CDNlyzer - A comprehensive measurement study of CDN usage for Alexa Top 200 websites

CSE 534 : Fundamentals of Computer Networks

Bhavesh Goyal
111482386
Stony Brook University
bgoyal@cs.stonybrook.edu

Rahul Sihag
111462160
Stony Brook University
rsihag@cs.stonybrook.edu

Abstract

Content Delivery Networks or Content Distribution Networks (CDNs) are a mechanism to distribute service spatially relative to end-users to provide high availability and high performance. CDNs serve a large portion of the Internet content today, including web objects (text, graphics and scripts), download-able objects (media files, software, documents), applications (e-commerce, portals), live streaming media, on-demand streaming media, and social networks.
Today CDNs are an inevitable solution for most websites looking for performance improvement and they have evolved as an entire layer in the internet ecosystem. Content owners such as media companies and e-commerce vendors pay hefty amounts to CDN operators for delivering their content to end users. Thus it is important for us to understand how the websites make use of this next-level optimization system. In this paper we discuss and analyze the CDN properties and traffic for Alexa top 200 websites. The compete code and results are present on our Github page. URL - `https://github.com/bhaveshgoyal/cdnlyzer`.

1 Introduction

A content delivery network (CDN) refers to a geographically distributed group of servers which work together to provide fast delivery of Internet content. A CDN allows for the quick transfer of assets needed for loading Internet content including HTML pages, javascript files, stylesheets, images, and videos. The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs, including traffic from major sites like Youtube, Facebook, Netflix, and Amazon.

While there are many CDN providers and majority of the internet traffic today is served by CDN, little work has been done on the extent to which CDNs are being used by individual websites and their impact on the performance benefits. Our work provides a comprehensive study analyzing today's CDNs as used under corporate environments and seeks to bridge this gap of insufficient analysis by gathering a extensive amount of data related to CDN usage by Alexa top 200 websites. Our contribution to this subject addresses:

1. What CDN techniques are employed and how the use of CDN techniques influence performance?
2. What is the extent of CDN usgae by the origin server sites
3. What is the nature of content served by the CDNs for the origin servers
4. Analyzing frequency at which a CDN serves an origin site

The format of the paper is organized as follows: Section 2 provides the CDN techniques and background on studied CDN redirection. Section 3 examines the usage of CDN in today's internet as per our study. Section 4 describes the performance improvements gained by using CDNs. Section 5 describes our methodology and origin level study technique. Section 6 summarizes the results of our analysis. Section 7 summarizes the concluding remarks and Section 8 describes the future work for our study.

2 CDN Redirection techniques

When the browser makes a DNS request for a domain name that is handled by a CDN, there is a slightly different process than with small, one-IP sites. The server handling DNS requests for the domain name looks at the incoming request to determine the best set of servers to handle it. At its simplest, the DNS server does a geographic lookup based on the DNS resolvers IP address and then returns an IP address for an edge server that is physically closest to that area. Thus if someone makes a request from the EastCoast to the DNS resolver, they would be routed to the closest EastCoast server(For instance, Virginia for Amazon's server). But, if we make the same request through a DNS resolver in West Coast(California), we would be redirected to an IP server on the West(Seattle for Amazon). But sometimes, we might not end up with a DNS resolver in the same geographic location from where you are making the request for other base servers who aren't using CDN at their base.

That's the first step of the process. But we also have to keep in mind that companies may optimize their CDNs in other ways as well, for instance, redirecting to a server that is cheaper to run or one that is sitting idle while another is almost at capacity. In any case, the CDN smartly returns the best possible IP address to handle the request.

2.1 CDN Hosting techniques

There exist two methods for redirecting client requests to a particular CDN server in a distributed network of content providers for fetching content for a webserver.

1. **DNS redirection** - In DNS redirection the authoritative DNS name server is controlled by the CDN. When the authoritative DNS server receives a DNS request from client it redirects the request by resolving the CDN server name to the IP address of one content server. The resolution is done on the basis of a number of factors such as the availability of resources, network conditions and proximity of server to the client. The reply has a TTL field that is used for load balancing among different servers. There are two different DNS redirection techniques: full site redirection (example is amazon.com being redirected to its CDN Amazon CloudFront) and partial site redirection (example is flipkart.com landing page). The origin server is hidden except to the CDN in case of full site redirection. All the requests for the origin server are redirected to the CDN and CDN serves the content. In case of partial site redirection the base url is served by the origin server and the embedded URLs for objects are modified to be served by the CDN.
2. **URL rewriting** - In URL rewriting the origin server rewrites URL links as part of dynamically generating pages to redirect clients to different content servers.

3 CDN Usage overview

The first part of our study examines how CDNs are used today to server the web content for Alexa Top 200 Websites. We analyzed the home page and all the embedded content of Alexa top 200 websites and found out that 74.9% of the Alexa top 200 websites have their base page served by a CDN. We also propose a DNS + Ping Response time based hybrid approach to arrive at these results. We make a DNS query to the base page and check if its a CNAME record pointing to a CDN. For DNS lookup we used the dig (Domain Information Groper) utility that looks for a CDN provider as the authoritative server. We have a list of top CDN providers to verify the match. We also check the average ping response time by pinging the origin server and the CDN server (if any). If it is a CDN, in most of the cases origin server will have a higher ping response time than the CDN server. We found out that on an average ping response time for CDN server is 31.13 ms less than the origin server. In terms of total bytes content in the entire page[Image + CSS + JS from (CDNs + Non-CDNs)], 47.6% were found to be the images bytes, about 4.88% of CSS bytes, 29% of the JS bytes. Extending our analysis to

CDN, we found that out of all the bytes served from CDN, images comprises of 58% of the total content, CSS comprises of 5.94% and JS about 36.06% of net CDN content.

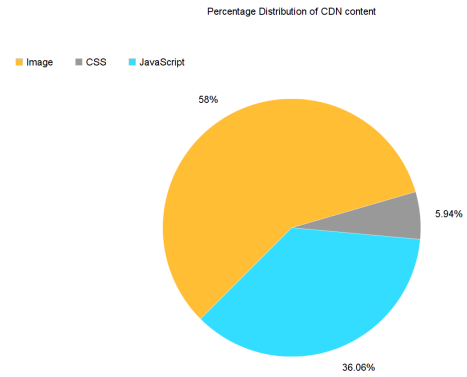

Figure 1: % distribution of CDN content

4 Performance study overview

The performance of CDNs can be measured in many ways. Following are some of the ways we calculated the performance of CDNs.

1. How many requests are offloaded from the origin servers

2. What is the impact of CDNs on client latency

3. Ability to load balance CDN requests

On an average, in terms of number of hops, distance to the CDN server is 65.14 hops whereas the distance to origin server is 78.21 hops, in terms of number of miles distance to the CDN servers is 2368.59 miles whereas distance to the CDN servers is 1838.06 miles. Average ping response time to origin server is 82.65 ms whereas the same for CDN server is 48.38 ms. Therefore average ping difference time between origin and CDN server is 34.26 ms.

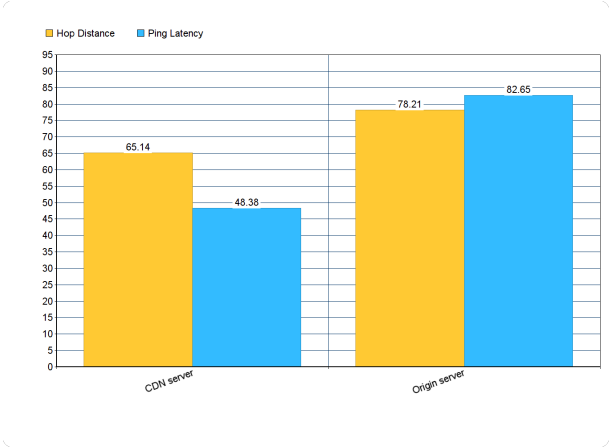

Figure 2: Average Hop Distance between CDN and origin server and Ping latency

For load balancing CDNs assign small DNS TTLs for the IP addresses and as a result client have to do frequent DNS lookups. And it helps CDNs determine which server they want the client to use. We observer an average TTL for a CDN server as per DNS record to be 50.03 seconds. The graph of TTL values for Alexa Top 20 sites is shown below. We can see that most of the TTL values are 60 seconds.

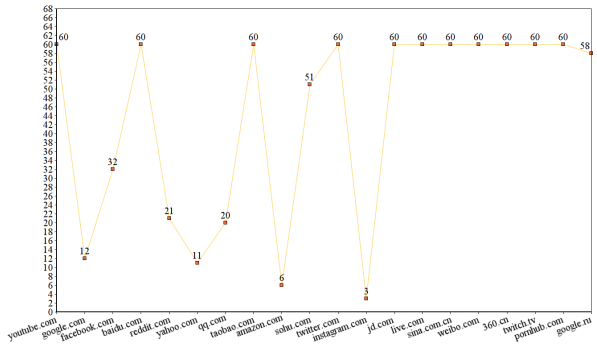

Figure 3: DNS TTL values for Alexa Top 20 sites

5 Methodology

5.1 Build Environment

We used two Amazon EC2 servers geographically distributed (East and West zones). The list of technologies/libraries used are as follows:

1. **Selenium drivers** - We could not use simple curl based approach to measure the time taken to load page when first byte of data is received. Existence of many third party links and resources require that we perform a real time rendering of webpages to estimate the

page load times.

2. **Mitmproxy-v3** - We used mitmproxy sitting in front of selenium browser. The proxy gathers all the requests made by a webpage and also captures the responses returned by server for latter analysis.

3. **BeautifulSoup** - To parse response from a third party API from cdnplanet and verify against our own results to detect CDN usage at origin.

4. **Geopy** - This library is used for IP location determination. We get the location coordinates (latitude, longitude) from an IP address and use these coordinates to find the distance in miles between two IP addresses.

5. **Pyping** - This library is used for fabricating ICMP messages to find the Ping response time from client to server. The library is also used for traceroute to determine the number of hops between client and server. Currently for our study we have a maximum hop count limit of 128.

6. **Python** - All the coding and experimental testing has been done in Python 2/3.6.

5.2 Origin Level Study Technique

First we identify Top Popular CDNs used by Alexa 200. For this, we made a series of DNS CNAME queries to websites and tried to map the hostname of the CNAME response to the CDN used. This domain-name to CDN mapping is submitted as part of our code which is used to identify popular CDNs. We also verified our responses and extended this approach to include some more popular CDNs using a third party API from CDNFinder. We queried their webpage for a given to ask for CDN and parse the response returned using BeautifulSoup to verify our response with theirs and include some domain names, if not in our popular domain mapping.

One of the challenges in this approach was identifying if the site use CDN for 'Non-popular' domains. For instance, Facebook recently switched from Akamai to its own CDN at domain '.facebook' which was not in our map. To overcome this challenge, we used a Hybrid DNS + Ping based approach. First we identified which of the other servers had CNAME records present at the origin. After identifying such records, we compare the response time of base domain and CNAME domain by pinging each of the domains separately. If we saw that the average ping response time differs substantially, we classify the CNAME as a CDN domain. Next, we performed CDN level analysis for each of the CDN domains obtained.

For each request to base domain of a server, CDN locates a server which is closest to the client to reduce the latency associated. The notion of closeness could be a distance metric, number of hops metric or ping response time metric. Thus in our study we determined the CDN server presence using the following three parameters:

1. **Geographically** - We used a distance based approach using Geoips to find the distance of origin server and CDN server by determining the location coordinates from IP addresses.

2. **Topologically** - We analyzed the traceroute paths (upto a maximum of 128 hops) to origin server and CDN server and compared hop counts of both.

3. **Latency** - We pinged the origin and CDN servers and calculated the average response times for 10 ICMP echo reply packets.

We also determined the CDN server update frequency using the average TTL in DNS response.

5.3 Site Level Study Technique

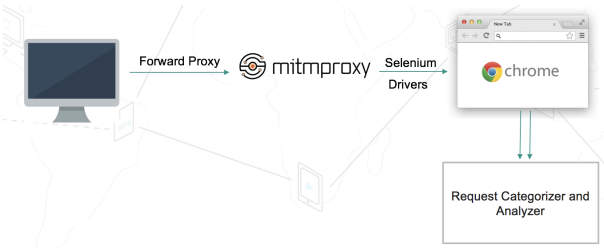

Figure 4: Site-Content Level Analysis Setup

To accurately perform a study over objects used by a web-server, we used Selenium drivers to develop a tool that initiates a Chrome instance behind mitmproxy(forward mode). When a webpage is loaded, we log all the requests and response to third party objects in a log file and analyze the log file for CDN usage.

Percentage of Alexa 200 sites using Popular CDNs for serving (Top 5)	
Google	32.4%
Akamai	10.8%
Amazon CloudFront	6.3%
Fastly	8.1%
CloudFare	1.8%

Table 1: Top popular CDNs usages. Note: We also observe that Google as CDN was found to be used highest since there were many Google domains in Top 200.

		Curr IP	Server IP	CDN IP	Trc Server	Trc CDN	Trc Diff.	Ping Server	Ping CDN	Ping Diff	CDN?	Hops Server	Hops CDN	Distance Server	Distance CDN	Distance Diff.	CDN TTL
2	youtube.com	13.57.229.187	216.58.194.206	216.58.192.14	10170.38941	10217.58747	-47.19805717	2.549	2.631	-0.082	Yes	16	16	10.68649768	10.68649768	0	60
3	google.com	13.57.229.187	216.58.194.174	172.217.3.196	10144.61517	10508.00943	-363.3942604	2.585	2.646	-0.061	Yes	16	21	10.68649768	10.68649768	0	12
4	facebook.com	13.57.229.187	31.13.76.68	31.13.70.36	352.6990414	380.6321621	-27.93312073	20.635	10.732	9.903	Yes	21	20	709.047468	305.4771129	403.5703551	32
5	baidu.com	13.57.229.187	220.181.57.216	104.193.88.77	531322.3097	565238.9705	-33916.66079	190.752	2.399	188.353	Yes	128	128	5959.945449	1318.353501	4641.591948	60
6	wikipedia.org	13.57.229.187	198.35.26.96	198.35.26.96	580107.0085	580280.2763	-173.2678413	3.219	3.105	0.114	No	128	128	41.58486272	41.58486272	0	60
7	reddit.com	13.57.229.187	151.101.65.140	151.101.189.140	580110.0585	580110.894	-0.835418701	2.16	1.977	0.183	Yes	128	128	40.52738652	0	40.52738652	21
8	yahoo.com	13.57.229.187	98.137.246.8	98.137.246.8	748.8634586	1168.045521	-419.1820621	54.615	32.359	22.256	Yes	19	19	8.7182104	8.7182104	0	11
9	qq.com	13.57.229.187	111.161.64.40	184.25.56.124	544166.9381	170.7653999	543996.1727	349.332	4.669	344.663	Yes	128	14	5971.075771	305.4771129	5665.598658	20
10	taobao.com	13.57.229.187	140.205.220.96	66.102.255.57	538996.1298	535799.607	3196.522713	270.626	10.353	260.273	Yes	128	128	6293.640162	2402.146223	3891.493939	60
11	amazon.com	13.57.229.187	176.32.98.166	54.192.141.59	486638.695	15904.96826	470733.7267	69.172	6.817	62.355	Yes	128	17	2396.50442	709.3337727	1687.170647	6
12	sohu.com	13.57.229.187	221.179.178.112	104.254.66.16	543008.1787	562831.4366	-19823.25792	253.355	3.199	250.156	Yes	128	128	6322.673697	1318.353501	5004.320197	51
13	tmall.com	13.57.229.187	140.205.94.193	66.102.255.18	531552.2785	535659.4038	-4107.125282	314.228	9.942	304.286	Yes	128	128	6293.640162	2402.146223	3891.493939	60
14	google.co.in	13.57.229.187	216.58.194.163	74.125.28.94	10201.48492	50352.40912	-40150.92421	2.549	24.207	-21.658	No	16	28	10.68649768	10.68649768	0	60
15	vk.com	13.57.229.187	87.240.129.189	87.240.129.72	363029.485	381958.4947	-18929.00968	194.531	194.547	-0.016	No	128	128	5908.564893	5908.564893	0	60
16	twitter.com	13.57.229.187	104.244.42.129	104.244.42.129	7907.770872	8404.667616	-496.8967438	5.226	5.545	-0.319	Yes	15	15	41.56604183	41.56604183	0	60
17	instagram.com	13.57.229.187	34.227.141.166	157.240.11.174	501847.0428	460.9246254	501386.1182	72.888	10.932	61.956	Yes	128	20	2396.50442	17.8193834	2378.685036	3
18	jd.com	13.57.229.187	106.39.167.118	192.229.173.173	527365.2687	943.8056946	526421.463	265.394	74.537	190.857	Yes	128	18	5959.945449	1318.353501	4641.591948	60
19	live.com	13.57.229.187	204.79.197.212	204.79.197.212	575120.0547	575125.1712	-5.116462708	2.717	2.372	0.345	Yes	128	128	713.8936637	713.8936637	0	60
20	sina.com.cn	13.57.229.187	202.108.33.107	36.51.254.37	14478.13559	8277.822733	6200.312853	399.808	191.514	208.294	Yes	25	23	5959.945449	5959.945449	0	60
21	weibo.com	13.57.229.187	123.125.104.197	123.125.104.197	24477.07987	7638.425589	16838.65428	274.526	275.316	-0.79	Yes	25	25	5959.945449	5959.945449	0	60
22	360.cn	13.57.229.187	101.198.193.22	101.198.193.22	17425.071	15384.54771	2040.523291	9.322	9.354	-0.032	Yes	22	22	5959.945449	5959.945449	0	60
23	yandex.ru	13.57.229.187	77.88.55.80	77.88.55.77	12274.33372	12087.0738	187.2599125	179.455	181.633	-2.178	No	32	32	5908.564893	5908.564893	0	60
24	twitch.tv	13.57.229.187	151.101.2.167	151.101.194.167	580117.5079	580114.5515	2.956390381	2.12	2	0.12	Yes	128	128	40.52738652	40.52738652	0	60
25	pornhub.com	13.57.229.187	216.18.168.16	216.18.168.16	565453.656	565304.6501	149.0058899	16.5	11.686	4.814	Yes	128	128	2674.28508	2674.28508	0	60
26	google.ru	13.57.229.187	172.217.0.35	74.125.197.94	235.7430458	45450.57678	-45214.83374	2.656	24.071	-21.415	No	16	27	10.68649768	10.68649768	0	58
27	google.co.uk	13.57.229.187	216.58.194.163	74.125.28.94	10234.36975	50367.34056	-40132.97081	2.692	22.652	-19.96	No	16	28	10.68649768	10.68649768	0	60
28	google.co.jp	13.57.229.187	172.217.0.35	172.217.0.35	246.2882996	646.8009949	-400.5126953	2.681	2.626	0.055	Yes	16	16	10.68649768	10.68649768	0	60
29	google.com.br	13.57.229.187	172.217.6.35	74.125.197.94	203.5467625	45442.64936	-45239.1026	2.718	22.535	-19.817	No	16	27	10.68649768	10.68649768	0	2
30	google.de	13.57.229.187	216.58.194.163	216.58.194.195	10814.99195	10193.73083	621.2611198	2.557	2.695	-0.138	Yes	16	16	10.68649768	10.68649768	0	60
31	google.com.hk	13.57.229.187	74.125.28.94	172.217.0.35	50370.78691	186.3541603	50184.43274	24.127	2.6	21.527	Yes	28	16	10.68649768	10.68649768	0	60
32	google.fr	13.57.229.187	216.58.194.163	216.58.216.131	10236.85479	10517.80772	-280.9529305	2.589	24.459	-21.87	No	16	21	10.68649768	10.68649768	0	60
33	yahoo.co.jp	13.57.229.187	182.22.59.229	183.79.249.252	551665.3309	561131.2401	-9465.909243	119.345	107.126	12.219	Yes	128	128	5189.106933	5334.996118	-145.8891848	60
34	bing.com	13.57.229.187	13.107.21.200	204.79.197.200	575121.4607	575251.4417	-129.981041	2.494	2.405	0.089	Yes	128	128	713.8936637	713.8936637	0	50
35	alipay.com	13.57.229.187	110.75.139.5	110.75.245.23	538300.6845	524059.2663	14241.41812	288.145	384.408	-96.263	No	128	128	6293.640162	6293.640162	0	60

Figure 5: Snapshot of Origin Level Analysis done on Alexa Top 200. The attributes compare the performace in terms of 3 parameters, Ping Response, GeoLocation and Traceroute, and estimates average CDN Frequency update through domain TTLs.

- We use string matching to identify object extension from the logged requests to categorize a link as a request for image, css, JS. We support the following image extensions as part of our study. (.jpeg, .jpg, .gif, .png, .svg). We use the results to identify percentage of objects hosted on a website. Next, we identify if the link is hosted as static CDN or dynamic DNS based method.

- Next for each of the link category urls we use url matching with CDN domain names to check if the link is hosted through static CDN. If not, we use our dynamic DNS + Ping based approach to identify if the link points to a CDN server and is hosted through DNS based method.

- Along with identifying the methodology of the hosted links, we also make an independent request to each of the links to fetch the response bytes and separately log the contents categorizing them into image, CSS, JS bytes coming from the CDN. Note, we also analyzed the total response bytes coming through mitmproxy to compare the category wise response bytes as a percentage measure.

6 Results

6.1 CDN Usage

As seen from Table-1, we identified the top 5 CDNs used amongst the Alexa 200 websites. These were ranked as follows:

- Google
- Akamai
- Amazon CloudFront
- Fastly
- CloudFlare

	Site Name	Total Bytes	Image Bytes	Image Percentage	CSS Bytes	CSS Percentage	JS Bytes	JS Percentage	Total Percentage
2	1688.com_res	3882006	2221839	0.57234301	62068	0.015988641	1340271	0.345252171	0.933583822
3	360.cn_res	2927449	2805879	0.958472376	0	0	119971	0.040981414	0.999453791
4	adobe.com_res	397360	0	0	0	0	7561	0.019028085	0.019028085
5	alibaba.com_res	631331	591937	0.93760167	0	0	35805	0.056713515	0.994315185
6	aliexpress.com_res	1931512	1545586	0.800194873	40012	0.020715377	299612	0.155117856	0.976028106
7	alipay.com_res	785826	691127	0.879491134	0	0	93100	0.118474064	0.997965198
8	amazon.co.jp_res	7963945	6328538	0.794648632	391757	0.049191324	747250	0.093829126	0.937669082
9	amazon.co.uk_res	9691895	8124685	0.838296845	391857	0.040431412	747485	0.077124752	0.955853009
10	amazon.com_res	8310709	6373295	0.766877411	413370	0.049739439	932286	0.112178877	0.928795726
11	amazon.de_res	8919109	7167608	0.80362377	391906	0.043940039	747380	0.083795366	0.931359175
12	amazon.in_res	7307793	5575689	0.762978508	394433	0.053974298	758444	0.103785644	0.92073845
13	amazonaws.com_res	9789284	2239532	0.228773831	390668	0.03990772	702114	0.071722712	0.340404262
14	aparat.com_res	3659264	3216488	0.878998618	0	0	438901	0.119942426	0.998941044
15	apple.com_res	1599	0	0	0	0	0	0	0
16	ask.com_res	24989	0	0	0	0	23390	0.936011845	0.936011845
17	avito.ru_res	1093453	504581	0.461456505	0	0	405855	0.371168217	0.832624722
18	babytree.com_res	1634	35	0.021419829	0	0	0	0	0.021419829
19	baidu.com_res	1599	0	0	0	0	0	0	0
20	bbc.co.uk_res	1786165	509647	0.285330303	28495	0.015953173	850127	0.47595099	0.777234466
21	bbc.com_res	570237	209799	0.367915446	335	0.000587475	272176	0.477303297	0.845806217
22	bing.com_res	50991	6196	0.121511639	0	0	0	0	0.121511639
23	bles.com_res	2098846	71242	0.033943415	57072	0.027192086	610285	0.290771691	0.351907191
24	bongacams.com_res	2901960	815276	0.280939779	433052	0.149227419	1651614	0.569137411	0.999304608
25	booking.com_res	4309439	3348188	0.776942892	309689	0.07186295	546042	0.126708372	0.975514214
26	bukalapak.com_res	4249920	1157356	0.272324185	1138577	0.267905514	1803792	0.424429636	0.964659335
27	buzzfeed.com_res	3356677	2447086	0.729020397	10816	0.003222234	714261	0.212788123	0.945030755
28	caijing.com.cn_res	17115	35	0.00204499	0	0	14353	0.838621093	0.840666082
29	chase.com_res	1599	0	0	0	0	0	0	0
30	chaturbate.com_res	1630500	1199661	0.735762649	70181	0.043042625	144805	0.088810181	0.867615455
31	china.com.cn_res	1599	0	0	0	0	0	0	0
32	chinadaily.com.cn_res	1599	0	0	0	0	0	0	0
33	cnet.com_res	3844124	994716	0.258762725	70227	0.018268661	2259260	0.587717774	0.86474916
34	cnn.com_res	3408619	1235035	0.362327089	170145	0.04991611	1492318	0.437807218	0.850050416
35	coccoc.com_res	1051601	2691	0.002558955	139899	0.133034297	816242	0.776189829	0.911783081

Figure 6: Snapshot of Percentage bytes in site coming through CDN for Alexa Top 200. The analysis was done by fetching all the links from the websites. Since we have the percentage bytes, we can fetch the throughput in terms of bytes/sec using the average Ping Response times from Figure-5. Since we know, average ping response times for CDNs are lesser, this implies more throughput to fetch content through CDNs than origins for the same number of bytes in webserver.

	Site	Percentage images	Static CDN Images	Dynamic CDN Images	Percentage CSS	Static CDN CSS	Dynamic CDN CSS	Percentage JS	Static CDN JS	Dynamic CDN JS
2	google.com	31.25	80	20	0			0		
3	youtube.com	32	0	100	5.333333333	50	50	18.66666667	35.71428571	64.28571429
4	facebook.com	29.41176471	0	100	0			38.23529412	0	100
5	baidu.com	50	0	100	0			10	0	100
6	wikipedia.org	37.5	0	100	0			25	0	100
7	reddit.com	58	0	96.55172414	0			26	3.846153846	96.15384615
8	yahoo.com	13.65187713	40	47.5	4.778156997	100	0	35.49488055	82.69230769	17.30769231
9	qq.com	51.83486239	0	74.33628319	0.917431193	0	100	11.9266055	0	88.46153846
10	taobao.com	60.78431373	0	100	2.941176471	0	100	10.78431373	0	100
11	amazon.com	72.96296296	0	100	2.592592593	0	100	9.259259259	12	76
12	twitter.com	0			15	0	100	30	0	100
13	google.co.in	26.66666667	75	25	0			0		
14	tmall.com	22.22222222	0	100	4.444444444	0	100	33.33333333	0	100
15	sohu.com	22.53521127	0	97.91666667	0.234741784	0	100	8.215962441	0	88.57142857
16	live.com	55	0	100	0			20	25	50
17	instagram.com	40.90909091	0	100	0			36.36363636	0	100
18	vk.com	22.44897959	0	100	16.32653061	0	100	34.69387755	0	100
19	jd.com	52.08333333	0	100	0			12.5	0	100
20	weibo.com	44.33962264	0	100	6.132075472	0	92.30769231	25.94339623	25.45454545	74.54545455
21	sina.com.cn	48.60499266	0	78.24773414	0.440528634	0	100	10.42584435	0	74.64788732
22	yandex.ru	27.77777778	0	100	5.555555556	0	100	13.88888889	0	100
23	360.cn	92.19858156	0	100	0			2.127659574	0	100
24	google.co.uk	26.66666667	75	25	0			0		
25	google.co.jp	26.66666667	75	25	0			0		
26	linkedin.com	0			0			10.25641026	0	100
27	google.com.hk	27.77777778	80	20	0			0		
28	google.ru	26.66666667	75	25	0			0		
29	google.com.br	26.66666667	75	25	0			0		
30	pornhub.com	41.17647059	0	100	5.882352941	0	100	28.23529412	0	100
31	google.de	26.66666667	75	25	0			0		
32	netflix.com	25	0	0	0			15	0	0
33	twitch.tv	12.5	0	100	4.166666667	0	100	20.83333333	0	100
34	google.fr	31.25	80	20	0			0		
35	office.com	4.166666667	0	0	12.5	0	0	25	0	0

Figure 7: Snapshot of percentage objects coming through CDN for Alexa Top 200. As seen, most websites prefer serving static images at CDNs that other object types. Also since static urls increase website management costs, most websites were seen to be using Dynamic CDN hosting, while older servers such as yahoo appeared to be using hosting method in approx 1:1 ratio.

Amongst the Top 200, we identified that almost 74.9% of origin websites (average over both zones) use CDN at their base server to redirect queries to appropriate servers. Amongst these, 68.55% of origins were seen to be using CDNs in the East Region, while 81.25% of sites were seen to be using CDN when repeated the experiment from West Region.

The KS-statistics report that for the ping distributions tested, 77% of origins(almost 154 of 200) rejected our NULL hypothesis meaning that they indeed use CDNs at the origin servers.

Please Note that the result metrics of origin based study in terms of parameters of TTL, Ping Response and Geo-location are summarized on Section-4.

6.2 Geo Analysis

When repeated the experiment from two different geographical locations(Stony Brook, N. California) we observed some interesting results for origin level analysis.

- For East Coast, average TTL (update frequency of CDNs) was estimated to be 4700s, while for the West Coast, the metrics was as small as 50s. The drastic shift could be attributed to the presence of some origins with non CDN usage who are based towards the West Coast and observed higher response time and higher TTL when analyzed.

- Both, our East and West Coast servers estimated almost same percentage of CDN Usage at base using out Dynamic DNS + ping based approach. For East Coast server, approx 68.55% of origins observed CDN usage at base while the West Coast experiment observed 81.25% of origins at base.

- Repeating the experiment using KS-Hypothesis testing by analyzing ping distribution, we observe that approx. 78.19% of origins tend to use CDN at base. The results is higher than our Hybrid approach due to possible cached responses and lower ping times from edge DNS servers.

Hosting Method	Type of Content		
	Images	CSS	JS
Percentage content	35.53%	3.14%	14.96%
URL Rewriting	15.59%	3.41%	7.67%
DNS based	66.95%	77.62%	72.95%

Table 2: Site Level Analysis (in percentage number)

6.3 Site Level Analysis

Observing the percentage of static content on CDN, we found that majority of content served on CDN was found to be images followed by JS, followed by CSS. This is in accordance with our understanding that the content that is expected to change the least over time is outsourced to CDNs the most. Another observation was that, since we see that the average amount of JS served on CDNs is low as compared to static images, it could be inferred that the majority of web-servers prefer other dynamic JS scripts(Angular, phantom scripts) to be hosted on their local servers. This is again in accordance with the fact that primary use of CDNs is meant to be as proxy caches and not as a means computational servers.

On analysis of the method of content hosting preferred by webservers for their content, we observe the following statistics as seen from *Table-2*.

From the statistics we also observe, that most website prefer use of Static URL + DNS based approach to serve content on websites. Such websites have dedicated urls set for serving static content which indirectly point to the CDN. This approach facilitates the convenience of not requiring to change the entire HTML of webpage every-time the CDN url change, but just the DNS entries. Thus it acts as a proxied URL that points towards CDN. Some websites example: Yahoo also had a 1:1 ratio of static vs DNS based usage.

We could also observe that most E-commerce websites such as amazon.com, netflix.com, alipayexpress.com preferred the majority of content on CDNs which could be attributed to the high availability requirement of the servers across entire country.

7 Conclusion

CDN servers are globally distributed across countries and thus facilitate reducing the communication distance between the users and the origin servers. They let a user to connect to a geographically closest data center than the website's base origin server thus reducing the latency and providing faster service and better end user experience. Better user experience also increases the number of visitors to a website thus also improving on the economic standpoint of the corporate environments.
As we observed, CDNs also provide a loading balancing setup by dynamically updating the server frequency for a particular origin. With its distributed nature, CDNs can handle more traffic and can withstand failure much better than the origin servers.
Moreover, we can see that CDNs also mitigate resource starvation attacks such as DoS, DDoS on the static content thus enhancing web security for an origin. The hosts have a choice of how much content to serve on a CDN and many hosts today prefer to keep statically served data on these large cloud based networks.

8 Future Work

- We can perform more comprehensive study by using longer streams of ping packets in our Hybrid DNS and ping approach. Due to time constraints, we were currently using ping streams of 5 packets for our site level analysis.

- We could repeating experiments with more EC2 resources across Southern regions of US to generate more detailed geographically dependent data. We currently used one EC2 N. California instance due to resource limitations.

- One of the most important aspect of analysis that could be further covered is analyzing interactions of CDNs among themselves and how they behave for hosting content for a particular webserver. For this, we could monitor publically available CDN logs and check how data is distributed across different CDNs from a same provider.

9 References

1. *Content Delivery Networks*
   `https://en.wikipedia.org/wiki/Content_delivery_network`.

2. *What is a CDN*
   `https://www.cloudflare.com/learning/cdn/what-is-a-cdn/`.

3. *Alexa Top Websites*
   `https://www.alexa.com/topsites`.

4. *Content Distribution Network Benefits*
   `https://www.akamai.com/us/en/resources/content-distribution-network.jsp`.

5. *How CDNs works*
   `https://www.incapsula.com/cdn-guide/what-is-cdn-how-it-works.html`.

6. *Kolmogorov Smirnov Test*
   `https://en.wikipedia.org/wiki/KolmogorovSmirnov_test`

7. *Third party tool for origin verification*
   `https://www.cdnplanet.com`

8. *Code/Library Maintained at*
   `https://github.com/bhaveshgoyal/cdnlyzer`