ISSN No. 0976-5697

Volume 6, No. 6, July - August 2015

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Request-Routing for Content Delivery Networks (CDN)

Omotunde, A. A[1],Okolie, S.O [2],Adekunle, Y.A[3],Izang, A.A[4],Ebiesuwa, O.O[5]

[1,2,3,4,5]Department of Computer Science and Information Systems,
School of Computing and Engineering Sciences, Babcock University,
Ilishan-Remo, Ogun State, Nigeria.

Abstract-A Content Delivery Network (CDN) is a distributed network of servers (that provide web content) and file storage devices deployed in various geographical locations such that when requests are being made by users, these requests are redirected via GEO-DNS to the closest content repository to the user for response. This work proposes an adaptive request routing algorithm to choose the replica server using network proximity and a combination of Quality of Service metrics. The metrics include: bandwidth, availability of server, and latency. This is because the original intention of CDNs is for surrogate servers to respond to request from clients that are closer in proximity. This is not always the best option because the closest surrogate may either be overloaded (making it unavailable) or the connection may be poor. Thus, proximity with a mixture of metric are combined to choose the best surrogate to respond to request. . In contrast to most works in existing literature, this study employs three metrics and five membership functions (Very high, High, Medium, Low and Very low) based on the rules guiding fuzzy logic to give an accurate measure for each of the three metrics used in order to determine without errors when to grant or deny a request from the users. This makes it an improvement on existing works that used some other metrics and only three membership functions (High, Medium and Low), this research uses two additional membership functions which help to give a more precise measure of the variables in question and hence improves the efficiency of the request delivery, thus giving users of the network more robust information as regards the quality of web content service delivery.

Keywords:Content Delivery Networks, Fuzzy Logic, Request routing, Surrogate server, Web content

I. INTRODUCTION

Over the years, the internet as a ubiquitous media for sharing, managing and delivering content has experienced tremendous growth and maturity. Driven by rapid acceptance of broadband access along with increase in system complexity and content richness in addition to migrating to a plethora of devices, the number of users continues to grow at a quick rate [11]. Lager audiences embracing the “digital lifestyle” are requesting greater and greater volumes of content on a daily basis [13]. Also, according to a report by [14], 18 million households in Europe came online between 2005 and 2008. In addition to that, there is increase in penetration of multimedia devices such as smart phones and this in turn has altered internet usage pattern from beingpartially online to alwaysonline. As a result of this growth and pervasiveness, there has been an unusually large growth in network traffic.

The need therefore arose for effective management and effective delivery for fast access of information. Also, the web is seen as a tool to deliver rich content to users and as such, efforts need to be made to ensure that there is no decline in the quality of service and also to ensure that access delay is reduced a minimum, hence over the years there has been an evolution of technologies that aim to improve content delivery and service provisioning over the web[21].

An approach that was employed was to modify traditional web architecture by upgrading the web server hardware adding a high-end processor, upgrading the memory and adding to the disk space. This approach tends to slightly helps to reduce the performance problem. However, it didn’t provide a lasting solution because of the fact that there is a constant surge in network traffic as more and more people surf the web on a daily basis. Moreover, this approach was not flexible as opined by [11] because it gets to certain point when small enhancements become impossible and the only option is to replace the entire web server system [21].

Another approach that was used was to improve performance and reducing bandwidth usage, especially for narrowband users, is by deploying caching proxies. Caching proxies may also be equipped with technologies to detect a server failure and maximize efficient use of caching proxy resources. Users often configure their browsers to send their Web request through these caches rather than sending directly to origin servers. When this configuration is properly done, the user’s entire browsing session goes through a specific caching proxy. Thus, the caches contain most popular content viewed by all the users of the caching proxies. A provider may also deploy different levels of local, regional, international caches at geographically distributed locations. Such arrangement is referred to as hierarchical caching which may provide additional performance improvements and bandwidth savings [3].

A different approach for better performance was the use of server farms. A server farm is a group of networked servers that distributes tasks in a way that maximized efficiency and minimizes the risk of losing data. According to [11], each server in the farm shares the burden of responding to requests for the same web site. Although server farms and hierarchical caching through caching proxies are useful techniques to address the Internet Web performance problem, they have limitations. In the first case, since servers are deployed near the origin server, they do little to improve the network performance due to network congestion. Caching proxies may be beneficial in this case. But they cache objects based on client demands. This may force the content providers with a popular content source to invest in large server farms, load balancing, and high bandwidth connections to keep up with the demand. To address these limitations, another type of content network has been deployed in late 1990s. This is termed as *Content Distribution Network* or *Content Delivery Network*, which is a system of computers networked together across the Internet to cooperate transparently for delivering content to end-users.

II. CONTENT DELIVERY NETWORKS

A Content Delivery Network (CDN) is a distributed network of servers (that provide Web content) and file storage devices deployed in various geographical locations such that when requests are being made by users, these requests are redirected via GEO-DNS to the closest content repository to the user for response. According to [27; 20], CDNs maximize bandwidth, increase accessibility and maintain correctness through content replication.

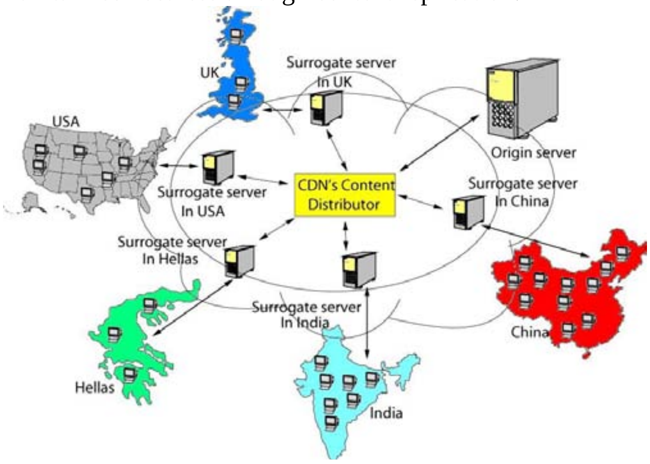

Figure 1: Content Delivery Network overview Source: [22]

A CDN has some combination of content-delivery, request-routing, distribution and accounting infrastructure. The ***content-delivery infrastructure*** consists of a set of edge servers (also called surrogates) that are distributed in various geographical locations to deliver copies of content to end-users. The ***request-routing infrastructure*** is responsible to directing client request to appropriate edge servers. It also interacts with the distribution infrastructure to keep an up-to-date view of the content stored in the CDN caches. The***distribution infrastructure*** moves content from the origin server to the CDN edge servers and ensures consistency of content in the caches. The ***accounting infrastructure*** maintains logs of client accesses and records the usage of the CDN servers. This information is used for traffic reporting and usage-based billing.

The conventional client-server communication is replaced by two communication flows: one between the requesting client and a surrogate server, and the other between the surrogate server and the origin server. This is illustrated in Figure 2.

There are three major key players in the CDN architecture. They are: content provider; CDN provider; and end-users. A content provider delegates the URI name space of the Web objects to be distributed. A CDN provider provides infrastructure facilities to content providers in order to deliver content in a timely and reliable manner. This content include static content (e.g. static HTML pages, images, documents, software patches), streaming media (e.g. audio, real time video), User Generated Videos (UGV), and varying content services (e.g. directory service, e-commerce service, file transfer service). Figure 3 depicts the various types of content distributed. *End-users* or *clients* are the entities who access content from the content provider's website.

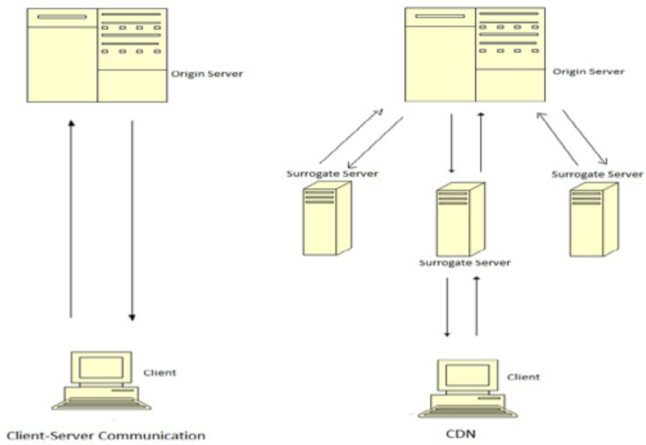

Figure 2: Client-Server Connection vs. CDN

Figure 3: Content Served by CDN [21]

The following are the basic functions of a CDN:

a. Direct a client's request to the closest suitable surrogate server so as to bypass congestion thus overcoming flash crowds.
b. Replicate content on surrogate servers on behalf of origin server.
c. Manage the network components, the accounting infrastructure and monitors report on content usage.

Major CDN problems are surrogate server implementation, request-routing from clients to either a surrogate server or the origin server, content distribution and synchronization from origin server to surrogate servers, client authentication, authorization and accounting (AAA), and the same level of security enforcement by the surrogate servers and the origin server. However, the focus of the research is on request-routing.

III. REQUEST ROUTING

The request-routing system consists of network elements that are responsible for routing clients request to an appropriate surrogate server for the delivery of content. [9], posited that if the response time for a web request exceeds 8 seconds, about 30% of users leave the request. The increase in response time is directly related to performance loss, congestion and a large number of users reloading the website, making access to the website worse. A common approach is to direct clients request to the replica server closest to the client. However, the closest server may not be the best to direct client's request to in order to maintain high quality of service as opined by [7]. Instead, a set of parameters could be considered during this selection process, such as distance, speed, available bandwidth and surrogate server load.

[21], noted that the content selection and delivery techniques (i.e. full-site and partial-site) used by a CDN have a direct impact on the design of its request-routing system. If the full-site approach is used by a CDN, the request-routing system assists to direct the client requests to the surrogate servers as they hold all the outsourced content. On the other hand, if the partial-site approach is used, the request-routing system is designed in such a way that on receiving the client request, the origin server delivers the basic content while surrogate servers deliver the embedded objects.

The request-routing system in a CDN has two parts: deployment of a request-routing algorithm [25], and use of a request-routing mechanism as explained in the work of [21].

Request-routing algorithms can be divided into two categories: adaptive algorithms and non-adaptive algorithms. In adaptive algorithms, the choice is made based on the server's status, requiring constant monitoring this it demonstrates high system roboustness [28] in the face of flash crowd events. Innon-adaptive algorithms, the choice is based on heuristics, and then a lightweight processing by not requiring monitoring.

Request-routing mechanisms inform the client about the selection of replica server, generated by the request-routing algorithms.

IV. RELATED WORKS

Request Routing Algorithms

The simplest method of distributing requests among a number of web servers is the Round Robin approach [30; 24; 26], in which each request is server by a different server following a cyclic order. It is generally assumed that the servers are placed in the same place [19], and they all have the same capacity and share the same network. The success of this algorithm is exhibited over small network subsets containing nodes that are close on network distance.

[10]. Considered an algorithm that takes into consideration the percentage of client requests that each replica server receives. In this algorithm assumption is made that the server receiving more requests is the more powerful thus client's requests are directed to the more powerful servers to achieve better resource utilization.

Another algorithm which is based on the non-adaptive technique is the random algorithm [10;17]. This randomly distributres requests to replica servers.

[12]. Proposed a non-adaptive algorithm that calculates a hashing function h from a large space of identifiers based on the URL content.

[8]. Proposed an adaptive algorithm referred to as Least Loaded that directs client's request to the server with the highest idle capacity. This is determined by observing the current state of the server via protocol as posited by [5].

[2; 1], proposed an adaptive algortihm that uses client-server latency for request routing by taking into account either client access logs or passive server-side latency measurement. Based on this algorithm, client's requests are redirected to a replica server that reports the minimal latency to the client.

[23]. Use an adaptive request-routing algorithm that selects the replica server closest to the clients in terms of network proximity. The metric estimation in Globule is based on path length which is updated periodically. The metric estimation service used in globule is passive, which does not introduce any additional traffic to the network.

An adaptive algorithm was proposed by [15], based on a mathematical model. Request arriving rate and the requisition arriving rate variation in a certain time interval are considered in decision-making.

In this paper, [18], proposed a new algorithm to choose the replica server in CDN networks using fuzzy logic. The fuzzy logic is feasible for this environment by simplifying the process modeling system, dispensing complex mathematical system, and leave the system closer to human thinking.

By simulations, they showed that the proposed algorithm gives good results comparing with other algorithms available in the literature. The main benefit was the lowest request response time obtained in three topologies tested. Furthermore, the algorithm presented the lowest standard deviation in these topologies, showing it gives a stable solution.

However, the simulation methodology could not show the performance of the proposed algorithm in real systems. Although Fuzzy logic produces a low impact in modern processor's performance, the researchers do not guarantee the algorithm scalability.

V. METRIC SELECTION

CDNs improve Web performance and have been proposed to maximize bandwidth, improve accessibility and maintain correctness through content replication [19; 3; 21; 22]. Requested web content spend less time on the network as surrogate servers are proximate to the clients making the request thus the geographical space that traversed is less [17]. Based on the aforesaid, one of the parameters that is considered when routing requests in CDN is proximity as obtained in the work of [16]. However, choosing surrogates based on proximity alone may not be the best choice because the closest surrogate may be over loaded at the time a request is being sent to it, thus increasing the response time of the request and increasing the user-percieved latency of the network. Hence, a combination of other Quality of Service metrics are considered. These are: 1. Availability of server and 2. Availabilty of bandwidth [21]. Availability of server implies that the server is reliable and being reliable means that the server is able to deliver a specified service. A popular method of indicating availability is calculate the fraction of the server's operational lifetime during which it has been accessible, yeilding a decimal representation that ranges between 0 and 1 [4].

This work proposes an adaptive algorithm to choose the replica server using network proximity and a combination of Quality of Service metrics. The metrics include: bandwidth, availability of server, and latency. This is because the original intention of CDNs is for surrogate servers to respond to request from clients that closer in proximity. This is not always the best option because the closest surrogate may either be overloaded (making it unavailable) or the connection may be poor. Thus, proximity with a mixture of metric are combined to choose the best surrogate to respond to request.

VI. FUZZIFICATION

For each one of the metrics used, there is defined membership function as proposed by [31], which maps the input parameters, generally numerical and accurate definition used in classic Boolean logic. This is to allow the request-routing mechanism to deal with subjective, imprecise and ambiguous information. All of the metrics used have five linguistic values attached to them:

a. Network proximity: Very high proximity, high proximity, medium proximity, low proximity, and very low proximity.
b. Availability of server: Very high availability, high availability, medium availability, low availability, and very low availability.
c. Bandwidth: Very high bandwidth, high bandwidth, medium bandwidth, low bandwidth and very low bandwidth.

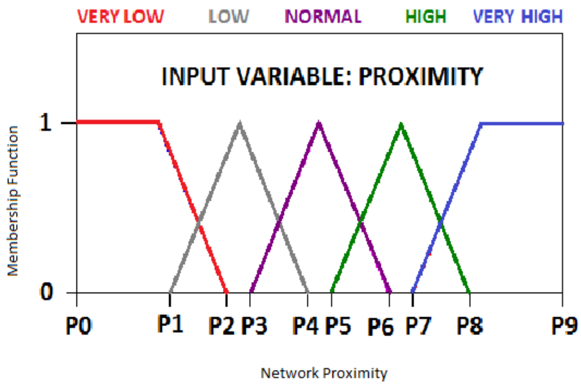

Figure 4: Membership function for Network Proximity

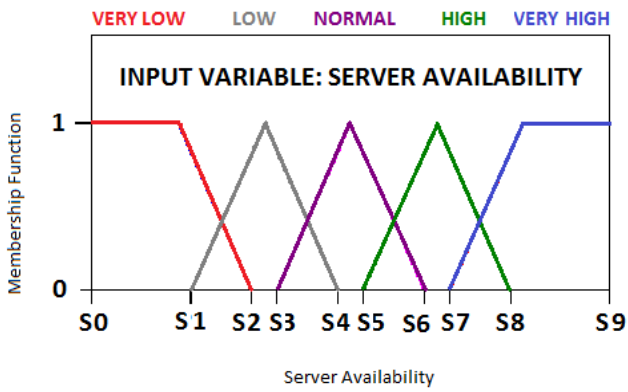

Figure 5: Membership function for Server Availability

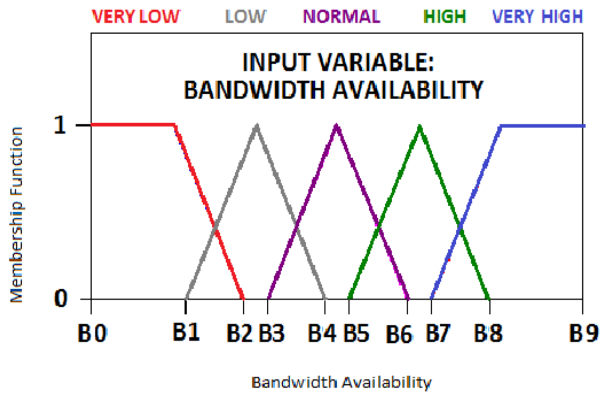

Figure 6: Membership function for bandwidth availability.

The membership functions for the three parameters i.e Network proximity, Bandwidth availability and server availability as shown in the figures above were obtained by varying the respective input variables using five membership functions (Very Low, Low, Medium, High and Very High) and ten values each to represent the ranges for the three parameters considered. The y-axis of the graphs is varied from 0 to 1 to depict a typical fuzzy logic scenario.

Because of the impreciseness and ambiguity of fuzzy logic, defining the inference rule is not a straightforward task. Hence, the Wang and Mendel's [29], fuzzy rule learning method is employed. This is because over the years, its high performance has been clearly demonstrated due to its comprehensiveness and simplicity [6],and it generates its Rule Base by looking for and selection the rules with the best individual performance.

Table 1: Rule Evaluation Table

Network Proximity	**Server Availability**	**Bandwidth Availability**	**Rule**
Very Low	Very Low	Very Low	Deny Request
Very Low	Very Low	Low	Deny Request
Very Low	Very Low	Medium	Deny Request
Very Low	Very Low	High	Deny Request
Very Low	Very Low	Very High	Deny Request
Very Low	Low	Very Low	Deny Request
Very Low	Low	Low	Deny Request
Very Low	Low	Medium	Deny Request
Very Low	Low	High	Deny Request
Very Low	Low	Very High	Deny Request
Very Low	Medium	Very Low	Deny Request
Very Low	Medium	Low	Deny Request
Very Low	Medium	Medium	Grant Request
Very Low	Medium	High	Grant Request
Very Low	Medium	Very High	Grant Request
Very Low	High	Very Low	Deny Request
Very Low	High	Low	Deny Request
Very Low	High	Medium	Grant

			Request
Very Low	High	High	Grant Request
Very Low	High	Very High	Grant Request
Very Low	Very High	Very Low	Deny Request
Very Low	Very High	Low	Deny Request
Very Low	Very High	Medium	Grant Request
Very Low	Very High	High	Grant Request
Very Low	Very High	Very High	Grant Request
Low	Very Low	Very Low	Deny Request
Low	Very Low	Low	Deny Request
Low	Very Low	Medium	Deny Request
Low	Very Low	High	Deny Request
Low	Very Low	Very High	Deny Request
Low	Low	Very Low	Deny Request
Low	Low	Low	Deny Request
Low	Low	Medium	Deny Request
Low	Low	High	Deny Request
Low	Low	Very High	Deny Request
Low	Medium	Very Low	Deny Request
Low	Medium	Low	Deny Request
Low	Medium	Medium	Grant Request
Low	Medium	High	Grant Request
Low	Medium	Very High	Grant Request
Low	High	Very Low	Deny Request
Low	High	Low	Deny Request
Low	High	Medium	Grant Request
Low	High	High	Grant Request
Low	High	Very High	Grant Request
Low	Very High	Very Low	Deny Request
Low	Very High	Low	Deny Request
Low	Very High	Medium	Grant Request
Low	Very High	High	Grant Request
Low	Very High	Very High	Grant Request
Medium	Very Low	Very Low	Deny Request
Medium	Very Low	Low	Deny Request
Medium	Very Low	Medium	Grant Request
Medium	Very Low	High	Grant Request
Medium	Very Low	Very High	Grant Request
Medium	Low	Very Low	Deny Request
Medium	Low	Low	Deny Request
Medium	Low	Medium	Grant Request
Medium	Low	High	Grant Request
Medium	Low	Very High	Grant Request
Medium	Medium	Very Low	Grant Request
Medium	Medium	Low	Grant request
Medium	Medium	Medium	Grant Request
Medium	Medium	High	Grant Request
Medium	Medium	Very High	Grant Request
Medium	High	Very Low	Grant Request
Medium	High	Low	Grant Request
Medium	High	Medium	Grant Request
Medium	High	High	Grant Request
Medium	High	Very High	Grant Request
Medium	Very High	Very Low	Grant Request
Medium	Very High	Low	Grant Request
Medium	Very High	Medium	Grant Request
Medium	Very High	High	Grant Request
Medium	Very High	Very High	Grant Request
High	Very Low	Very Low	Deny Request
High	Very Low	Low	Deny Request
High	Very Low	Medium	Grant Request
High	Very Low	High	Grant Request
High	Very Low	Very High	Grant Request
High	Low	Very Low	Deny Request
High	Low	Low	Deny request
High	Low	Medium	Grant Request
High	Low	High	Grant Request
High	Low	Very High	Grant Request
High	Medium	Very Low	Grant Request
High	Medium	Low	Grant Request
High	Medium	Medium	Grant Request
High	Medium	High	Grant Request
High	Medium	Very High	Grant Request
High	High	Very Low	Grant Request
High	High	Low	Grant Request
High	High	Medium	Grant Request
High	High	High	Grant Request
High	High	Very High	Grant Request

High	Very High	Very Low	Grant Request
High	Very High	Low	Grant Request
High	Very High	Medium	Grant Request
High	Very High	High	Grant Request
High	Very High	Very High	Grant Request
Very High	Very Low	Very Low	Deny Request
Very High	Very Low	Low	Deny Request
Very High	Very Low	Medium	Grant Request
Very High	Very Low	High	Grant Request
Very High	Very Low	Very High	Grant Request
Very High	Low	Very Low	Deny Request
Very High	Low	Low	Deny Request
Very High	Low	Medium	Grant Request
Very High	Low	High	Grant Request
Very High	Low	Very High	Grant Request
Very High	Medium	Very Low	Grant Request
Very High	Medium	Low	Grant Request
Very High	Medium	Medium	Grant Request
Very High	Medium	High	Grant Request
Very High	Medium	Very High	Grant Request
Very High	High	Very Low	Grant Request
Very High	High	Low	Grant Request
Very High	High	Medium	Grant Request
Very High	High	High	Grant Request
Very High	High	Very High	Grant Request
Very High	Very High	Very Low	Grant Request
Very High	Very High	Low	Grant Request
Very High	Very High	Medium	Grant Request
Very High	Very High	High	Grant Request
Very High	Very High	Very High	Grant Request

For this work, the Network Proximity is determined using the rate of return when a packet is sent; the Server Availability is determined using the number of requests granted while the Bandwidth Availability is determined using the speed at which the hosts on the network are able to access the internet or network.

VII. CONTRIBUTION TO KNOWLEDGE

This work employed the use of five membership functions namely Very high, High, Medium, Low and Very Low to give a measure of the three parameters used in this paper which are Network Proximity, Server Availability and Bandwidth Availability. In contrast to most works in existing literature, this work employs three metrics to determine request delivery which makes it an improvement on existing works that used other metrics where only three membership functions (High, Medium and Low) were used, this work uses two additional membership functions which help to give a more precise measure of the variables in question and hence improves the efficiency of the request delivery, thus giving users of the network more robust information as regards the denial or grant of requests.

VIII. CONCLUSION

[9], posited that if the response time for a web request exceeds 8 seconds, about 30% of users leave the request.

The increase in response time is directly related to performance loss, congestion and a large number of users reloading the website, making access to the website worse. A common approach is to direct clients request to the replica server closest to the client. However, the closest server may not be the best to direct client's request to in order to maintain high quality of service as opined by [7]. Instead, a set of parameters could be considered during this selection process, such as distance, speed, available bandwidth and surrogate server load. In this paper, three parameters were used to determine the result of the request; they are Network proximity, Server availability and Bandwidth availability. By using fuzzification which is what wasemployed in this paper, it is possible to determine to what extent each of the parameters is rated for aparticularinstanceand based on this we can be able to determine the minimal requirement expected of the three parameters in order for the request to be granted or denied.

For further work another passionate researcher can implement the fuzzy logic algorithm using a computer program and then compare the performance of the fuzzy logic algorithm with other algorithms to determine how efficient our fuzzy logic algorithm is in comparison with other algorithms used in request routing for web content in Content Delivery Networks.

IX. REFERENCES

[1]. Andrews, M., Shepherd, B., Srinivasan, A., Winkler, P., & Zane, F. (2002). Clustering and Server Selection Using Passive Monitoring. *In Proceedings of IEEE INFOCOM.* NY, USA.

[2]. Ardaiz, O., Freitag, F., & Navarro, L. (2001). Improving the Service Time of Web Clients Using Server Redirection. *ACM SIGMETRICS Performance Evaluation Review, 29*(2), 39-44.

[3]. Bartolini, N., Casalicchio, E., & Tucci, S. (2003). A Walk through Content Delivery Networks. *Proceeding of MASCOTS, 2965*, 1-25.

[4]. Broadwell, P. (2004). *Response Time as a Performability Metric for Online Services.* University of California at Berkley, Computer Science Division, Berkley.

[5]. Cardellini, V., Colajanni, M., & Yu, P. (1999). Redirection algorithms for load sharing in distributed web-server systems. *In the Proceedings of the 19th IEEE International Conference on Distributed Computing Systems*, (pp. 528-535).

[6]. Casillas, J.; Cordon, O.; Herrera, F. (1992, August). COR: A Methodology to Improve Ad Hoc Data-Driven Linguistic Rule Learning Methods by Inducing Cooperation Among Rules. *IEEE Transactions on Systems, Man, And Cybernetics - Part B: Cybernetics, 32*(4), 526-537.

[7]. Chen, C. M., Ling, Y., Pang, M., Chen, W., Cai, S., Suwa, Y., & Altintas, O. (2005). Scalable Request-Routing with Next Neighbor Laod Sharing in Multi-Server Environments. *International Conference on Advanced Information Networking and Applications, IEEE Computer Society*, (pp. 441-446). Washington D.C.

[8]. Dahlin, M. (2001). Interpreting stale load information. *IEEE Transactions on Parallel and Distributed Systems, 11*(10), 1033-1047.

[9]. Davison, B. (2001). A web caching primer. *Internet Computing IEEE, 5*(4), 38-45.

[10]. Delgadillo, K. (1997). Cisco DistributedDirector. *Cisco White Paper.* Cisco System, Inc.

[11]. Hofmann, M., & Beaumont, L. R. (2005). *Content Networking: Architecture, Protocols, and Practice.* San Francisco, CA, USA: Morgan Kaufmann Publishers.

[12]. Karger, D., Sherman, A., Berkheimer, A., Bogstad, B., Dhanidina, R., Iwamoto, K., . . . Yerushalmi, Y. (1999). Web Caching with Consistent Hashing. *Computer Networks, 31*(11-16), 1203-1213.

[13]. Katsaros, D., Pallis, G., Stamos, K., Vakali, A., Sidiropoulos, A., and Manolopoulos, Y (2009). *CDNs Content Outsourcing via Generalized Communities.* IEEE Transactions on Knowledge and Data Engineering, VOL. 21, NO. 1, January 2009

[14]. Kearney, A.T.(2010). *A Viable Future Model for the Internet*

[15]. Manfredi, S., Oliviero, F., & Romano, S. (2012). A distributed control law for load balancing in content delivery networks. *IEEE/ACM Transactions on Networking, PP*(99), 1.

[16]. Mao, Z. M., Cranor, C. D., Boughs, F., Rabinovich, M., Spatscheck, O., & Wang, J. (2002). A Precise and Efficient Evaluation of the Proximity between Web Clients and their Local DNS Servers. *In Proceedings of the USENIX 2002 Annual Technical Conference*, (pp. 229-242). Monterey, CA, USA.

[17]. Motwani, R., & Raghavan, P. (2010). *Randomized Algorithms.* Chapman & Hall/CRC.

[18]. Oliveira, T. Q., & Fernandez, M. P. (2013). Fuzzy Redirection Algorithm for Content Delivery Network (CDN). *12th International Conference on Networks* (pp. 137-143). IARA.

[19]. Pai, V., Aron, M., Banga, G., Svendsen, M., Druschel, P., Zwaenepoel, W., & Nahum, E. (1998). Locality-Aware Request Distribution in Cluster-Based Network Servers. *ACM SIGPLAN Notices, 33*(11), 205-216.

[20]. Pallis, G and Vakali, A. (2006).*Insight and Perspectives for content delivery networks.*Communications of the ACM January 2006/Vol. 49, No. 1

[21]. Pathan, A., & Buyya, R. (2008). A Taxonomy and Survey of Content Delivery Networks.

[22]. Pathan, M. (2011, July 5). *Ongoing Trends and Future Directions in Content Delivery Networks (CDNs).*Retrieved December 17, 2013, from amkpathan: http://amkpathan.wordpress.com/article/ongoing-trends-and-future-directions-in-3uxfz2buz8z1w-2/

[23]. Pierre, G., & van Steen, M. (2006). Globule: A Collaborative Content Delivery Network. *IEEE Communications, 44*(8).

[24]. Radware. (2002). *Web Server Director.* Radware Inc.

[25]. Sivasubramanian, S, Halderen, B. van & Pierre, G (2004). *Globule: a User-Centric Content Delivery Network*, Dept. of Computer Science, Vrije Universiteit, Amsterdam

[26]. Szymaniak, M., Pierre, G., & Van Steen, M. (2003). Netairt: A DNS-based Redirection System for Apache. *In Proceedings of International Conference WWW/Internet.* Algrave, Portugal.

[27]. Vakali, A and Pallis, G (2003). *Content DeliveryNetworks: Status and Trends***.** November • December 2003 Published by the IEEE Computer Society 1089-7801/03/$17.00©2003 IEEEInternet Computing

[28]. Wang, L. (2003). The WM Method Completed: A Flexible Fuzzy System Approach to Data Mining. *IEEE Transactions on Fuzzy Systems, 11*(6), 768-782.

[29]. Wang, L., & Mendel, J. (1992). Generating fuzzy rules by learning from examples. *Systems, Man and Cybernetics, Transactions on IEEE, 22*(6), 1414-1427.

[30]. Xu, Z., & Huang, R. (2009). *Performance Study of Load Balancing algorithms in distributed web server systems.* CS213 Parallel and Distributed Processing Project Report.

[31]. Zadeh, L. (1965). Fuzzy Sets. *Information and Control, 8*(3), 338-353.

1

