Protick Bhowmick

protick@vt.edu

CDN Performance analysis

Overview: We are given the the Top 1000 sites from Tranco domain list [1]. The objective of this exercise is to find out which which domains are served by CDNs from that list and have comparative analysis of different CDNs in respect of different performance metrics.

Identify domains served by CDNs: CDNs usually employ two methods for redirecting domains to CDNs. Firstly, it can use **CNAME**s: for example: in case of **www.reddit.com**, a **CNAME** redirects it to another domain controlled by **Fastly** (Shown in **Figure 1**) [2].

```
;; ANSWER SECTION:
www.reddit.com.         5802    IN      CNAME   reddit.map.fastly.net.
reddit.map.fastly.net.  15      IN      A       151.101.1.140
```

Figure: 1

Secondly, domains can also delegate their name servers to **CDN**s by replacing their **NS** records to the **CDN**s' nameservers [3]. For example in **figure 2**, we can see that **npmjs.com**'s name server points to **Cloudflare**'s own controlled name server. And if we look at the owner of the ip 104.16.92.83, we would see that it is owned by Cloudflare Inc.

```
;; ANSWER SECTION:
npmjs.com.              21600   IN      NS      bayan.ns.cloudflare.com.
npmjs.com.              300     IN      A       104.16.92.83
```

Figure: 2

So, to find out if a domain belongs to a **CDN**, we manually looked into the CNAME patterns of different **CDN**s. We also looked at the organization name owning the IP prefix of the A record. Thus, we complied a list of reference strings mapping to different **CDN**s, and then we looked for those strings in the **CNAME** expansions as well as the IP prefix owner names. With these methodology, we could identify **458** domains to be using **CDN**s. It is notable that some Organizations provides both CDN and VPS services (ex: Google) that might lead to some false positives.

Now in the following table, we sort the CDNs according to number of domains observed:

CDN	Number of domains
Akamai	137
Cloudflare	133
Fastly	83
Google	59
Cloudfront	38
Edgecast	8

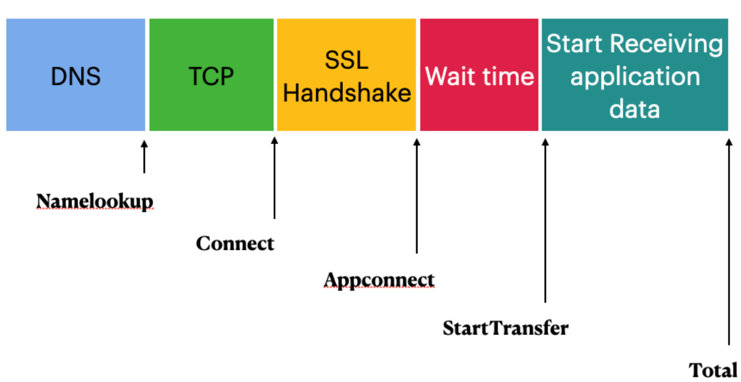
Table: 1

We can see that **Akamai** and **Cloudflare** have the major share of the domains in the top 1000 list.

Performance Evaluation: For evaluation, we would be using the following performance metrics. [For all the metrics, 10 samples are collected for each website under each CDN and the median value is calculated.]

a) **DNS resolution time**: For DNS resolution time, we used a python library **pydig** [4] that is a wrapper around the dig command tool. It is to be notable here that the full **DNS recursive query** includes multiple requests starting from the root server to the leaf authoritative DNS server. We have only considered the DNS resolution time to get the authoritative response from the CDN's authoritative server and excluded the time of the fetching NS records from the root, tld name servers.

b) We use **PucURL** [5] (a python wrapper for **libcurl**) for finding some other connection timings for our performance evaluation. As we see in following figure, it gives us different timing information [6] about a connection to a server. We do not use the **DNS** timing from



this source as it uses the default DNS resolver. From these attributes, we find out the **TCP connection time** (connect - namelookup), **SSL handshake time** (appconnect - connect) and **Time to first byte** (starttransfer) for all CDNs. We do not compare the time-to-last-byte (total time) as we are fetching different size pages for different websites and the results can not be generalized.

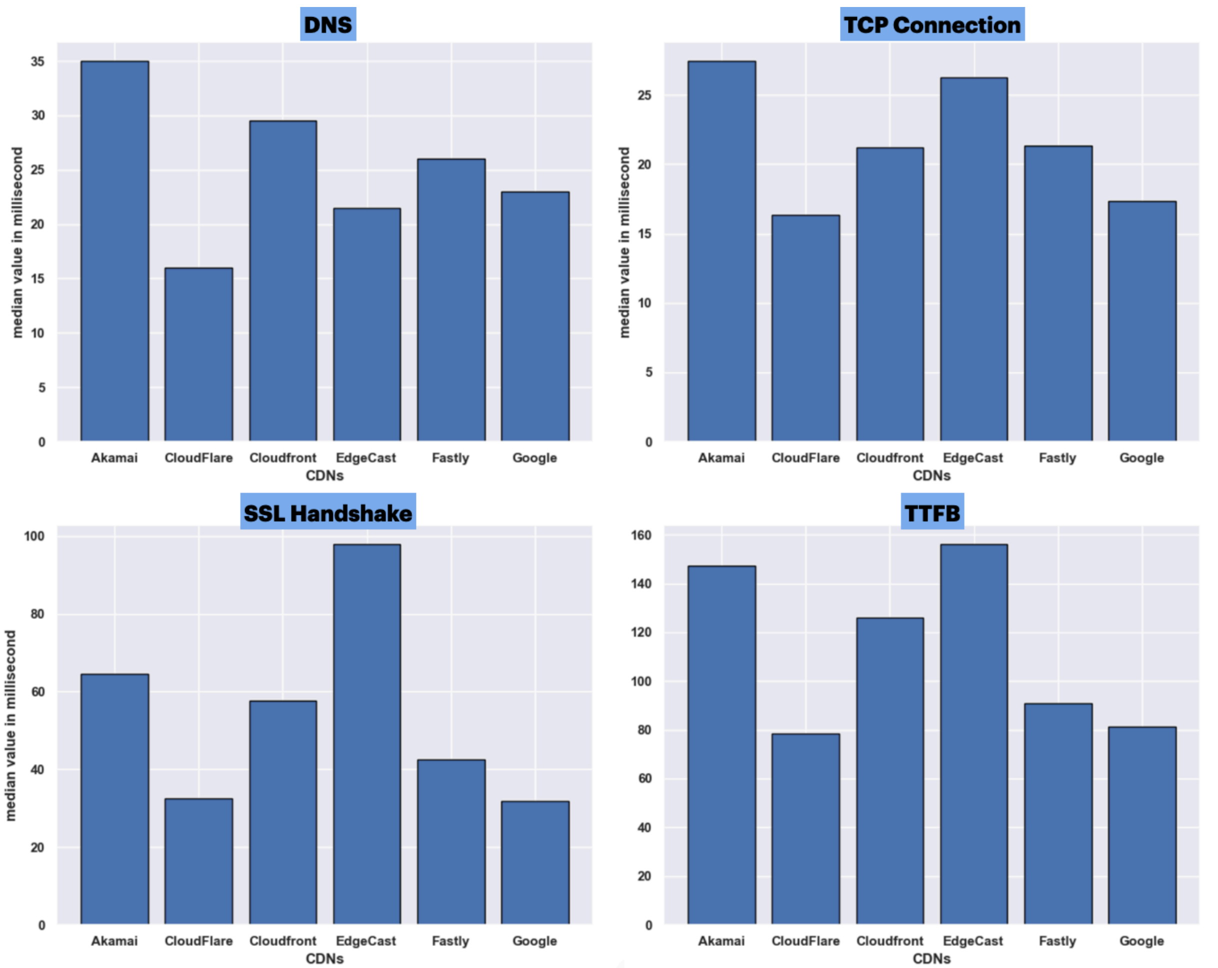


Figure: 3

Findings: In **figure 3**, we plot the median time of different connection timings (DNS, SSL handshake, TTFB and TCP connection). The quick takeaway from the figure is that **Cloudflare** and **Google** have relatively better performance than other CDNs on all metrics. Also, **Edgecast** has relatively worst performance considering all metrics closely followed by **Akamai**. Also, we see that the distribution was relatively similar across all metrics. In practice CDNs can host **DNS** servers in different datacenter than the HTTP proxy servers. As we see in case of **Cloudfront**, the DNS performance was relatively weaker in comparison to the other metrics.
In summary, **Cloudflare** and **Google** has better performance and resultantly should have better user experience in terms of **lower latency** and **faster page load**.

How would you expand your analysis: One major caveat of this experiment is that the measurement was done from a **single vantage point** (located in Virginia, USA). **CDN**s have distributed Point of presence (PoPs) all over the world so that clients can fetch content from nearby PoPs [7]. So, to have a more stronger inference on our findings, we need to conduct our measurement from multiple vantage points. Only by measuring the same metrics from multiple vantage points, we can truly evaluate the distributed architecture of CDNs in terms of global reachability and performance. Our results can be biased as **Cloudflare/Google** might have a nearby PoP from our vantage point which distorts our findings. Also, it would be interesting to evaluate how CDNs perform under different kind of content fetching (ex: video streaming). For example: Facebook's own CDN has different layers of caches for static content and video content [8].

To have a better understanding leveraging multiple **vantage point**s, we have measured Round-Trip-Time (**RTT**) [9] for the CDNs with randomly selected 100 Ripe Atlas probes [10] around the world. **RTT** is a key factor that impacts latency and page load time. We only focus on the top 5 CDNs found from our previous measurement.

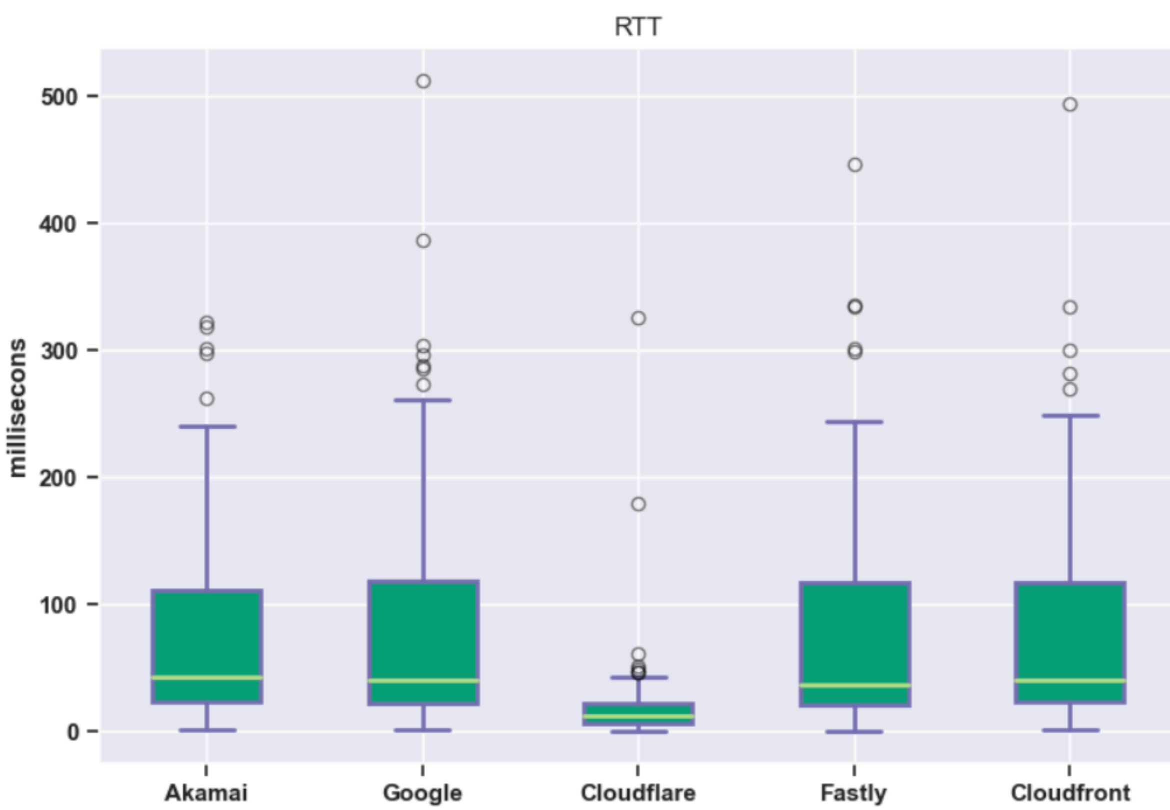


Figure: 4

In **figure 4**, we draw the box plot of the **RTT**’s measured from the 100 ripe atlas probes. We see that Cloudflare has the **lowest median RTT** along with **lowest variance** among all the CDNs. This leads us to believe that Cloudflare PoPs are more **distributed** around the world ensuring **faster response** to clients globally. On the other hand, other CDNs have more variable **RTT,** which means depending on the geolocation, you can experience vastly differing round trip time while connecting to these CDNs.

We carry out a separate experiment to **cross-check** our assumptions. Our findings indicate that Cloudflare’s PoPs are relatively widely distributed around the world. Some **CDN**s provide **geohints** in the HTTP response headers about the location of the PoP contacted by the client [11]. **Cloudflare**, **Fastly** and **Cloudfront** returns such **geohints** in their http response, they header keys are shown in **table 2**. All three of them report back the iata [12] of the PoP.

CDN	Geo-hint HTTP header	Hint Type
Cloudflare	CF-Ray	IATA of nearest airport
Fastly	X-Served-By	IATA of nearest airport
Cloudfront	X-Amz-Cf-Pop	IATA of nearest airport

Table: 2

So, now we use a residential proxy called **Brightdata** [13] to find out the distribution of PoPs for these three CDNs. We send randomly choose a website from each of the CDNs and send **GET** request through residential exitnodes [14] of Brightdata situated in 238 different countries. Then we analyze the geo-hints (iata) returned in these responses to find out the closest PoPs of the exit nodes around the world. As we see from **table 3**, we have found out that Cloudflare have more PoPs in comparison with the other two. One thing is to take note is that this number is a **strict lower bound** of actual number of PoPs as a) there can be multiple PoPs mapping to same iata, b) there can be many PoPs in one country.

CDN	Distinct Countries of exit nodes	Total PoPs	PoP located in same country as the exitnode
Cloudflare	237	80	47
Fastly	237	51	32
Cloudfront	237	60	39

Table: 3

References:

1) https://tranco-list.eu/

2) Wang, Z., Huang, J., & Rose, S. (2017). Evolution and challenges of DNS-based CDNs. Digital Communications and Networks, 4(4), 235–243. https://doi.org/10.1016/j.dcan.2017.07.005

3) https://developers.cloudflare.com/dns/zone-setups/full-setup/setup/

4) https://pypi.org/project/pydig/

5) https://pypi.org/project/pycurl/#description

6) https://curl.se/libcurl/c/curl_easy_getinfo.html

7) Huang, C., Wang, A., Li, J., & Ross, K.W. (2008). Measuring and Evaluating Large-Scale CDNs.

8) https://ooni.org/post/jordan-measuring-facebook-interference/

9) https://www.cloudflare.com/learning/cdn/glossary/round-trip-time-rtt/

10) https://atlas.ripe.net/probes/

11) Huang, R., Zhou, M., Guo, T., & Chen, Y. (2022b). Locating CDN edge servers with HTTP responses. Proceedings of the SIGCOMM ’22 Poster and Demo Sessions. https://doi.org/10.1145/3546037.3546051

12) https://en.wikipedia.org/wiki/IATA_airport_code

13) https://brightdata.com/

14) https://brightdata.com/proxy-types/residential-proxies