Recent Developments in Content Delivery Network: A Survey

Jiaming Zhao[1], Pingjia Liang[1], Weiming Liufu[1], and Zhengping Fan[2](✉)

[1] School of Data and Computer Science, SUN Yat-Sen University, Guangzhou, China
[2] School of Intelligent Systems Engineering, SUN Yat-Sen University, Guangzhou, China
fanzhp@mail.sysu.edu.cn

Abstract. With the development of computer networks, the amount of network information continues to grow. To facilitate the transfer of the increasing information, a content distribution network (CDN) is developed by adding an intermediate layer on the existing network. Technically, Caching strategy of CDN is the most important mechanism, which heavily impacts the CDN performance. On the other hand, considering the cost of operating CDN, some strategies have been proposed, aiming to save the CDN cost in terms of, e.g., power energy. This paper makes a brief review on the recent developments of CDN in terms of its caching strategy and operation cost, and discusses some potential development directions of CDN.

Keywords: Content delivery network · Cache strategy · Cost

1 Introduction

In the past decades, with the wide use of the Internet and the mobile network, the amount of Internet information has been increasing at an explosive rate. A report by Cisco network company [1] has shown that the Internet traffic may grow nearly three times in the next five years. It also indicates that by 2021 the Internet traffic of the entire world will be 127 times of that of 2005. Meanwhile, the traffic carried by CDN [2] will account for 71% of all Internet traffic by then. Technically, CDN caches files near the user location, which significantly accelerates the response speed and thus reducing the delayed time.

The key strategies of CDN are the placement of replica server and content caching strategy. The placement of the Replica [3] is to find the best location from some candidate client nodes for the Replica by maximizing the CDN performance. For the content caching strategy, contents have different popularity, CDN needs to determine which caches should be chosen to store a given content such that the end users can reach the content with a high speed and less delayed time. For example, a fluid queue model in CDN was proposed by balancing the content caching strategy with redirect proximity in [4]. For each redistributed request, CDN utilizes the difference between the computational caches to select the appropriate replica server. In addition, since the algorithm limits the migration distance of each request, the latency cost is also greatly reduced. Note that the optimization method can also be used to determine the content location.

H. Shen and Y. Sang (Eds.): PAAP 2019, CCIS 1163, pp. 98–106, 2020.
https://doi.org/10.1007/978-981-15-2767-8_9

For example, the method of optimizing the CDN content caching strategy was proposed in [5], which significantly improves the CDN cache hit rate. A hybrid integer linear programming (MILP) optimization model was proposed in [6], which considered three issues of replica server placement and content caching and allocations.

Operation cost is another factor that should be considered in designing CDN. Although some strategies such as increasing server cache capacity, can improve the performance of CDN, it definitely increases the operating cost of service providers, creating an additional burden. Therefore, reducing cost of CDN is also very important for CDN providers.

In this paper, we give a brief review on the recent developments of CDN by focusing our attention on the impact of CDN caching strategy and operation cost. The paper is organized as follows. In Part Two, we investigate several caching strategies. In Part Three, we discuss the effect of operating costs on CDN and explore how CDN performance can be optimized at a limited cost. Finally, we conclude our review in Part Four.

2 Cache Strategy

In CDN, nodes often cooperate with each other. One of the most important things is to choose which node to cache which file. Common caching strategies are LCE, LCD, etc. [7]. Most of the caching nodes selected by these strategies are based on the path of content transmission. For example, LCE caches content at every point of the content transmission path. These strategies are intuitive and easy to implement, but the performance is often not satisfied. In order to adapt to the network structure to achieve better cache performance, some new strategies are proposed, in which cache nodes are chosen according to centrality, or have cooperation among caches.

2.1 Centrality-Measures Based Algorithm

It is shown [8] that better performance can be achieved by caching on a subset of content routers rather than on all routers in the content delivery path. The selection of a subset of content routers should achieve the goal of maximizing cache performance. Based on this observation, an algorithm based on centrality measure was proposed. The centrality can be determined by:

Closeness-centrality is the inverse of farness. The farness of a content router (CR) is defined as the summation of its shortest path distances to all other CRs. a CR which contains high Closeness-centrality is the most central CR in the network

Reach-centrality (RC) defines how many numbers of hops, a particular CR reaches to another CR in a network topology. A CR having a high value of RC indicates that it can reach to other CRs in a less hop count. A CR having a high value of RC can easily reach to the consumers with less number of hops.

Degree-centrality plays a very important role in Centrality-measures. This value of centrality defines how central a CR is in terms of nearby CRs. A CR having high degree contains a large number of the CRs in its range, and so it can easily distribute content items among various CRs to satisfy the customer requests.

Betweenness-centrality defines how many times a particular CR lies on the path between a consumer and server.

When delivering a user's request package, the maximum centrality value and the corresponding CR's ID that the package has passed through are recorded. All passed CRs need to check and decide whether to modify the request package information according to centrality. When the request packet is responded, the data packet will have the CR's ID with the largest centrality. When the data packet returns along the original path, if CR is the same router as the ID stored in the data packet, it caches the content, otherwise it does not cache, but only transfers packet. For example, in the Fig. 1, the request package is sent from A to D, and the Table 1 shows the changes in the package content during the process.

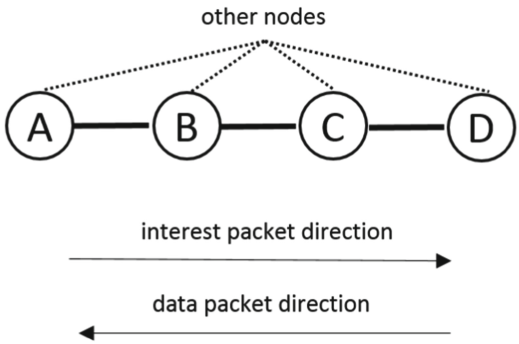

Fig. 1. Packet delivery path

Table 1. Changes in the contents of the packet

Node	A	B	C	D
Centrality	2	3	5	4
(ID, centrality) in interest parket	(A,2)	(B,3)	(C,5)	(C,5)
Data packet cached	×	×	√	×

As can be seen from the above, if the network structure is taken into account, the caching performance is improved. Note that in a real network it may be grouped into some clusters. In such networks, one can develop a community-based caching strategy, which may further improve the CDN performance.

2.2 Cooperative Caching

In [9], the author studies the user-centered cooperative edge caching problem in content delivery networks to improve the quality of experience by utilizing service provisioning at the edge of the network and minimizing end-to-end latency. They introduced a caching algorithm, a group of small base stations (SBS) collaboratively shared storage, and jointly decided on the caching strategy to cache as much content as possible under capacity constraints. In collaborative caching, nearby SBS forms a group and caches files based on user preference matrix. The purpose is to use the storage capacity of all SBS in the group to cache all files. A portion of each SBS storage space is used to cache the most popular files, while the rest is used to cache the most popular files that are not cached by other nodes (Fig. 2).

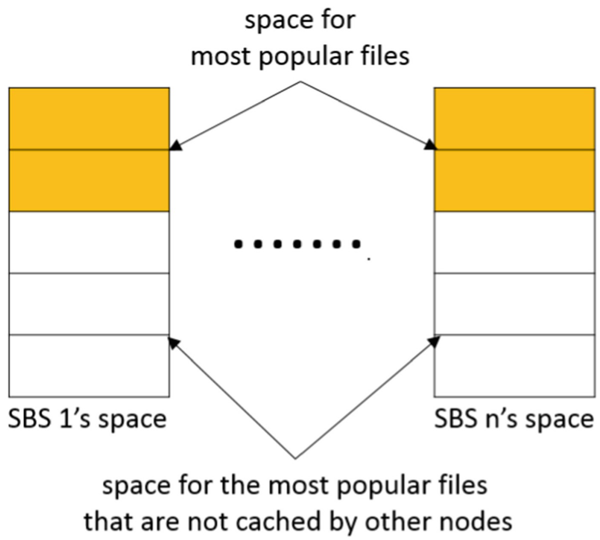

Fig. 2. SBS caching algorithm

In addition, in the user-centered delivery scheme, an improved matching theory is used to match users and SBS to ensure that more users can be served satisfactorily. Usually, a user requests a service from the nearest SBS, and if it has the requested file and sufficient service channels, the service will serve. Otherwise, other SBSs in the pre-allocation group will automatically check whether they meet the two requirements of serving the user. If there is no SBS in the group that can provide such services, the user request will be directed to the path that is not cached.

2.3 Mobile Edge Caching

The traditional CDN-content-based allocation mechanism is usually designed for the traditional wired communication network architecture. However, in today's increasingly mobile networks, resources (e.g., storage, bandwidth and computing power) and the location of deployed servers are limited. More importantly, due to content dynamics, user mobility and the limitation of the number of users in the cell, the hit rate of cached content in the mobile network may be very low. In [10], the author studies content distribution in the rapidly developing mobile network. They proposed a learning-based cooperative caching framework for mobile edge caching servers, which does not require prior knowledge of content requirements and content popularity matrices. They first use artificial neural networks to observe real-time requirements over a period of time, and then represent the content cache of the minimum latency problem as a $0-1$ integer programming problem. Furthermore, they prove that the problem is NP-hard and propose a greedy algorithm to solve it.

In the Vehicular content networks (VCN), it is similar to the mobile network. It is pointed out that due to the mobility of vehicles, it is quite inefficient to establish end-to-end connections in VCNs [11]. Therefore, content packages are usually sent back to requesting nodes through different paths in VCNs. The network performance of VCNs can be improved if the vehicle acts as a relay and carries data by using the mobility of nodes. In order to achieve this, the urban area is divided into different hot spots according to the way users travel, and these areas can be adjusted according to the dynamic vehicle density. Finally, popular content are only cached on nodes that frequently visit hot spots.

In VCN, the roadside unit (RSU) caches content at the edge of the network to facilitate timely delivery of content to the train when requested [12]. Here a model for vehicles is developed to determine whether to obtain the requested content from other mobile vehicles or fixed RSUs on the edges of VCN. When mobile vehicles issue content requests, mobile vehicles can intelligently select other vehicles or fixed RSUs to connect to retrieve the content of requests, and thus greatly reducing transmission delay. For a fixed RSU, an edge caching scheme based on cross-entropy is proposed to determine the content to be replaced when the space is full on the basis of the request decision of the vehicle in its coverage. When the RSU cannot provide the requested content solely, it can identify and recommend its collaborative RSU that the content should be transferred to, and then this collaborative RSU send the content to the vehicle.

3 The Cost of CDN

In CDN deployment, it definitely needs energy. How to save energy is another very important issue in operating CDN.

3.1 The Energy Consumption Cost of CDN

A CDN is a large distributed system that consists of hundreds of thousands of servers [13]. These servers are implemented as clusters, which consume a lot of power in the content delivery system [14]. An intuitive way to reduce energy consumption is to reduce the number of CDN servers, but this will lead to performance degradation and other cost increases accordingly. The most of methods in reducing energy consumption of traditional CDN is to reduce cluster energy consumption by "adjusting" its service capacity [13, 15–17]. Its basic idea is to switch idle servers to energy-saving mode when the load is low, so as to reduce energy consumption [13]. However, it is pointed out [13] that in traditional CDN where a large number of edge servers are deployed, reducing service capacity will increase the flow of data between ISPs, which results in more cross-ISP traffic expenditure. He et al. [13] proposed a capacity allocation algorithm based on the workload prediction, especially considering ISP traffic expenditure. Through this method, the overall operating cost of CDN is reduced and frequent server switching is effectively avoided. That is, the traffic between ISPs is reduced. Another scheme is proposed on the basis of smart grid technology [14]. Here the smart grid technology with low complexity is integrated with online Lyapunov optimization for an energy/QoE efficient CDN. Compared with other methods without real-time energy management, this method can converge to the optimal convergence point at a faster speed. In fact, the dynamic prediction and real-time management methods are very common ideas to solve the problem of CDN energy consumption. Especially with the emergence and development of cloud-based CDN, real-time management method has more development space. The main reason is that cloud-based vCDNs are more flexible so that the size of CDNs can be dynamically adjusted to reduce energy consumption. Liao et al. [18] proposed an approximate algorithm of maximum flow prediction (MMF) by combining dynamic prediction and real-time management. This method can determine the best capacity of CDN components in real time, and dynamically adjust the scale of CDN to reduce energy consumption.

3.2 The Delivery Cost and Storage Cost of CDN

In CDN, when a request asking for a movie arrives at some node v, the CDN may select any other node u (as in a P2P network) which currently has a copy of that movie, and instructs u to send a copy to v [19]. In this process, both the movie content sent by u to v and the control information sent by CDN to u (although this is almost negligible compared with the bandwidth occupied by the content sent by u) occupy bandwidth, which constitutes the delivery cost of CDN. In order to reduce the delivery cost, two solutions are often given (as showed in Fig. 1): (1) to make u closer to v [19], and (2) to choose a better transit path [20], such as a transit route with better performance and lower price. In solution one, due to the limited caching space of the nodes near v, when there are many requests for different files, some of the cached contents need to be replaced. If more files are wanted to be close to v, the storage capacity of node v has to be increased, which leads to an increase in storage costs.

On the other hand, in the solution one, there is an obvious problem- the balance between the delivery cost and storage cost. It has been shown that finding a cache placement method that minimizes both costs is NP-hard [19], which is also confirmed by [21]. To deal with this, an O (log δ)- competitive algorithm is proposed in [19], where δ is the normalized diameter of the network. The caching strategy is: If node v gets a copy of the file from u, the time of the file kept in node v is positively correlated with the distance between u and v. In [21], a genetic algorithm (GA) is applied to solve the problem of dynamically placing copies to minimize the total cost including storage and delivery costs. Here the needed solved problem is expressed as a mixed integer programming (MIP) problem that takes into account the service level agreement (SLA) of CDN and the multicast transfer feature for the delivery. Compared with the current popular optimization algorithms, including random add, random delete, random delete all, zero greedy delete and one greedy delete, the GA algorithm is superior to these algorithms in reducing the total cost of delivery and storage (Fig. 3).

For the solution two, that is, to choose a better transit path, this method has nothing to do with the storage cost, but can reduce the delivery cost. It is clearly that the solution two is compatible with the solution one, which indicates that the solution two can be further

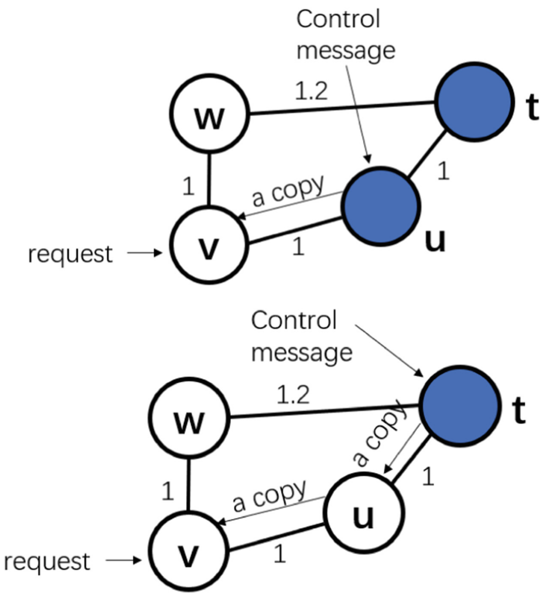

Fig. 3. The node t-w is CDN servers. The dark node indicates that the requested file is cached by the node. The white node indicates that the node has no requested file. The weight of the connection between the nodes indicates the cost of the file when it passes through the path. In Idea 1, considering the left and right graphs, the cost of the left graph is 1, and the cost of the right graph is 2. When the node nearer to v has the file, the delivery cost is lower. However, the cost of route is also very important. According to idea 2, considering the right figure, when t sends information to v, the cost of selecting path t-u-v is lower than that of path t-w-v.

optimized on the basis of the solution one. One application of such idea is discussed in [20]. In the Internet, there are many Internet Exchange Points (IXPs), which connect a large number of ISPs. But as mentioned in [13], this kind of content transmission across ISPs incurs a high cost, and thus it is particularly important to choose a lower cost transmission path. Through the optimal selection strategy, CDN deployed on IXPs can reduce transmission cost by 57% on average without sacrificing performance.

4 Conclusion

Nowadays, CDN has evolved from the original blank to the current bloom, and the role of CDN has become more prominent. In latest years, there have also been derivations of peer-assisted CDN [22], cloud-based CDN [23, 24], etc., all aimed at improving the performance of CDN. However, No matter what CDN have been developed, the most two important technical issues are the placement of the replica server and the selection of the content caching strategy. On the other hand, from the standpoint of CDN providers, the operating cost should be reduced as much as possible. In this paper, we have given a brief survey on CDN in terms of these topics. In the days to come, there may be more variants of CDN. For example, we can combine various topological properties of complex networks into CDN to improve the performance of CDN. The question is: Which topological property of a network should be chosen to achieve the best performance of CDN? It is worth of further investigation in our work.

References

1. Cisco. https://www.cisco.com/c/dam/global/zh_cn/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360-CN.pdf?oid=wprsi001573&elqTrackId=8c75641ee1b443fea18d8278a77ea927&elq=&elqaid=5964&elqat=2&elqCampaignId=&elqcst=272&elqcsid=2374&_gscu_=69512337n7nuz341&_gscs_=t69815000lsxmmb41. Accessed 20 Sept 2019
2. Stocker, V., Smaragdakis, G., Lehr, W., Bauer, S.: The growing complexity of content delivery networks: challenges and implications for the internet ecosystem. Telecommun. Policy **41**(2), 1003–1016 (2017)
3. Sahoo, J., et al.: A survey on replica server placement algorithms for content delivery networks. IEEE Commun. Surv. Tutorials **19**(2), 1002–1026 (2016)
4. Shuai, Q., Wang, K., Miao, F., Jin, L.: A cost-based distributed algorithm for load balancing in content delivery network. In: 2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, pp. 11–15. IEEE (2017)
5. Kyryk, M., Pleskanka, N., Pleskanka, M.: The analysis of the optimal data distribution method at the content delivery network. In: 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), Polyana, Ukraine, pp. 1–4. IEEE (2019)
6. Xu, K., Li, X., Bose, S.K., Shen, G.: Joint replica server placement, content caching, and request load assignment in content delivery networks. IEEE Access **6**(2), 17968–17981 (2018)
7. Zhang, G., Li, Y., Lin, T.: Caching in information centric networking: a survey. Comput. Netw. **57**(16), 3128–3141 (2013)
8. Lal, K.N., Kumar, A.: A centrality-measures based caching scheme for content-centric networking (CCN). Multimedia Tools Appl. **77**(14), 17625–17642 (2018)
9. Tang, S.Y., Alnoman, A., Anpalagan, A., Woungang, I.: A user-centric cooperative edge caching scheme for minimizing delay in 5G content delivery networks. Trans. Emerg. Telecommun. Technol. **29**(8), e3461 (2018)
10. Sun, S.S., Jiang, W., Feng, G., Qin, S., Yuan, Y.: Cooperative caching with content popularity prediction for mobile edge caching. Tehnicki Vjesnik-Technical Gazette **26**(2), 503–509 (2019)
11. Yao, L., Chen, A.L., Deng, J., Wang, J.B., Wu, G.W.: A cooperative caching scheme based on mobility prediction in vehicular content centric networks. IEEE Trans. Veh. Technol. **67**(6), 5435–5444 (2018)
12. Su, Z., Hui, Y.L., Xu, Q.C., Yang, T.T., Liu, J.Y., Jia, Y.J.: An edge caching scheme to distribute content in vehicular networks. IEEE Trans. Veh. Technol. **67**(6), 5346–5356 (2018)
13. He, H.J., Zhao, Y., Wu, J.F., Tian, Y.: Cost-aware capacity provisioning for internet video streaming CDNs. Comput. J. **58**(12), 3255–3270 (2015)
14. Simulation-transactions of the society for modeling and simulation international. http://sage.cnpereading.com/paragraph/article/10.1177/0037549719862023. Accessed 30 Sept 2019
15. Lin, M.H., Wierman, A., Andrew, L.L.H., Thereska, E.: Dynamic right-sizing for power-proportional data centers. IEEE/ACM Trans. Networking **21**(5), 1378–1391 (2013)
16. Mathew, V., Sitaraman, R.K., Shenoy, P.: Energy-aware load balancing in content delivery networks. In: IEEE INFOCOM, vol. 12, pp. 954–962. IEEE Press, Orlando (2012)

17. Tchernykh, A., Cortes-Mendoza, J.M., Pecero, J.E., Bouvry, P., Kliazovich, D.: Adaptive energy efficient distributed VoIP load balancing in federated cloud infrastructure. In: 3rd IEEE International Conference on Cloud Networking, pp. 1–6. IEEE Press, Luxembourg (2014)
18. Liao, D., Sun, G., Yang, G.H., Chang, V.: Energy-efficient virtual content distribution network provisioning in cloud-based data centers. Future Gener. Comput. Syst. Int. J. Sci. **83**, 347–357 (2018)
19. Bar-Yehuda, R., Kantor, E., Kutten, S., Rawitz, D.: Growing half-balls: minimizing storage and communication costs in content delivery networks. SIAM J. Discrete Math. **32**(3), 1903–1921 (2018)
20. Ahmed, F., Shafiq, M.Z., Khakpour, A.R., Liu, A.X.: Optimizing internet transit routing for content delivery networks. IEEE-ACM Trans. Netw. **26**(1), 76–89 (2018)
21. Fatin, H.Z., Jamali, S., Fatin, G.Z.: Data replication in large scale content delivery networks: a genetic algorithm approach. J. Circ. Syst. Comput. **27**(12), 1850189 (2018)
22. Tseng, L., DeAntonis, J., Higuchi, T., Altintas, O.: Peer-assisted content delivery network by vehicular micro clouds. In: 2018 IEEE 7th International Conference on Cloud Networking (CloudNet), Tokyo, Japan, pp. 1–3. IEEE (2018)
23. Salahuddin, M.A., Sahoo, J., Glitho, R., Elbiaze, H., Ajib, W.: A survey on content placement algorithms for cloud-based content delivery networks. IEEE Access **6**(8), 91–114 (2018)
24. Mahesh, G., Maheswara Rao, V.V.R., Shankar, R.S., Sirisha, G.V.G.: Primal-dual parallel algorithm for optimal content delivery in cloud CDNs. In: 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, pp. 1–6. IEEE (2017)