



Investigating Performance Metrics for Evaluation of Content Delivery Networks

Seyed Jalal Jafari^{1(✉)}, HamidReza Naji², and Masoumeh Jannatifar³

¹ University of Zanjan, Zanjan, Iran
sj.jafari@znu.ac.ir

² Kerman Graduate University of Advanced Technology, Kerman, Iran

³ AmirKabar University of Technology, Tehran, Iran

Abstract. Content Delivery Networks are one of the most common services in order to overcome performance problems caused by massive data requests in popular web applications. CDNs improve clients' perceived quality of service by placing replica servers scattered around the globe and consequently redirecting users to closer servers. While CDNs' ultimate goal is to improve the performance of data delivery, their own efficiency can also be an issue to investigate. Due to the complexity of these services, plenty of factors can impact the performance of CDNs. As a result, the efficiency of CDNs can be measured using various metrics. In this paper we review some of the well-known performance metrics in the literature for evaluating CDNs. We also present some other measures including Fairness and Content Travel. In order to attain an overall insight about a CDN, a Cost Function is also presented which incorporates most of the metrics in a single formula.

Keywords: Content Delivery Networks · Performance Measurement · Metrics · QoS

1 Introduction

Recently, Internet-based services have turned into an inseparable part of people's everyday life. The rapid growth in the popularity of some services causes them to face performance issues and bottlenecks in terms of latency, bandwidth consumption, etc. In order to avoid performance related concerns as well as improving QoS and QoE for end users, large-scale web applications deliver contents through Content Delivery Networks. CDNs act as a trusted overlay network that offers high-performance delivery of common Web objects, static data, and rich multimedia content by distributing load among servers that are close to the clients [1]. CDNs provide services that improve network performance by maximizing bandwidth, improving accessibility and maintaining correctness through content replication [2]. This is achieved by spreading some surrogate servers across a geographic area. When a user issues a request for some content,

the surrogate server, which is more proper than the others, will respond to that request. Figure 1 shows a typical CDN architecture [3].

The very last few years have seen an astonishing development in CDNs' technology, and today's Internet content is largely delivered by major CDNs like Akamai or Google CDN [4]. Facebook contents, for example are mainly hosted by Akamai CDN servers [4]. Despite the commercial stability of CDNs, researches to improve these systems are still ongoing. There are different research aspects in CDNs e.g. Replica Server Placement, Request Routing Mechanisms, Caching Policies, etc. which can in turn lead to improvements in the performance of CDNs. However, due to complexity and intricate structure of CDNs, measuring the performance of them can also be a subject of great interest. There are plenty of factors which impact the performance of CDNs. As a consequence, several performance metrics can be employed to investigate efficiency from different angles. RTT (Round Trip Time), for example, is one of the most considered metrics for evaluating CDNs in the literature. Although RTT can provide an acceptable overview of how well CDNs performs, it does not necessarily reflect all performance subtleties in these systems. In this paper we will discuss the existing performance metrics which are currently used to evaluate CDNs in details. Furthermore, some new performance metrics will be presented.

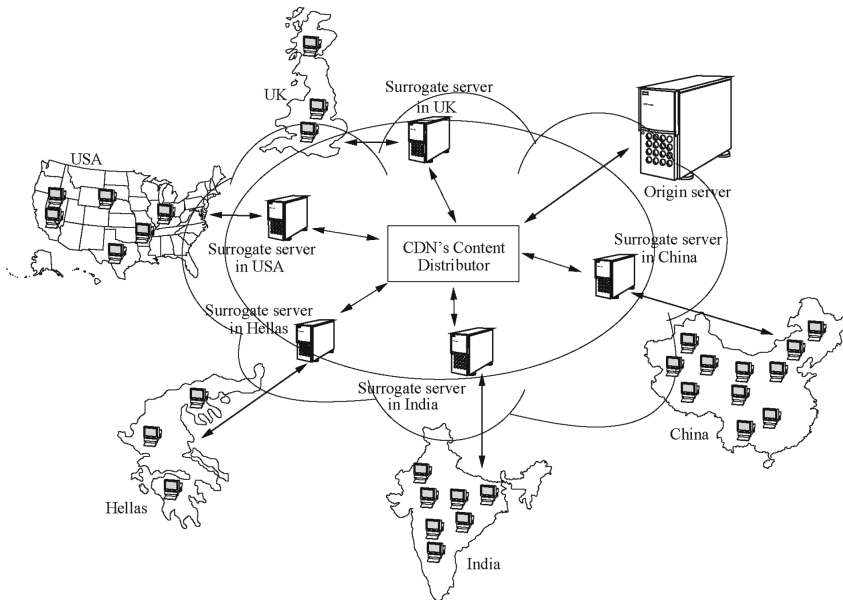


Fig. 1. A typical CDN's architecture [4]

2 Literature Review

From the early ages of CDNs to the time being, clients' perceived latency (AKA: response time, Round Trip Time – RTT) has been the top priority metric for researchers when measuring CDNs' performance [4–6]. Akhtar et al. [7] employ statistical functions operating basically on latency in order to evaluate users' perceived performance in different commercial CDNs. While reducing latency is the ultimate goal of CDNs, the performance of CDNs can also be measured from other points of views.

Looking at some recent works, Hours et al. [8] examine the impact of DNS resolving methods in CDNs on the performance of web browsing in terms of External TTL and also Throughput (Mbps). Although geographical distance can affect the performance of CDNs, few works use this metric for evaluations. However, this metric has been employed in some recent works. In [9], authors take physical distance between clients and servers as a metric to measure the performance of AnyCast DNS resolving. Mapping distance is the term which Chen et al. use to indicate the great circle distance between a client and the server as a metric for evaluating CDNs [10]. They also introduce time to first byte (TTFB) as another parameter which is basically the duration from when the client makes a HTTP request for the base web page to when the first byte of the requested web page was received by the client.

In [2], Pathan et al. mention performance measurement as an issue in CDNs. They consider Cache hit ratio, Reserved bandwidth, Latency, Surrogate server utilization and Reliability (packet loss) as important measures to investigate.

3 Metrics Discussion

In this section we present and discuss a variety of metrics which can be used to evaluate CDNs' performance. In abstract, some of the measures can be seen from the clients' point of view e.g. RTT (latency) and Throughput while others belong to the internal architecture of CDNs like server cache misses, fairness, etc.

Some metrics mentioned in this section have been employed in the literature before, however we discuss them here in order to establish a comprehensive image on the issue. We also introduce some other performance metrics to evaluate CDNs including fairness, Content Travel and CDN Cost.

3.1 Latency (RTT)

As it was mentioned before, latency is the most straightforward metric for evaluating CDNs' performance. In a large number of works, Round Trip Time is considered as an appropriate measure to indicate users' perceived latency. RTT is the amount of time that takes an IP packet to travel from the source machine to the target machine plus the time of receiving an ACK (Acknowledgement) for that packet.

RTT can be seen as a metric in different places and various forms, each of which indicating the performance from a specific angle. Although RTT is usually measured on the client side, it might be interesting if we take it into account on the server side too. RTT on the client side depicts the amount of time users wait for their requests, however this metric on the server side can be interpreted as a ground to measure how fast are the communications of a server with respect to its clients. This can lead to decisions like changing a server's location or strengthening its links.

Investigating RTT can be useful at different levels. As it is stated in [4], RTT to any specific IP address consists of both the propagation delay and the processing delay. Considering a large number of packet exchange, min RTT can be assumed as an approximation for propagation delay. In other words, min RTT can correspond to network distance between clients and servers. Mean RTT can also be another noticeable variation of RTT which can be also a suitable factor to evaluate the response time of clients and servers. In other words, mean RTT indicates the average amount of time that clients or the servers wait for their requests to be fulfilled.

Processing delay is a hidden metric which lies within RTT. We can assume the difference between max RTT and min RTT in every TCP flow to approximate processing delay for a given network element e.g. a replica server. Equation 1 indicates this metric. Mean processing delay of each network element equals to the mean processing delay of all TCP flows toward that element. It can help to evaluate how busy the servers are, for example.

$$\overline{PD}_{element} = \frac{(\sum_{i=1}^{n_{flow}} \max RTT_{flow_i} - \min RTT_{flow_i})}{n_{flow}} \quad (1)$$

3.2 Cache Miss

Caching is a key element in CDNs. Improvement in content delivery is achieved by caching web objects on surrogate servers which are located somewhere close to the request source. Whenever a client is redirected to a surrogate server but the requested object does not exist in that server, a cache miss occurs and the surrogate server has to retrieve the object from origin server. Cache misses can affect the performance of content delivery dramatically. There are plenty of factors which influence cache miss ratio in surrogate servers. Cache size, caching policies, prefetching mechanisms [11, 12] and server congestion can be considered as some of these factors. Not only does lower cache miss improve the quality of services, it also indicates that server has imposed lower load on the network. Depending on the investigation scenario, cache miss can be a proper metric to measure surrogate servers' performance in CDNs.

3.3 Throughput (Average Bits/Sec)

In computer networks throughput generally indicates the performance of network elements in terms of data transmission rate per a time unit. It is usually expressed

as average sent and/or received bits/sec. Interpreting throughput in CDNs may not be as plain as other metrics. In fact, higher throughput can be considered as both a negative or a positive phenomenon depending on the scenario conditions.

When higher throughput in servers results in lower latencies, we can claim that servers have put more effort to deliver better quality services. On the other hand, we can imagine a scenario in which overall throughput in servers is high while no significant change is seen in the latency numbers. In this case we can say that servers may have been uselessly busy because of improper topology or inefficient caching. In [13] authors state that “even though most clients are served by a geographically nearby CDN node, a sizeable fraction of clients’ experience latencies several tens of milliseconds higher than other clients in the same region. Second, we find that queueing delays often override the benefits of a client interacting with a nearby server.” This indicates higher throughput of servers can lead to lower response time in some cases. Similarly, as it is mentioned in [14], latency can also be affected by throughput bottlenecks along the path between client and server. In this case rethinking path selection mechanisms can be a solution.

3.4 Geographical Distance

Sometimes the distance between clients and servers is approximated with min RTT [4]. Although it can indicate the delay between a server and a client but it may not be stable due to congestion or throughput bottlenecks. Geographical location of clients and servers can be employed as a solid factor to measure the distance between clients and servers. IP geolocation services [15] can be used to provide this data. In a CDN evaluation scenario, if we provide the geographical coordinates of clients and servers, we can eventually extract the average physical distance of surrogate servers from their clients. Average client distances can tell us how efficiently the surrogate servers are scattered in a given area. As this value is higher the effectiveness of CDN drops.

3.5 Fairness

As it was mentioned before, there are multiple surrogate servers in a CDN. It would be ideal to distribute the load among them equally. The worst case scenario occurs when some servers work with their maximum capacity while there are other idle servers available in the CDN. We can say that if the load on servers is distributed approximately equal, the requests will be routed to the surrogate servers fairly. The number of served requests by each server can be used as a basis to calculate the fairness measure. In order to calculate fairness we use Jain’s fairness index [16]:

$$J(S_1, S_2, \dots, S_n) = \frac{(\sum_{i=1}^n S_i)^2}{n \times \sum_{i=1}^n S_i^2} \quad (2)$$

In this equation n is the number of servers and S_i is the load amount tolerated by server i (precisely, the number of requests served by server i but normalized

to a value between 0 and 1). The result is a number between 0 and 1. As the result of this equation is closer to 1, the load is distributed among the servers more fairly.

3.6 Overall Consumed Bandwidth

CDN topology directly impacts the routes on which packets travel in the network. As the surrogate servers are farther from users, the packets travel a longer distance in the network. Therefore, more equipment (like routers) should be involved in the process of request fulfilling and also more control packets should be generated. Hence overall consumed bandwidth will rise. On the other hand, it is rational to say that perceived response time by final users is directly proportional to overall consumed bandwidth in the network. Under normal conditions, more bandwidth consumption can be interpreted as the fact that the packets have traveled longer routes, so the users must have tolerated more delays. Therefore, the amount of overall bandwidth used in a network can be regarded as another decent measure to evaluate CDNs' performance.

3.7 Content Travel Measure

As it was stated, it is desired that contents travel shorter routes through the CDN network. If the overall delivered contents travel longer paths to reach their destination, there will be some consequences for this incident:

- Obviously there will be an increase in average content delivery latency;
- More network equipment (e.g. routers) must be involved in content delivery process. Therefore, more processing resources will be used;
- More bandwidth will be consumed in the whole network infrastructure.

As a result, we can say that when contents travel longer routes in the network, CDNs performance diminishes in terms of latency, resource usage and bandwidth consumption. If location information for the clients is provided for a CDN scenario, it is possible to define a factor to measure this event. Mean travelled distance by packets multiplied by overall contents size served in the network will give us a measure for evaluating CDN's performance for this phenomenon which we call "Content Travel" measure. In a content delivery process, it gives us an insight about the path length between a surrogate server and its clients and also the content size served by that server, all integrated into a single value. As the Content Travel value is higher, it can be said that the massive contents have traveled longer routes in the network, therefore CDN has been affected in terms of performance measures discussed above. One of the goals can be to minimize this factor. Equation 3 describes this measure. First the mean distance between request sources and each server must be calculated. \overline{D}_s is the mean traveled distance for requests (D_{req_i}) destined to server S in kilometers. n_{req} indicates the number of requests which have been sent to a specific server. C_{earth} is a constant value which is considered to calculate the great circle distance between

two points instead of a simple Euclidean distance. This value usually is set to 111 [16]. Then the Content Travel measure can be calculated for all servers in the network. n_S is the number of servers available in CDN and $ServedSize_S$ indicates the size of content served by Server S.

$$\begin{aligned} \bar{D}_S &= \frac{\sum_{i=1}^{n_{req}} D_{req_i}}{n_{req}} \times C_{earth} \\ ContentTravel &= \frac{\sum_{S=1}^{n_S} (\bar{D}_S \times ServedSize_S)}{n_S} \end{aligned} \quad (3)$$

3.8 CDN Cost

Finally, a cost function can be defined to summarize different parameters (from different aspects). Its value shows how well the CDN has performed in a scenario. Here we have picked some of the important metrics discussed above to build this function. The CDNCost function is defined as follows:

$$\begin{aligned} CDNCost &= w_1 RTT_{Clients} + w_2 RTT_{Servers} \\ &+ w_3 Throughput + w_4 ContentTravel + w_5 (1 - Fairness) \end{aligned} \quad (4)$$

- $RTT_{Clients}$ and $RTT_{Servers}$ are the mean perceived RTT measure by Clients and Servers.
- Throughput is the mean bits transferred in a second by all the devices working in the network (clients, server and routers).
- Content Travel and Fairness are the parameters which were discussed earlier. The fairness value is subtracted from 1 because we desire lower values for CDN Cost measure while higher fairness values indicate better performance in terms of this measure.

All the parameters in Eq. 4 must be scaled to a value between 0 and 1. CDN Cost value is also a number between 0 and 1. As it is closer to 0, it means that CDN is performing better. Every parameter in this formula has a weight coefficient which reflects the importance of that parameter. Sum of all weights must be equal to 1. For example, if we want to pay equal attention to all parameters we should set all the weights equal to 0.2. By changing the weight values any parameter can be bolded or faded out according to the desires of experimenter.

4 Experiments

In this section we employ some of the important metrics discussed in previous sections for evaluating an example experiment. This experiment aims to investigate Replica Server Placement problem by simulating some approaches from the literature including hotspot [5] and GeoIP clustering [17]. It is assumed that in the CDN topology we have at most three replica servers for which we need to choose a place (besides the one fixed origin server). Three different approaches have been employed in order to determine a place for the replica servers:

1. Random selection: replica server places are selected randomly. This approach is never used in reality but the results can give us an insight about the effectiveness of other approaches.
2. HotSpot [5]: the main idea behind this approach is to put replica servers where higher request rates are observed.
3. GeoIP Subtractive [17]: this approach uses client’s geographical coordinates to cluster the users. It employs subtractive clustering for this purpose.

In the following we will execute the aforementioned approaches in a simulation environment (using INET Framework under OMNet++) and then we will evaluate the results. Six-month access log of a Swedish webapp is used to create content and clients’ datasets for all scenarios. The dataset is called googlecreeper and represents the search history of Swedish users. In all experiments the origin server is placed in the US.

4.1 Scenario #1: Random Selection

The first scenario chooses two random Routers in the network infrastructure and connects the surrogate servers to them. There is no rationale behind this approach and it is only executed to be compared with other schemes. Suppose that a router in Australia and another router in Iran are chosen as replicas for this scenario. The origin server is connected to a router in the USA. Table 1 indicates the result of simulation using these configurations.

4.2 Scenario #2: HotSpot

HotSpot considers the places where most of the requests come from as a suitable choice for placing the replica servers. With the given dataset and in a classic client-server network, simulation results indicated that the most congested routers are somewhere in Sweden, Canada and Mexico. These are the top three routers which receive the highest number of requests in the first hop. Hence, HotSpot elects those areas to place replica servers. Table 2 shows the result of simulation with this configuration.

Table 1. The result of Random Replica Server Selection

Module name	Mean RTT (ms)	Max RTT (ms)	Min RTT (ms)	Receive throughput (bit/sec)	Send throughput (bit/sec)	Served web objects	Served content size (MB)	Cache misses	Average distances (KM)
originServer	230	870	180	3.25	23.30	353	22.21	-	-
surrogateServer1 (AU)	240	1090	066	46.49	321.28	681	315.19	383	6105
surrogateServer2 (IR)	280	1109	120	210.63	1809.44	5063	1518.76	663	9546
Clients	230	1220	60	34.31	3.7	-	-	-	-
Routers	-	-	-	27.58	27.58	-	-	-	-
Overall average	250	1220	65	28.42	28.42	-	-	-	-

Table 2. The result of HotSpot Replica Server Selection

Module name	Mean RTT (ms)	Max RTT (ms)	Min RTT (ms)	Receive throughput (bit/sec)	Send throughput (bit/sec)	Served web objects	Served content size (MB)	Cache misses	Average distances (KM)
originServer	172	680	90	4.55	34.51	467	33.28	-	-
surrogateServer1 (SE)	242	1320	66	143.10	1189.23	4017	1171.70	812	5106
surrogateServer2 (CA)	187	900	90	59.40	442.30	804	435.87	121	666
surrogateServer3 (MX)	174	1100	85	65.89	500.94	923	493.47	227	3219
Clients	178	1360	65	34.32	3.71	-	-	-	-
Routers	-	-	-	13.32	13.34	-	-	-	-
Overall average	200	1360	65	15.45	15.45	-	-	-	-

4.3 Scenario #3: GeoIP Subtractive

GeoIP Subtractive [17] is another approach that can be employed for replica server placement problem. This scheme clusters clients according to their geographical location and places the servers near the cluster centers. Applying this method on the given dataset gives us some coordinates in Canada, Sweden and China as the best candidates to place replica servers. The results of simulation using this configuration can be seen in Table 3.

Table 3. The result of simulation for GeoIP Subtractive Replica Server Selection

Module name	Mean RTT (ms)	Max RTT (ms)	Min RTT (ms)	Receive throughput (bit/sec)	Send throughput (bit/sec)	Served web objects	Served content size (MB)	Cache misses (KM)	Average distances (KM)
originServer	179	0790	83	4.57	34.52	496	33.29	-	-
surrogateServer1 (SE)	200	860	66	82.26	638.61	2436	628.12	562	888
surrogateServer2 (CA)	172	1100	63	114.02	941.524	1727	929.23	241	2886
surrogateServer3 (CN)	271	1055	65	72.13	552.45	1581	543.68	386	888
Clients	157	1140	65	33.778	3.655	-	-	-	-
Routers	-	-	-	9.073	9.087	-	-	-	-
Overall average	186	1140	63	11.55	11.55	-	-	-	-

5 Discussion

As it was demonstrated in the previous section, we employed some of the discussed metrics in this paper to evaluate three different scenarios for replica server placement problem. In this section we will discuss these scenarios by scrutinizing each of those metrics.

5.1 RTT

Figure 2 indicates the observed RTT in different scenarios. Besides the mean RTT of all modules in the network, RTT measure is also calculated for different network elements which can give us useful information to analyze the network components separately. For example, mean RTT among all clients indicates the average response time tolerated by end users. RTT in replica servers can indicate their distance from the machines they communicate with. In other words, as the replica servers are closer to the clients, the RTT in replica servers will be lower. Lower RTT in replica server tells us that they are placed in proper locations. Beside the distance parameter, higher RTT in servers can also be a sign of longer packet processing time. RTT in origin server shows the communication overhead between the origin server and replica servers. This measure can influence of object fetching when a cache miss occurs. Max and Min RTT exhibit the worst and the best cases in terms of response time. As the simulation results show, scenario #3 performs better in terms of all aspects of RTT measure. The reason is that this approach has placed the replica servers where the average distance between them and clients is minimized. Processing load has been insignificant in these experiments.

5.2 Fairness

As it was mentioned in Sect. 3.5, fairness is another factor which can indicate how the network’s load is distributed among replica servers. We can say that it is unfair if a server is congested with massive amount of traffic while other servers are idle. Since the request routing mechanism in these scenarios chooses the nearest server in terms of network distance, placing servers in farther locations

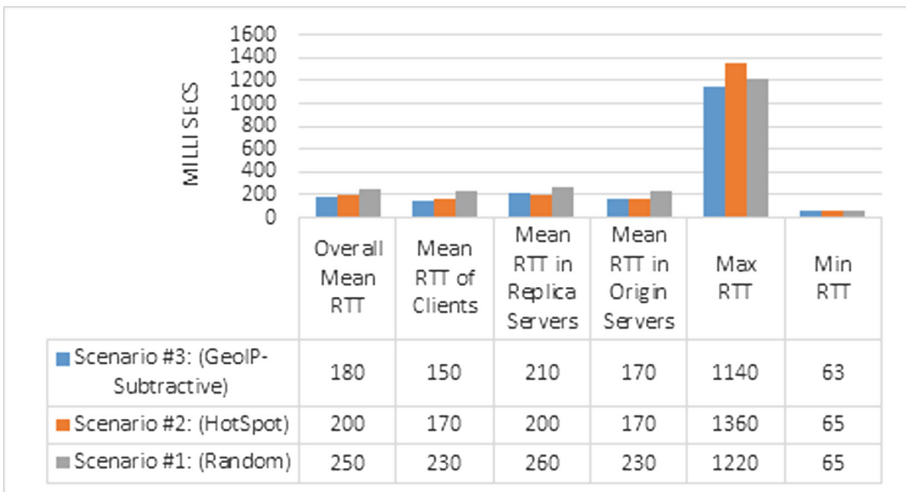


Fig. 2. RTT in different scenarios

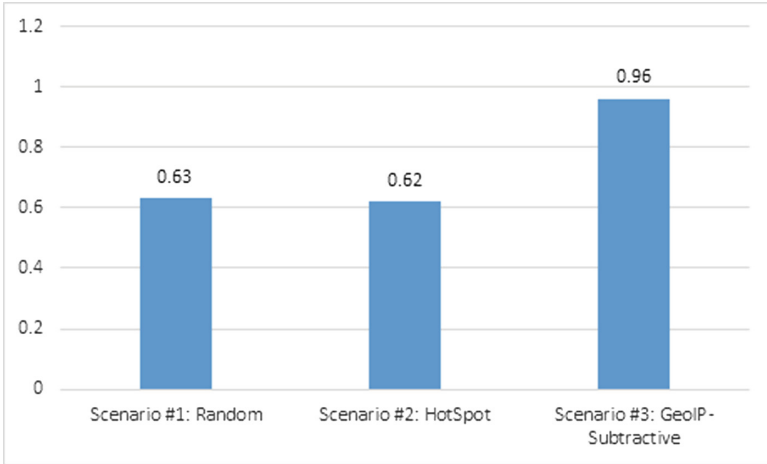


Fig. 3. Jain's fairness index

will result in pressure on some servers while others remain idle. On the other hand, if the servers are scattered around the network appropriately, the load is distributed among them fairly.

Jain's fairness index can give us a good insight here. Fairness measure gives a number between 0 and 1 for each scenario. As this measure is closer to 1, the load is distributed among replica servers more fairly. Figure 3 illustrates Jain's fairness index for the simulated scenarios. As it stands out from the graph, Scenario #3 has operated more fairly than the others in distributing load among servers equivalently.

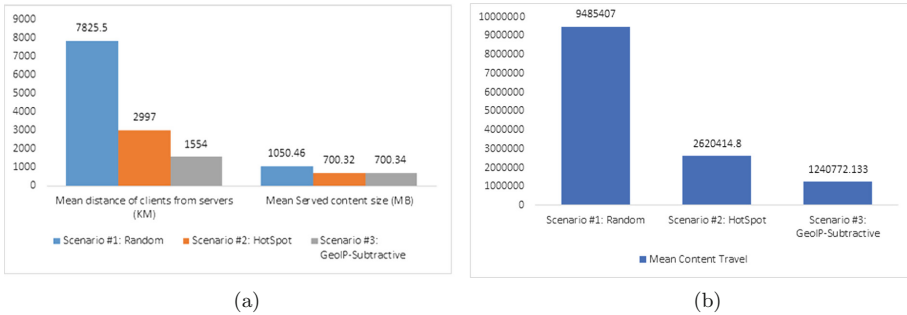
5.3 Content Travel Measure

In this section we investigate Content Travel measure (explained in Sect. 3.7) for the simulated scenarios. Each row in Table 4 demonstrates Content Travel measure for a surrogate server in one scenario. More specifically it tells us the mean distance of clients from that replica server, Served Content size by that server and finally the calculated Content Travel measure for that server. As it was mentioned before lower values in this measure indicate better performance of a replica server in CDN.

Figure 4a indicates mean distance of clients from replica servers and mean served content size by replica servers in two column groups. Also mean Content Travel measure for each scenario can be seen in Fig. 4b. As the result shows, scenario #3 has performed better in terms of Content Travel. In other words, massive contents have traveled shorter paths in the aforementioned scenario. This means the resources of CDN have been used more efficiently.

Table 4. Content Travel statistics for simulated scenarios

Servers in different scenarios	Mean distance of clients from servers (KM)	Served content size (MB)	Content Travel measure (MB KM)
Scenario #1: (Random) surrogateServer1 (AU)	6105	315.19	1924235
Scenario #1: (Random) surrogateServer2 (IR)	9546	1785.73	17046579
Scenario #2: (HotSpot) surrogateServer1 (SE)	5106	1171.70	5982700
Scenario #2: (HotSpot) surrogateServer2 (CA)	666	435.87	290289.4
Scenario #2: (HotSpot) surrogateServer3 (MX)	3219	493.4	1588255
Scenario #3: (GeIP-FCM) surrogateServer1 (SE)	5106	1171.70	5982700
Scenario #3: (GeIP-FCM) surrogateServer2 (CA)	2886	929.25	2681816
Scenario #4: (GeIP-Subtractive) surrogateServer1 (SE)	888	628.12	557770.6
Scenario #4: (GeIP-Subtractive) surrogateServer2 (CA)	2886	929.23	2681758
Scenario #4: (GeIP-Subtractive) surrogateServer3 (CN)	888	543.68	482787.8

**Fig. 4.** Mean distance of clients from replica servers in different scenarios (a), Mean Content Travel measure for each scenario (b)

5.4 Overall CDN Cost

Using various performance measures, CDN's performance was evaluated from different perspectives. In order to attain an insight about the overall performance of a CDN topology, the CDN Cost measure was proposed above. All the incorporated factors in this formula are normalized to a value between 0 and 1. The impact of each factor can be determined by a weight coefficient. Sum of weights must be equal to 1. As the CDN Cost value is lower in a scenario it means that CDN has performed better under the configurations of that scenario. Figure 5 depicts the normalized values for different metrics in the scenarios we have discussed. The last column group indicates CDN Cost measure. The weights for all factors is assumed to be equal ($=0.2$). This means that no factor has priority over the others.

The results show that scenario #3 has performed better than the others. As it is expected scenario #1, which had no rationale behind, is the worst.

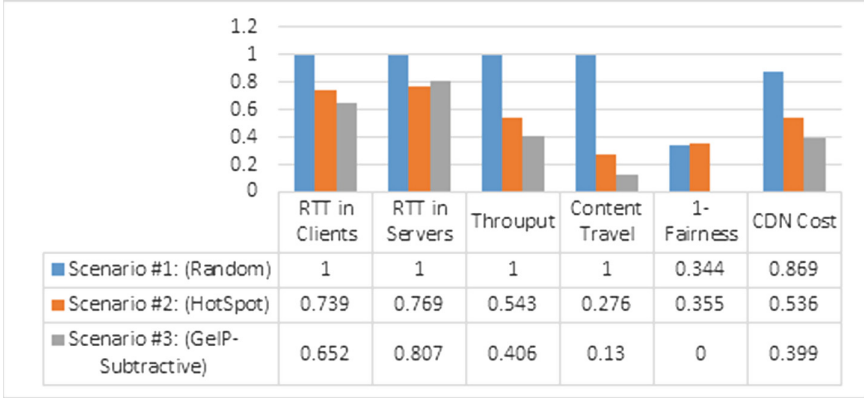


Fig. 5. CDN cost for the experimented scenarios

6 Conclusions

In this paper we reviewed and discussed common metrics for evaluating Content Delivery Network services. Furthermore, we introduced some other metrics for this purpose including fairness, Content Travel and finally an overall Cost Function to attain a big picture of CDN performance.

In order to compare the metrics in action we designed three simulation scenarios for Replica Server Selection problem. Key measures were extracted from the simulation results. The experiments showed that investigating and improving performance of CDNs is not limited to simply optimizing latencies. Depending on scenario, different factors should be taken into account to analyze performance of these services.

7 Motivating Scenario and Benefits for Organizations

In the past years, owners of large-scale web applications have been seeking solutions to reduce the latency of their services which is inevitably caused by massive requests. Content Delivery Networks offer a solution for this issue. CDN vendors and researchers, consequently, have been working hard to come up with new ideas for improving service qualities. In this path, measurement of quality has relied mostly on the latency and delay which clients experience. However, there are plenty of factors which impact the performance of CDNs. As the volume and variety of contents being transmitted over the Internet increases, CDNs themselves might not work efficiently enough. This imposes extra costs for CDN vendors and consequently for application owners. Investigating the performance in CDNs from different angles can help organizations utilize their resources while delivering high quality services. The metrics discussed in this paper can be employed by CDN stakeholders to achieve clearer pictures about the performance of these systems.

References

1. Vakali, A., Pallis, G.: Content delivery networks: status and trends. *IEEE Internet Comput.* **7**(6), 68–74 (2003)
2. Pathan, A.-M.K., Buyya, R.: A taxonomy and survey of content delivery networks. Grid Computing and Distributed Systems Laboratory, University of Melbourne, Technical report, 4 (2007)
3. Buyya, R., Pathan, M., Vakali, A.: Content Delivery Networks. *LNEE*, vol. 9. Springer, Heidelberg (2008). <https://doi.org/10.1007/978-3-540-77887-5>
4. Fiadino, P., D'Alconzo, A., Casas, P.: Characterizing web services provisioning via CDNs: the case of facebook. In: 2014 International Wireless Communications and Mobile Computing Conference (IWCMC), pages 310–315. IEEE (2014)
5. Qiu, L., Padmanabhan, V.N., Voelker, G.M.: On the placement of web server replicas. In: Proceedings of the IEEE Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001, vol. 3, pp. 1587–1596. IEEE (2001)
6. Akhtar, Z., Hussain, A., Katz-Bassett, E., Govindan, R.: DBit: assessing statistically significant differences in CDN performance. *Comput. Netw.* **107**, 94–103 (2016)
7. Hours, H., Biersack, E., Loiseau, P., Finamore, A., Mellia, M.: A study of the impact of DNS resolvers on CDN performance using a causal approach. *Comput. Netw.* **109**, 200–210 (2016)
8. Calder, M., Flavel, A., Katz-Bassett, E., Mahajan, R., Padhye, J.: Analyzing the performance of an anycast CDN. In: Proceedings of the 2015 ACM Conference on Internet Measurement Conference, pp. 531–537. ACM (2015)
9. Chen, F., Sitaraman, R.K., Torres, M.: End-user mapping: next generation request routing for content delivery. *ACM SIGCOMM Comput. Commun. Rev.* **45**, 167–181 (2015)
10. Sidiropoulos, A., Pallis, G., Katsaros, D., Stamos, K., Vakali, A., Manolopoulos, Y.: Prefetching in content distribution networks via web communities identification and outsourcing. *World Wide Web* **11**(1), 39–70 (2008)
11. Ariyasinghe, L.R., Wickramasinghe, C., Samarakoon, P.M.A.B., Perera, U.B.P., Prabhath Buddhika, R.A., Wijesundara, M.N.: Distributed local area content delivery approach with heuristic based web prefetching. In: 2013 8th International Conference on Computer Science & Education (ICCSE), pp. 377–382. IEEE (2013)
12. Krishnan, R., Madhyastha, H.V., Srinivasan, S., Jain, S., Krishnamurthy, A., Anderson, T., Gao, J.: Moving beyond end-to-end path information to optimize CDN performance. In: Proceedings of the 9th ACM SIGCOMM Conference on Internet Measurement Conference, pp. 190–201. ACM (2009)
13. Yu, M., Jiang, W., Li, H., Stoica, I.: Tradeoffs in CDN designs for throughput oriented traffic. In: Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies, pp. 145–156. ACM (2012)
14. MaxMind LLC. GeoIP (2010)
15. Jain, R., Chiu, D.-M., Hawe, W.R.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System, vol. 38. Eastern Research Laboratory, Digital Equipment Corporation Hudson, MA (1984)

16. Veness, C.: Calculate distance and bearing between two latitude/longitude points using Haversine formula in Javascript. Movable Type Scripts (2011)
17. Jafari, S.J., Naji, H.: GeoIP clustering: solving replica server placement problem in content delivery networks by clustering users according to their physical locations. In: 2013 5th Conference on Information and Knowledge Technology (IKT), pp. 502–507. IEEE (2013)