A PROTOTYPE APPLICATION FOR END-TO-END PATH DISCOVERY IN A CONTENT DELIVERY NETWORK

PRESENTATION BY **TAHIR HUSSAIN**

EMAIL **tahir.bvs@gmail.com**

OUTLINE

1. Brief introduction, architecture and advantages of CDNs
2. Problem description
3. Aims and objectives
4. Technical methodology
5. Discussion of results
6. Future Work
7. Conclusion

HAS YOUR YOUTUBE VIDEO EVER BUFFERED?

CONTENT DELIVERY NETWORKS (CDN)

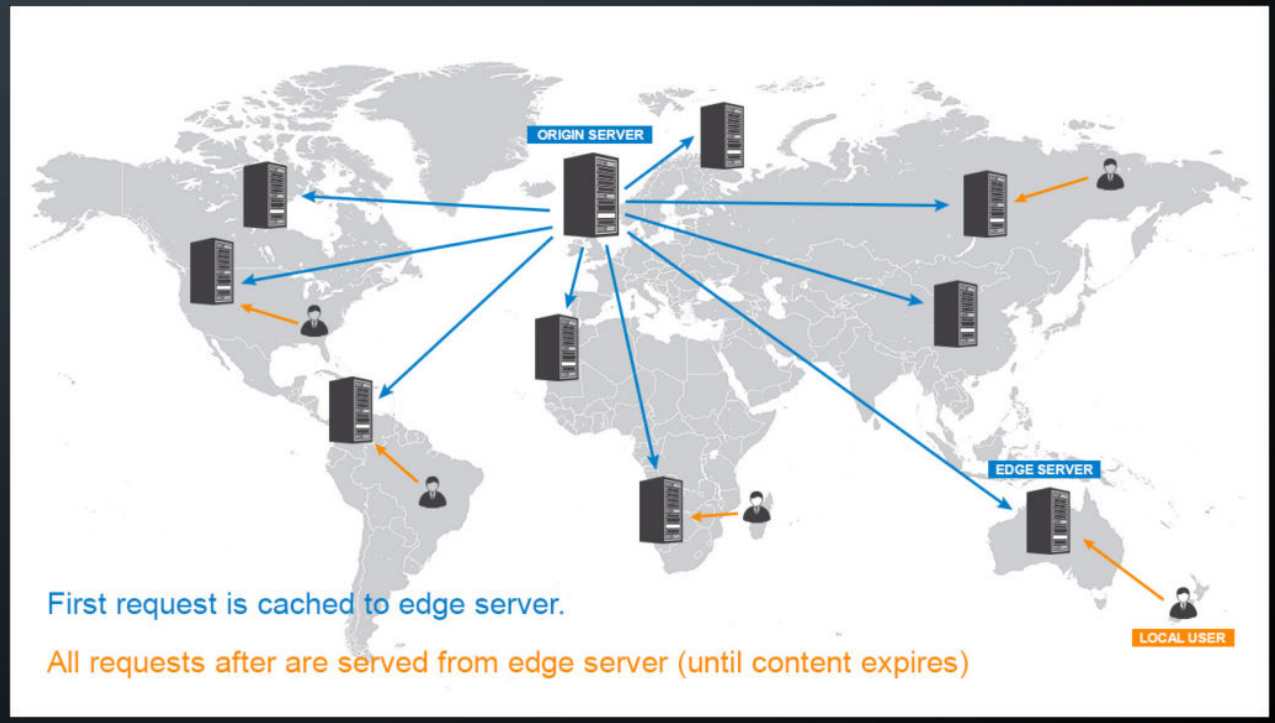
- What is a CDN ?
  - A network of high end servers strategically distributed around the globe.
- Why is it Important ?
  - To deliver the data as efficiently, quickly, reliably and cheaply as possible.
- How does it accomplish it ?
  - The system intelligently replicates the content on to the distributed servers.

WHAT IS CDN ?

- The major components make up a Content Delivery Network
  - Origin Server
    - Where the original copy of the data exist or where the data was first uploaded.
  - Surrogate Server
    - Surrogate Server or Surrogates are the nodes or the servers where the content is replicated.
- The benefits of a CDN networks are
  - Optimal Latency
  - Capacity management
  - High availability

A BRIEF VIEW OF CDN ARCHITECTURE

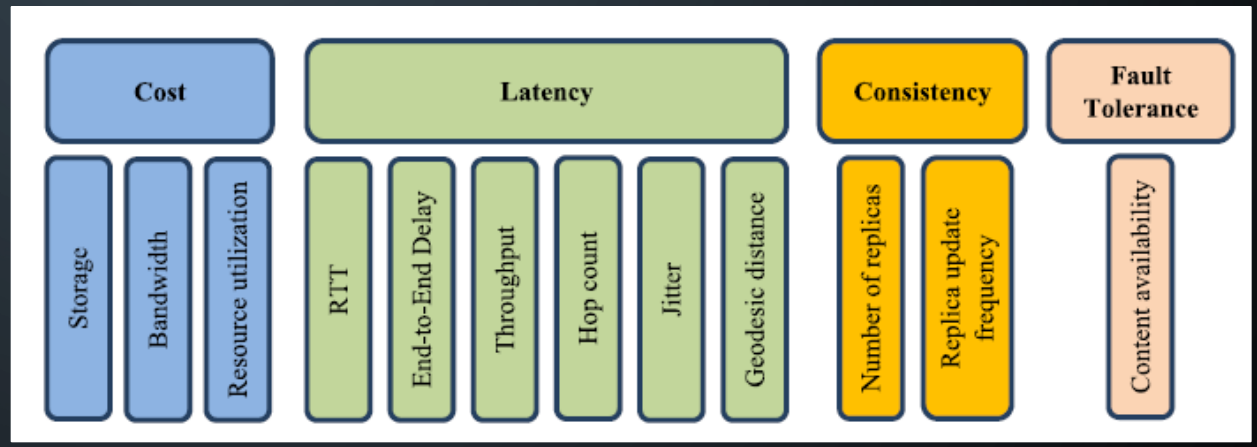
- The origin server distributes the content to the CDN servers or surrogate servers where user within proximity can easily fetch it reducing the delay and unnecessary use of transit bandwidth
- The surrogate servers or edge servers are the CDN nodes on which the Web content is replicated which is pulled from the origin server.



RESEARCH IN INEFFICIENT TRAFFIC REDIRECTION

The latency can be inflated because of an issue in either forward path from the CDN node to the prefix or on the reverse path back.

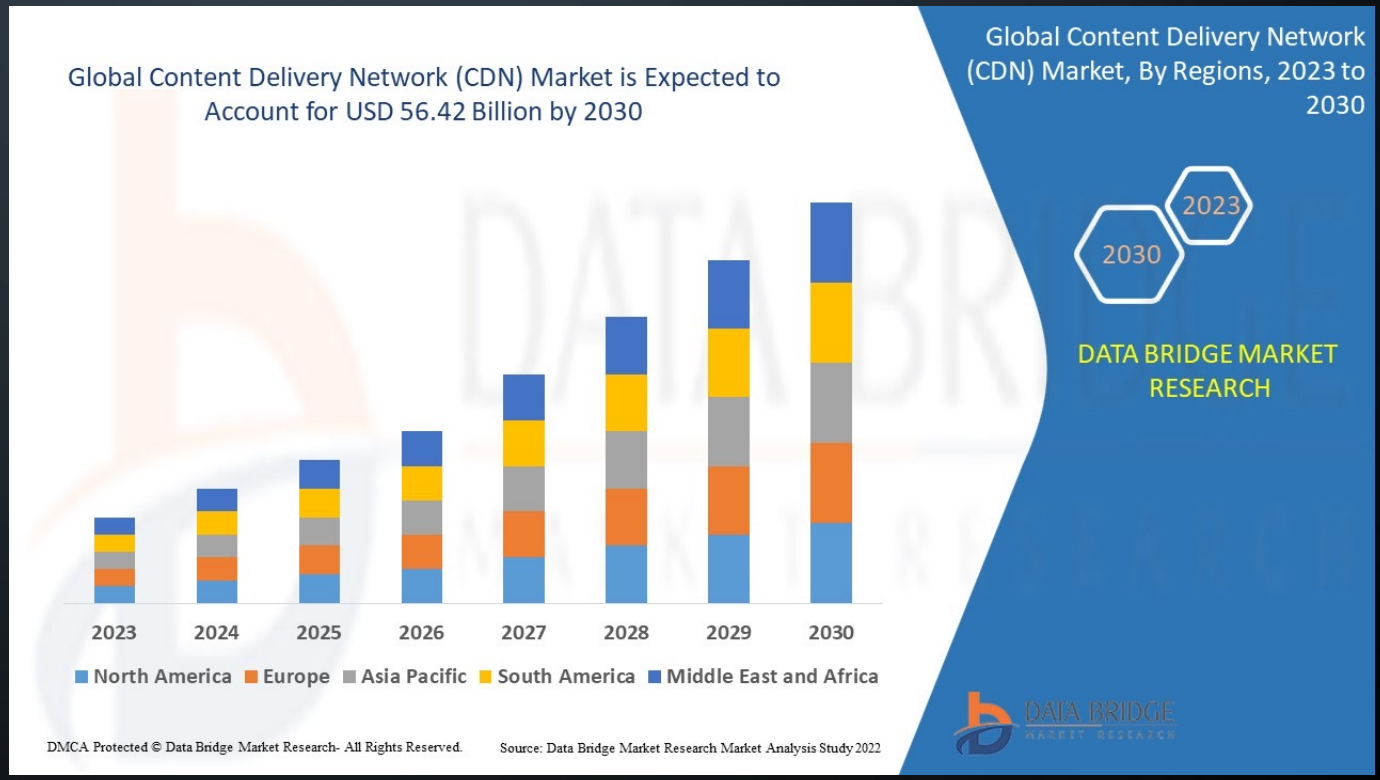
Few of the path inflating parameters are highlighted in the model frame work shown where the attributes contributing in latency inflation are RTT, End to End delay, throughput, hop count, jitter, and geodesic distance.



CDN model framework

REPORTS AND TRENDS

- Google (U.S.)
- Akamai Technologies (U.S.)
- Microsoft (U.S.)
- IBM (U.S.)
- AWS (U.S.)
- AT&T (U.S.)
- Cloudflare
- Lumen Technologies (U.S.)
- Deutsche Telekom AG (Germany)
- Comcast Cable Communications Management
- Tata Communications (India)



REPORTS AND TRENDS

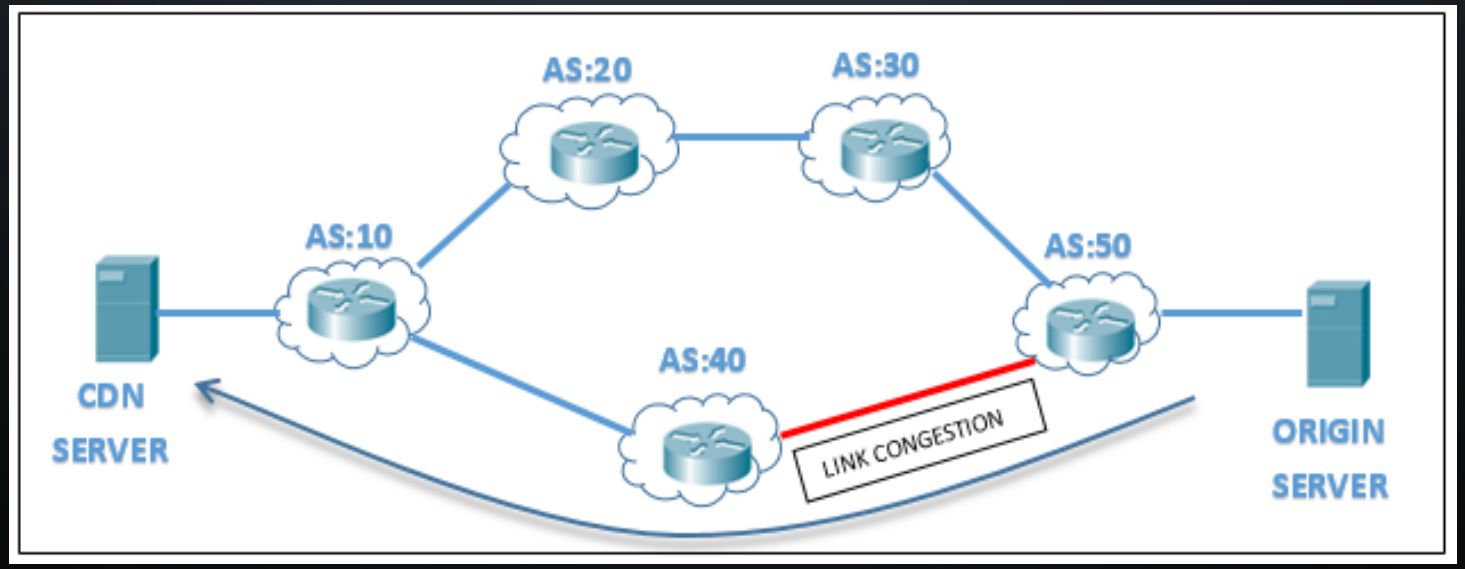
- The global content delivery network market size was estimated at USD 21.36 billion in 2023 and is expected to register a CAGR of 17.7% from 2024 to 2030.
- The application of digital solutions based on IoT networks and artificial intelligence across various industries is driving market players to develop and promote customized industry-specific CDN solutions.
- One-third of all online activities on the internet comprises of watching videos [1].
- Cisco had estimated that content delivery network traffic will carry nearly two-thirds of all Internet video traffic by 2030 [1].
- Netflix alone is reportedly streaming over 2 billion hours of video each month which is equivalent to almost 7,200,000 Terabytes of video traffic [1].

PROBLEM DESCRIPTION

- Inefficient Redirection
  - Internet routing follows the destination-based routing principle and the entire routing table of the internet is made up of updates received from neighbours using the BGP protocol. BGP itself follows path selection process based on the metric, which for BGP is shortest AS path length.
  - This may not be correct in terms of the shortest path every time, as the destination having the shortest AS path might not have the most optimal path towards it which in turn can give rise to latency.

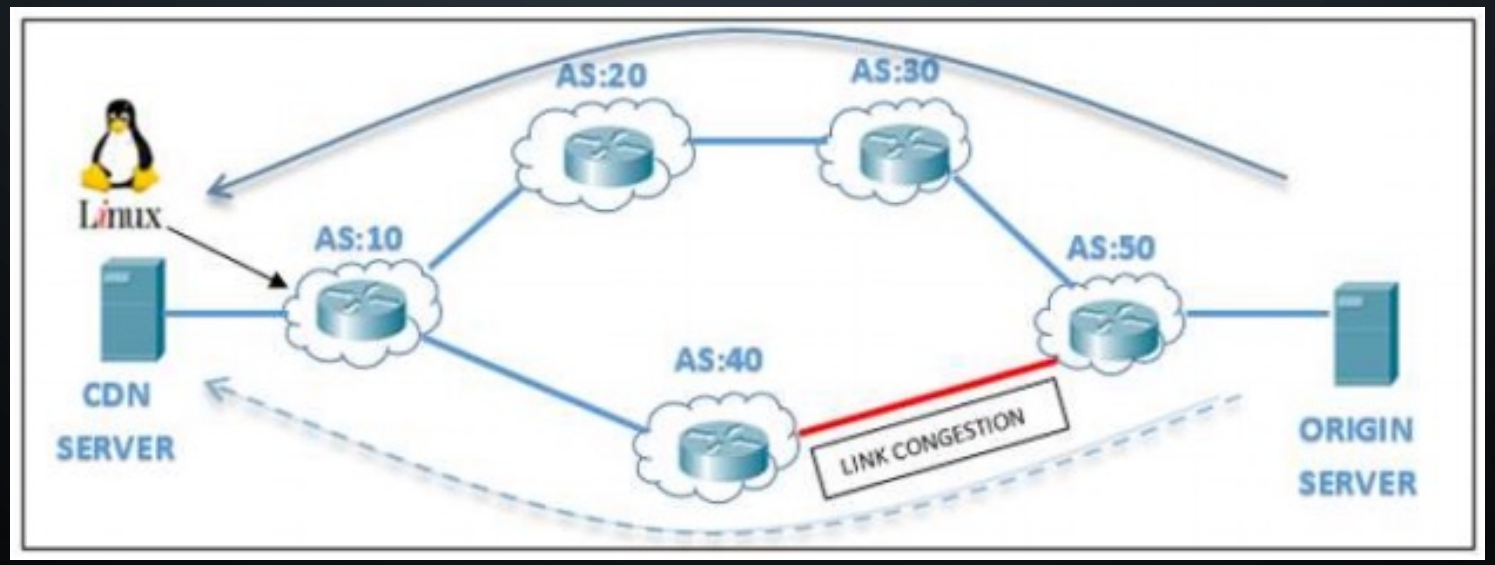
PROBLEM DESCRIPTION

- Inefficient Redirection
  - Traffic being redirected to nodes having degraded path attributes



PROJECT AIM

- A prototype application to resolve inefficient redirection
  - Traffic being monitored and intelligently shifted to paths having better path attributes



OBJECTIVES

- Primary Objectives
  - Sourcing and Initializing Hardware and Software
  - Topology Construction
  - Virtual Machine Integration
  - Designing the flow chart of the algorithm
  - Translating the designed logic algorithm
  - Extracting the results
  - Plotting the results
- Secondary Objectives
  - Analysing BGP path selection criteria
  - Studying qualitative research in inefficient traffic redirection in CDNs
  - Research to extract the link gauging parameters
  - Research Data Analytics Software

METHEDOLOGY

The technical methodology of the project is based on a Cisco PPDIOO approach that defines a life cycle approach of a service within a network.

The PPDIOO approach stands for:

- P Prepare
- P Plan
- D Design
- I Implement
- O Operate
- O Optimize.

METHEDOLOGY

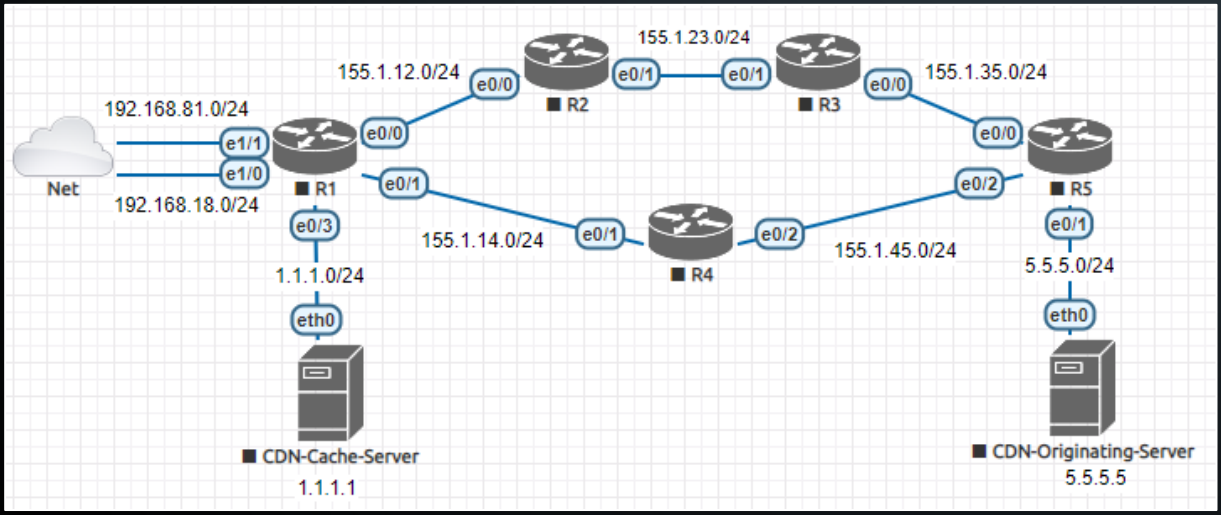
Preparing and planning the fundamentals and tools for the project

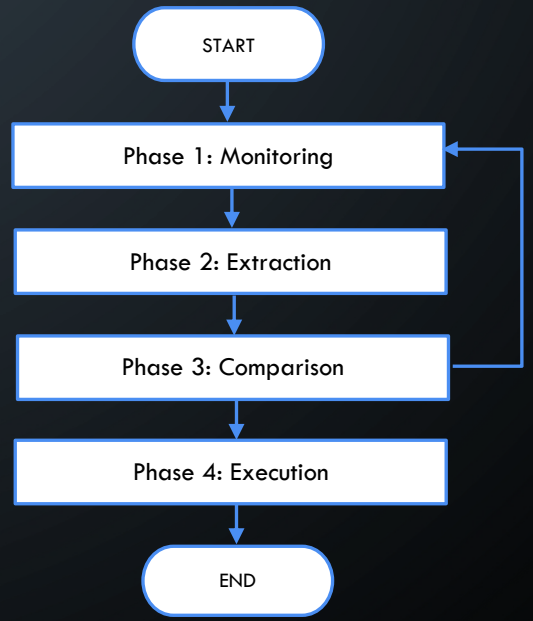
1. Sourcing and Initializing Hardware and Software	VMware Workstation
2. Installation of a network emulator as a virtual machine	EVE-NG
3. Topology Construction on the network emulator	Test Topology
4. Virtual Machine Integration	Ubuntu 18.0.4 LTS
5. Installation of programming language	Bash
6. Extracting and plotting the result	Smokeping

METHEDOLOGY

Designing application flow chart

1. Application Planning Flow Chart

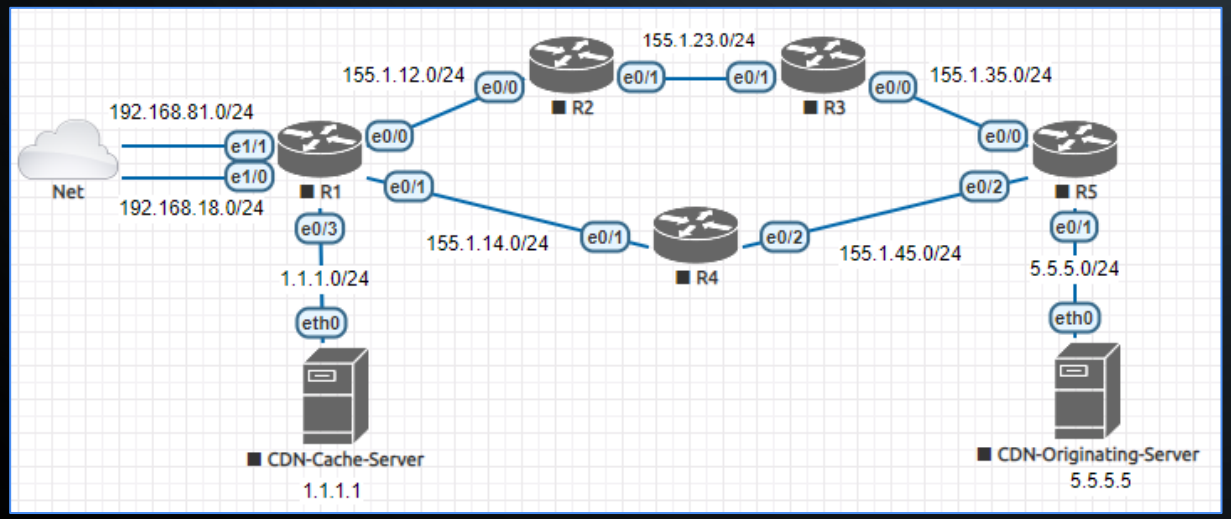


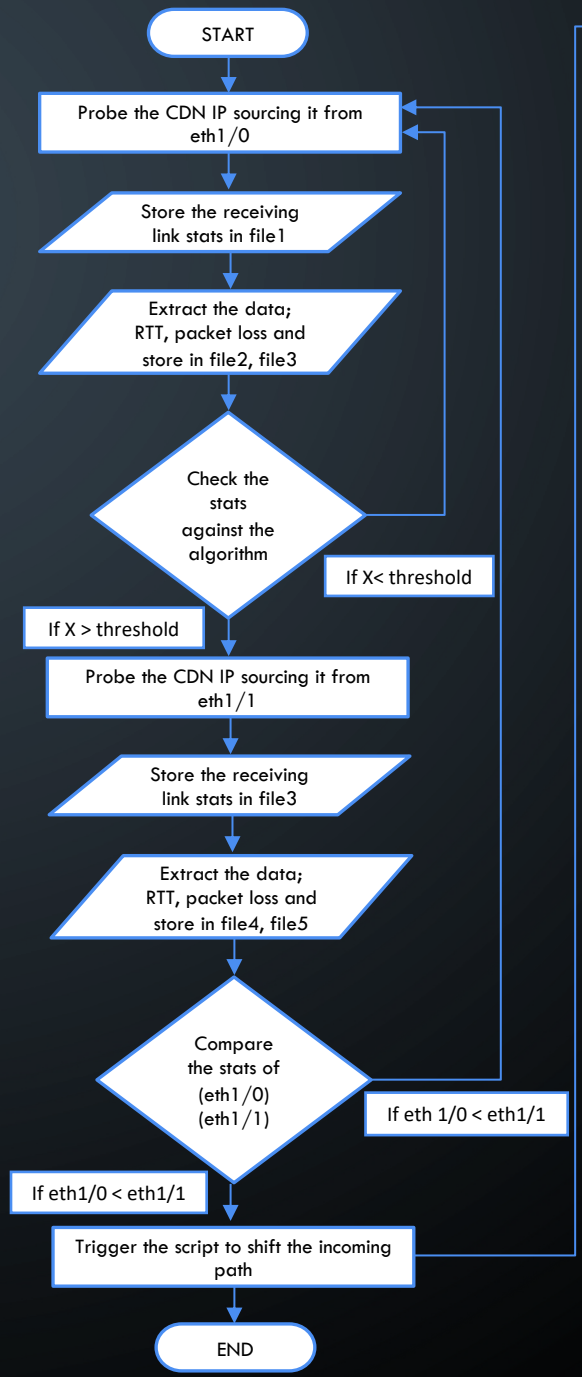


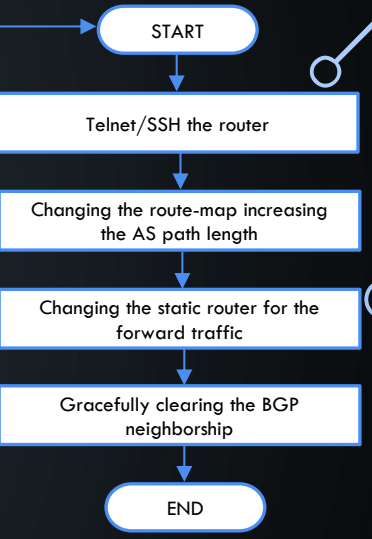
METHEDOLOGY

Designing application flow chart

1. Application Execution Flow Chart



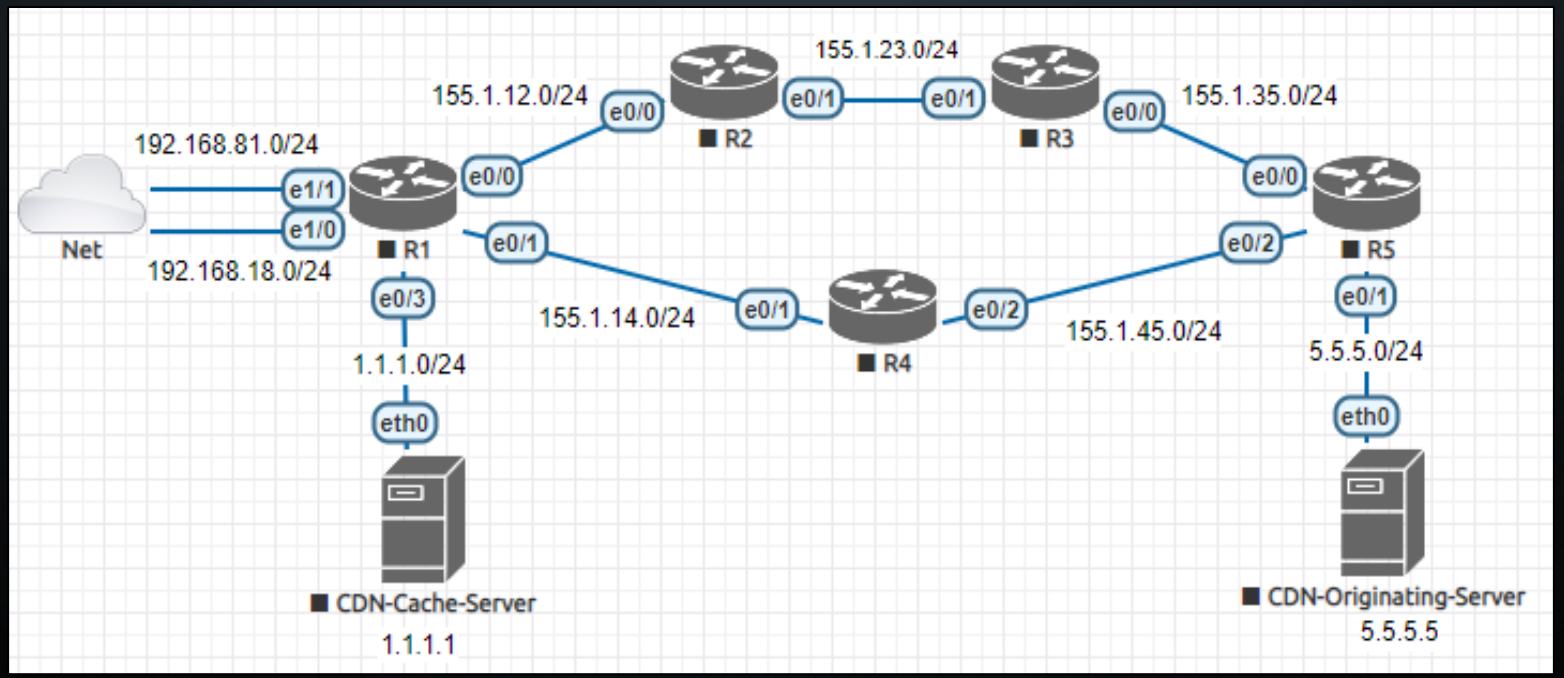




METHEDOLOGY

Implementation, operation and execution

Topology:



METHEDOLOGY

Implementation, operation and execution

Pings

```
aie@aie:~$ ping 5.5.5.5
PING 5.5.5.5 (5.5.5.5) 56(84) bytes of data.
64 bytes from 5.5.5.5: icmp_seq=1 ttl=61 time=0.946 ms
64 bytes from 5.5.5.5: icmp_seq=2 ttl=61 time=0.879 ms
64 bytes from 5.5.5.5: icmp_seq=3 ttl=61 time=1.09 ms
64 bytes from 5.5.5.5: icmp_seq=4 ttl=61 time=1.08 ms
64 bytes from 5.5.5.5: icmp_seq=5 ttl=61 time=0.972 ms
^C
--- 5.5.5.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4029ms
rtt min/avg/max/mdev = 0.879/0.995/1.095/0.087 ms
```

My traceroute

```
                                My traceroute  [v0.92]
aie (192.168.18.128)                                2019-08-21T18:49:56+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                     Packets               Pings
 Host                              Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 192.168.18.150                  0.0%     6    0.5   0.6   0.5   0.7   0.1
 2. 155.1.14.4                      0.0%     6    0.6   0.9   0.6   1.1   0.2
 3. 155.1.45.5                      0.0%     5    0.9   1.0   0.9   1.1   0.1
 4. 5.5.5.5                         0.0%     5    1.2   1.3   1.1   1.7   0.2
```

METHEDOLOGY

Implementation, operation and execution

Implementing Phase 1: Monitoring

```
ping -c 10 5.5.5.5  > ./output/step1-ping-results
```

```
aie@aie:~/scripts$ more ./output/step1-ping-results
PING 5.5.5.5 (5.5.5.5) 56(84) bytes of data.
64 bytes from 5.5.5.5: icmp_seq=1 ttl=61 time=2.57 ms
64 bytes from 5.5.5.5: icmp_seq=2 ttl=61 time=5.97 ms
64 bytes from 5.5.5.5: icmp_seq=3 ttl=61 time=6.63 ms
64 bytes from 5.5.5.5: icmp_seq=4 ttl=61 time=6.31 ms
64 bytes from 5.5.5.5: icmp_seq=5 ttl=61 time=7.42 ms
64 bytes from 5.5.5.5: icmp_seq=6 ttl=61 time=7.22 ms
64 bytes from 5.5.5.5: icmp_seq=7 ttl=61 time=7.13 ms
64 bytes from 5.5.5.5: icmp_seq=8 ttl=61 time=8.32 ms
64 bytes from 5.5.5.5: icmp_seq=9 ttl=61 time=18.8 ms
64 bytes from 5.5.5.5: icmp_seq=10 ttl=61 time=6.49 ms

--- 5.5.5.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9003ms
rtt min/avg/max/mdev = 2.577/7.697/18.882/3.998 ms
aie@aie:~/scripts$
```

METHEDOLOGY

Implementation, operation and execution

Implementing Phase 2: Extraction

```
cat ./output/step1-ping-results | grep rtt > ./output/step2-rtt
```

```
aie@aie:~/scripts$ cat ./output/step1-ping-results | grep rtt
rtt min/avg/max/mdev = 2.577/7.697/18.882/3.998 ms
aie@aie:~/scripts$
```

```
cat ./output/step1-ping-results | grep loss > ./output/step3-loss
```

```
aie@aie:~/scripts$ cat ./output/step1-ping-results | grep loss
10 packets transmitted, 10 received, 0% packet loss, time 9003ms
aie@aie:~/scripts$
```

METHEDOLOGY

Implementation, operation and execution

Implementing Phase 2: Extraction

```
rtt=$(cat ./output/step2-rtt | cut -d ' ' -f4 | cut -d '/' -f2)
```

```
aie@aie:~/scripts$ rtt=$(cat ./output/step2-rtt | cut -d ' ' -f4 | cut -d '/' -f2)
aie@aie:~/scripts$ echo $rtt
7.697
aie@aie:~/scripts$
```

```
loss=$(cat ./output/step3-loss | cut -d ' ' -f6 | sed 's/%//')
```

```
aie@aie:~/scripts$ loss=$(cat ./output/step3-loss | cut -d ' ' -f6 | sed 's/%//')
aie@aie:~/scripts$ echo $loss
0
aie@aie:~/scripts$
```

METHEDOLOGY

Implementation, operation and execution

Implementing Phase 3: Comparison

```
if [ $loss -gt 7 ] || [ $loss -eq 7 ] || [$rtt -gt 20]
then
    echo "loss is more"
    expect ./shifter-r4-to-r3.sh
fi
```

METHEDOLOGY

Implementation, operation and execution

Implementing Phase 4: Execution

```
#!/usr/bin/expect

spawn telnet 192.168.18.150

expect "Username: "
send "admin\n"
expect "Password: "
send "cisco\n"

expect "R1#"
send "config terminal\n"

expect "R1(config)#"
send "route-map r4-outbound permit 10\n"
expect "R1(config-route-map)#"
send "set as-path prepend 1 1 1 1 1\n"
expect "R1(config-route-map)#"
send "do clear ip bgp 155.1.14.4 soft out\n"
expect "R1(config-route-map)#"
send "exit\n"
expect "R1(config)#"
send "ip route 5.5.5.0 255.255.255.0 155.1.12.2\n"
expect "R1(config)#"
send "exit\n"
expect "R1#"
send "exit\n"
```

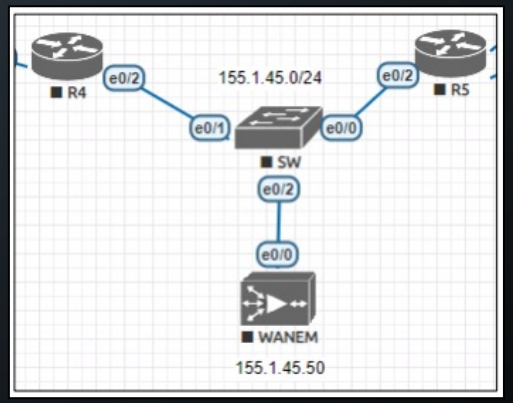
METHEDOLOGY

Optimization

1. Optimizing the monitoring phase

```
ping -c 100 -i 0.2 5.5.5.5  > ./output/step1-ping-results
```

2. Optimizing the Comparison phase
3. Introducing path degradations for application testing



METHEDOLOGY

Application Testing

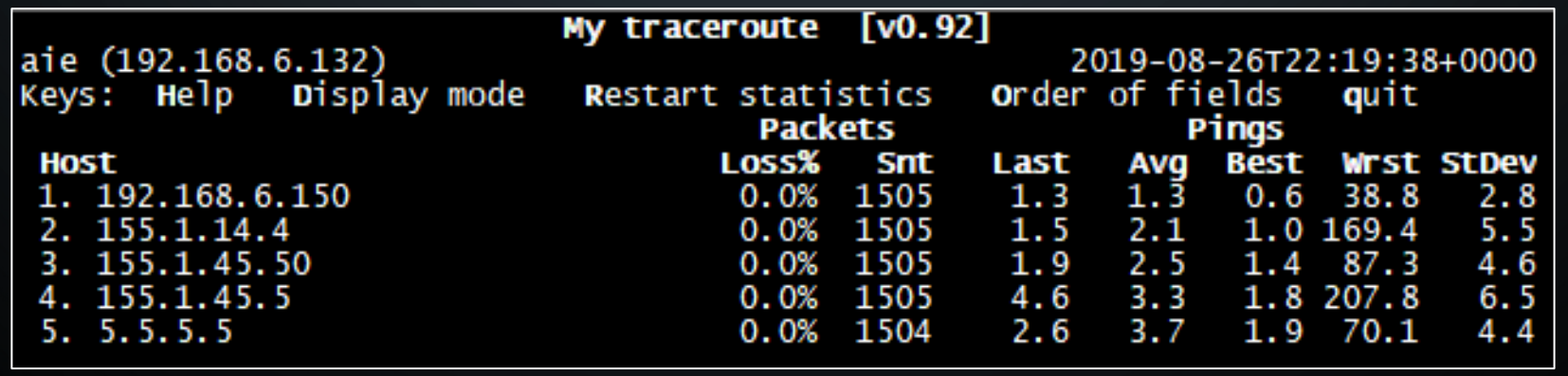
Test Scenarios	Packet Loss	RTT Value	MTR Probe	Smokeping Duration
Test 1	0%	<10ms	~1500 pings	>20 mins
Test 2	~3%	<10ms	~1500 pings	>20 mins
Test 3	~5%	<10ms	~1500 pings	>20 mins
Test 4	~7%	<10ms	~1500 pings	>20 mins
Test 5-a	~0%	<10ms	~1500 pings	>40 mins
Test 5-b	~3%	<40ms	~1500 pings	>40 mins
Test 5-c	~5%	<60ms	~1500 pings	>40 mins
Test 5-d	~7%	<80ms	~1500 pings	>40 mins

RESULTS

Application Test 1: 0% packet loss & < 20ms RTT

Application Test 2: 3% packet loss & < 20ms RTT

Application Test 3: 5% packet loss & < 20ms RTT

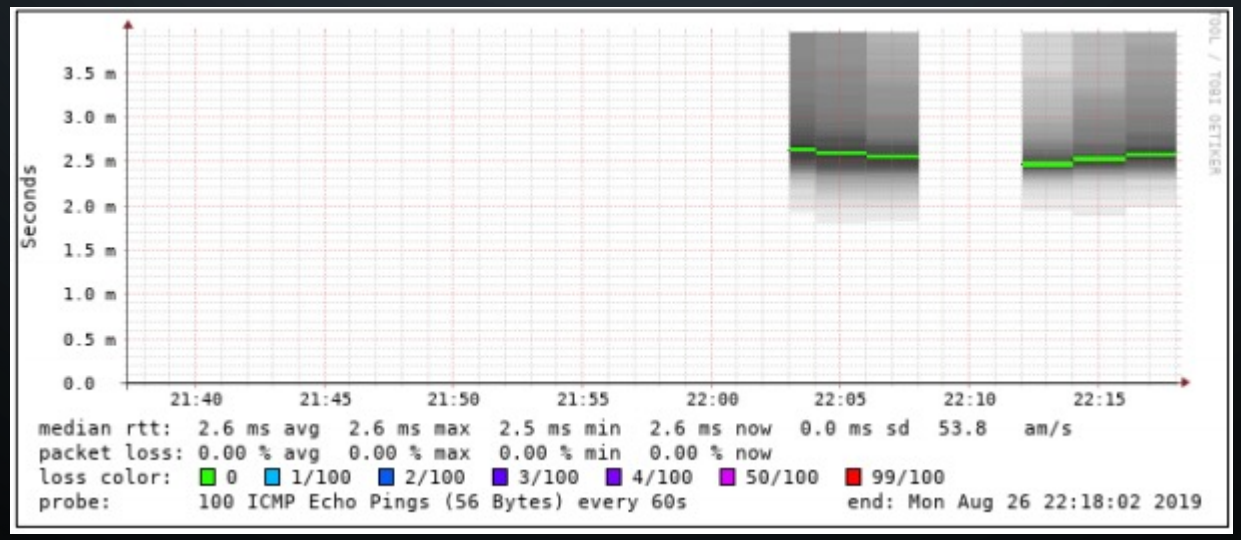


RESULTS

Application Test 1: 0% packet loss & < 20ms RTT

Application Test 2: 3% packet loss & < 20ms RTT

Application Test 3: 5% packet loss & < 20ms RTT



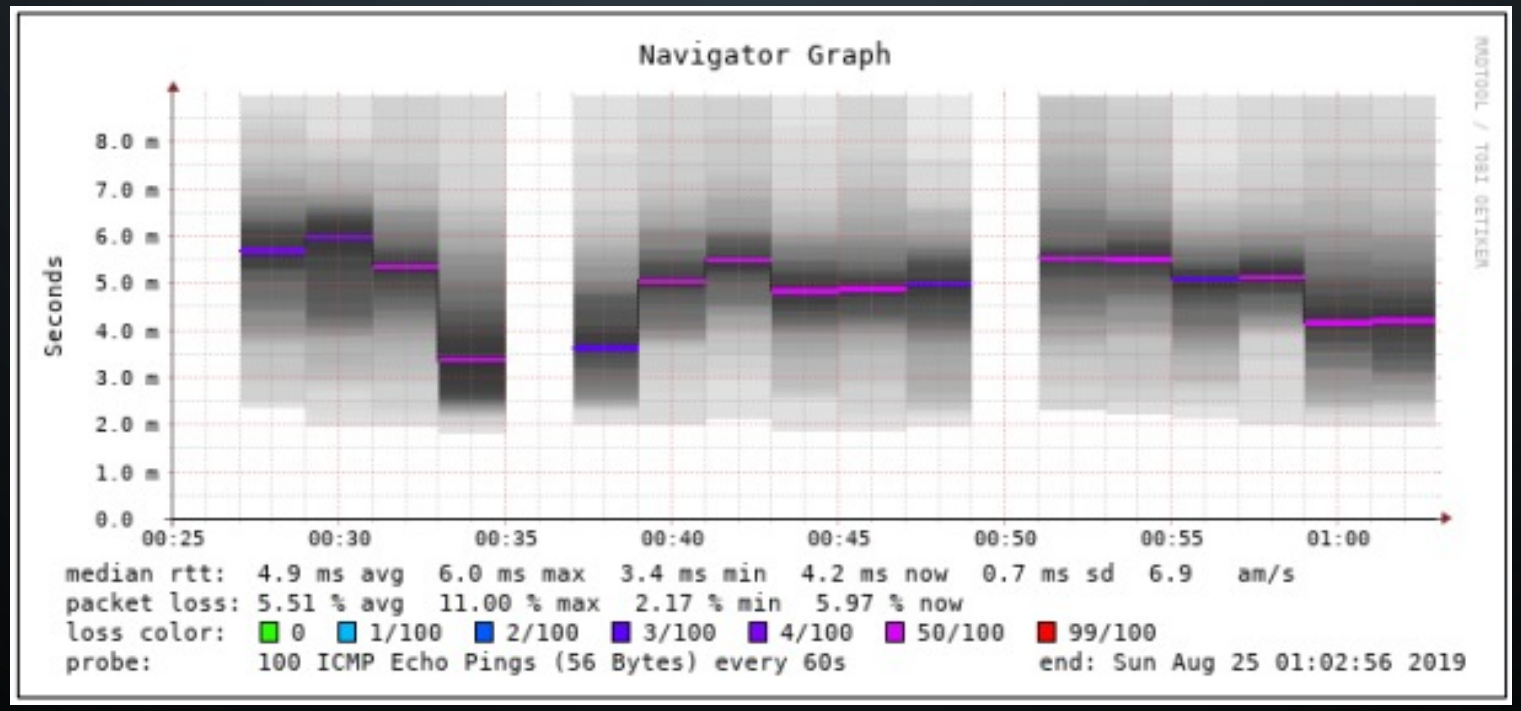
RESULTS

Application Test 4: 7% packet loss & < 20ms RTT

```
                    My traceroute  [v0.92]
aie (192.168.6.132)                              2019-08-25T01:02:11+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                 Packets                 Pings
 Host                         Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 192.168.6.150              0.0%  1416    1.0   1.8   0.7  46.1   2.0
 2. 155.1.14.4                 0.0%  1416    1.8   3.0   1.1 201.8   5.6
 3. 155.1.45.50                6.2%  1416   37.2   3.8   1.6  52.8   3.7
 4. 155.1.45.5                 7.3%  1416    3.3   4.6   1.8  52.5   3.0
 5. 5.5.5.5                    7.8%  1415    3.9   5.3   2.2  39.0   2.5
```

RESULTS

Application Test 4: 7% packet loss & < 20ms RTT



RESULTS

Application Test 4: 7% packet loss & < 20ms RTT

My traceroute [v0.92]

aie (192.168.6.132) 2019-08-25T01:13:00+0000

Keys: Help Display mode Restart statistics Order of fields quit

Host	Packets Loss%	Snt	Pings Last	Avg	Best	Wrst	StDev
1. 192.168.6.150	0.0%	2065	2.2	1.6	0.7	46.1	2.3
2. 155.1.14.4	0.0%	2065	7.8	2.7	1.1	201.8	4.8
155.1.12.2							
3. 155.1.45.50	6.8%	2065	2.2	3.5	1.5	52.8	3.2
155.1.23.3							
4. 155.1.45.5	7.2%	2064	5.6	4.3	1.8	52.5	3.0
155.1.35.5							
5. 5.5.5.5	7.4%	2064	7.6	5.0	2.0	39.0	2.6

My traceroute [v0.92]

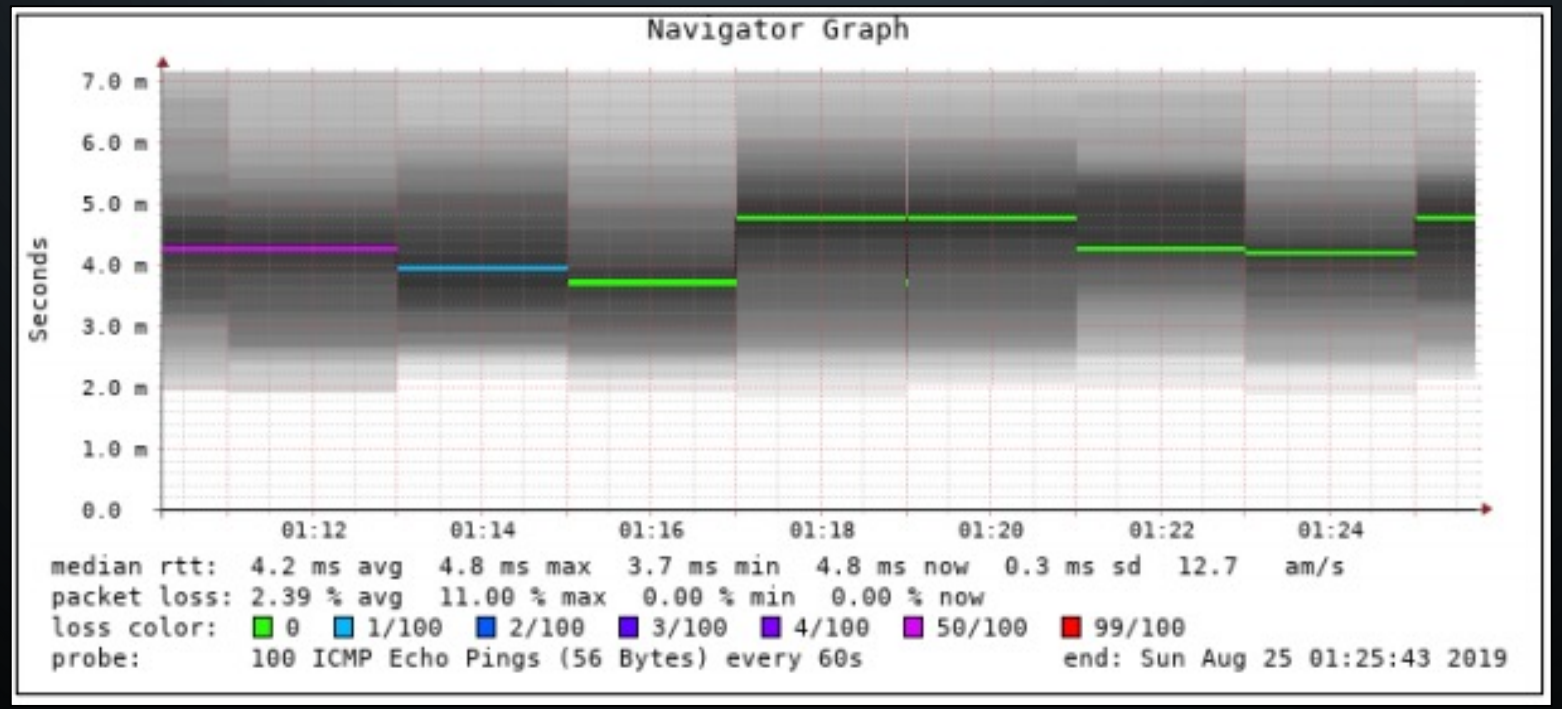
aie (192.168.6.132) 2019-08-25T16:53:53+0000

Keys: Help Display mode Restart statistics Order of fields quit

Host	Packets Loss%	Snt	Pings Last	Avg	Best	Wrst	StDev
1. 192.168.6.150	0.0%	823	0.8	1.8	0.7	51.2	2.7
2. 155.1.12.2	0.0%	822	1.6	2.6	1.0	20.8	1.8
3. 155.1.23.3	0.0%	822	2.3	3.9	1.5	134.4	5.5
4. 155.1.35.5	0.0%	822	2.7	4.1	1.7	47.5	2.9
5. 5.5.5.5	0.0%	822	35.2	4.9	1.9	78.9	4.0

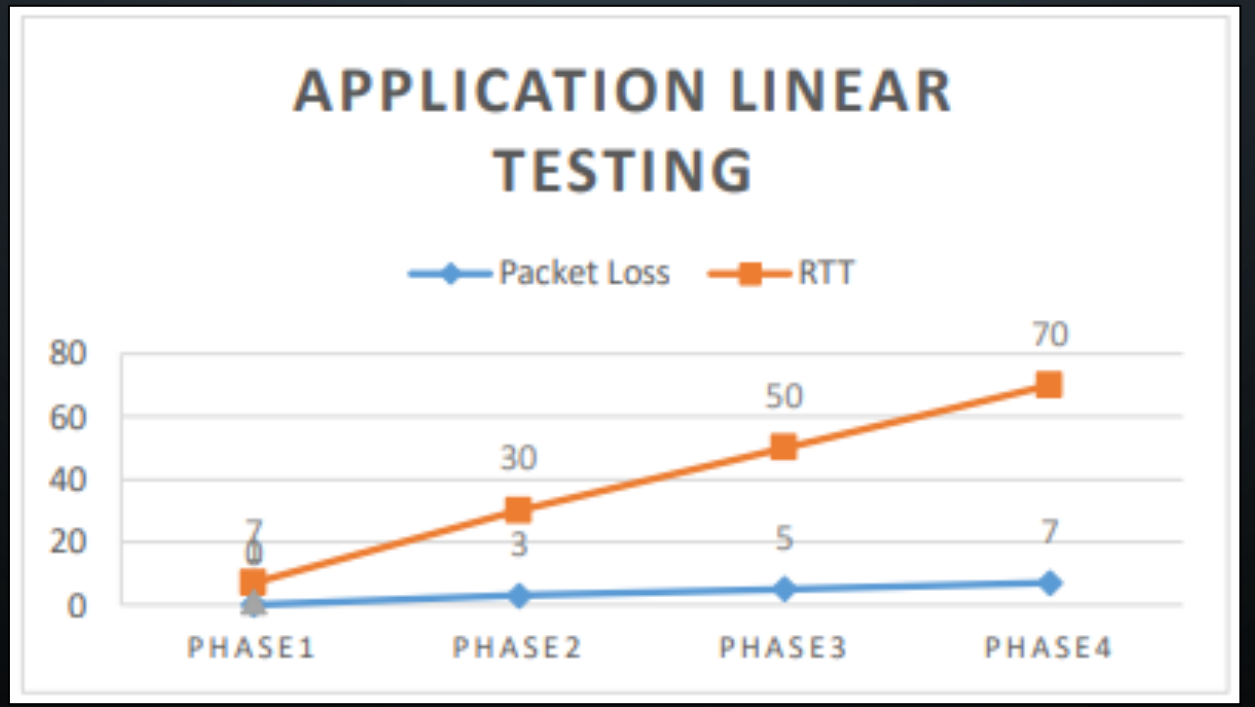
RESULTS

Application Test 4: 7% packet loss & < 20ms RTT



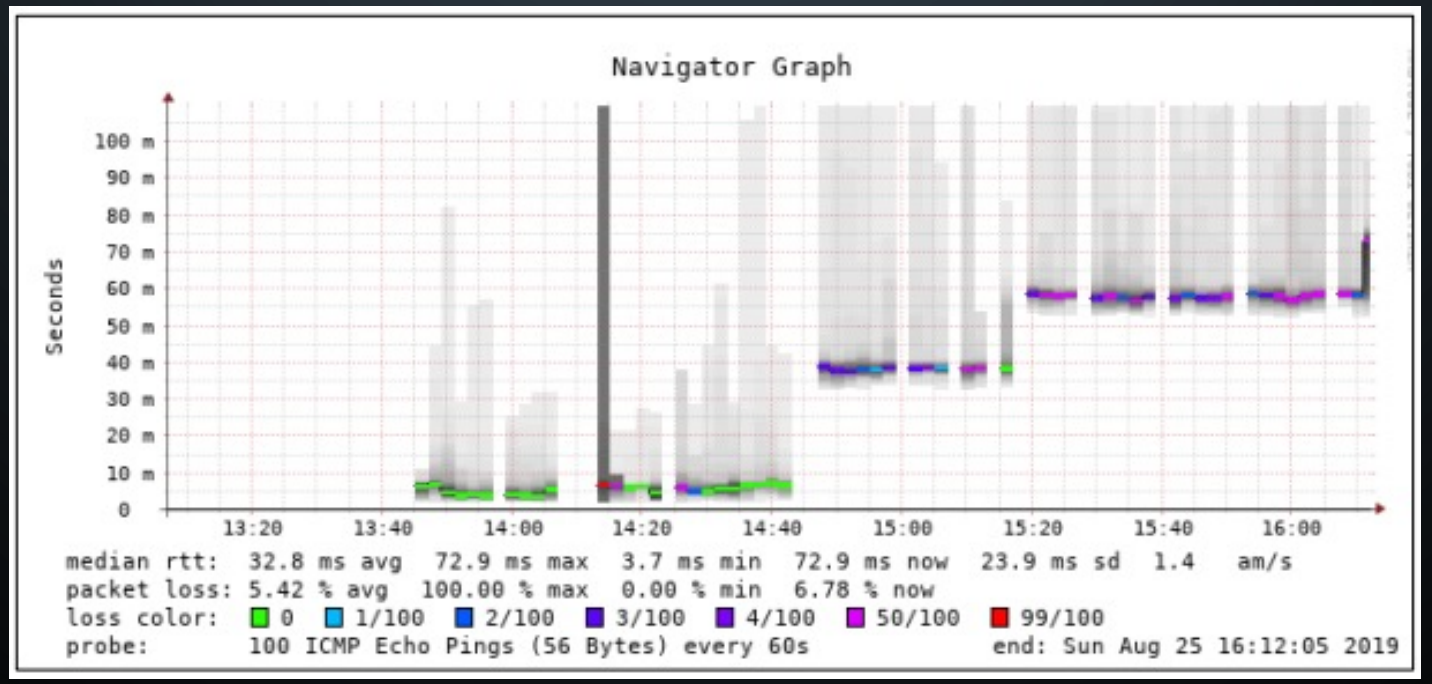
RESULTS

Application Test 5: Linear Increase in Packet Loss and RTT



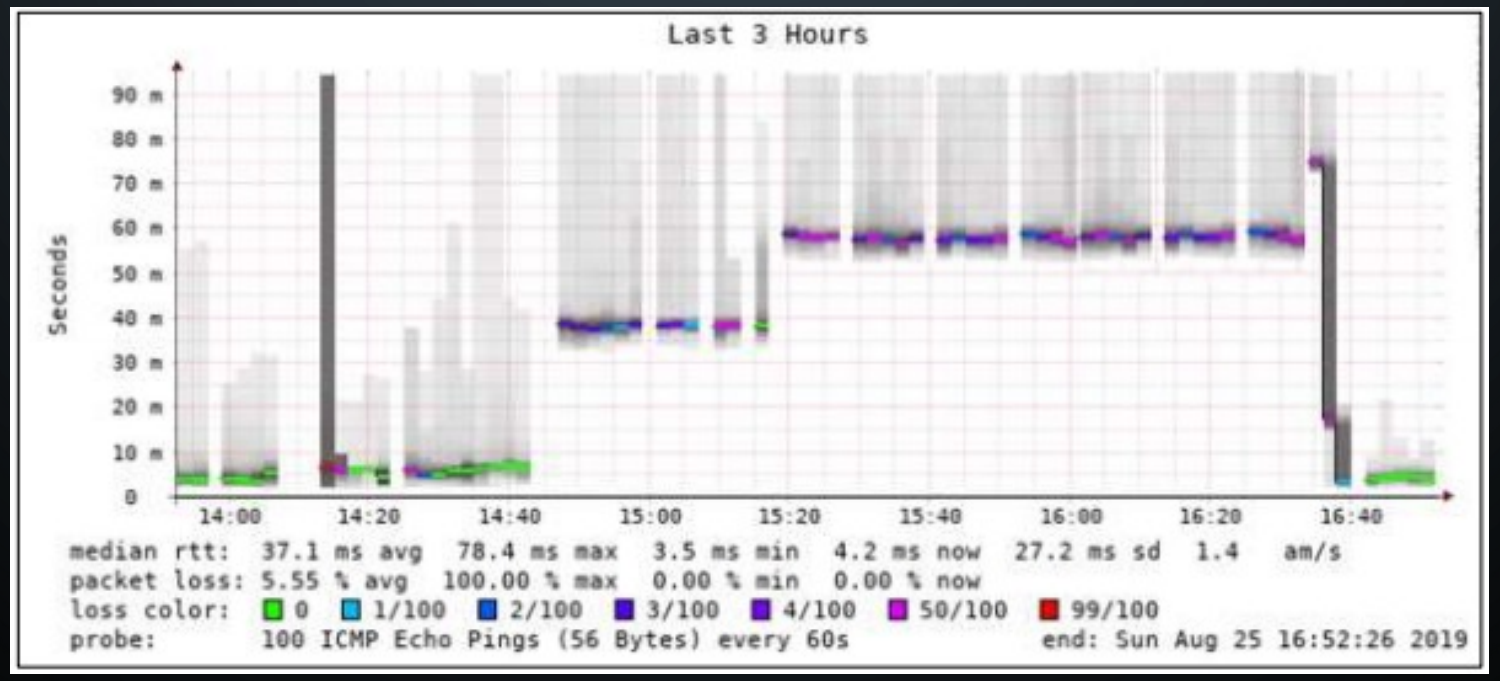
RESULTS

Application Test 5: Linear Increase in Packet Loss and RTT



RESULTS

Application Test 5: Linear Increase in Packet Loss and RTT



FURTHER WORK

- Updating the path calculation algorithm
- Creating GUI for the Application
- Integration of Artificial Intelligence
- Containerization of the application
- Including it in BGP RFC

CONCLUSION

- Content delivery networks have helped reducing the buffering and capacity problems.
- This project successfully implemented a prototype application installed on a Linux operating system
- The results from the performance evaluation testing showed a successful run of the application under stressful and abrupt situations proving its worth and shifting the path as soon as the threshold was crossed
- To conclude, the application is useful and beneficial in a service provider environment where the content is fetched and served to the local users at a very high capacity

REFERENCES

- [1] N. Anjum, D. Karamshuk, M. Shikh-Bahaei, and N. Sastry, “Survey on peer-assisted content delivery networks,” *Computer Networks*, vol. 116. pp. 1339–1351, 2017.
- [2] R. Krishnan *et al.*, “Moving beyond end-to-end path information to optimize CDN performance,” in *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference - IMC ’09*, 2009, p. 190.
- [3] Y. Zhu, B. Helsley, J. Rexford, A. Siganporia, and S. Srinivasan, “LatLong: Diagnosing wide-area latency changesfor CDNs,” *IEEE Trans. Netw. Serv. Manag.*, vol. 9, no. 3, pp. 333–345, 2012.
- [4] "Cisco - Global Home Page", Cisco, 2019. [Online]. Available: https://www.cisco.com/.

QUESTIONS ?