ISSN: 2321-7782 (Online)

Volume 2, Issue 12, December 2014

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Content Delivery Network Approach to Improve Web Performance: A Review

Meenakshi Gupta[1]
Research Scholar
MMICT&BM (MCA)
Maharishi Markandeshwar University
Mullana, Haryana – India

Atul Garg[2]
Associate Professor
MMICT&BM (MCA)
Maharishi Markandeshwar University
Mullana, Haryana – India

Abstract: The explosive growth of web traffic has affected the efficient delivery of contents forcing it to become a major concern. Due to the huge data and usage of web servers, servers are becoming overwhelmed with the increasing number of users and escalating volume and size of contents. This is having an adverse affect on web service providers and organizations relying on the web. Content Delivery Network is an effective approach to alleviate the congestion on network and servers to improve the response to end-users. It optimizes the content delivery by replicating the contents on surrogate servers placed at the edge of Internet. Apart from increasing web traffic, flash crowd is a new congestion phenomenon on the Internet. Flash crowd, different from Internet congestion, suddenly causes heavy workload towards particular websites. Hence, it becomes crucial to maintain web performance in such unpredictable situations. Further, streaming media objects are steadily becoming an increasing fraction of the contents transmitted on the Internet. These objects demand for higher bandwidth and consistency. This paper analyzes the existing strategies for content replication, request routing, flash crowd mitigation and streaming media contents for design and implementation of efficient content delivery networks.

Keywords: Content delivery network; content replication; request routing; flash crowd; media streaming.

I. Introduction

Web has emerged as a universal medium for exchanging information and services. It is now, not only a medium for accessing information, rather it is becoming a platform for business and society. More and more activities of our life are moving online and web users are growing at a fast pace. With the increasing number of web users and web services, retrieval of information from the web is posing various challenges such as latency, network bottleneck, security and reliability problems. Further, the web users are no longer passive users; rather they have become active contributors to the web. They are creating web contents by posting on social networking sites, blogs, wikis and feedbacks.

The increasing use of web in the existing client-server (CS) networking model is resulting in poor performance for popular websites. These websites usually floods with web requests that may either cause delay in giving the response to end-users or no response at all. This results in negative response and end-users may switch to some other websites resulting in loss to the website owners. Even with the constant improvement in Internet infrastructure and increasing capacity of various servers, web users are still suffering from very significant access delays [9]. To alleviate this problem, various approaches have been suggested by researchers such as proxy cache, clustering, multihoming and mirroring. In spite of the usefulness of these approaches, the limits with these approaches have resulted in the development of Content Delivery Network (CDN). The concept of CDN is based on placing the replica of contents closer to the end users in order to increase scalability, availability and accessibility of the contents and as a result improves the user-perceived performance in receiving the requested web contents. It transparently delivers the content to the end-users on behalf of the origin server. Request redirection algorithm is used to select the best replica server and user's request is redirected to that server. A CDN also supports to enhance the performance of web during burst traffic. It is also used for delivering streaming services economically and reliably.

In this paper, various approaches proposed by the renowned researchers to improve web performance are discussed in section 2. Section 3 describes the basic architecture and working of CDN. A detailed discussion on content distribution and request routing system is provided in section 4 and 5. Section 6 introduces the strategies to mitigate the effect of flash crowd, while strategies for streaming media contents are described in section 7. Section 8 concludes the paper.

II. Related Work

In centralized model of servicing web requests, a single physical location is used to provide information to all its users. This approach is not suitable for popular websites to handle increasing volume of contents and web users. The centralized model is not scalable and server becomes easily overloaded causing failure of requests. It adversely affects the performance of web. Various approaches have been suggested by researchers to improve the web performance. Some of these are discussed below [19]:

Increase server capacity: A solution to the problem may be improving server capacity by adding more memory, storage capacity and improving speed of server. However this approach is scalable only to a particular limit.

Proxy Caching: In this approach the web contents that are frequently or recently accessed by the end-users are stored in cache memory of proxy servers. The future requests for these contents are satisfied through proxy server rather than sending these requests to origin server. This reduces network traffic, load on web server and response time [34]. In [3] Portable Extended Cache Approach (PECA) is proposed to store frequently used data at user-end in an extended cache memory to enhance the computational performance of web service. The extended cache memory may be in the form of pen drive, compact disk (CD), Digital Versatile Disk (DVD) or any other secondary memory. However in cache memory contents are placed after the users request them. Further the cache must be properly updated otherwise the users may get stale contents. Further these days, users usually access the web to find something new, as a result cache hit ratio tends to be low. This hinders in improving the performance of web content delivery to end-users.

Clustering: Local clustering can improve reliability and scalability of the server. However, it does not help either in reducing the latency of web request or the corresponding response. Further if the data center or ISP fails, the entire cluster is inaccessible to end-users. It is also difficult to scale cluster to thousand of servers.

Mirroring: To solve the problems of clustering, mirroring can be used in which clusters are deployed at different locations and the contents are copied on these mirrors. However mirroring approach is expensive and complex as it requires the cost of establishing the mirrors and maintaining the consistency of contents on them.

Multihoming: It uses multiple ISPs to connect to the Internet in order to improve the reliability of accessing the contents. In multihoming, it may be the case that two independent links may use the same transmission line and this reduces the reliability.

Although these approaches are useful to improve user-perceived web performance, but they aim at one or more aspects rather than addressing all the issues to improve performance of web content delivery. Therefore, CDN approach has become more popular to overcome the limitations of these approaches and further improve the performance of web content delivery to end-users. The interest in CDNs has also flourished due to the successful emergence of various CDN service providers such as Akamai, DigitalIsland, MirrorImage, EdgeCast, Limelight Networks etc. Some content providers that make use of CDN services to boost the delivery of contents to their users are – Yahoo, Twitter, Rediff, Adobe, MTV, Kellogg's, Pinterest, Kodak, StumbleUpon, Facebook, BBC, Toyota, Samsung, LinkedIn, McAfee etc.

III. Architecture and Working of CDN

A content delivery network is a system of distributed surrogate servers (also called replica servers) to deliver web contents to end-users on behalf of the origin server. The contents of the origin server are replicated on the surrogate servers. The requests from end-users are redirected to surrogate severs closer to them as is shown in Figure 1. As a result load on the origin server is reduced and network bandwidth expands.

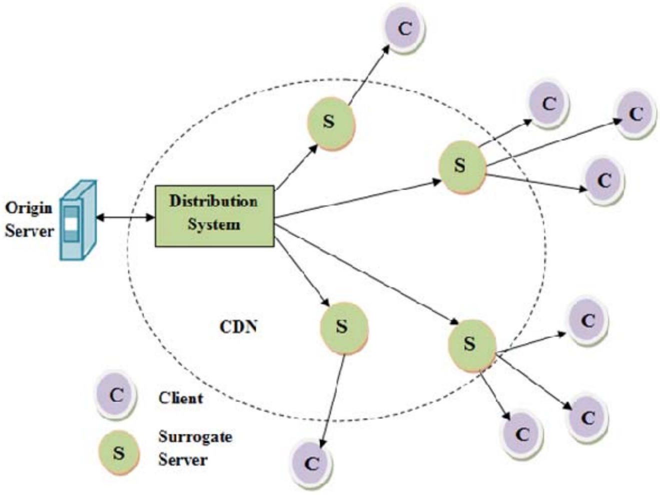

Figure 1: A Model of Content Delivery Network

The rationale behind using CDN is to improve the web content delivery to end-users. This also increases availability and accessibility of contents. It offers fast and reliable applications and services to end-users on the behalf of the origin server. The basic components and working of the Content Delivery Network are shown in figure 2.

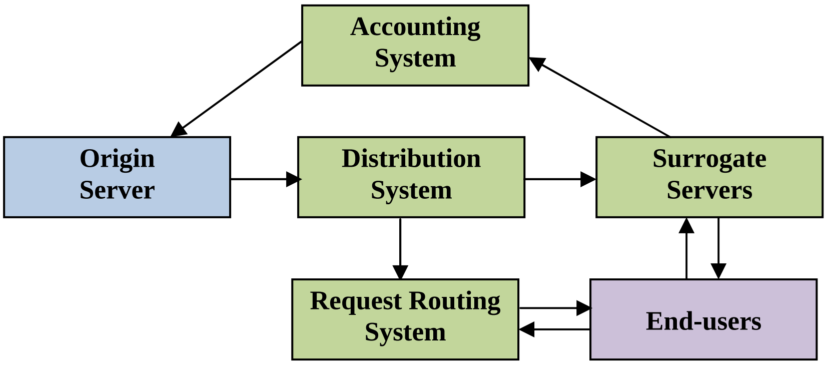

Figure 2: Basic Components and Working of Content Delivery Network

The basic working of CDN is as follows:

1. Origin server provides the contents to be replicated to the Distribution System.
2. The Distribution System replicates the contents on surrogate servers and also maintains the consistency of data at the surrogate severs.
3. The Distribution system also provides the information about replication to request routing system to help in surrogate server selection for redirecting end-users requests.
4. The request for the contents from end-user is directed to Request Routing System.
5. Request Routing System redirects the request to suitable surrogate server. This process is transparent to end-users.
6. Selected surrogate server satisfies the end-user request on the behalf of the origin server.
7. Surrogate server also sends the log of data transferred to Accounting System.

8. Accounting System aggregates this information for use by the origin server and for billing purpose according to the agreement with content provider.

9. Origin server uses this aggregated information for decision about that which contents should be replicated and where in order to further improve web performance as well to minimize the cost.

IV. Content Distribution System

Content distribution system deals with placement of surrogate servers, selection of surrogate servers for content replication, content distribution and consistency management. The purpose is to maximize the throughput and minimize the cost, hence making the optimum utilization of network resources. An optimal distribution of contents on surrogate servers assists to improve the network performance. This facilitates in designing scalable, reliable and efficient content delivery system.

A. Surrogate Server Placement:

The efficient operation of CDN requires the decision about the number of replica servers, their location and capacity. Various approaches have been proposed for the purpose considering one or more factors such as proximity with end-users, load on surrogate server and cost of updating the contents. Generally, the problem is to place M number of surrogate servers among N different sites where N>M in a way that leads to the lowest cost. Several algorithms have been proposed for surrogate servers placement such as Random, Greedy, Hot spot, Super Optimal, Tree based, Topology informed etc. [4, 23 & 33]. However we eschew going in details of decision regarding placement of surrogate servers as most of the CDN service providers already have their well established infrastructure. Therefore we will mainly consider how this infrastructure can be better utilized to further improve the web content delivery performance.

B. Content Replication:

An important decision related to design and implementation of CDNs is the optimal replication of contents on surrogate servers as the number and capacity of surrogate servers is limited. This requires the decision about what contents should be replicated and on what surrogate servers. The surrogate servers for replication of contents are selected in a way so that the cost of placing and accessing the contents is minimized and user-perceived performance of web content delivery is maximized.

The problem of content replication is considered as NP-Complete. Several heuristics have been proposed to solve this problem to improve CDN performance. These heuristics mostly concentrated on efficient replication of the contents from one origin server. J. Kangasharju et al., [21] considered a global case for replication of contents from several origin servers. They considered content replication as a combinatorial optimization problem and showed that this optimization problem is NP-complete. They assumed nodes with finite storage capacity and made replication decision on per-object granularity taking into consideration the cooperation between CDN servers. They have developed four natural heuristics namely Random, Popularity, Greedy-Single, Greedy-Global for possible best placements of contents and showed that the best performing heuristic is Greedy-Global which has all the CDN servers cooperating.

Most of the object replication algorithms assume static placement of replicas and do not consider adaptation with the changes in client’s access pattern. Scalable Content Access Network (SCAN) [40] is a scalable replica management framework that dynamically places a minimal number of replicas to meet client Quality of Service (QoS) and server capacity constraints. It tries to minimize the number of replicas while meeting these constraints. Adaptive Genetic Replication Algorithm (AGRA) [37] is a hybrid genetic algorithm that combines the features of both static and dynamic algorithms. It uses the current replica distribution as input and calculates a new one using the network attributes and the changes occurred. AGRA adapts to changing environment very quickly. Adaptive Distributed Request Window (ADRW) [25], an adaptive object replication algorithm, dynamically adjusts the allocation scheme of objects to minimize the total servicing cost of the arriving requests in a distributed system.

Latency based object placement (Lat-cdn) [12] is a network-adaptive technique to replicate the contents to surrogate servers, which does not require any prior knowledge of request statistics. This algorithm uses object's latency to make replication decision using cooperative push-based scheme. However this approach does not consider the load of the objects. Therefore it is possible that during a flash crowd event some surrogate servers may be overloaded. Later this technique was improved as il2p algorithm [13] that integrates both the network latency and object load on servers to improve the response time of the requests significantly.

CDN outsources the contents on behalf of origin server and in return charges according to the usage. Constraint P-median (CPM) [17] is an optimization model based on Multiple Minimum Cost Flow Model. CPM algorithms consist of three parts; replication algorithm preprocess, constraint P-median model and algorithm of solving constraint P-median problems with the iteration method. It allocates replicas of files on storage capacity constrained servers in order to minimize the total cost. Whereas in [10] CDN utility metric is proposed as a parameter to pricing policy. The metric specifies the relation between the number of bytes of the served content against the number of bytes of pulled content. The purpose is to place the requested objects to surrogate servers with respect to CDN utility metric. The best surrogate server is the one that produces the maximum CDN utility value and improves CDN's performance. Figure 3 shows different content placement strategies:

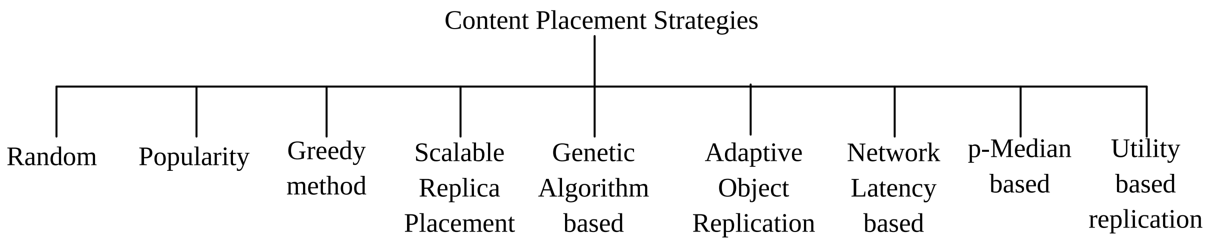
Content Placement Strategies
Random
Popularity
Greedy method
Scalable Replica Placement
Genetic Algorithm based
Adaptive Object Replication
Network Latency based
p-Median based
Utility based replication

Figure 3: Content Placement Strategies

Various approaches suggested for content replication have focused on different parameters such as web objects size, server and network limits, coarse/fine grain replication, static/dynamic replication and cooperation between surrogate servers. Selecting a proper mix of these parameters for content replication will have a profound effect on web access performance through CDN.

C. Content Consistency Management:

The contents replicated on surrogate servers must be consistent with the origin server. The end-users must get the updated contents otherwise no scheme for content replication will be effective. Various approaches used for this purpose are as follows [11]:

Cooperative push-based: Initially contents are pushed from origin server to surrogate servers. Surrogate servers cooperate to minimize the replication and update cost. CDN maintains a mapping between content and surrogate servers. Web request is routed to the closest surrogate server. If the surrogate sever does not have requested web content then the request is directed to the origin server.

Uncooperative pull-based: Web request is routed to the closest surrogate server. If surrogate server does not have requested web contents then the contents are pulled from the origin server to satisfy the request.

Cooperative pull-based: In this approach surrogate servers are cooperative with one another. If the contents are not available on the surrogate server then it pulls them from nearby surrogate server having the requested contents.

V. Request Routing System

A request routing system redirects the client's request to a suitable surrogate server. The redirection decision is complementary to the decision about placement of surrogate servers. The policies used for placing the surrogate servers are implemented during request redirection. The Request routing system consists of redirection algorithms and redirection mechanisms. Redirection algorithm decides the selection of surrogate server to satisfy client request whereas redirection mechanism informs the client about selection [35].

A. Request Routing Algorithms:

Request redirection approaches [35] can be classified as either client-side or server-side or somewhere in the network between these two. The classification is based on the point of redirection decision. In client-side redirection approach, decision is taken by a client-side proxy where as in server-side redirection approach, request is first sent to origin server which then redirects it to suitable surrogate server.

One of the important factors in the efficient utilization of surrogate servers is to redirect client's request to the best server based on some optimality criteria. The simplest approach to select server for redirection is random approach. This approach requires less computation complexity. However it is not efficient as it does not take into consideration server load and client latency. Round robin approach tries to balance the load on servers but still does not take into account proximity with the client. In [42] a light-weight server push approach has been combined with client-probe approach to estimate the expected response time. The goal is to allocate a server that minimizes client's response time. However this approach requires the installation of proxies to act as probing agents.

As the contents are replicated on multiple surrogate servers, therefore Transmission Control Protocol (TCP) parallel access scheme [32] has been suggested to retrieve file from multiple servers at the same time instead of using complex algorithms for selection of a suitable surrogate server. In this scheme, the transfer time is enhanced by providing the addresses of best servers to the clients. In [29] replicated servers are ranked based on prediction of content transfer time while considering the load on servers as well as the characteristics of path between the client and servers. The clients can download from the best server or a subset of servers at the top can be used in parallel.

The performance of redirection scheme directly or indirectly depends upon replication scheme. Hence replication and redirection approaches have been considered jointly. Co-Operative Cost Optimization Algorithm (COCOA) is a placement and retrieval algorithm [16] that is based on hybrid architecture. It is as scalable as the popularity-local algorithm and provides a similar performance as the greedy- global replica placement algorithm (RPA). Randomized and Next Neighbor (RNN) algorithm [7] is also scalable to increasing number of surrogate servers and request load. Each server maintains the information about its load and its next-neighbor load. It randomly selects a server and makes a choice between that server and its next-neighbor depending on their load.

Typically there is a trade-off between load-aware and locality-aware approaches. Locality-aware approaches usually lack load-balancing and vice-versa. Load-Aware Network Coordinates (LANCs) approach [30] tries to make a balance between these two. Popular contents are replicated among nearby content servers with low load to balance the request between content servers and assign clients to nearby servers.

An efficient redirection approach may be the one that considers multiple factors at a time such as proximity with the client, availability of the contents on the surrogate server, load on the surrogate server and network conditions. FuzzyCDN, an adaptive redirection algorithm, based on fuzzy logic to choose the best replica server has been proposed in [38]. It takes into consideration queue size, service time and response time metrics simultaneously. On the other hand, Hybrid Network Heuristic (HNH) [39] is a hybrid method that solves Replica Placement and Request Distribution Problem (RPRDP) jointly. It considers exact and heuristic constraints such as server disk space and bandwidth, QoS requirements of requests and changes in the network conditions simultaneously.

Various request redirection algorithms can be categorized as adaptive or non-adaptive. The adaptive algorithm selects a server for satisfying the request on the basis of current state of surrogate servers. In non-adaptive algorithm decision is based on some heuristics without any overhead about the current state of surrogate servers [35 & 2].

B. Request Routing Mechanisms:

According to the decision taken by the request routing algorithm, request routing mechanism is used to redirect the client's request to selected surrogate server. Various request routing mechanisms [9] have been proposed such as:

Client Multiplexing: The client gets the addresses of suitable surrogate servers and selects one of them to send the request.

HTTP Redirection: The request is sent to the origin server which in return redirects it to a new Uniform Resource Locator (URL). The end-user gets the response from this new URL.

DNS Indirection: The request is sent to Domain Name System (DNS) server which in return sends the Internet Protocol (IP) address of one of a set of surrogate servers. This technique is transparent to the client.

Anycasting: An IP anycast address is used to represent a set of servers providing the same service. The client requests for the contents using this anycast address. Anycast aware routers direct this request to one of the servers identified by anycast address.

Peer-to-Peer Routing: In this mechanism, a peer node in adhoc-network has the information about contents available on some other peer nodes in the network. On the basis of this request is redirected to one of the candidate peer nodes.

VI. Flash Crowd Mitigation

Flash crowd refers to a situation when a very large number of users simultaneously access a popular website. The reason may be to get information about some popular event. Such type of events may be known in advance such as the sports events like Olympics or the world cup. However there may be flash crowd without any advanced warning such as September 2001 terrorist attack [15]. In such circumstances, the network traffic pattern to the website is different from the usual one and often in burst mode. Though this is usually for a short term, however, for that period it may make the CDN inefficient. During flash crowd, CDN may not be able to handle such sudden increase in web requests and its performance may deteriorate considerably. The end-users may not be able to access the website or it may take too long due to network congestion. The strategies that are used to direct the web requests to surrogate servers during normal workload may not be fit to handle flash crowd. Therefore CDN requires some specific strategies to handle such situations in order to satisfy the web requests.

Denial-of-Service (DoS) attacks also create the same situation as flash crowds. However flash crowds are different from DoS attacks. Flash crowds are legitimate requests where as DoS attacks are malicious requests with the intention to degrade the normal functioning of the website. Identifying DoS attacks helps websites to make a provision for discarding malicious attacks and efficiently handle the legitimate requests.

Main challenges to address the issue of flash crowd events are:

- Monitoring CDN to find that a flash crowd situation has occurred
- Adjusting the policies and resources to handle the situation

Different strategies to handle flash crowd events are categorized based on network architecture as server-layer, intermediate-layer and client-layer [31]. At server layer over provisioning is used to handle flash crowd events in static CDNs. However, large-scale over provisioning in CDN is costly and inefficient approach due to unpredictable nature of flash crowds.

At intermediate-layer, caching techniques are used to curb the server load during flash crowds. At client-layer, clients also play the role of server to reduce the load on servers. The request from a client is redirected to another client that has recently downloaded the contents. Different mechanisms based on client-layer have been proposed to handle flash crowd such as CoopNet, PROOFS etc.

The decision regarding redirection of request to suitable surrogate server is critical in handling flash crowd. Redirection algorithms that adjust dynamically the number of replica servers [24] for a given object, makes the system better to support load without affecting the user-perceived response latency. These algorithms do not require any perfect information about sever load and work well under a wide range of loads and are robust during flash crowd and Distribute Denial-of-Service (DDoS) attacks. An adaptive CDN architecture is proposed in [15] based on dynamic delegation of requests to handle flash crowds. In this approach, surrogate servers are organized into groups. Within a group, one server is considered as primary for a given website and rest of the servers are considered primary for other websites. Client's requests are redirected only to primary servers in the groups. During flash event, when load on the primary servers increases, the requests are distributed among other servers in the group called delegates. When the flash event ends, the delegate servers are released. In [1], a prototype is proposed for dynamic allocation of existing resources to effectively handle flash crowds with different characteristics without over-provisioning. According to this prototype, to handle very sharp growth in load, the dynamic allocation scheme must be either extremely responsive or employ low overhead mechanism. Whereas gradually increasing flash crowds can be equally handled with larger overheads and slower reaction times.

Caching and replication techniques are mainly implemented in proxy servers and CDNs respectively. In [22] surrogate servers are used both as replicators and proxy caches to make the CDN system robust during a flash crowd event. In this approach, storage capacity at surrogate servers is partitioned into static cache and proxy cache. Static cache is used for replicating contents statically and proxy cache for running a cache policy replicating contents dynamically. With this integration, CDN may take the benefit of dynamic nature of caching while using replication for availability, reliability and bounded update propagation cost. CDN has also been combined with peer-to-peer (P2P) network to handle flash event. Peer Assisted Peer-Allocation (PAPA) algorithm [8] is based on collaboration between servers and peers. In PAPA, proactive strategies are used to handle sudden workload on the system. Based on the users preferences on similar files published earlier, the servers pre-allocate the contents to idle peers. PAPA is not applicable to live streaming where the contents are not available before release. An adaptive CDN, Flash Crowd Alleviation Network (FCAN), is proposed in [6] that changes its network structure dynamically between client-server and cache proxy P2P mode based on load fluctuation to deal with flash crowd.

The existing approaches show that a flash crowd can be handled effectively, if its occurrence can be predicted in advance. However, still CDNs need some specific strategies to handle a flash crowd situation.

VII. Media Contents Streaming

The nature of contents on web is changing from text and images to multimedia. With the rise of online shopping, social networking, gaming, education, music listening, movie watching etc, there is an explosion of multimedia applications. Further with the advancement of broadband technology and cheaper internet access, the users prefer streaming more than downloading. It can be live or on demand. In streaming, play back begins along with downloading and users do not have to wait for the complete downloading. The media streaming has dominated the Internet traffic. However it requires consistent performance so that end-users can listen and view without any jitter. Efficient media streaming requires coherent decisions regarding content selection and placement, request routing and delivering of contents to end-users. A main issue is to design content distribution system that is scalable, places replica servers closer to clients and minimizes the total server and network delivery cost [20].

Media streaming can be unicast or multicast based. In unicast streaming, shortest path routing is used to minimize network bandwidth and total server bandwidth usage. It does not take into account the number of replicas and their placement. In multicast streaming, contents are delivered to multiple interested receivers simultaneously. However, the issue with multicasting is that if the minimum distance is considered then it reduces the number of clients that can be served simultaneously and vice-versa. In [20] greedy min-cost tsp (tree of shortest path) heuristic for placement and shortest path routing/greedy ordered min-cost heuristic for routing client requests and multicast streams has been suggested to produce the best near-optimal solution. In Multiple Description–Content Delivery Network (MD-CDN) architecture, [18] Multiple Description Coding (MDC) is combined with path diversity to achieve reduced clients response time, servers load balancing, scalability and robustness in streaming media CDN.

Multimedia files are usually large in size. Taking into consideration the characteristics of multimedia files, deploying as many replicas as possible is always not a good strategy. It will increase the cost of replicating the contents and may also degrade the performance of clients. Therefore considering the cost of distributing contents, optimal number of servers should be selected from potential servers for replication [28]. Further instead of placing entire file on n nodes, the file can be subdivided into n subfiles of equal size and each subfile can be placed on a different node. FastReplica [26] algorithm is based on this approach for scalable and reliable replication of large files to speed up the overall download time. For media files encoded with MDC, different descriptions can be treated as subfiles.

To fulfill the users expectations and to improve Hyper Text Transfer Protocol (HTTP) media streaming quality, bit rate adaptation metric is used in Dynamic Adaptive Streaming over HTTP (DASH) architecture [5]. The metric detects the network congestion and spare network capacity. The step-wise switch up and multi-step switch down strategies adapts the bit rate to match the end-to-end network capacity for fetching of serial and parallel media segments.

In general, users watch the first part of video and then switch to another one. The performance of CDN can be improved by storing only the first part of video in cache memory with reduced extra cost. In [36], SSD (Solid State Drive) caching scheme is used to store the first part of videos for increasing the accessibility of popular contents for greater number of users simultaneously as data rate and access time of it is multiple times faster than HDD (Hard Disk Drive).

Flash crowd can also occur in media streaming. FCAN, an adaptive CDN, for static content delivery [6] has been extended for handling flash crowds during live and on-demand streaming by adding dynamic resizing and quality restriction to enhance the resilience of the system [41].

Hybrid CDN-P2P networks (HCDN) have been used for handling media streaming. Hybrid Replica Placement Mechanism (HRPM) [27] is a dynamic economical mechanism to optimize the number and placement of replica servers for HCDN. The streaming content delivery services are based on recursive hierarchical push-based cooperative replica placement strategy. End-users can receive the service in either CS or P2P mode. The streams are replicated with different qualities for CS and P2P end-users. Content providers are charged less for delivery in P2P mode. Therefore, it takes into account not only the content delivery costs and popularity of contents but also revenue of the HCDN. TrustStream architecture [14] combines the best features of scalable coding, CDN and P2P networks to achieve security, scalability, heterogeneity and certain QoS simultaneously. In this video is encoded in two layers- base and enhanced. Base layer is delivered through CDN-featured single-source multi-receiver P2P network to guarantee a minimum level of quality. Enhanced layer is delivered through pure multi-source multi-receiver P2P network to achieve maximum scalability and bandwidth utilization.

Streaming media files are usually larger objects and have specific characteristics such as unequal access to different parts of file, consistent delivery and more bandwidth requirement. Therefore distribution system for media streaming should be designed to fulfill these characteristics for better quality of experience to end-users.

VIII. CONCLUSION AND FUTURE WORK

Content Delivery Network approach is based on optimizing the delivery of contents by replicating the contents on surrogate servers placed at the edge of Internet. This helps in reducing the bandwidth consumption and improving the user-perceived latency. In this paper, the basic strategies used for content delivery networks have been analyzed. This analysis helps in getting the thorough insight into content delivery networks. This shows that the design and implementation of a scalable, reliable and efficient content delivery network entails focusing on a number of technical aspects such as what contents should be replicated and where, which is the appropriate server on which the request should be redirected, how to handle the load in case of flash crowd events, what strategy should be used to deliver streaming media contents. A lot of research work has been done on these fronts. However still there is a scope for improving these strategies of content delivery networks for enhancing web performance. As future work we intend to propose a new framework for efficient distribution of contents over surrogate servers.

References

1. Abhishek Chandra, Prashant Shenoy, "Effectiveness of Dynamic Resource Allocation for Handling Internet Flash Crowds", TR03-37, Department of Computer Science, University of Massachusetts, USA, Nov. 2003.
2. Al-Mukaddim Khan Pathan, Rajkumar Buyya, “A Taxonomy and Survey of Content Delivery Networks”, *Technical Report, GRIDS-TR-2007-4, Grid Computing and Distributed Systems Laboratory, The University of Melbourne, Australia, 2007.*
3. Atul Garg, Anil Kapil, “Potable Extended Cache Memory to Reduce Web Traffic”, International Journal of Engineering Science and Technology, Vol. 2(9), pp. 4744-4750, 2010.
4. Bo Li, Mordecai J. Golin, Giuseppe F. Italiano, Xin Deng and Kazem Sohraby, “On the Optimal Placement of Web Proxies in the Internet,” In Proceedings of IEEE INFOCOM’99, pp. 1282-1290, 1999.
5. Chenghao Liu, Imed Bouazizi, Miska M. Hannuksela, Moncef Gabbouj, “Rate Adaptation for Dynamic Adaptive Streaming over HTTP in Content Distribution Network”, Signal Processing: Image Communication, Elsevier, 27(4), 288–311, 2012.
6. Chenyu Pan, Merdan Atajanov, Mohd. Belayet Hossain, Toshihiko Shimokawa, Norihiko Yoshida, “FCAN: Flash Crowds Alleviation Network using Adaptive P2P Overlay of Cache Proxies”, IEICE Trans. On Communications, Vol. 89, No. 4, pp. 1119–1126, 2006.
7. Chung-Min Chen, Yibei Ling, Marcus Pang, Wai Chen, Shengwei Cai, Yoshihisa Suwa, Onur Altintas, "Scalable Request Routing with Next-Neighbor Load Sharing in Multi-server Environments", 19th International Conference on Advanced Information Networking and Applications, AINA’05, IEEE, Vol. 1, pp. 441-446, 2005.
8. Dan Huang, Min Zhang, Yi Zheng, Changjia Chen, Yan Huang, "Pre-allocation based Flash Crowd Mitigation Algorithm for Large-scale Content Delivery System", Peer-to-Peer Networking and Applications, pp. 1-8, 2014.
9. Gang Peng, “CDN: Content Distribution Network”, CoRR, arXiv:cs/0411069v1, Technical Report TR-125 of Experimental Computer Systems Lab, Stony Brook University, NY, 2004.
10. George Pallis, “Improving Content Delivery by Exploiting the Utility of CDN Servers”, In Proc. of the 5th Int. Conf. on Data Management in Cloud, Grid and P2P Systems (Globe). LNCS, Springer, 88–99, 2012.
11. George Pallis, Athena Vakali, “Insight and Perspectives for Content Delivery Networks”, Communications of the ACM - Personal information management, Volume 49 Issue 1, pp. 101-106, 2006.
12. George Pallis, Athena Vakali, Konstantinos Stamos, Antonis Sidiropoulos, Dimitrios Katsaros, Yannis Manolopoulos, “A Latency-Based Object Placement Approach in Content Distribution Networks,” In Proceedings of the 3rd Latin American Web Congress (La-Web 2005), IEEE Press, Buenos Aires, Argentina, pp. 140-147, October 2005.
13. George Pallis, Konstantinos Stamos, Athena Vakali, Dimitrios Katsaros, Antonis Sidiropoulos, Yannis Manolopoulos, “Replication Based on Objects Load under a Content Distribution Network”, 22nd International Conference on Data Engineering Workshops (ICDEW’06), IEEE, p. 53, 2006.
14. Hao Yin, Chuang Lin, Qian Zhang, Zhijia Chen, Dapeng Wu, "TrustStream: A Secure and Scalable Architecture for Large-scale Internet Media Streaming", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 18, No. 12, pp. 1692-1702, 2008.
15. Jaeyeon Jung, Balachander Krishnamurthy, Michael Rabinovich, “Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites”, Proceedings of the 11th International Conference on World Wide Web, ACM, pp. 293-304, 2002.
16. Jan Coppens, Tim Wauters, Filip De Turck, Bart Dhoedt, Piet Demeester, "Evaluation of Replica Placement and Retrieval Algorithms in Self-organizing CDNs", Proceeding of the IFIP/IEEE International Workshop on Self-Managed Systems & Services (SelfMan'05), 2005.
17. Jing Sun, Suixiang Gao, Wenguo Yang, Zhipeng Jiang, “Heuristic Replica Placement Algorithms in Content Distribution Networks”, Journal of Networks. Vol. 6, No. 3, pp. 416-423, March 2011.
18. John Apostolopoulos, Tina Wong, Wai-tian Tan, Susie Wee, “On Multiple Description Streaming with Content Delivery Networks”, IEEE INFOCOM, Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, Vol. 3, pp. 1736-1745, 2002.

19. John Dilley, Bruce Maggs, Jay Parikh, Harald Prokop, Ramesh Sitaraman, Bill Weihl, “Globally Distributed Content Delivery”, Internet Computing, IEEE, Vol. 6, No. 5, 50–58, 2002.
20. Jussara M. Almeida, Derek L. Eager, Mary K. Vernon, Stephen J. Wright, “Minimizing Delivery Cost in Scalable Streaming Content Distribution Systems”, IEEE Transactions on Multimedia, 6(2), 356–365, 2004.
21. Jussi Kangasharju, James Roberts, Keith W. Ross, “Object Replication Strategies in Content Distribution Network”, Computer Communicaiton, Elsevier, Vol. 25, No. 4, pp. 376-383, 2002.
22. Konstantinos Stamos, George Pallis, Athena Vakali, "Integrating Caching Techniques on a Content Distribution Network", In Advances in Databases and Information Systems, Springer, pp. 200-215, 2006.
23. Lili Qiu, Venkata N. Padmanabhan, Geoffrey M. Voelker, “On the Placement of Web Server Replicas”, Proc. IEEE INFOCOM’01, Vol. 3, 1587-1596, 2001.
24. LiminWang, Vivek Pai, Larry Peterson, “The Effectiveness of Request Redirection on CDN Robustness”, ACM SIGOPS Operating Systems Review, 36, SI, 345-360, 2002.
25. Lin Wujuana, Bharadwaj Veeravalli, “Design and Analysis of an Adaptive Object Replication Algorithm in Distributed Network Systems”, Computer Communications, Elsevier, 31(10), pp. 2005-2015, 2008.
26. Ludmila Cherkasova, Lee Jangwon, “FastReplica: Efficient Large File Distribution within Content Delivery Networks”, In Proceedings of the 4th USENIX Symposium on Internet Technologies and Systems, 2003.
27. Mehran Garmehi, Morteza Analoui, Mukaddim Pathan, Rajkumar Buyya, “An Economic Replica Placement Mechanism for Streaming Content Distribution in Hybrid CDN-P2P Networks”, Computer Communications, Elsevier, Vol. 52, pp. 60–70, 2014.
28. Mengkun Yang, Zongming Fei, “A Model for Replica Placement in Content Distribution Networks for Multimedia Applications”, Proceedings of IEEE international conference on communications (ICC’03), vol. 1, pp. 557–561, 2003.
29. Mohammad Malli, Chadi Barakat, Walid Dabbous, "An Efficient Approach for Content Delivery in Overlay Networks", Consumer Communications and Networking Conference, IEEE, pp. 128-133, 2005.
30. Nicholas Ball, Peter Pietzuch, “Distributed Content Delivery using Load-Aware Network Coordinates”, Proceeding of ACM CoNEXT Conference, 2008.
31. Norihiko Yoshida, “Dynamic CDN against Flash Crowds”, in Content Delivery Networks, Springer, pp. 275-296, 2008.
32. Pablo Rodriguez, Ernst W. Biersack, “Dynamic Parallel-Access to Replicated Content in the Internet”, IEEE/ACM Transactions on Networking, 10(4), 455-465, 2002.
33. Pavlin Radoslavov, Ramesh Govindan, Deborah Estrin, “Topology-Informed Internet Replica Placement”, Computer Communications, Elsevier, 25(4), 384–392, 2002.
34. Radhika Malpani, Jacob Lorch, David Berger, "Making World Wide Web Caching Servers Cooperate", In Proceedings of the Fourth International World Wide Web Conference, pp. 107-117, 1995.
35. Swaminathan Sivasubramanian, Michal Szymaniak Szymaniak, Guillaume Pierre, Maarten Van Steen, “Replication for Web Hosting Systems,” ACM Computing Surveys (CSUR), Vol. 36, No. 3, pp. 291–334, 2004.
36. Taekook Kim, Eui-Jik Kim, “Hybrid Storage-based Caching Strategy for Content Delivery Network Services”, Multimedia Tools and Applications, Springer, 1-13, 2014.
37. Thanasis Loukopoulos, Ishfaq Ahmad, “Static and Adaptive Distributed Data Replication using Genetic Algorithms,” Journal of Parallel and Distributed Computing, Elsevier, 64(11), pp. 1270-1285, 2004.
38. Thiago Queiroz de Oliveira, Marcial P. Fernandez, “Fuzzy Redirection Algorithm for Content Delivery Network (CDN)”, ICN 2013, The Twelfth International Conference on Networks, pp.137-143, 2013.
39. Tiago Neves, Luiz Satoru Ochi, Célio Albuquerque, “A New Hybrid Heuristic for Replica Placement and Request Distribution in Content Distribution Networks”, Optimization Letters, Springer, pp. 1-16, 2014.
40. Yan Chen, Randy H. Katz, John D. Kubiatowicz, “SCAN: A Dynamic, Scalable and Efficient Content Distribution Network”, In Proceedings of First International Conference on Pervasive Computing 2002, LNCS 2414, Springer, pp. 282-296, 2002.
41. Yuta Miyauchi, Noriko Matsumoto, Norihiko Yoshida, Yuko Kamiya, Toshihiko Shimokawa, “Adaptive Content Distribution Network for Live and On-Demand Streaming”, ARCS Workshops, pp. 27-37, 2012.
42. Zongming Fie, Samrat Bhattacharjee, Ellen W. Zegura, Mostafa H. Ammar, “A Novel Server Selection Technique for Improving the Response Time of a Replicated Service", in Proceedings IEEE INFOCOM'98, Vol. 2, 1998.

AUTHOR(S) PROFILE

Meenakshi Gupta received degree of Master of Computer Applications from IGNOU, New Delhi and M. Phil. (Comp. Sc.) from Periyar University in 2005 and 2008 respectively. Currently, she is working as an Assistant Professor at Maharaja Agrasen Institute of Management and Technology, Jagadhri, Haryana. She has 8 years of teaching experience and has several national and international publications to her credit. Her area of interest is fuzzy logic based systems and web optimization.

Atul Garg received degree of Master of Computer Applications from Kurukshetra University, Kurukshetra in 2004 and completed his Ph. D degree from Maharishi Markandeshwar University, Mullana (Ambala) in 2013. Currently, he is working as an Associate Professor at M.M.I.C.T.&B.M., Maharishi Markandeshwar University, Mullana (Ambala), Haryana. He is a Senior Member of the association of Universal Association of Computer & Electronics Engineers (UACEE), Australia, member in the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering (ICST), Belgium and Member in the International Association of Engineers, Hong Kong. His area of interest is web, Query Optimizations and mobile ad hoc networks.