COMP2207 Coursework 2

IPv4/IPv6 Performance Measurement

This exercise involved testing network performance using a provided list of the top 100 websites in the world. It gave the opportunity to demonstrate knowledge of using the ping tool to measure performance across protocols as well as display technical scripting skills in order to obtain and analyse data.

A) Gathering the Data

To gather the data I needed, I used the csv file of the top 100 websites from alexa and used scripts to manipulate this file and execute the necessary commands. I chose to write my scripts in Java, although this may not have been the most efficient option, it was the programming language I was most familiar with so felt it would give me the necessary amount of control to execute the task.

I started by writing a PerformanceTest class with the main method taking the csv file as an argument in order that the program could be run from the command line. I split the lines in the file by the comma separating the site and the site rank in order to obtain just the domain part of each line. A second class named PingTest was then responsible for taking the domain as an argument and constructing the ping command with the input and desired protocol ready to execute at runtime. I chose to run the command three times with both IPv4 and IPv6 as I thought this would be sufficient to give a good statistical average of the results. After running the command, the program then calculated an average of the times from the ping data and added them to an array.

In the PingTest class, I used a method writeCsv() to write the array of averages back to a separate csv file depending on the protocol used before compiling both the IPv4 and IPv6 results into one file ready to record the data.

Even though the method I chose was found to be a little time consuming and potentially quite bulky in terms of code, I found that in the end it gave results for all sites and displayed them in a good manner which then made it easier to start going about analysing the data recorded.

I ran my tests from my home network as after initially running on eduroam, I found that there was a significant amount of packet loss, giving a very small number of results. I had to cater for the occurrences where there was no ping data, for example in the case when the domains were not yet able to support IPv6, so implemented a statement checking for if the data read by the program was null.

B) Measurement Results

Rank	Site	Ping with IPv4	Ping with IPv6
1	google.com	5	6
2	youtube.com	12	5
3	facebook.com	6	5
4	baidu.com	188	0
5	wikipedia.org	13	16
6	yahoo.com	82	93
7	google.co.in	5	6
8	twitter.com	6	0
9	amazon.com	79	0
10	qq.com	189	0
11	google.co.jp	5	17
12	live.com	0	0
13	taobao.com	207	0
14	vk.com	56	58
15	instagram.com	79	0
16	linkedin.com	79	81
17	hao123.com	190	0
18	sohu.com	176	0
19	sina.com.cn	0	0
20	360.cn	260	0
21	weibo.com	212	0
22	google.de	6	6
23	tmall.com	209	0
24	reddit.com	6	0
25	google.co.uk	18	5
26	yahoo.co.jp	251	0
27	google.com.br	28	5
28	google.fr	6	5
29	yandex.ru	40	40
30	google.ru	5	5
31	onclickads.net	12	0
32	soso.com	176	0
33	t.co	10	0
34	ebay.com	0	0
35	wordpress.com	6	0
36	bing.com	6	0
37	blogspot.com	6	7
38	google.it	5	6
39	tumblr.com	0	0
40	google.es	6	23
41	msn.com	0	0

42	stackoverflow.com	15	0
43	gmw.cn	5	0
44	aliexpress.com	147	0
45	microsoft.com	0	0
46	apple.com	0	0
47	google.com.mx	6	23
48	netflix.com	0	29
49	pinterest.com	5	0
50	google.ca	18	25
51	imgur.com	6	0
52	paypal.com	0	0
53	imdb.com	80	0
54	mail.ru	70	59
55	google.com.hk	6	5
56	naver.com	0	0
57	popads.net	11	0
58	amazon.co.jp	0	0
59	pornhub.com	12	0
60	ok.ru	55	0
61	github.com	79	0
62	blogger.com	6	6
63	google.co.kr	12	6
64	amazon.in	42	0
65	office.com	0	0
66	cnzz.com	207	0
67	whatsapp.com	115	0
68	google.co.id	6	5
69	google.com.tr	6	7
70	fc2.com	148	0
71	google.com.au	6	6
72	dropbox.com	149	0
73	youku.com	172	0
74	google.com.tw	5	7
75	adobe.com	113	0
76	xinhuanet.com	0	0
77	microsoftonline.com	0	0
78	googleusercontent.com	0	0
79	xhamster.com	10	10
80	google.pl	5	5
81	tianya.cn	242	0
82	cnn.com	10	0
83	xvideos.com	0	0
84	twitch.tv	0	0
85	amazon.de	0	0

86	pixnet.net	294	0
87	wikia.com	6	0
88	people.com.cn	28	0
89	adf.ly	6	6
90	alibaba.com	164	0
91	coccoc.com	279	0
92	craigslist.org	79	0
93	google.com.eg	6	5
94	quora.com	0	0
95	diply.com	23	7
96	google.com.ar	7	6
97	so.com	194	0
98	bbc.com	6	0
99	google.com.sa	6	19
100	360.com	124	0

Figure 1: Tabulated data showing IPv4 and IPv6 ping time averages for each site

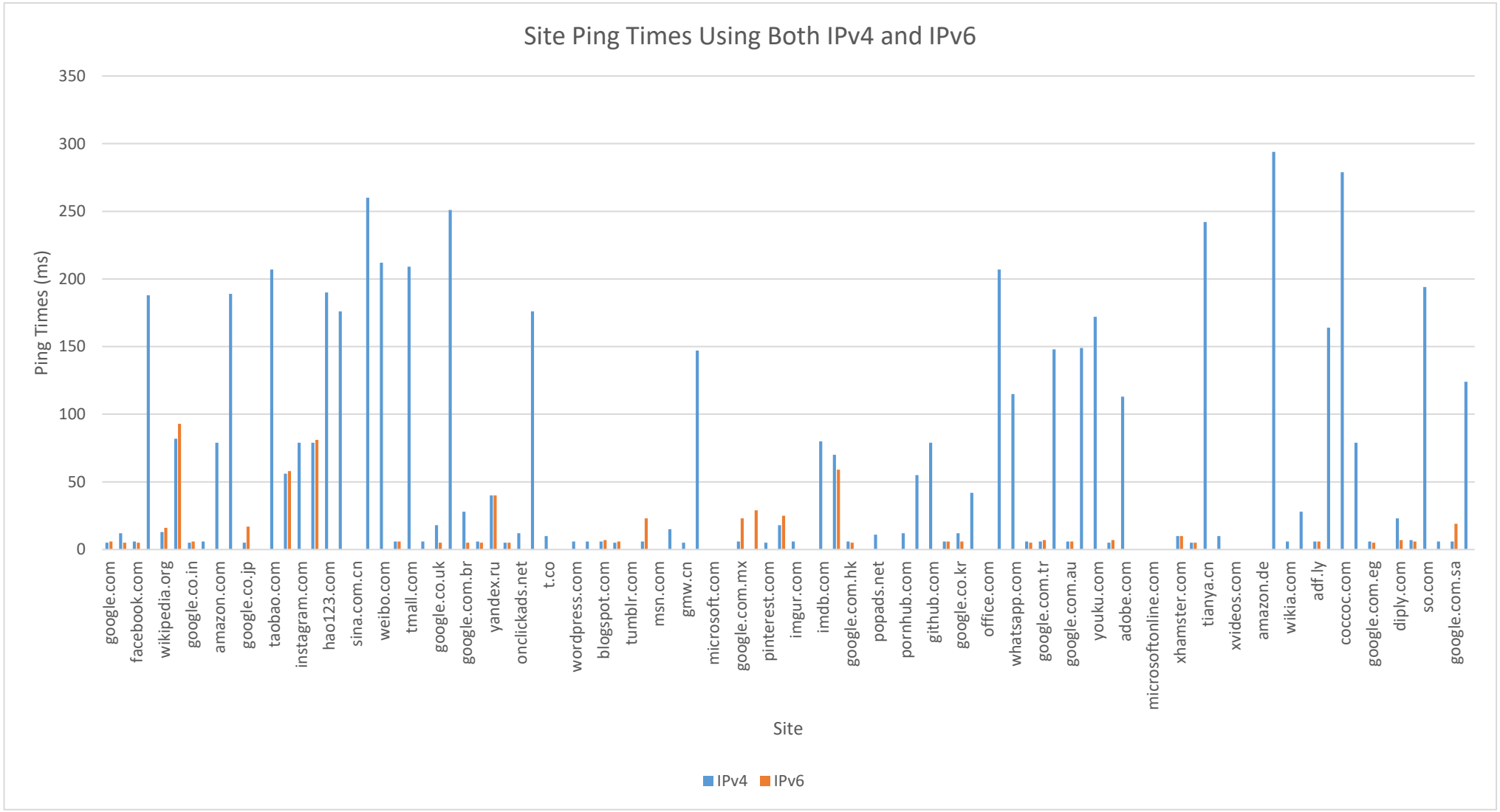

Figure 2: Bar chart comparing the IPv4 and IPv6 ping times of each site

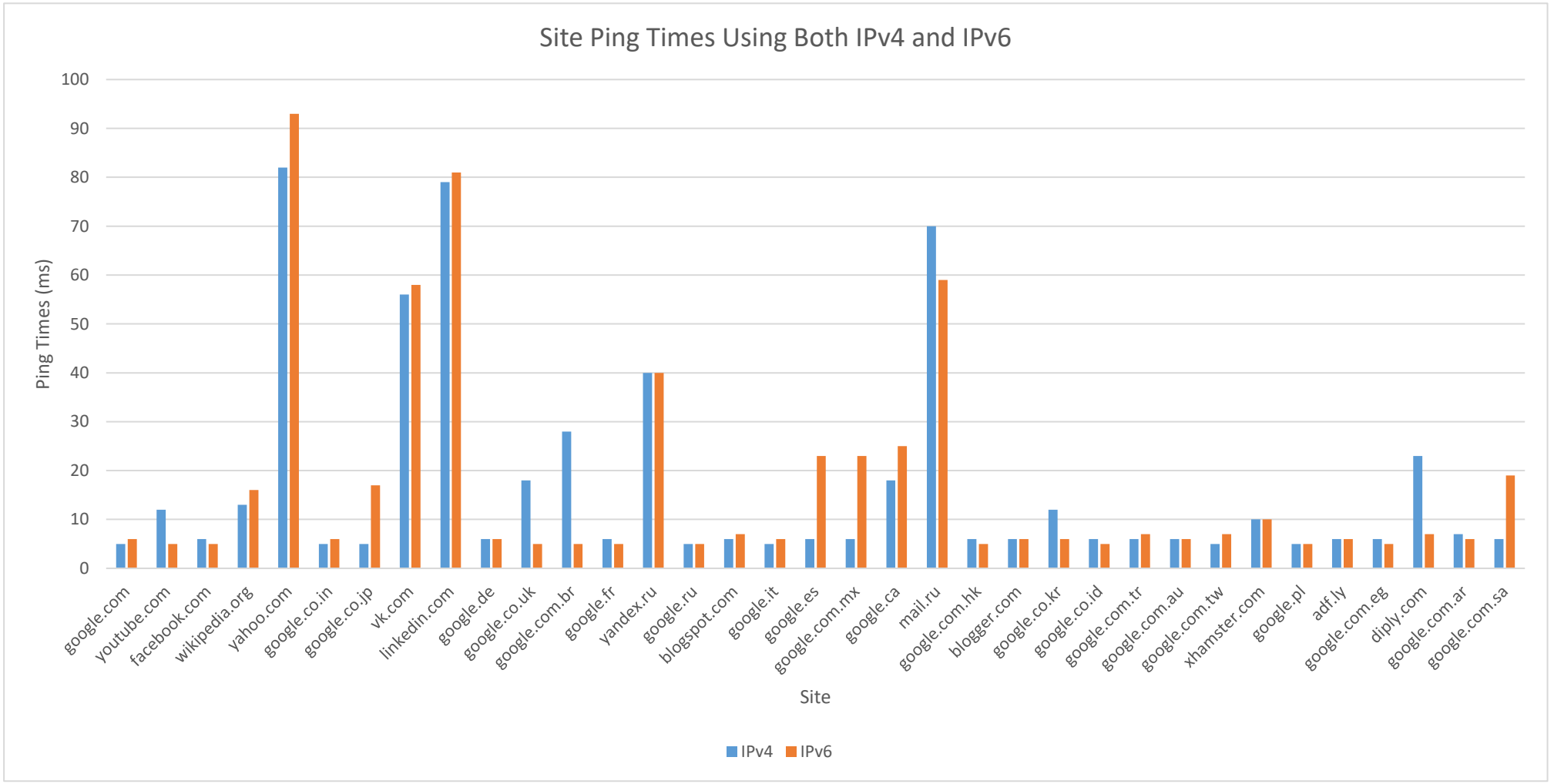

Figure 3: Bar chart comparing the IPv4 and IPv6 ping times of each site, only displaying results with data from both tests

Figure 4: Pie charts comparing the percentage of sites that were able to be pinged

Standard Deviation of IPv4 Results:

$\sigma = 80.86$ (2 d.p.)

Standard Deviation of IPv6 Results:

$\sigma = 15.59$ (2 d.p.)

c) Discussion

After recording all the data gathered from the ping tests on the site list, shown in Figure 1, I planned on how I was going to graph the results in the most effective manner possible in order to make it simple for me to draw conclusions from the visual representations. Firstly, I noticed there were many sites that did not return data when pinged using IPv6, this could have been either due to security reasons or the fact that the particular site was not IPv6-enabled. This is shown in Figure 2 where there are many bars representing times for the IPv4 data but a severe lack of bars showing IPv6, this is what lead me to produce the second bar chart in Figure 3 showing only the sites that could be pinged using both protocols to give a clearer comparison between the two.

From figure 3, I could conclude that there wasn't a significant amount of difference in the performance for the two protocols. The sites with the highest IPv4 ping times were also the ones with the highest times with IPv6, this was also the case for the sites with the lowest times. Another observation I made was the fact that from the total set of one hundred sites, only 35 responded with both protocols, a significant proportion of which belonged to Google meaning the data returned could have potentially been biased. If there had been a much larger initial set of sites, this would have most likely made the data more reliable. When calculating the average times from the data excluding any sites with 0 values, the IPv6 average was only slightly higher than the IPv4 average showing the sites gave consistent responses to both pings.

After comparing the times using the bar charts, I decided to use pie charts, shown in Figure 4, to compare the general response from the sites when being pinged with each protocol. The charts gave a much clearer indication that the percentage of sites that could be pinged using IPv4 was much higher than that of IPv6 with 64 out of the 100 sites not supporting IPv6. When looking at Figure 2, it shows these sites had particularly high ping times with IPv4 suggesting that should these sites have responded, they would have had a similarly long length response time.

Furthermore, I calculated the standard deviations of both sets of results in order to be able to carry out a comparison of the measure of spread in each set. I discovered that there was considerably more variation with the IPv4 results than for the IPv6, this is most likely due to the factor of there being more data available for the calculation whereas the majority of the numbers in the IPv6 set were 0 values. If there had been more sites in the initial data that had responded to the IPv6 pings then the standard deviation values would have turned out to be largely similar.