in the following graphs. The graphs give a better representation of the data collected.
8.2.1 OBSERVATIONS AND INFERENCES

Figure 10 represents the increasing memory utilization with the increase in BGP routes. This increase is almost linear and indicates that the memory utilization of the router is directly proportional to the number of routes. Thus it is essential to make sure that routers have enough memory before they are loaded with large routing tables.


8.2.2 OBSERVATIONS AND INFERENCES

Figure 11 represents packet processing time against BGP route count. During the start of this experiment, I had expected the packet processing time to be fairly constant around the 1 – 2 ms second mark since this was the behavior shown during the static route testing until the 200000 route mark. There were a few unexplained spikes in the packet processing time, but the overall trend is similar to the static route test at similar route counts. The packet processing time at 220000 should have been theoretically infinite as the router was never able to process as many routes. We can infer that the packet processing times are fairly good at route counts around 200000.
8.2.3 OBSERVATIONS AND INFERENCES

Figure 12 represents the jitter against BGP router count. The jitter value can be seen to be fairly constant around 2 – 3 ms until about 180,000 – 200,000. Loading the router with 200,000 BGP routes is only possible by turning CEF (Cisco Express Forwarding) off. Similar to the static route case, once CEF is turned off the jitter value skyrocket to about 30 – 35ms. This in turn means that packet delay variation has increased significantly now that the router has to go through the actual routing table before forwarding packets.
PHASE 1 RESULT COMPARISON (STATIC Vs BGP)

FIGURE 13 a

Route Count Vs Routing Table Memory

FIGURE 13b

Route Count Vs Processing Time
The above three graphs are comparisons of the various parameters studied in Phase 1. Overlaying the results from the static route case study and the BGP case study gives us a clear picture on how the parameters stack against each other at similar route counts.
PHASE 2

The goal of Phase 2 case studies is to analyze whether frequently used routes or routes that comprise the top section of the routing table have faster packet processing time compared to ones that are not used frequently.

8.3 PHASE 2 CASE STUDY 1

The following table represents raw packet processing times for 50 IP address samples selected from the 400,000 routing table entries on router R2. The samples have been selected at regular increments of 8000 routes from each other to obtain packet processing time across the entire routing table.

Table 3

<table>
<thead>
<tr>
<th>Sample IP Addresses</th>
<th>Packet Processing Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>200.102.174.4</td>
<td>5.345</td>
</tr>
<tr>
<td>193.136.193.5</td>
<td>111.643</td>
</tr>
<tr>
<td>50.0.4.58</td>
<td>7.389</td>
</tr>
<tr>
<td>50.0.12.240</td>
<td>9.301</td>
</tr>
<tr>
<td>193.103.63.7</td>
<td>7.035</td>
</tr>
<tr>
<td>193.70.135.89</td>
<td>27.593</td>
</tr>
<tr>
<td>200.86.23.6</td>
<td>15.01</td>
</tr>
<tr>
<td>116.60.0.56</td>
<td>12.987</td>
</tr>
<tr>
<td>193.81.104.8</td>
<td>19.916</td>
</tr>
<tr>
<td>193.57.246.3</td>
<td>10.069</td>
</tr>
<tr>
<td>193.160.212.1</td>
<td>21.514</td>
</tr>
<tr>
<td>175.140.0.23</td>
<td>15.635</td>
</tr>
<tr>
<td>193.1.176.78</td>
<td>4.468</td>
</tr>
<tr>
<td>193.169.161.9</td>
<td>12.007</td>
</tr>
<tr>
<td>200.62.87.29</td>
<td>18.054</td>
</tr>
<tr>
<td>193.50.147.9</td>
<td>13.237</td>
</tr>
<tr>
<td>173.74.4.3</td>
<td>13.108</td>
</tr>
<tr>
<td>128.78.0.231</td>
<td>19.311</td>
</tr>
<tr>
<td>93.188.0.245</td>
<td>122.196</td>
</tr>
<tr>
<td>200.98.73.76</td>
<td>10.434</td>
</tr>
<tr>
<td>111.171.0.89</td>
<td>24.23</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>-------</td>
</tr>
<tr>
<td>57.82.0.43</td>
<td>3.281</td>
</tr>
<tr>
<td>10.0.183.183</td>
<td>3.027</td>
</tr>
<tr>
<td>10.0.180.168</td>
<td>5.55</td>
</tr>
<tr>
<td>10.0.176.152</td>
<td>13.522</td>
</tr>
<tr>
<td>10.0.172.136</td>
<td>2.996</td>
</tr>
<tr>
<td>10.0.168.152</td>
<td>48.405</td>
</tr>
<tr>
<td>10.0.164.232</td>
<td>5.563</td>
</tr>
<tr>
<td>10.0.160.248</td>
<td>18.324</td>
</tr>
<tr>
<td>10.0.156.200</td>
<td>227.021</td>
</tr>
<tr>
<td>10.0.152.248</td>
<td>21.002</td>
</tr>
<tr>
<td>10.0.149.233</td>
<td>18.083</td>
</tr>
<tr>
<td>10.0.142.6</td>
<td>14.768</td>
</tr>
<tr>
<td>10.0.120.252</td>
<td>15.894</td>
</tr>
<tr>
<td>10.0.104.248</td>
<td>7.108</td>
</tr>
<tr>
<td>10.0.81.253</td>
<td>11.843</td>
</tr>
<tr>
<td>10.0.56.128</td>
<td>7.42</td>
</tr>
<tr>
<td>10.0.31.171</td>
<td>8.221</td>
</tr>
<tr>
<td>10.24.217.8</td>
<td>7.419</td>
</tr>
<tr>
<td>10.1.34.254</td>
<td>5.64</td>
</tr>
<tr>
<td>10.0.245.28</td>
<td>79.402</td>
</tr>
<tr>
<td>10.0.196.33</td>
<td>17.498</td>
</tr>
<tr>
<td>10.0.152.105</td>
<td>196.181</td>
</tr>
<tr>
<td>95.248.7.9</td>
<td>7.604</td>
</tr>
<tr>
<td>193.134.113.9</td>
<td>15.28</td>
</tr>
<tr>
<td>183.121.0.2</td>
<td>2.444</td>
</tr>
<tr>
<td>106.172.0.169</td>
<td>9.233</td>
</tr>
<tr>
<td>200.58.24.1</td>
<td>2.787</td>
</tr>
<tr>
<td>193.125.170.5</td>
<td>2.614</td>
</tr>
<tr>
<td>193.2.71.98</td>
<td>4.04</td>
</tr>
</tbody>
</table>

The data from this is represented in a graphical format as shown below,
8.3.1 OBSERVATIONS AND INFERENCES

Figure 14 represents packet processing times for various IP address samples. The majority of the IP address samples have packet processing time around the 20 – 30 ms mark. In addition, there are a few packets that have high packet processing times varying from around 50 – 230ms. The most important observation in this experiment is the positioning of the samples in the routing table. As seen in the graph, the high packet processing times are not located just at the end of the routing
table but are also present at the beginning of the routing table. This is contrary to the assumption at the start of this experiment.

At the start of this experiment, my assumption was that the router would take more time to process packets at the end of the routing table since it would have go through more entries to reach there. The result of this experiment proved that theory wrong, and one possible reason could be that the router does not use the routing table in the sequence that is displayed when using the *show ip route* command.

The second part of this experiment was to determine whether more frequently used routes are processed faster than the ones which are not.

The following are the raw results for some of the samples,

<table>
<thead>
<tr>
<th>IP 10.0.152.105</th>
</tr>
</thead>
</table>
| silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.152.105  
traceroute to 10.0.152.105 (10.0.152.105), 30 hops max, 40 byte packets  
1  192.168.100.1  3.988 ms  3.740 ms  3.641 ms  
2  10.0.152.105  3.557 ms  3.481 ms  3.404 ms  |
| silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.152.105  
traceroute to 10.0.152.105 (10.0.152.105), 30 hops max, 40 byte packets  
1  192.168.100.1  1.823 ms  13.967 ms  13.876 ms  
2  10.0.152.105  13.793 ms  13.716 ms  13.638 ms  |
| silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.152.105  
traceroute to 10.0.152.105 (10.0.152.105), 30 hops max, 40 byte packets  
1  192.168.100.1  1.337 ms  5.368 ms  4.729 ms  
2  10.0.152.105  4.621 ms  4.541 ms  3.941 ms  |
| silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.152.105  
traceroute to 10.0.152.105 (10.0.152.105), 30 hops max, 40 byte packets  
1  192.168.100.1  3.543 ms  2.881 ms  2.786 ms  
2  10.0.152.105  2.700 ms  2.623 ms  3.294 ms  |
IP 10.0.196.33

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.196.33
traceroute to 10.0.196.33 (10.0.196.33), 30 hops max, 40 byte packets
1 192.168.100.1  7.260 ms  7.003 ms  6.916 ms
2 10.0.196.33  6.833 ms  6.756 ms  6.677 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.196.33
traceroute to 10.0.196.33 (10.0.196.33), 30 hops max, 40 byte packets
1 192.168.100.1  16.587 ms  16.318 ms  16.230 ms
2 10.0.196.33  16.132 ms  16.051 ms  15.974 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.196.33
traceroute to 10.0.196.33 (10.0.196.33), 30 hops max, 40 byte packets
1 192.168.100.1  4.819 ms  4.560 ms  4.473 ms
2 10.0.196.33  4.389 ms  4.295 ms  4.216 ms

8.3.2 OBSERVATIONS AND INFERENCES

The above outputs are the result of generating traffic for a particular destination in quick
succession. Successive trace routes with less than a second between each try provided us with the
above results.

Trying to trace to a particular IP address makes the router perform lookups for the particular
destination. As seen in the output, we can see that the packet processing times at first hop, which is
the router loaded with 400000 routes. The result of this experiment proves my original hypothesis
is wrong. According my hypothesis, the router should have taken more time to look up a destination for the first time and then use lesser time on successive tries. The results above are contrary to this notion. I observed much greater packet processing times for some instances. In one case there is an 800% increase in the packet processing time for the successive test. This behavior remains unexplained as the only traffic flowing through the router is the experimental traffic, which means the load on the router is constant throughout the various trace outputs. It is impossible to predict the packet processing time based on the usage of particular routes.

8.4 PHASE 2 CASE STUDY 2

This case study is similar to the previous case study with the only difference being that BGP is used rather than static routes to load the router.

Table 4

<table>
<thead>
<tr>
<th>Sample IP addresses</th>
<th>Packet Processing Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>193.187.122.8</td>
<td>3.035</td>
</tr>
<tr>
<td>193.166.89.7</td>
<td>6.074</td>
</tr>
<tr>
<td>193.116.14.6</td>
<td>2.331</td>
</tr>
<tr>
<td>193.43.9.4</td>
<td>9.943</td>
</tr>
<tr>
<td>193.15.93.9</td>
<td>4.328</td>
</tr>
<tr>
<td>172.3.121.1</td>
<td>28.109</td>
</tr>
<tr>
<td>172.19.46.5</td>
<td>2.663</td>
</tr>
<tr>
<td>193.157.245.4</td>
<td>6.862</td>
</tr>
<tr>
<td>193.54.141.9</td>
<td>14.519</td>
</tr>
<tr>
<td>193.24.97.6</td>
<td>4.652</td>
</tr>
<tr>
<td>10.1.62.54</td>
<td>2.983</td>
</tr>
<tr>
<td>10.2.120.122</td>
<td>16.501</td>
</tr>
<tr>
<td>10.2.239.246</td>
<td>11.743</td>
</tr>
<tr>
<td>10.0.80.77</td>
<td>2.972</td>
</tr>
<tr>
<td>10.0.195.215</td>
<td>3.797</td>
</tr>
<tr>
<td>10.1.57.23</td>
<td>6.001</td>
</tr>
<tr>
<td>10.2.114.86</td>
<td>65.358</td>
</tr>
<tr>
<td>10.2.231.220</td>
<td>8.752</td>
</tr>
<tr>
<td>10.0.72.123</td>
<td>6.897</td>
</tr>
<tr>
<td>10.0.187.241</td>
<td>7.453</td>
</tr>
<tr>
<td>10.1.41.101</td>
<td>8.113</td>
</tr>
<tr>
<td>10.2.92.26</td>
<td>7.94</td>
</tr>
<tr>
<td>10.2.202.151</td>
<td>9.615</td>
</tr>
<tr>
<td>10.0.36.117</td>
<td>2.963</td>
</tr>
<tr>
<td>10.0.144.248</td>
<td>3.339</td>
</tr>
<tr>
<td>10.0.253.158</td>
<td>2.751</td>
</tr>
<tr>
<td>10.1.107.16</td>
<td>4.784</td>
</tr>
<tr>
<td>10.2.153.230</td>
<td>9.911</td>
</tr>
<tr>
<td>10.19.103.6</td>
<td>5.189</td>
</tr>
<tr>
<td>10.0.102.232</td>
<td>2.267</td>
</tr>
<tr>
<td>10.0.215.86</td>
<td>3.681</td>
</tr>
<tr>
<td>10.1.73.208</td>
<td>5.917</td>
</tr>
<tr>
<td>10.2.125.236</td>
<td>12.789</td>
</tr>
<tr>
<td>10.2.233.65</td>
<td>5.838</td>
</tr>
<tr>
<td>10.0.67.239</td>
<td>11.311</td>
</tr>
<tr>
<td>10.0.170.13</td>
<td>175.074</td>
</tr>
<tr>
<td>10.1.22.169</td>
<td>2.436</td>
</tr>
<tr>
<td>10.1.130.50</td>
<td>7.543</td>
</tr>
<tr>
<td>10.2.178.120</td>
<td>18.135</td>
</tr>
<tr>
<td>10.0.11.201</td>
<td>10.554</td>
</tr>
<tr>
<td>10.0.120.189</td>
<td>3.688</td>
</tr>
<tr>
<td>10.0.229.57</td>
<td>2.547</td>
</tr>
<tr>
<td>10.1.84.130</td>
<td>6.436</td>
</tr>
<tr>
<td>10.2.125.145</td>
<td>9.02</td>
</tr>
<tr>
<td>10.2.223.60</td>
<td>2.233</td>
</tr>
<tr>
<td>10.0.48.203</td>
<td>77.95</td>
</tr>
<tr>
<td>10.0.149.103</td>
<td>9.34</td>
</tr>
<tr>
<td>10.0.246.3</td>
<td>2.253</td>
</tr>
<tr>
<td>193.171.96.6</td>
<td>7.463</td>
</tr>
<tr>
<td>193.56.220.5</td>
<td>9.283</td>
</tr>
</tbody>
</table>
Table 4 shows the raw data collected from this experiment using BGP. The data from the table is represented on the graph below.

**FIGURE 15**

**Packet Processing time (ms)**

8.4.1 OBSERVATIONS AND INFERENCES

Figure 15 is the graphical representation of packet processing times against the 50 samples from the 200000 route BGP routing table. Each sample is collected at 4000 route increment from the previous sample. The majority of the samples have fairly low packet processing times <10ms. Similar to the static route testing, this experiment also produced some lookup times that where
unexplained. The most important observation in this experiment is the positioning of the samples in the routing table. As seen in the graph, the high packet processing times are not located just at the end of the routing table but are also present at the beginning of the routing table. This is contrary to the assumption at the start of this experiment.

At the start of this experiment my assumption was that the router would take more time to process packets at the end of the routing table since it would have go through more entries to reach there. The result of this experiment proves that theory wrong and one possible reason could be that the router does not use the routing table in the sequence which is displayed when using the `show ip route` command.

The second part of this experiment was to determine whether more frequently used BGP routes are processed faster than the ones which are not.

The following are the raw results for some of the samples,

```
<table>
<thead>
<tr>
<th>IP: 193.136.193.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>silvers@silvers-desktop:~$ sudo traceroute -I -n 193.136.193.5</td>
</tr>
<tr>
<td>traceroute to 193.136.193.5 (193.136.193.5), 30 hops max, 40 byte packets</td>
</tr>
<tr>
<td>1 192.168.100.1 12.220 ms 11.980 ms 11.890 ms</td>
</tr>
<tr>
<td>2 193.136.193.5 11.804 ms 11.725 ms 11.643 ms</td>
</tr>
<tr>
<td>silvers@silvers-desktop:~$ sudo traceroute -I -n 193.136.193.5</td>
</tr>
<tr>
<td>traceroute to 193.136.193.5 (193.136.193.5), 30 hops max, 40 byte packets</td>
</tr>
<tr>
<td>1 192.168.100.1 12.807 ms 12.518 ms 12.418 ms</td>
</tr>
<tr>
<td>2 193.136.193.5 12.333 ms 12.256 ms 12.179 ms</td>
</tr>
<tr>
<td>silvers@silvers-desktop:~$ sudo traceroute -I -n 193.136.193.5</td>
</tr>
<tr>
<td>traceroute to 193.136.193.5 (193.136.193.5), 30 hops max, 40 byte packets</td>
</tr>
<tr>
<td>1 192.168.100.1 5.253 ms 4.996 ms 4.910 ms</td>
</tr>
<tr>
<td>2 193.136.193.5 4.827 ms 4.749 ms 4.671 ms</td>
</tr>
<tr>
<td>silvers@silvers-desktop:~$ sudo traceroute -I -n 193.136.193.5</td>
</tr>
<tr>
<td>traceroute to 193.136.193.5 (193.136.193.5), 30 hops max, 40 byte packets</td>
</tr>
<tr>
<td>1 192.168.100.1 3.007 ms 2.688 ms 2.596 ms</td>
</tr>
<tr>
<td>2 193.136.193.5 2.514 ms 2.436 ms 2.359 ms</td>
</tr>
</tbody>
</table>
|```
silvers@silvers-desktop:~$ sudo traceroute -l -n 192.168.100.1
traceroute to 192.168.100.1 (192.168.100.1), 30 hops max, 40 byte packets
1 192.168.100.1 2.388 ms 2.107 ms 1.991 ms
2 193.163.193.5 1.904 ms 14.267 ms 14.189 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 193.163.193.5
traceroute to 193.163.193.5 (193.163.193.5), 30 hops max, 40 byte packets
1 192.168.100.1 106.235 ms 106.640 ms 106.542 ms
2 193.163.193.5 21.284 ms 21.207 ms 21.129 ms

IP: 10.0.172.152

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.172.152
traceroute to 10.0.172.152 (10.0.172.152), 30 hops max, 40 byte packets
1 192.168.100.1 13.522 ms 13.250 ms 13.146 ms
2 10.0.176.152 13.061 ms 12.982 ms 13.105 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.172.136
traceroute to 10.0.172.136 (10.0.172.136), 30 hops max, 40 byte packets
1 192.168.100.1 1.348 ms 2.996 ms 2.780 ms
2 10.0.172.136 13.126 ms 13.047 ms 12.971 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.168.152
traceroute to 10.0.168.152 (10.0.168.152), 30 hops max, 40 byte packets
1 192.168.100.1 47.841 ms 48.405 ms 48.298 ms
2 10.0.168.152 50.172 ms 51.010 ms 51.757 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.164.232
traceroute to 10.0.164.232 (10.0.164.232), 30 hops max, 40 byte packets
1 192.168.100.1 5.563 ms 5.293 ms 5.198 ms
2 10.0.164.232 5.113 ms 13.078 ms 13.001 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.160.248
traceroute to 10.0.160.248 (10.0.160.248), 30 hops max, 40 byte packets
1 192.168.100.1 18.324 ms 17.992 ms 17.894 ms
2 10.0.160.248 17.801 ms 17.717 ms 17.639 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.156.200
traceroute to 10.0.156.200 (10.0.156.200), 30 hops max, 40 byte packets
1 192.168.100.1 203.235 ms 202.932 ms 227.021 ms
2 10.0.156.200 226.938 ms 226.869 ms 226.745 ms

silvers@silvers-desktop:~$ sudo traceroute -l -n 10.0.156.200
traceroute to 10.0.156.200 (10.0.156.200), 30 hops max, 40 byte packets
1 192.168.100.1 19.248 ms 18.719 ms 18.628 ms
2 10.0.156.200 18.543 ms 18.464 ms 18.386 ms
8.4.2 OBSERVATIONS AND INFERENCES

The above outputs are the result of generating traffic for a particular destination in quick succession. Successive trace routes with less than a second between each try provide us with the above results.

Trying to trace a particular IP address makes the router perform lookups for the particular destination. As seen in the output, one can see that the packet processing times at first hop, which is the router loaded with 200000 BGP routes. The result of this experiment proves the original hypothesis of the study to be wrong. According my hypothesis, the router should have taken a greater time to look up a destination for the first time and then use lesser time on successive tries. The results above are to the contrary. I observe much greater packet processing times for some instances. In one case there is an increase in the packet processing time that is 40 times the previous test. This behavior remains unexplained as the only traffic flowing through the router is the experimental traffic, which means the load on the router is constant throughout the various trace outputs. It is impossible to predict the packet processing time based on the usage of particular routes.
PHASE 2 RESULT COMPARISON (STATIC Vs BGP)

The above comparative graph for the first 25 samples from the static routing table and BGP routing table used in Phase 2 of this experiment reiterates the randomness of packet processing times. In neither of the cases were packet processing times for earlier routes faster than routes that were at the bottom of the table.
9. CONCLUSIONS

The findings of this research conclude that large routing tables have a significant impact on the router performance. The other important conclusion is the uncertainty in predicting the behavior of routing table processing of a router.

Phase 1 of this research clearly indicates that the increasing routing table size on a router increases the packet processing times on the router. Post CEF failure, the jitter values also become fairly high, indicating the performance degradation of the network. Based on this part of the research, it can be concluded that it is very necessary to provision the router correctly if we need good network performance. We should always keep in mind that Internet routing tables can be unstable at times and inject unusually large amounts of routing information to routers. In case the routers do not have enough memory and processor cycles to process these spikes, they would crash causing large downtimes, as these routers need a lot of time to come to a stable state and start processing packets after a crash or a reboot.

Phase 2 of the research indicates that predicting the route lookup behavior is difficult. The routers do not process routes at the top of the table any faster than at the bottom. Routes at the bottom of the routing table have processing time similar to ones at the top. The other important conclusion from this research is that routers do not process frequently used routes faster than the routes which are not fast. This behavior of routers makes it almost impossible to formulate an exploitation script to make use of the route processing procedure.


10. FUTURE WORK

This research can be further extended by using different router models as well as routers from different vendors like Juniper Networks. This research is performed on a router that was not processing packets other than the experimental packets. A good next step would be to study routers with significant traffic other than the experimental traffic to take the study even closer to actual Internet routers. The sheer number of packets being processed affects the throughput of Internet routers, and this research could be extended to study the effect of different packets per second on a router.

Traffic generators and advanced visualization tools would enhance this study and could be used when this study is taken to the next step.
11. BIBLIOGRAPHY


12. APPENDIX

1. Configuration generation script.

```perl
#!/usr/bin/perl
#Arnav Ghosh
#cisco config.pl

open FILE,"<"."/home/silvers/Desktop/putty.log" or die;
@origfile = <FILE>;
close FILE;

for ($i=0;$i<=$#origfile;$i++)
{
  if ($origfile[$i] =~ /.*Loopback0.*/) 
  {
    open FILE,">>","/home/silvers/Desktop/confop200k";
    for ($j=0;$j<3;$j++)
    {
      for ($k=0;$k<255;$k++)
      {
        for ($l=0;$l<255;$l++)
        {
          $loopbackctr++;
          print FILE " ip route 201.1$k.$l.0 255.255.255.0 FastEthernet0/0 192.168.1.10\n";
          if ($loopbackctr == 200000)
          {
            $l=256;
            $k=256;
            $j=4;
          }
        }
      }
    }
  }
}
```

2. CEF [11]: Cisco’s Express Forwarding (CEF) technology for IP is a scalable, distributed, layer 3 switching solution designed to meet the future performance requirements of the Internet and Enterprise networks.

3. Iperf [12]: Iperf is a tool to measure the bandwidth and the quality of a network link.
4. Router R2 Config

!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R2
!
boot-start-marker
boot-end-marker
!
enable password cisco
!
no aaa new-model
!
resource policy
!
!
no ip cef
!
!
!
password encryption aes

!
!
!
interface FastEthernet0/0
  ip address 192.168.1.1 255.255.255.0
duplex auto
  speed auto
!
interface FastEthernet0/1
  ip address 192.168.100.1 255.255.255.0
duplex auto
  speed auto
!
  ip route 10.0.0.0 255.255.255.255 FastEthernet0/0 192.168.1.10
  ip route 10.0.0.1 255.255.255.255 FastEthernet0/0 192.168.1.10
  ip route 10.0.0.2 255.255.255.255 FastEthernet0/0 192.168.1.10
  ip route 10.0.0.3 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.4 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.5 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.6 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.7 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.8 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.9 255.255.255.255 FastEthernet0/0 192.168.1.10
ip route 10.0.0.10 255.255.255.255 FastEthernet0/0 192.168.1.10
.........
ip route 200.110.144.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 200.110.145.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 200.110.146.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 200.110.147.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 200.110.148.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 200.110.149.0 255.255.255.0 FastEthernet0/0 192.168.1.10
ip route 192.168.200.0 255.255.255.0 FastEthernet0/1 192.168.100.100
!
!
ip http server
no ip http secure-server
!
!
!
control-plane
!
!
!
!
!
line con 0
length 0
line aux 0
line vty 0 4
password cisco
login
line vty 5 7
password cisco
login
!
scheduler allocate 20000 1000
!
end
5. Raw traceroute data sample

silvers@silvers-desktop:~$ sudo traceroute -I -n 10.0.176.152
traceroute to 10.0.176.152 (10.0.176.152), 30 hops max, 40 byte packets
 1 192.168.100.1  13.522 ms 13.250 ms 13.146 ms
 2 10.0.176.152  13.061 ms 12.982 ms 13.105 ms

silvers@silvers-desktop:~$ sudo traceroute -I -n 10.0.172.136
traceroute to 10.0.172.136 (10.0.172.136), 30 hops max, 40 byte packets
 1 192.168.100.1  1.348 ms  2.996 ms  2.780 ms
 2 10.0.172.136  13.126 ms 13.047 ms 12.971 ms

6. Raw traceroute and iperf data sample

silvers@silvers-desktop:~/Desktop$ sudo traceroute -I -n 10.0.0.5
[sudo] password for silvers:
traceroute to 10.0.0.5 (10.0.0.5), 30 hops max, 40 byte packets
 1 192.168.200.1  1.081 ms  0.972 ms  1.264 ms
 2 192.168.100.1  166.722 ms 166.735 ms 166.832 ms
 3 10.0.0.5  1.568 ms  1.699 ms  1.683 ms

silvers@silvers-desktop:~/Desktop$ iperf -c 10.0.99.2 -u -b 10m
_________________________________________________________
Client connecting to 10.0.99.2, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size: 109 KByte (default)
_________________________________________________________
[ 3] local 192.168.200.50 port 52955 connected with 10.0.99.2 port 5001
[ 3] 0.0-10.0 sec 11.9 MBytes 10.0 Mbits/sec
[ 3] Sent 8505 datagrams
[ 3] Server Report:
[ 3] 0.0-10.0 sec 28.7 KBytes 23.5 Kbits/sec 31.432 ms 8485/ 8504 (1e+02%)

7. Router memory allocation failure with BGP

*Jan 31 20:26:18.783: %CONTROLLER-5-UPDOWN: Controller T1 0/0/0, changed state to down (LOS detected)
*Jan 31 20:26:48.559: %BGP-5-ADJCHANGE: neighbor 192.168.2.10 Up
*Jan 31 20:41:28.975: %SYS-2-MALLOCFAIL: Memory allocation of 65536 bytes failed from 0x400AA81C, alignment 0
Pool: Processor Free: 64740 Cause: Not enough free memory
Alternate Pool: None Free: 0 Cause: No Alternate pool
-Process= "BGP Router", ipl= 0, pid= 166 -Traceback= 0x41182E94 0x4009A648
0x4009F254 0x4009F750 0x400AA824 0x400AB754 0x40E12710 0x40E46600
0x40E47288 0x40E254BC 0x42196BE4 0x42196BC8
*Jan 31 20:41:47.167: %BGP-5-ADJCHANGE: neighbor 192.168.2.10 Down No memory
*Jan 31 20:45:45.075: %SYS-2-MALLOCFAIL: Memory allocation of 65536 bytes failed from 0x400AA81C, alignment 0
Pool: Processor Free: 6638312 Cause: Memory fragmentation
Alternate Pool: None Free: 0 Cause: No Alternate pool
-Process= "Exec", ipl= 0, pid= 3 -Traceback= 0x41182E94 0x4009A648 0x4009F254
0x4009F750 0x400AA824 0x400AB754 0x41850CA4 0x4185BA30 0x41872BBC
0x41872DEC 0x41873A50 0x41874A1C 0x411C35C0 0x411DCFB4 0x411DD184
0x411DD2FC
*Jan 31 20:46:15.075: %SYS-2-MALLOCFAIL: Memory allocation of 65536 bytes failed from 0x400AA81C, alignment 0
Pool: Processor Free: 6638060 Cause: Memory fragmentation
Alternate Pool: None Free: 0 Cause: No Alternate pool
-Process= "Exec", ipl= 0, pid= 3 -Traceback= 0x41182E94 0x4009A648 0x4009F254
0x4009F750 0x400AA824 0x400AB754 0x4187390C 0x41873A50 0x41874A1C
0x411C35C0 0x411DCFB4 0x411DD184 0x411DD2FC 0x411DD3A8 0x4161D3EC
0x4161D7D10
5671366 bytes copied in 1255.144 secs (4518 bytes/sec)