Poster: Optimized Cache Pollution Attack: A DDoS Vector in Content Delivery Networks

Jiaqi Liu
Graduate School of Information Science and
Engineering Ritsumeikan University
Ibaraki, Osaka, Japan
is0705vf@ed.ritsumei.ac.jp

Kamiyama Noriaki
College of Information Science and Engineering
Ritsumeikan University
Ibaraki, Osaka, Japan
kamiaki@fc.ritsumei.ac.jp

ABSTRACT

Content Delivery Network (CDN) enhances Internet scalability and resilience against traditional Distributed Denial-of-Service (DDoS) attacks by deploying cached edge servers close to end-users. However, CDNs remain vulnerable to unique attack vectors, such as cache pollution attack (CPA). While existing research focuses on attack detection and mitigation, this paper investigates optimized CPA strategies from an adversarial perspective, specifically targeting CDN origin servers to bypass edge server defenses and amplify DDoS impacts. We analyze the attack mechanisms, evaluate performance degradation under varying CPA parameters, and quantify the scalability of this DDoS vector. Experimental results demonstrate that optimized CPA can exhaust origin server resources with fewer malicious requests, highlighting a pressing need for robust CDN architectural countermeasures.

CCS CONCEPTS

• **Security and privacy → Denial-of-service attacks**.

KEYWORDS

CDN, CPA, DDoS, Hierarchical Cache System

ACM Reference Format:
Jiaqi Liu and Kamiyama Noriaki. 2025. Poster: Optimized Cache Pollution Attack: A DDoS Vector in Content Delivery Networks. In *The 31st Annual International Conference on Mobile Computing and Networking (ACM MOBICOM '25), November 4–8, 2025, Hong Kong, China.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3680207.3765658

ACM MOBICOM '25, November 4–8, 2025, Hong Kong, China
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1129-9/2025/11
https://doi.org/10.1145/3680207.3765658

1 INTRODUCTION

A Content Delivery Network (CDN) is a globally distributed server network designed to deliver content to users more efficiently and rapidly. It achieves this by strategically deploying multiple nodes worldwide to cache content. The CDN operates by distributing copies of content to servers across diverse geographical locations. When a user requests content, the CDN serves it from the nearest server to the user, thereby reducing data transmission distance and accelerating content delivery.

Key components of a CDN include cache servers located near users to rapidly deliver cached content; caches, which temporarily store recently accessed content to reduce retrieval latency; and the origin server, which is only queried when requested content is unavailable in edge caches. While caching is critical to CDN functionality, it introduces vulnerabilities. Although cache servers face risks similar to other servers, such as Distributed Denial-of-Service (DDoS) attacks, the origin server is shielded from direct attacks due to its exclusive communication with cache servers. However, CDNs are particularly susceptible to cache-specific threats, such as Cache Pollution Attack (CPA). Once a CDN cache is compromised, its performance and response time may degrade significantly, while unexpected traffic surges redirected to the origin server impose sudden operational pressure.

Despite extensive research on detecting and mitigating CPA[5][1], analyzing attack characteristics from an adversarial perspective may yield novel insights. Based on [4], this paper will conduct research on the hierarchical cache system by using the genetic algorithm.

Overall, our contributions are as follows:

- We proposed an M/M/1 model applicable to hierarchical cache system.
- We proposed a genetic algorithm scheme for optimizing the attacker's strategy.
- We analyzed the distribution characteristics of the attacker's requests and other features under the optimal attack strategy.

2 ANALYTICAL MODEL

2.1 M/M/1 queue model

We define μ as the mean service time following an exponential distribution, M as the number of content items provided within the CDN, and λ_i^e as the Poisson arrival rate of requests for content i in cache server. Thus, we can use the $M/M/1$ queueing model formula, which is widely used for simulating system operation, to derive the average response time W for the CDN by

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

The request rate redirected from cache servers to the origin server for content i, λ_i^o can be calculated using the cache hit ratio h_i by

$$\lambda_i^o = h_i \lambda_i^e. \tag{2}$$

We use the *Che* approximation to estimate the cache hit ratio h_i for content i on the cache server[2]. Let C represent the cache size of the server, meaning the maximum number of content it can store. According to the *Che* approximation, the cache hit ratio h_i can be expressed as

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

where t_c is the characteristic time of the cache server. With equation (3), t_c can be calculated using $\sum_{i=1}^{M} h_i = C$.

2.2 Hierarchical Cache System

Compared to conventional CDNs, the hierarchical cache system introduces a central server between edge servers and the origin server to buffer requests from edge servers as shown in Fig.1. Notably, requests reaching the central server can be considered analogous to those targeting the origin server in standard CDNs.

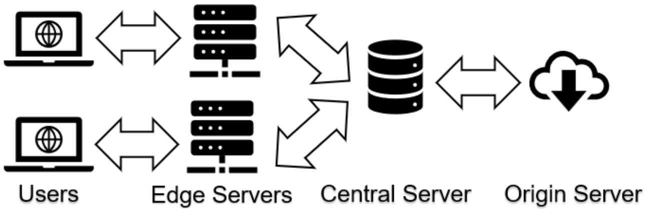

Figure 1: CDN of hierarchical cache model

Network latency constitutes a significant portion of the average response time. We define T as the network latency between the cache server and the origin server. The latency between edge servers and the central server is defined as T_1, and the latency between the central server and the origin server is defined as T_2. As validated in [3], *Che* approximation is applicable to multi-layer LRU caching systems. Based on Equations (1) and (2), the average response times are derived as W_e for edge servers, W_c for the central server, and W_o for the origin server. Consequently, the average response time for a user requesting content from an edge server falls into three scenarios: edge server cache hit r_e, central server cache hit r_c, and origin server access r_o. It's obvious that $r_e = W_e$. r_c and r_o can be expresses by

$$\begin{aligned} r_c &= W_e + W_c + T_1, & (4) \\ r_o &= W_e + W_c + Wo + T_1 + T_2. & (5) \end{aligned}$$

Let h_i^e denote the cache hit ratio for content i at the edge server, and h_i^c represent the cache hit ratio for content i at the central server. The expected average response time R_i for content i under different scenarios is expressed as

$$R_i = h_i^c r_e + (1 - h_i^c) h_i^e r_c + (1 - h_i^e)(1 - h_i^c) r_o. \tag{6}$$

By incorporating the popularity distribution of content, Rm the overall expected average response time across all content is given by

$$R = \sum_{i=1}^{M} \frac{\lambda_i R_i}{\sum_{i=1}^{M} \lambda_i}. \tag{7}$$

3 OPTIMUM ATTACK AGAINST CDN CACHES

Let λ_T represent attack capacity, which is the total request rate that the attacker can generate, and s_i^n represent the proportion of requests the attacker sends for content i in edge server n relative to the total request rate. Thus, the request rate λ_i^n for content i in edge server n sent by the attacker can be expressed as follows

$$\lambda_i^n = s_i^n \lambda_T. \tag{8}$$

The optimal attack strategy will reflect how the attacker allocates limited resources across different servers to achieve their objective. In the GA, the average response time of the origin server W_o is adopted as the fitness, reflecting the attacker's objective to maximize the volume of requests reaching the origin server. We use algorithm 1 to implement the GA and obtain the attacker's optimal strategy.

Algorithm 1 Optimum allocation of sending rate of attack packets on each CDN edge server

```
Randomly set $s_i^n$ as the initialized chromosomes
To achieve crossover, set any two chromosomes as a group and randomly exchange of $s_i^n$ in groups
To achieve mutation, randomly change two $s_i^n$
To achieve selection, add attacker's request to the normal request and calculate $W_o$ as fitness. Then select the best chromosomes from the parents and children
Repeat steps 2, 3 and 4 until $N$ generations
```

4 NUMERICAL EVALUATION

We conduct computer simulations to evaluate CDN performance variations under cache pollution attacks. Our hierarchical cache system model contains 10 edge servers, each receiving 200 normal requests per second. The system provides 10 content items of equal size, with content popularity following Zipf's distribution($\theta = 4$). Cache capacities are set to 3 for edge servers and 6 for central servers. Service times are configured at 1.67ms for edge/central servers and 3.33ms for the origin server. Network latency is modeled with 500ms between edge and central servers(T_1), and 300 ms between central and origin servers(T_2). All cache servers implement LRU replacement policies, and content request rates are calculated based on the Zipf popularity distribution.

4.1 Attack Request Distribution

With 5 edge servers, Fig.2 shows two λ_T-dependent attack strategies: beyond threshold, concentrated targeting of specific edge servers (a); below threshold, uniform distribution across all servers (b). Iterative λ_T subdivision determines this strategic transition threshold.

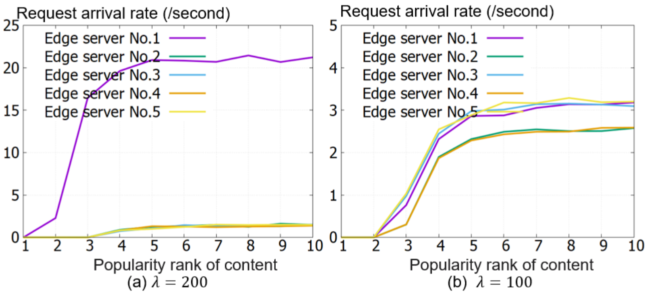

Figure 2: Attack request distribution

Both strategies prioritize targeting low-popularity but attackers avoid saturating requests toward a single unpopular item. Instead, they distribute requests smoothly across multiple low-popularity items to prevent high traffic that might reduce attack efficacy.

4.2 Cache hit ratio and GR

To quantify the DDoS impact of CPA, the analysis focuses on the request rates redirected to the origin server and the central server. The growth rate of average response time (GR) is introduced to measure performance degradation under attack scenarios. The GR is defined as $GR = \frac{R-R^n}{R^n}$ where R^n and R denote the average response time under normal and attack conditions, respectively.

Figure 3(a) shows cache hit ratio variations for the top-four content items under CPA. Without attack ($\lambda_T = 0$), ratios follow Poisson distribution across all ten items. Under attack ($\lambda_T > 0$), ratios for rank 2-3 content decrease substantially, while rank-1 content shows marginal degradation. The consistent behavior observed in lower-ranked items empirically demonstrates CPA-induced cache efficiency degradation and upstream traffic amplification.

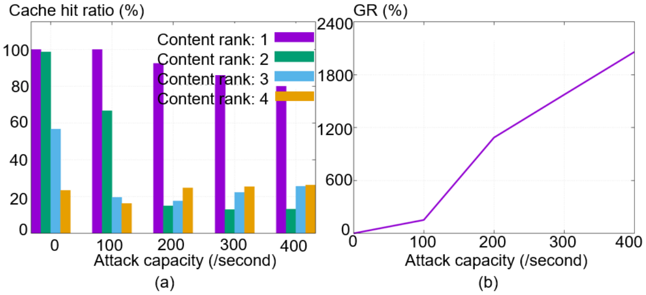

Figure 3: (a) Cache hit ratio and (b) GR against attack capacity

Figure 3(b) analyzes GR versus λ_T. At $\lambda_T = 100$, uniform attack distribution yields low GR. When $\lambda_T = 200$, optimal strategy shifts to single-server targeting, triggering sharp GR surge on victim servers. Nontargeted servers show linear GR increase from residual attack traffic.

5 CONCLUSION

This study demonstrated that CPA pose a significant threat to hierarchical CDNs, even under normal operational conditions with minimal upstream server traffic. By optimizing attack strategies via a genetic algorithm, we revealed that attackers can bypass the isolation of edge servers in CDN through CPA and carry out DDoS attacks on the origin server.

ACKNOWLEDGMENTS

This research was supported by JSPS KAKENHI Grant Numbers 25K03113 and 23K28078.

REFERENCES

[1] Takakazu Ashihara and Noriaki Kamiyama. 2021. Detecting Cache Pollution Attacks Using Bloom Filter. In *2021 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN)*. 1–6. https://doi.org/10.1109/LANMAN52105.2021.9478804

[2] Hao Che, Ye Tung, and Zhijun Wang. 2002. Hierarchical Web caching systems: modeling, design and experimental results. *IEEE Journal on Selected Areas in Communications* 20, 7 (2002), 1305–1314. https://doi.org/10.1109/JSAC.2002.801752

[3] Christine Fricker, Philippe Robert, and James Roberts. 2012. A versatile and accurate approximation for LRU cache performance. In *2012 24th International Teletraffic Congress (ITC 24)*. 1–8.

[4] Jiaqi Liu and Noriaki Kamiyama. 2024. Investigating Impact of DDoS Attack and CPA Targeting CDN Caches. In *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. 1–6. https://doi.org/10.1109/NOMS59830.2024.10575854

[5] Lin Yao, Zhenzhen Fan, Jing Deng, Xin Fan, and Guowei Wu. 2020. Detection and Defense of Cache Pollution Attacks Using Clustering in Named Data Networks. *IEEE Transactions on Dependable and Secure Computing* 17, 6 (2020), 1310–1321. https://doi.org/10.1109/TDSC.2018.2876257