一般社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
IEICE Technical Report
CQ2024-40(2024-06)

遺伝的アルゴリズムを用いた最適キャッシュ攻撃

劉　　甲奇† 　上山　憲昭†

† 立命館大学 情報理工学部
〒567-8570 大阪府茨木市岩倉町 2-150
E-mail: †is0705vf@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

あらまし　近年，コンテンツ配信ネットワーク (cdn) の普及に伴い，cdn に対する攻撃も増加している．cdn は幅広い分野で使用されており cdn を用いた配信はネットワークトラフィックの大部分を占めるため，攻撃から cdn を保護することは非常に重要である．cdn に対するキャッシュ攻撃については多くの研究が行われているが，攻撃者の視点で分析することは稀である．そこで本稿では，遺伝的アルゴリズムを用いて攻撃者の最適な攻撃戦略を計算し，cdn プロバイダがシステム上の脆弱な箇所を発見し，より効率的に防御する方法を発見することを目的とする．本稿では，多層 cdn モデルを例に，遺伝的アルゴリズムを用いてモデルの弱点と防御を強化する方法を提案する．cdn プロバイダの場合，遺伝的アルゴリズムを使用して独自の cdn を分析し，強化することができる．

キーワード　CDN, CPA, LRU, キャッシュ

Optimum Cache Attacks Using Genetic Algorithm

Jiaqi LIU† and Noriaki KAMIYAMA†

† College of Information Science and Engineering, Ritsumeikan University
2-150 Iwakuracho, Ibaraki, Osaka 567-8570
E-mail: †is0705vf@ed.ritsumei.ac.jp, ††kamiaki@fc.ritsumei.ac.jp

Abstract　In recent years, as the content delivery network (CDN) is more widely used, the attacks against CDN are also increasing. Since CDN have been used in a wide range of fields and occupy a large amount of network traffic, it is very important to protect CDN from attacks. Although there has been a lot of research on cache attacks against CDN, it is rare to analyze them from the attacker's perspective. Therefore, this paper aims to calculate the optimal attack strategy of attackers through genetic algorithm to help CDN providers find vulnerable points in the system and find out how to defend more efficiently. This paper takes the multi-layer CDN model as an example and uses genetic algorithm to analyze the weakness of the model and the method of strengthening defense. For CDN providers, genetic algorithm can be used to analyze and strengthen their own CDN.

Key words　CDN, CPA, LRU, Cache

1. Introduction

Content delivery network (CDN) usually consists of geographically distributed servers to cache and efficiently deliver Internet contents, such as HTML pages, images, and videos. CDN have been used in a wide range of fields and occupy a large amount of network traffic. According to a survey [1], CDN market value is expected to rise from $11.76 billion in 2019 to $49.61 billion in 2025, which shows that CDN has great development potential and application prospect. However, CDN also faces the threat of cyber attacks, such as Distributed Denial of Service (DDoS), which affect CDN services and user experience. Because CDN has multiple distributed servers, CDN has a certain defense effect against DDoS attacks compared with the common server. There are also attacks that specifically target caches, such as Cache Pollution Attack (CPA), which can reduce CDN cache performance and causes users to suffer longer response times. Since response time greatly affects the user experience [4], it is a very important performance metric for CDN providers. Therefore, CDN needs to be well protected against cache attacks.

Although there are many existing works which investigate the impact of each of cyber attacks against CDN [2] [3], just a

few works analyzed attacks from the attacker's point of view. To effectively allocate resources to defend against attacks, we should understand the attacker's strategy. We have found the best attack strategy assuming multilayer cache server (CS) [7].

In this paper, considering that many CDNs consist of multiple CSes with single or multiple layers [10], we still use the CDN model with multiple CSes in 2 layers and propose a method of optimally allocating the attack rate against each CS using the genetic algorithm (GA). We use the queuing model to calculate the load of each CS and link in the multilayer CSes. In Section 2., we propose a method to compute the response time of CS. In Section 3., we build the multilayer CS model, and we propose a method of optimally allocating the attack rate against each CS using the GA in Section 4.. In Section 5., we show the numerical results with finding the factors that affect the effect of the attack. Finally, we conclude this paper in Section 6..

2. Analytical Model

The impact of CPA can be measured by the increase of the response time of contents because the cache miss results in increasing the response time. So we use response time to measure the CDN performance.

We use the M/M/1 queue model to derive the average response time of contents. We assume that server requests follow zipf law, and let M denote the number of contents provided by CDN. Let λ_i denote the Poisson arrival rate of request for content i, and we define $1/\mu$ as the mean of the exponentially distributed service time of CS. Using the M/M/1 queue model, we can obtain W, the average response time, by

$$W = \frac{1}{\mu - \sum_{i=1}^{M} \lambda_i}. \tag{1}$$

In a network with CS, there are two cases based on whether the requested content is cached or not, as shown in Fig. 1. The origin server stores all the provided contents, and the CS stores a part of contents. When the content requested by a user does not exist in the CS, which is called *cache miss*, the CS will obtain the content from the origin server, store the content according to the cache replacement policy, and send the content to the user. On the other hand, when the content requested by the user exist in the CS, which is called *cache hit*, the CS will update the cache storage according to the cache replacement policy and deliver the content to the user. Since the LRU (least recently used) is a common cache replacement policy in CDN [6], this paper assumes that all CSes adopt the LRU.

Since the latency between users and CDN CSes depends on the networks between them, and it will not be affected by attacks against CSes, we do not consider the latency between users and CSes. However, we consider the latency between the CSes and the origin server because this latency will affect the impact of the attack strategyon the effect of attack. We define T as the latency between a CS and the origin server which is the latency between the time instance that the CS sends a request to the origin server and the time instance that the CS receives the requested content from the origin server. The average response time of the source server and the cache server will be different according to the request rate, so we define the average response time of the origin server as W_o and the average response time of the CS as W_c. When the requested content exists in the CS, i.e., cache hit, the average response time is W_c, whereas the average response time is $W_c + T + W_o$ when the requested content dose not exist in the CS, i.e., cache miss.

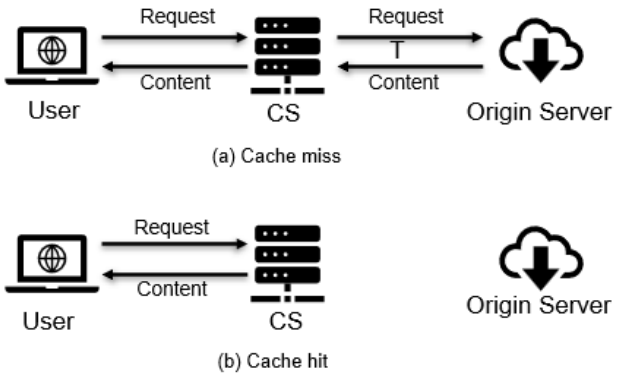

Fig. 1: Workflow of CS

Because each content has a different cache hit ratio, let h_i denote the cache hit ratio of content i. We can obtain the average response time of content i, W_i, by

$$W_i = h_i W_c + (1 - h_i)(W_c + T + W_o). \tag{2}$$

We use the Che-approximation [8] to predict the hit ratio h_i of each content i on the CS. Let C denote the capacity of the CS, and the maximum number of contents that can be stored in the CS is C. We assume that the average request arrival rate of content i is λ_i, and from the Che-approximation, we can obtain the cache hit ratio of content i, h_i by

$$h_i \approx 1 - e^{-\lambda_i t_c}, \tag{3}$$

where t_c is the characteristic time of the CS, and it is obtained by solving

$$\sum_{i=1}^{M} h_i = C. \tag{4}$$

3. Multilayer CDN Model

Multilayer CDN are designed to provide faster service and to defend against DDoS attacks to some extent [5], and we focus on this model to evaluate and analyze the CPA and DDoS against CSes in this paper.

The multilayer CDN model is composed of multiple independent CSes with multiple layers, and we assume CSes of two layers, L1 and L2, as shown in Fig.2. When a user requests a content, the CS accommodating the requesting user at L1 checks whether the requested content exists or not in its cache storage. If the requested content exists in the cache storage, the CS of L1 sends the requested content to the user. Otherwise, the request is forwarded to the CS of L2 connecting to the CS of L1. If the requested content does not exist in the CS of L2, the origin server which stores all M contents sends the requested content to the user.

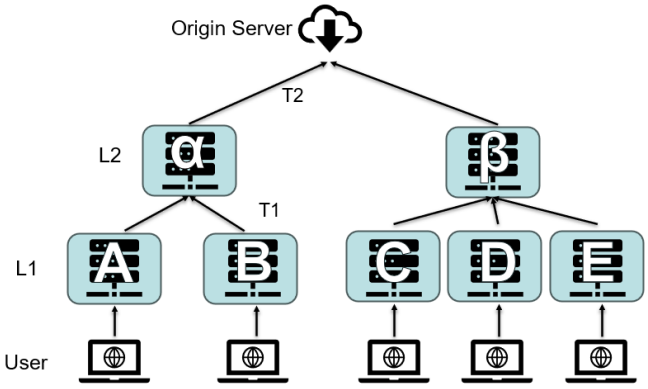

Fig. 2: Multilayer CDN Model

We further assume that there are two CSes, α and β, at L2, two CSes, A and B, at layer 1 connecting CS α, and three CSes, C, D, and E, at layer 1 connecting CS β. Let T_1 denote the latency between L1 CSes and L2 CSes, and T_2 denote the latency between L2 CSes and the origin server. Moreover, let W_A, W_α, and W_o denote the average response time of contents at CS A, CS α, and the origin server, respectively.

When the requested content exists in the CS A at L1, i.e., cache hit, the average response time in the model, r_A, is W_A. We define r_α as the average response time when the request is cache miss in CS A and forwarded to the CS α at L2. Moreover, we also define r_o as the average response time when the request is forwarded to the origin server. We can obtain r_α and r_o by

$$r_\alpha = W_A + W_\alpha + T_1, \tag{5}$$

$$r_o = W_A + W_\alpha + Wo + T_1 + T_2. \tag{6}$$

Let h_i^A denote the cache hit ratio of content i in CS A and h_i^α denote the cache hit ratio of content i in CS α. The average response time of content i when request arrives at CS A, $R_A(i)$, is obtained by

$$R_A(i) = h_i^A r_A + (1-h_i^A)h_i^\alpha r_\alpha + (1-h_i^A)(1-h_i^\alpha)r_o \tag{7}$$

By summing $R_A(i)$ among all i in M, we can obtain the average response time of request when accessing CS A by

$$R_A = \frac{\sum_{i=1}^{M} R_{a_i}}{M}. \tag{8}$$

In the same way, we can also obtain the average response time when accessing each of other CSes.

4. Genetic Algorithm

To investigate the potential threat of the CPA, we need to find th optimal strategy of attackers to maximize the average response time over all CSes in the system. Therefore, in this section, we propose a method of optically allocating the attacking resources among target CSes using the Genetic algorithm (GA). The GA is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms. Genetic algorithms are commonly used to generate high-quality solutions to optimization and search problems by relying on biologically inspired operators such as mutation, crossover and selection. Genetic algorithms rely on biologically inspired operators such as mutation, crossover and selection to achieve optimization which have been widely used in computer science [11] [12].

Algorithm 1: Genetic algorithm optimizing attacking strategy

1: Randomly set s_i^n as the initialized chromosome, which $0 < s_i^n < 1$
2: To achieve crossover, set any two chromosomes as a group and randomly exchange of s_i^n in groups
3: To achieve mutation, randomly change a s_i^n, which $0 < s_i^n < 1$
4: To achieve selection, add attacker's request to the normal request and calculate the average of R of each CS as fitness. Then select the best chromosomes from the parents and children.
5: Repeats steps 2, 3 and 4 until N generations.

In this paper, we use the GA to optimize an attacker's strategy to achieve the maximum damage an attacker can cause. Then, by comparing the best attack strategies under different parameters, we can find more efficient defense methods.

Let λ_T denote the maximum requests sent by the attacker, and s_i^n represents the proportion of requests that the attacker allocates to server n from λ_T. We can obtain the request λ_i^n

by

$$\lambda_i^n = \frac{s_i^n}{\sum_{n=A}^{E}\sum_{i=1}^{M} s_i^n}. \quad (9)$$

The optimal attack strategy can reflect how an attacker distributes requests to different servers in different content. We get the optimal attack strategy of the attacker by using the algorithm1.

5. Numerical Evaluation

We numerically evaluate the proposed optimization method through computer simulations, and we set the parameters of the simulation based on the convenience of the experiment and the authenticity of the data.

The setting parameters of the experiments are shown in Table 1. We assume contents provided by CDN occupy the same amount of space in the cache and have different request arrival rate depending on zipf law. We set the latency in basic case based on the realistic data [9].

Tab. 1: Simulation parameter setting in basic case

Paramater	Value
Number of contents provided, M	5
Cache size	2
Zipf law parameter	3
Total requests rate to CS, λ	100 /second
Attacker total requests rate, λ_T	150 /second
Av. service time of L1 CSes, $1/\mu_1$	3.3 ms
Av. service time of L2 CSes, $1/\mu_2$	3.3 ms
Av. service time of origin server, $1/\mu_o$	2.5 ms
Latency between L1 and L2, T_1	80 ms
Latency between L2 and origin server, T_2	40 ms

5.1 Definition of Attack and Evaluation Criterion

The optimal attack strategy can be obtained by using the proposed GA algorithm. We assume that requests sent by the CPA are indistinguishable from legitimate requests and cannot be detected. Because the CPA will make many requests for low-popularity content, the cache hit ratio of popular content will decrease, resulting in a longer response time for most users, or even the server cannot process the request.

Because CDNs are designed to reduce the latency, the latency does not have a noticeable impact on the user's experience. However, the CPA attacks will weaken the latency reduction effect of CDN, so the CPA makes obvious effects. According to the optimal attack strategy, the attacker's requests are distributed to different contents on different CSes and sent to the CDN disguised as normal requests. The optimal attack strategy will change with the change of other parameters. We evaluate the potential threat of the optimum attack strategy against CDN by P, the increase ratio of average response time (IRAR). Let R denote the average response time over all CSes, and we obtaine R by

$$R = \frac{R_A + R_B + R_C + R_D + R_E}{5}. \quad (10)$$

Let R^n denote the average response time without attack, and let R^a denote the average response time with optimal attack. We can obtain the IRAR P by

$$P = \frac{R^a - R^n}{R^n}. \quad (11)$$

5.2 Effect of Latency

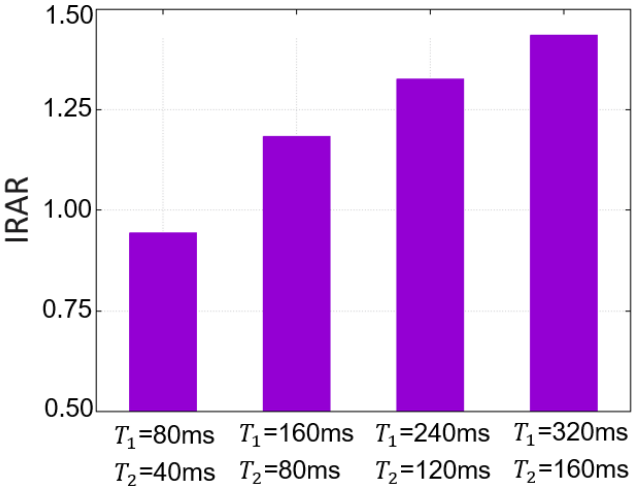

Fig. 3: Effect of Latency

Figure 3 shows the IRAR for four settings of latency T_1 and T_2. In the base case, the latency $T_1 = 80ms$, $T_2 = 40ms$, the average response time increased by 94.45% under the optimal attack strategy.

As the latency increased, the IRAR also increased. However, as the latency increased, the increase ratio of the IRAR decreased. So, in this CDN model, provider should try to keep latency at a low level, which can significantly reduce the impact of the attack. By analyzing the s_i^n in GA, we find that the optimal attack strategy for an attacker is to focus resources on a particular CS, usually a CS linked to CS β connecting more CSes at layer 1.

5.3 Effect of Service Rate

Service rate is often considered a measure of a server's performance, and it also affects the cost of a CDN. In order to save costs, many CDN provider do not set up very large performance redundancies, but according to our research, it can have a positive impact when faced with the optimal attack strategy. However, after our analysis in GA result, we found the reason of this result.

Figure 4 shows the IRAR under different service rate. Remarkably, as the service rate increased, the IRAR also increased. This seems very unreasonable, because in general, the performance of the server can be better defense against attacks.

Due to the very fast service rate, the response time W of each server was very small. If there was no attacks, this can significantly reduce the average response time. However, in the face of the attack, due to the low cache hit ratio, many requests had to be sent to the origin server, which will spend time on latency T_1 and T_2. We find that as the service rate increased, the average response time in the absence of the attack decreased significantly, while the average response time under attack decreased limitedly, so the IRAR increased.

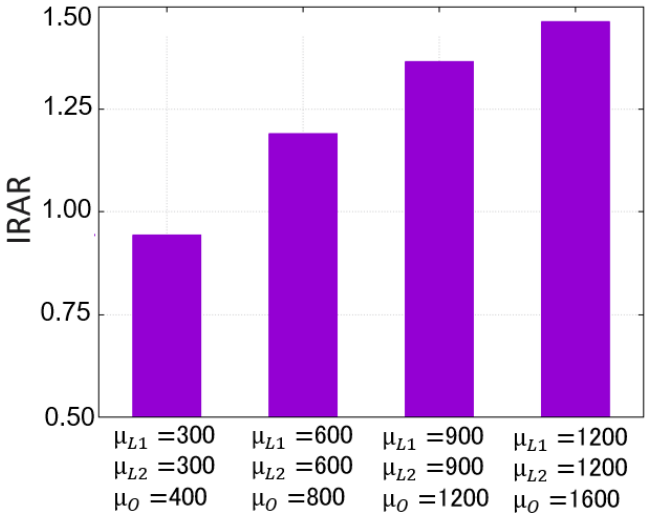

Fig. 4: Effect of service rate

By analyzing the optimal attack strategy, we found that with the increase of the processing rate, the strategy changed from attacking a single CS to attacking all CSes equally, which indicated that the multilayer CDN model can defend the attack to some extent in this case.

While increasing service rates generally improves CDN performance, it is important to be aware of the buckets effect in the face of attacks. Especially in time-sensitive CDN, a significant change in response time when attacked can lead to a bad experience.

5.4 Effect of Attack-Request Rate

The attacker's request rate is often considered to be the most direct factor affecting the effectiveness of the attack, and very high request rates can cause the same effects as DDoS attacks, but also increase the risk of detection.

Figure 5 shows the IRAR for four values of λ_T, the total request rate of attacker. The response time increased exponentially with the increase of the attacker's total request rate. When λ_T reached 200, the request rate of the server is greater than the processing rate and the server cannot process the request. Since a high attacker request rate directly causes the server to be unable to process the request, and the impact is exponential, we believe it is important to keep the attacker request rate low.

However, attackers who want to achieve higher request rates also bear a greater risk of detection. Because many existing studies have proposed to detect CPA based on the characteristics of the attacker's request which also shows that it is very important for CDN providers to deploy CPA detection mechanisms [13] [14]

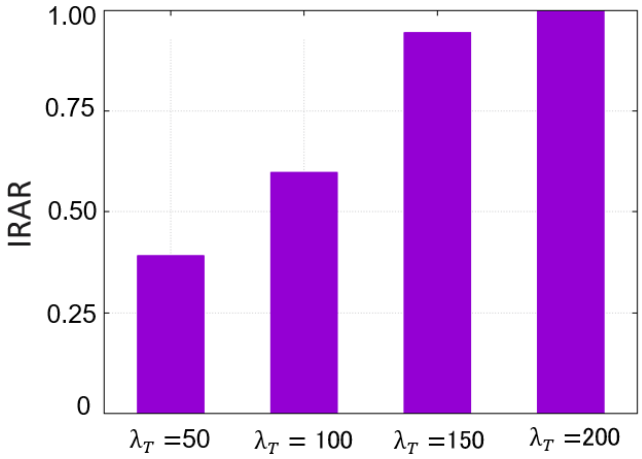

Fig. 5: Effect of total request rate of attacker

5.5 Effect of Zipf Parameter

Zipf law parameter is often used to reflect whether the popularity of the content is concentrated, and the greater the parameter, the more concentrated the request is in the popular contents. It usually depends on the type of content provided.

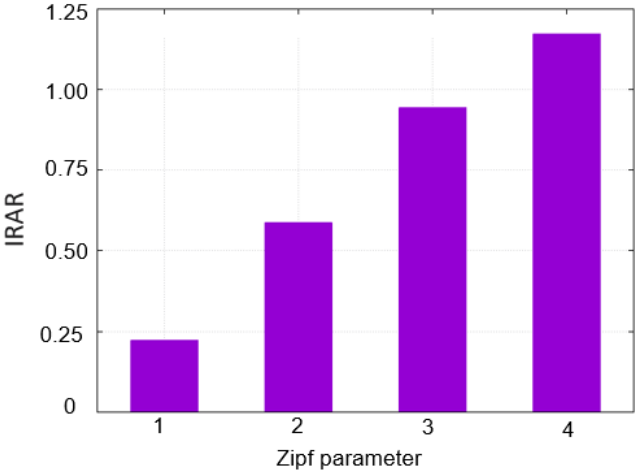

Fig. 6: Effect of zipf parameter

Figure 6 shows the IRAR for four values of Zipf parameter. As the Zipf parameters increased, the impact of the attack also increased. The greater the parameter, the more concentrated the popularity; otherwise, the more dispersed the popularity. When the popularity is concentrated, it will benefit more for the CSes, and it will be more vulnerable to CPA.

By analyzing the results of the GA, we found that the attacks were more inclined to concentrate when the Zipf parameter was high, and more inclined to disperse when the Zipf parameter was low. Although it is difficult for a CDN provider to change the Zipf parameter, the cost of defending against attacks can be adjusted based on the Zipf parameter.

6. Conclusion

This paper focused on a multi-layer CDN model and used the M/M/1 queuing model to derive average response time for content requests. The impact of the CPA was measured by the increase ratio of average response time, considering both cache hits and cache misses, with assuming the LRU cache replacement policy.

The genetic algorithm (GA) was employed to find the optimal attack strategy that maximized the average response time by distributing attack requests strategically across different contents and servers. The GA operated through initialization, crossover, mutation, and selection processes, iterating through generations to evolve the optimal attack strategy. Numerical evaluations were conducted under various scenarios to understand the factors influencing the effectiveness of CPA and we can summarize the main finding as follows

1. Latency: Higher latency between CDN layers and the origin server increased the impact of CPAs. The optimal attack strategy often targeted specific cache servers to exploit this latency.

2. Service Rate: Surprisingly, increasing the service rate of servers can result in a higher proportional increase in response time under attack. This occurred because faster service rates significantly reduced the response times under normal conditions, making the relative impact of attacks more pronounced.

3. Attacker Request Rate: The response time increases exponentially with the attacker request rate. High request rates can overwhelm servers, leading to a denial of service, although they also increased the risk of detection.

4. Zipf Parameter: Higher Zipf parameter, indicating more concentrated popularity of content, resulted in greater vulnerability to CPAs. This is because popular content benefited more from caching, making them more susceptible to cache pollution.

This paper concluded that understanding the attacker optimal strategy through GA can help CDN providers identify weak points and enhance their defense mechanisms. By analyzing various parameters, CDN providers can better allocate resources to mitigate the impact of CPAs, ultimately ensuring more robust and resilient CDN services.

ACKNOWLEDGEMENTS

This work was supported by JSPS KAKENHI Grant Number 23K21664 and 23K21665.

References

[1] M. Ghaznavi, E. Jalalpour, M. A. Salahuddin, R. Boutaba, D. Migault and S. Preda, "Content Delivery Network Security: A Survey," in IEEE Communications Surveys & Tutorials, vol. 23, no. 4, pp. 2166-2190, Fourthquarter 2021

[2] L. Yao, Z. Fan, J. Deng, X. Fan and G. Wu, "Detection and Defense of Cache Pollution Attacks Using Clustering in Named Data Networks," in IEEE Transactions on Dependable and Secure Computing, vol. 17, no. 6, pp. 1310-1321, 1 Nov.-Dec. 2020

[3] K. Kim, Y. You, M. Park and K. Lee, "DDoS Mitigation: Decentralized CDN Using Private Blockchain," 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, 2018

[4] Emma Ryan, "Website Load Time Statistics: Why Speed Matters in 2024" , Website Builder Expert, 2021. [Online]. Available: https://www.websitebuilderexpert.com/building-websites/website-speed. [Accessed: Jan- 2024]

[5] Sankalp Basavaraj , "Multi-Cloud, Multi-CDN: The Future of the Internet" , Medium, 2022. [Online]. Available: https://labs.ripe.net/author/sankalp-basavaraj/multi-cloud-multi-cdn-architecture-a-deceptive-future-of-the-internet. [Accessed:Jan- 2024]

[6] Z. Zeng and H. Zhang, "A Study on Cache Strategy of CDN Stream Media," 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2020, pp. 1424-1429,

[7] J. Liu and N. Kamiyama, "Investigating Impact of DDoS Attack and CPA Targeting CDN Caches," 2024 IEEE/IFIP International Wworkshop on Analytics for Nerwork and Service Management (AnNet), Seoul, Korea, May 2024

[8] H. Che, et al., "Hierarchical Web Caching Systems: Modeling, Design and Experimental Results," IEEE J. Selected Areas of Commun., vol.20, no.7, Sep. 2002

[9] R. K. Thelagathoti, S. Mastorakis, A. Shah, H. Bedi and S. Shannigrahi, "Named Data Networking for Content Delivery Network Workflows," 2020 IEEE 9th International Conference on Cloud Networking (CloudNet), Piscataway, NJ, USA, 2020

[10] K. Wang, Y. Wu, J. Chen and H. Yin, "Reduce Transmission Delay for Caching-Aided Two-Layer Networks," 2019 IEEE International Symposium on Information Theory (ISIT), Paris, France, 2019

[11] P. Guo, X. Wang and Y. Han, "A Hybrid Genetic Algorithm for Structural Optimization with Discrete Variables," 2011 International Conference on Internet Computing and Information Services, Hong Kong, China, 2011

[12] N. Rikatsih and W. F. Mahmudy, "Adaptive Genetic Algorithm Based on Crossover and Mutation Method for Optimization of Poultry Feed Composition," 2018 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, Indonesia, 2018

[13] L. Yao, Y. Zeng, X. Wang, A. Chen and G. Wu, "Detection and Defense of Cache Pollution Based on Popularity Prediction in Named Data Networking," in IEEE Transactions on Dependable and Secure Computing, vol. 18, no. 6, pp. 2848-2860, 1 Nov.-Dec. 2021

[14] Q. Xu, Z. Su, K. Zhang and P. Li, "Intelligent Cache Pollution Attacks Detection for Edge Computing Enabled Mobile Social Networks," in IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 3, pp. 241-252, June 2020