Distributed Systems And Networks Coursework-2

Anish Katariya
Student ID:27561879
Email: ak7n14

Index

A) Gathering Data:

For the given CSV file to run network performance test commands I wrote a script in python that took in the CSV file and looped through each line in the file and pinged the URL 10 times using both PING and PING6 commands and saved the result to two separate text files one for IPv4 and one for IPv6.
The python script for the same is given below

```
#Developed by Anish Katariya
import subprocess
import csv

#Defines the function to take the given url
#Ping for ipv4 it and save the relevant result to a text file
def pingSite(URL):
    ping = subprocess.Popen(["ping", "-c", "10",URL],stdout = subprocess.PIPE,stderr = subprocess.PIPE)
    out, error = ping.communicate()
    out = out.strip()
    error = error.strip()
    output = open("PingResults.txt",'a')
    output.write(str(out))
    output.write(str(error))
    output.write("\n\n====================\n\n")
    print(out)
    print(error)
    output.close()
#Defines the function to take the given url
#Ping for ipv6 it and save the relevant result to a text file
def pingSite6(URL):
    ping = subprocess.Popen(["ping6", "-c", "10",URL],stdout = subprocess.PIPE,stderr = subprocess.PIPE
    out, error = ping.communicate()
    out = out.strip()
    error = error.strip()
    output = open("PingResults6.txt",'a')
    output.write(str(out))
    output.write(str(error))
    output.write("\n\n====================\n\n")
    print(out)
    print(error)
    output.close()
#Goes through each line in the given csv file
#Extracs the URL and send it to both files
with open('Top100sites.csv') as csvfile:
    readCsv = csv.reader(csvfile, delimiter = ',')
    for row in readCsv:
        URL=row[1]
        pingSite(URL)
        pingSite6(URL)
```

The results were then scanned for the relevant parts using grep and store in an CSV format and opened in Google Sheet for graphing.

The assumptions made for the data is that these were average cases for the website and there was not high traffic on both networks(website network and Users Network).

B) Measurements:

To analyse the data better I made use of graphs. Some of them are given below

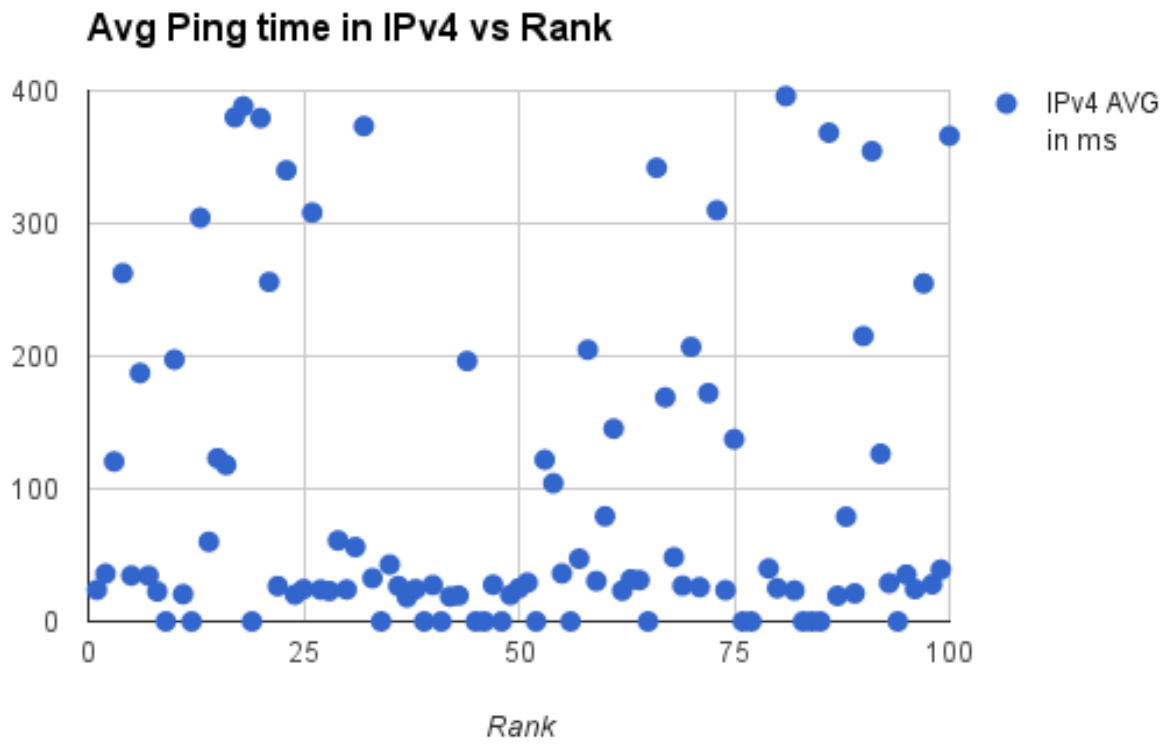


Figure 1(Average Ping time in IPv4 vs Rank)

This graph(Figure 1) showed that there was a great variation on the average time for the packet to return from the website even though they were all really popular websites . This showed that even though some websites got a high amount of traffic everyday they were still using slower servers hence showing a great amount of difference in the avg ping time.

An other interesting finding from this graph was that 18% of the websites showed **100% package loss**. On investigation it was found out that to avoid being DDoSed certain websites use firewalls that do not respond to ping ICMP packets.

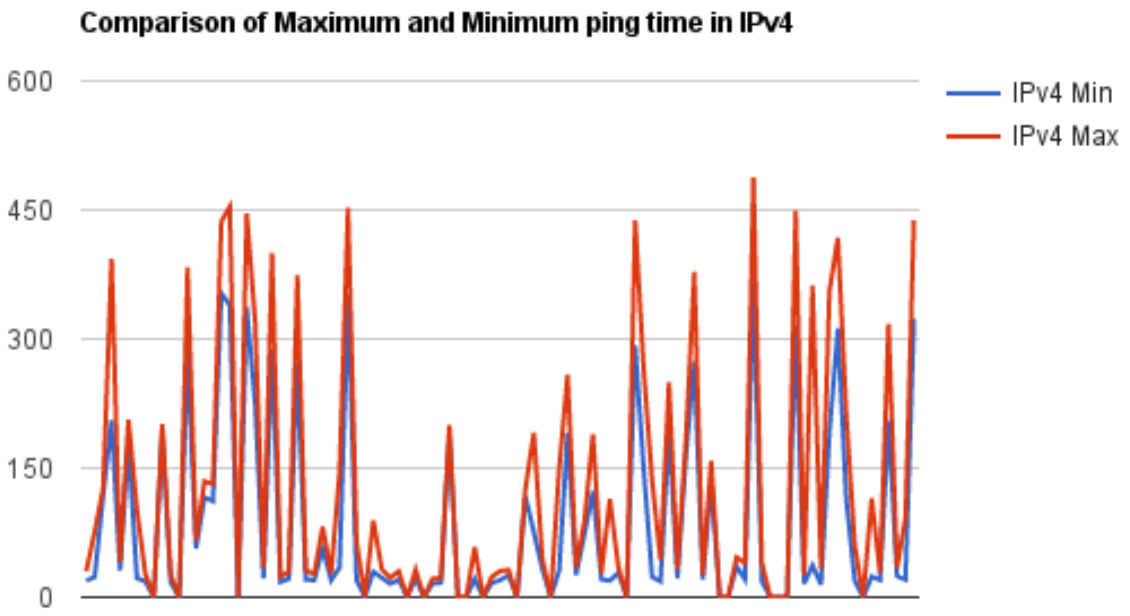


Figure 2(Comparison of maximum and minimum ping time in IPv4)

From the above graph it was easy to conclude that there was not a lot of difference in the minimum and maximum time taken for a IPv4 Ping packet to return. However this test can not be fully relied upon as the data collected for this test was taken in quick succession and there would have been no sufficient change on the traffic on the websites in such a small window of time to effect the response time of the websites.

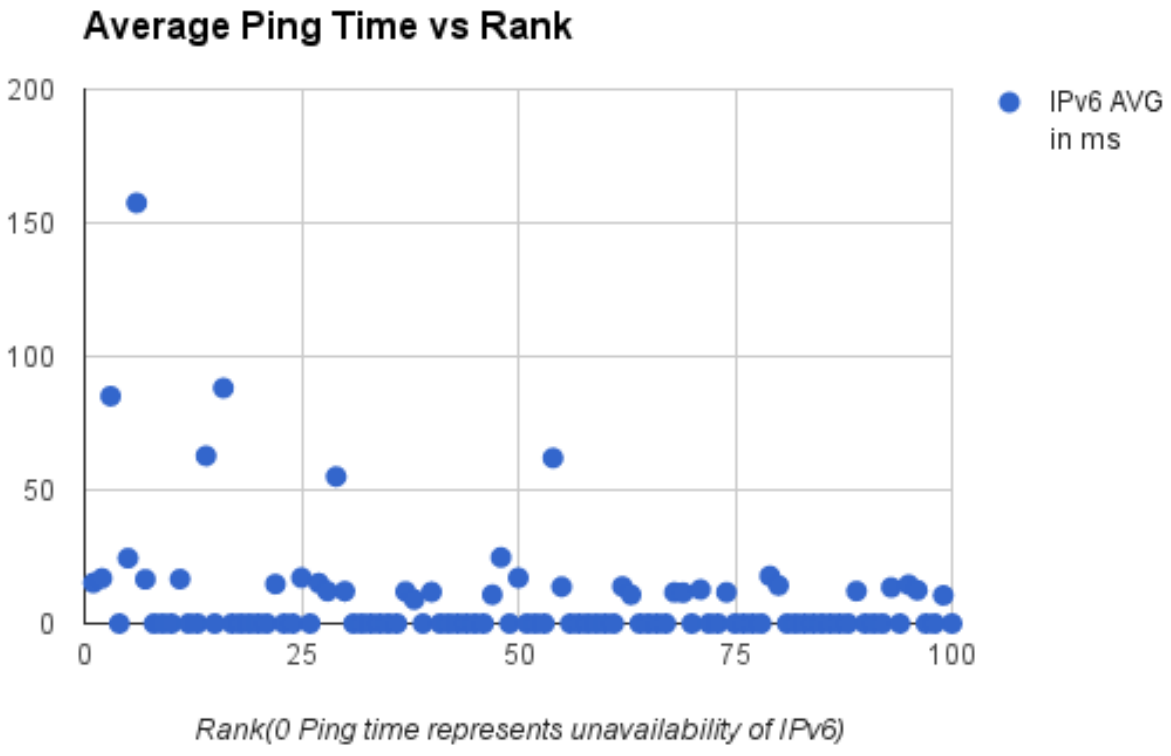
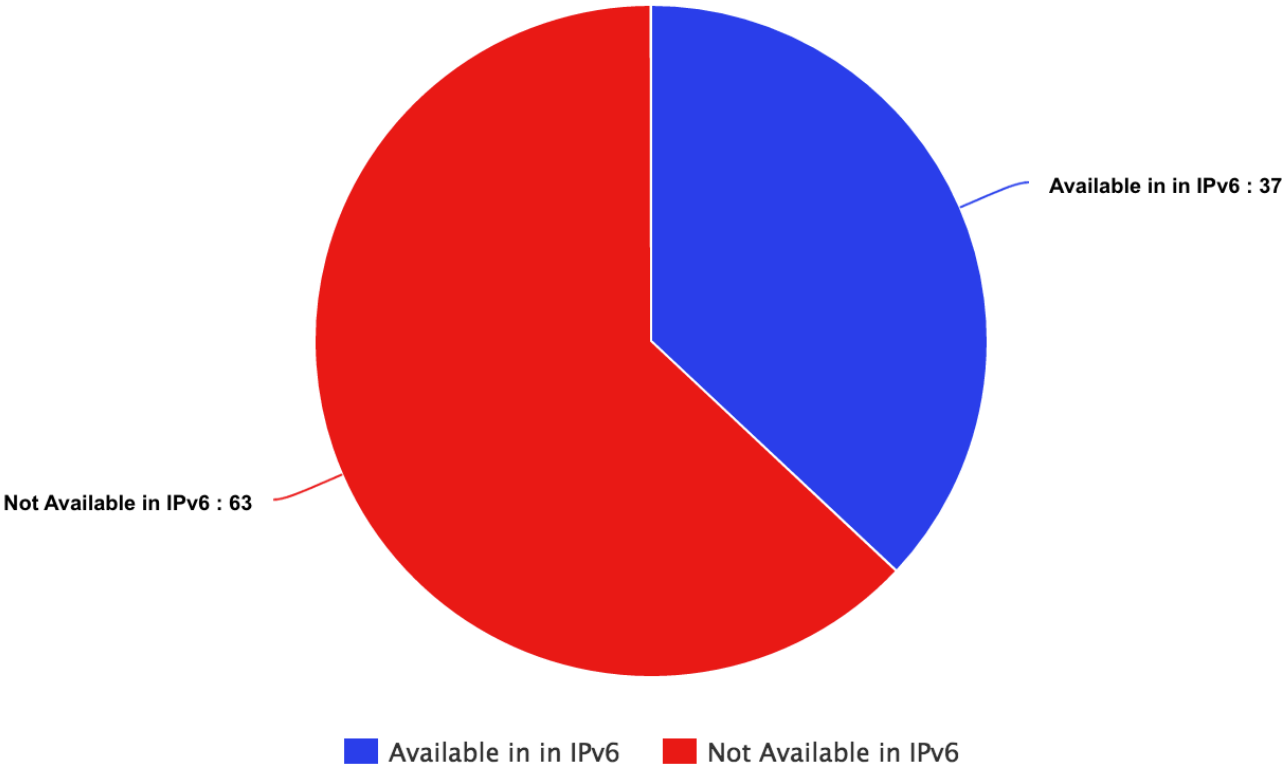


Figure 3 (Average ping time vs Rank in IPv6)



From the above graph(figure 3) it was easy to conclude that IPv6 was easy to conclude that a very few websites had IPv6 available. In fact out of the given sites only 37 websites were shown to have the availability of IPv6.

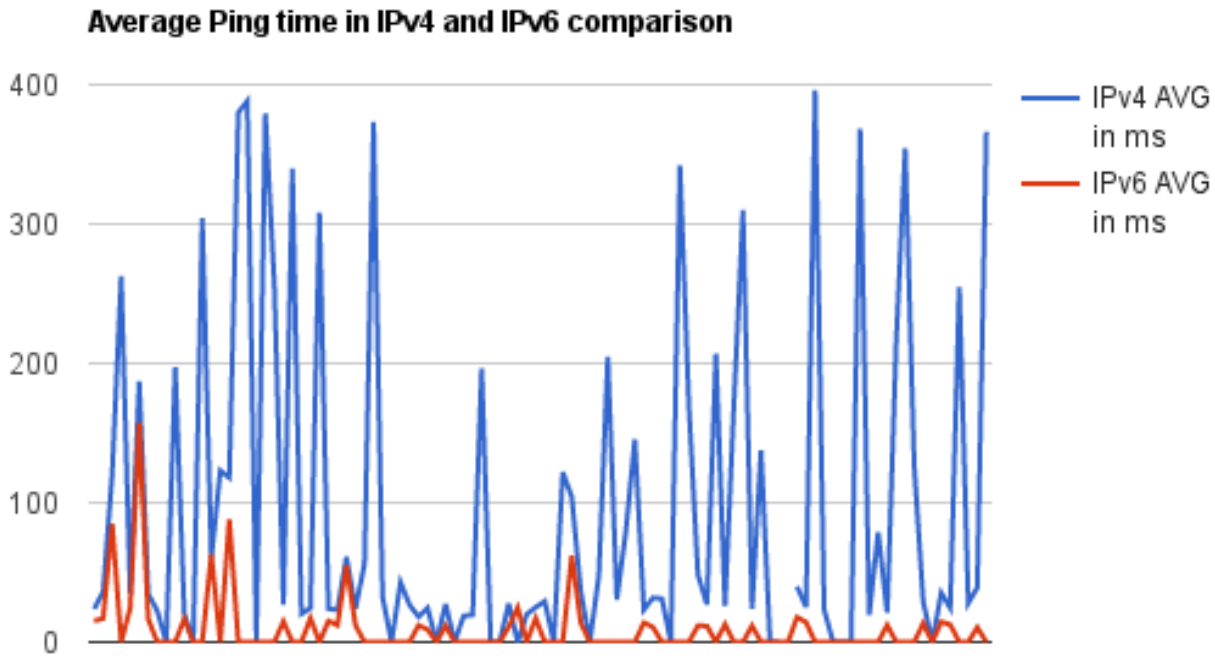


Figure 4 (Average Ping time in IPv4 and IPv6)

From the above graph it was very east to draw the conclusion that IPv6 had a much faster response time than than IPv4 with the. However it was also easy to see that a lot of websites were still only available in IPv4 and did not have the availability of IPv6 yet.

C) Discussion:

Throughout the test, I pinged the websites using a script I developed in python however this approach is not the best way to test the quality of the websites and there could certainly be a lot of improvements made to this approach of testing the performance of a network some of them are given below.

C.1) Potential Improvements to the Test

1. The test currently only shows the minimum time, the maximum time and the average time of the network response. In most of the cases all comparisons have been made on the average time taken by the network to perform. However if the value of the max time goes really high due to an anomaly in the server response it throws off the average value by a lot. To counter this I believe that the mean value should be calculated and used for comparisons .

2. The test does not take into consideration the geographic distance between the user and the website server. The further the server is geographically the more routers the package has to go through hence increasing the ping time taken.To counter this a test has to be performed that takes this into consideration

3. The test carried out pings the server 10 times in quick succession hence there is no significant difference in the traffic of the server. To counter this the test should not be taken in succession but in random times throughout the day

C. 2) Interesting Findings

1. Only 37% of the websites had IPv6 available

2. Some Ping results shows 100% package loss in the result. on investigation it was found out that some websites used firewalls to not respond to PING ICMP packets to avoid DDoS attacks.

3. The googleusercontents.com did not resolve in the ping. On investigation it was found that it has to be provided with a subdominant to be resolved.

4. All google websites had very similar IPv6 address and on investigation it was found out that google resolves to its on server as a result all its IPv6 addresses are similar