Fuzzy Redirection Algorithm for Content Delivery Network (CDN)

Thiago Queiroz de Oliveira
Universidade Estadual do Ceará (UECE)
Av. Paranjana 1700
Fortaleza - CE - Brazil
thiagoq@larces.uece.br

Marcial P. Fernandez
Universidade Estadual do Ceará (UECE)
Av. Paranjana 1700
Fortaleza - CE - Brazil
marcial@larces.uece.br

***Abstract*—Content Delivery Network (CDN) is a large distributed system of servers deployed in multiple data centers on the Internet. Its main goal is to serve requests from users providing high availability and performance. It also reduces the system failure risk providing redirection to many replica servers. It can provide load balancing between servers, avoiding network bottlenecks and, therefore, ensuring greater performance and QoE (Quality of Experience) to the end user. One of the critical issues involving CDN networks is the algorithm used to choose the replica server, because it directly influences the performance and scalability of the network. In this paper, an algorithm for choosing the best replica server is proposed. The proposal is compared in simulation against other algorithms in the literature. Finally, we show the effectiveness of the proposed algorithm to improve the choice of the best replica server in CDN.**

Keywords-Content Delivery Network; Fuzzy Logic; Web Server.

I. INTRODUCTION

Content Delivery Network (CDN) is a large distributed system deployed in multiple data centers on the Internet [1]. CDN provides fast and reliable services distributing content by replica servers. The origin to the term CDN was in the 90's with the intention to provide website service with performance, scalability, replication and load balancing [1]. Websites with high-volume traffics, such as e-commerce sites, can present bottlenecks and slowdowns when it is implemented on a single server.

Research has shown that if the response time for a web request exceeds 8 sec, about 30% of users leave the request [2]. 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.

The site replication in different location's aims to: (1) reduce the response time to the nearest user, (2) eliminate a single point of failure, and (3) balance the load among multiple servers. A common approach is to redirect the user to the server closest to him, thus minimizing the bandwidth used, depending on the server's load; this way one can get a shorter response time [3].

One of the critical issues related to CDN concerns which replica server must be used. The closest server to the user is not always the best. Instead, a set of parameters could be considered during this selection process, such as distance, speed, available bandwidth and server load.

This type of algorithm, also known as request routing algorithm, can be divided into two categories: adaptive algorithms and nonadaptive algorithms [4]. In adaptive algorithms, the choice is made based on the server's status, requiring constant monitoring. In nonadaptive algorithms, the choice is based on heuristics, and then a lightweight processing by not requiring monitoring.

This paper proposes a new adaptive algorithm based on fuzzy logic to choose the best replica server. This algorithm considers the following parameters: (1) size of the service queue for each replica server; (2) time needed to answer a request from a given URL on the replica server, (3) response time of replica server (Round-Trip Time (RTT)).

The proposed algorithm can be classified as adaptive, because there is status information exchange between servers. The proposal evaluation shows the reduction of minimum and average response time for a request comparing to other models, validating the proposal.

The evaluation was done in the Network Simulator 2 (ns-2) [5], using the CDN module proposed by Cece et al. [6]. The evaluations were done in three real network topologies obtained in *Topology-zoo* [7]. The proposed algorithm is compared against three different algorithms to redirect the request: two nonadaptative, round-robin and random, and one adaptive algorithm, the *least-loaded* [8]. The metrics used to evaluate are minimum time, maximum time and average time of service request and degree of network unbalancing.

The rest of the paper is structured as follows. In section II, we present some related works, Section III introduces the Content Delivery Networks concepts and fuzzy logic. In Section IV we present our proposal, the FuzzyCDN. Section V shows the proposal evaluation, Section VI shows the results and Section VII concludes the paper.

II. RELATED WORKS

Chen Liao [9] proposes a fuzzy logic based algorithm to choose the replica server in a CDN network. However, in this proposal, the algorithm only act in case of congestion, considering the server bandwidth and the packets drop rate.

 ISBN: 978-1-61208-245-5

Manfredi, Oliviero and Roman [10] proposed an adaptive algorithm based on a mathematical model. In this algorithm, it is considered the request arriving rate and the requisition arriving rate variation in a certain time interval. These parameters are used in the decision-making process, where the last measured time has a weight x, and the average time for the whole period has a weight y, where x is greater than y. The results showed that this algorithm has a good performance.

One of the oldest non-adaptive algorithms is Round-Robin (RR) [11]. Each request is served by a different server, following a cyclic order. This algorithm performs well in homogeneous environment, e.g., the servers have similar capacities; they are in the same place; they share the alike network, and the requests generate a similar workload. However, when one of the conditions is not satisfied, RR algorithm gives bad results.

Another classical algorithm is the Random [12]. As its name suggests, the server will always chose at random. Generally, the system workload is much less than system capacity. Due to its stochastic characteristic, it is not possible to predict its performance.

Dahlin [13] proposes an algorithm called *Least-Loaded* that redirects the request to the server with the lowest load. Another approach would be to choose the server with the shorter response time. Least-Loaded Routing (LLR) algorithm is an algorithm that attempts to distribute the requisition to servers with the largest idle capacity, i.e., the least loaded server. The least loaded server is discovered by monitoring the current server state via protocol [14].

III. Content Delivery Networks

A CDN is a collection of network devices arranged for delivery contents to end users. A CDN network could be implemented in many architectures and topologies, which can be centralized, hierarchical, infra-structured with administrative control and decentralized [15].

The CDN provides better performance by offering content caching and server replication scattered strategically in order to address requests overloads for web content, which is called *flash crowd* [16] or *SlashDot effect* [17].

The design of a CDN network is quite complex [18]. To build a CDN network, some strategic issues must be defined [19]: (1) how many replica servers should be used and where they should be located [20]; (2) what content should be replicated and in which server should be replicated [21]; (3) what strategy should be used to keep the replica server's contents consistent; (4) how it chooses the replica server, and what mechanism should be used to redirect the client to the replica server.

The routing request process is responsible for performing customer's requisition routing to the best replica server. One solution is to redirect the request to the nearest server replica. However, not always the closest replica server is the best to serve a request [3]. The ideal is to consider another metrics to perform routing: network proximity, connection latency, distance and the replica server load.

The routing request should be divided into two mechanisms: (1) routing request algorithm, which is used to define the best replica server to answer a request; and, (2) request routing mechanism, that is responsible for performing the redirection from client to the server [18].

A. Routing Request Algorithm

The routing request algorithms are divided into two categories: adaptive algorithms and non-adaptive algorithms. The adaptive algorithms consider the current state of the replica servers before define the server to be used. The non-adaptive algorithms do not consider the server's state, i.e., no overhead is introduced to exchange status information between servers. In adaptive algorithms, the choice of replica server is based on its current state. Then, it is necessary to monitor the server's load and network congestion. As the network and server state can vary very fast, these algorithms consume many bandwidth. In non-adaptive algorithms, the replica server choice could be made using heuristics. Such solution is less complex, consumes less bandwidth, but are not efficient as the adaptive algorithms.

B. CDN Performance

To evaluate a CDN network performance, some metrics can be used [18] [22]: (1) temporal metrics, the time client expects to have his request served [23]; (2) space metrics, can be geographic distance or number of hops RTT [24]; (3) network use metrics, associated to network resources consumed, it can be internal, when communication is among servers to network management and software updates, and external, related to communication between clients and servers [18]; (4) cost metrics: they relate to server's acquisition and maintenance costs; (5) consistency metrics, related to consistency of the content accessed by users.

IV. FuzzyCDN: A Fuzzy algorithm to choose CDN replica server

This work proposes an adaptive algorithm to choose the replica server based on fuzzy logic [25]. The fuzzy logic unlike classical Boolean logic, can assign intermediate values between true and false and also, making a decision based on in-between inputs. Working with a logic that permits dealing with subjective information, imprecise and ambiguous, opens many possibilities to develop solutions to problems that classical logic is not able to solve.It considers following variables as input:

1) Queue size: queue length on replica server.
2) Service time: time to answer a request from a given URL on the replica server.

 ISBN: 978-1-61208-245-5

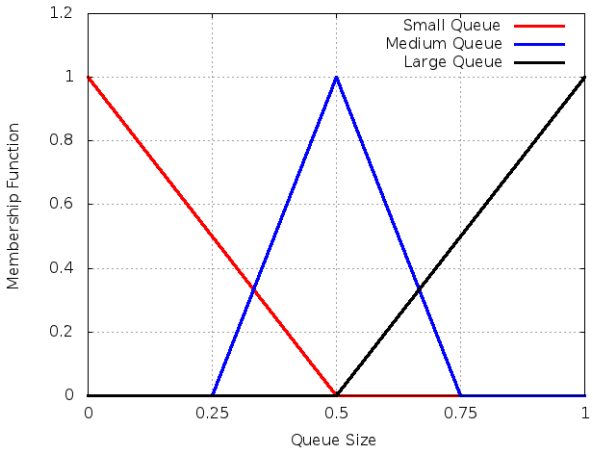

Figure 1. Membership functions to the variable queue

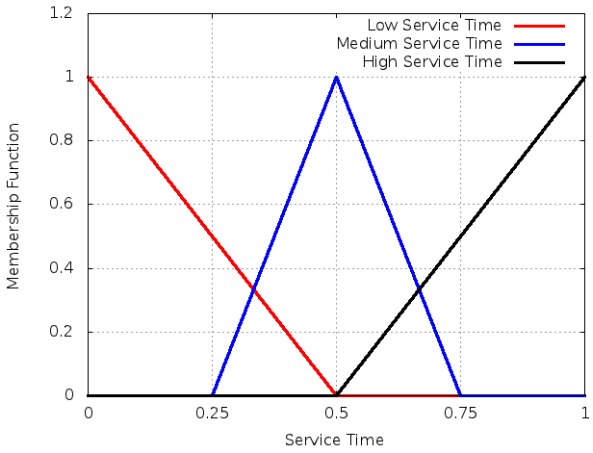

Figure 2. Membership functions to the variable queue service time

3) Response time: replica server response time (Round-Trip Time (RTT)).

Every server has a list of neighbor's servers. The servers periodically exchange status information with its neighbors. With this information, when a server receives a request, it performs the normalization of neighbor's status and using the inference mechanism, which will be described in the following sections, decide which server will attend the request.

A. *Fuzzification*

In this step, the mapping of input parameters, generally numerical and accurate, for fuzzy sets is realized using the membership functions. The input membership functions are triangular, and all of then have the same characteristics within the function range. The queue size variable has three linguistic values associated; they are: small queue, medium queue and large queue. The Figure 1 show graphically this membership functions, along with their respective ranges.

The service time variable also has three membership functions associated; they are: low service time, medium service time and high service time. Figure 2 show graphically this membership function. The variable response time also has three membership functions (Low, Medium and High) as shown in Figure 3.

The output consists of five membership functions using triangular functions (Very Good, Good, Normal, Bad, Very Bad), as shown in Figure 4.

B. *Rule evaluation*

The fuzzy controller inference rules definition is not a simple task, because it can produce inconsistent results. To avoid creating wrong rules, it was used the Wang-Mendel algorithm [26], in order to create consistent rules. The Wang-Mendel algorithm provides a method to create fuzzy logic rules by five phases:

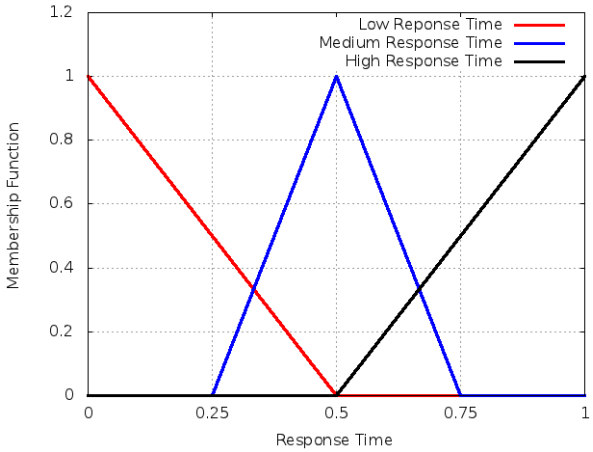

Figure 3. Membership functions to the variable queue response time

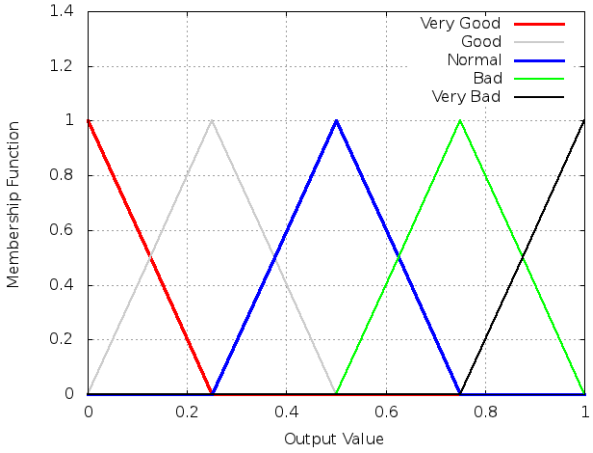

Figure 4. Membership functions to the output variable

1) Divides the input and output spaces from a given

numerical data into fuzzy regions.
2) Generates fuzzy rules based upon the given data.
3) Assigns a degree for each generated rules for resolving conflicts.
4) Creates a fuzzy rule set based on the generated rules and linguistic rules.
5) Defines a mapping from input space to output space based on the combined fuzzy rule base using the defuzzifying procedure.

The first four steps generate the knowledge base and compose the training stage. The last step generates the output values for the possible entries from the knowledge base. The inference rules obtained are shown in Table I.

Table I
FUZZY INFERENCE TABLE

Queue Size	Service Time	Response Time(RTT)	Output
Small	Low	Low	Very Good
Small	Low	Medium	Very Good
Small	Low	High	Very Good
Small	Medium	Low	Good
Small	Medium	Medium	Good
Small	Medium	High	Good
Small	High	Low	Good
Small	High	Medium	Normal
Small	High	High	Normal
Medium	Low	Low	Good
Medium	Low	Medium	Good
Medium	Low	High	Good
Medium	Medium	Low	Normal
Medium	Medium	Medium	Normal
Medium	Medium	High	Normal
Medium	High	Low	Bad
Medium	High	Medium	Bad
Medium	High	High	Bad
Large	Low	Low	Normal
Large	Low	Medium	Normal
Large	Low	High	Bad
Large	Medium	Low	Bad
Large	Medium	Medium	Bad
Large	Medium	High	Bad
Large	High	Low	Very Bad
Large	High	Medium	Very Bad
Large	High	High	Very Bad

C. Defuzzification

The Algorithm 1 shows the controller operation in the decision-making process. The algorithm gets the URL request as input and gives the server chosen as output. In line 3, it gets the servers list for a given URL. Then, from lines 4 to 13, it calculates the fuzzy value for each replica server using as parameters the server queue, service time and response time.

Then, the parameter values are mapped to fuzzy sets according to their membership functions as described above. After this step, the inference rules from Table I are applied, resulting in a fuzzy set output, showed on Table 4. Finally, it is held deffuzifaction that returns a quantitative value using the center area method. The algorithm returns the server with the lowest fuzzy value.

Algorithm 1: FuzzyCDN algorithm

```
1  s_min ← NULL;
2  server ← NULL;
   Input: A request R for a given URL
3  n_s ← number of servers having the URL;
4  while i = 1, ..., n_s do
5      f_i ← queue size of replica server i;
6      ts_i ← service time of server i for a given URL;
7      tr_i ← response time of server i (RTT);
8      s ← calculate the fuzzy value(f_i, ts_i, tr_i);
9      if (s < s_min) or (s_min = NULL) then
10         s_min ← s;
11         server ← i;
12     end
13 end
14 return the most appropriate server
```

V. PROPOSAL EVALUATION

The simulator used in the experiments was the ns-2 (*Network Simulator*network simulator), version 2.33 [5]. It was used the ns-2 CDN module developed by [6].

The client sends the request to the nearest server containing the desired content. The replica server choice mechanism is decentralized, where any server can decide whether it serves the request or for which server should forward the request.

Every server has an associated service time, which represent to the processing time required to serve a given request. Any request received is inserted into a queue, and the server will decide whether the request will be answered or redirected. The queuing model may be D/D/1 or M/M/1 for deterministic or exponential distribution, respectively.

Each server maintains a list of neighbors. The servers exchange periodically status information with their neighbors. This information is used to select the server to a given request.

A. Evaluation Topologies

For the tests, it was used three topologies of real networks, obtained at the Topology-zoo site [7], which are: Claranet [27], GridNet [28] and RNP [29].

B. Experiments Details

In the experiments, the customers' number is the same of the servers, and they vary according to the topology. The clients always send the original request to the nearest server, this request follows a Poisson process with arrival rate λ_i. Each server has a service rate μ_i.

The time that servers exchange information status is 1s. The traffic generated to the servers is CBR, the packet sent is HTTP and the simulation time for each experiment is 300 sec.

 ISBN: 978-1-61208-245-5

The first two are considered non-adaptive algorithms, and the last one is an adaptive algorithm. The round-robin algorithm chooses a different server every decision. The random algorithm uses a stochastic process to choose the server, and the least-loaded algorithm chooses the server with the smaller queue.

C. Traffic Model

The GridNet network is composed of nine servers, located in the southern U.S. In the experiments, nine clients generate traffic to nine servers, one server for each client. Table II shows the characteristics of each server based on parameters used by Manfredi [10].

Table II
TRAFFIC CHARACTERISTICS: GRIDNET

	1	2	3	4	5	6	7	8	9
$\lambda_i[req/s]$	12	10	7	10	13	11	11	14	17
$\mu_i[req/s]$	12	13	14	17	10	11	11	7	10

In Table II, λ_i is the requests arrival rate to the server i and μ_i is the service rate of server i. According to [10], this traffic pattern described in Table II represents a real CDN network. The *flash-crowd* effect is also simulated by increasing the arrival rate at Server 7, λ_7 of 11 requests per second to 200 requests per second, between the time $t_0 = 200sec$ and $t_1 = 250sec$. The other topologies have a similar traffic model, changing only the number of servers. Then, it will not be shown.

VI. RESULTS

The metrics used for evaluation are: minimum time, medium time and maximum time for answering a request; standard deviation of answering request and network degree of unbalancing. The network unbalancing degree is the standard deviation of the queues size in all servers along the time. Therefore, smaller network unbalancing degree is better.

The following figures show graphically the results of each metric for the four algorithms on these three topologies. Despite not having much precision in differentiating values, these graphs enable you to have an assessment of each metric.

In Figure 5, the minimum time was obtained by the fuzzy algorithm. The others' algorithms have similar results. In contrast, according to Figure 6, the proposed fuzzy algorithm presented the worst maximum time on these three topologies.

In Figure 7, it was observed that the adaptive algorithms showed better results for the average time on the GridNet and RNP. In all three topologies, the proposed algorithm showed the best results. Figure 8 shows that adaptive algorithms have the lowest standard deviation in the three topologies, especially the fuzzy algorithm which had the lowest standard deviation in all tests. This good result of

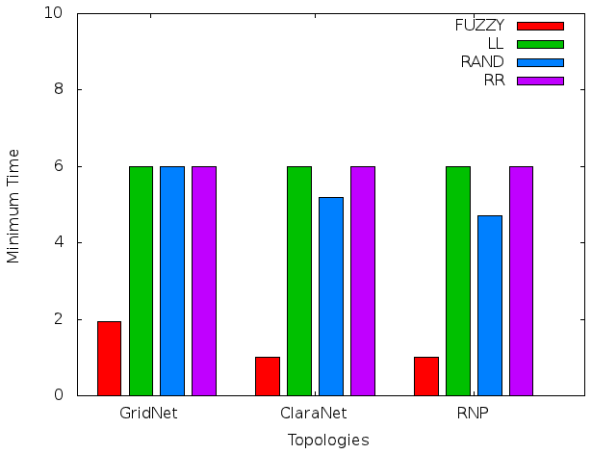

Figure 5. Minimum Response Time

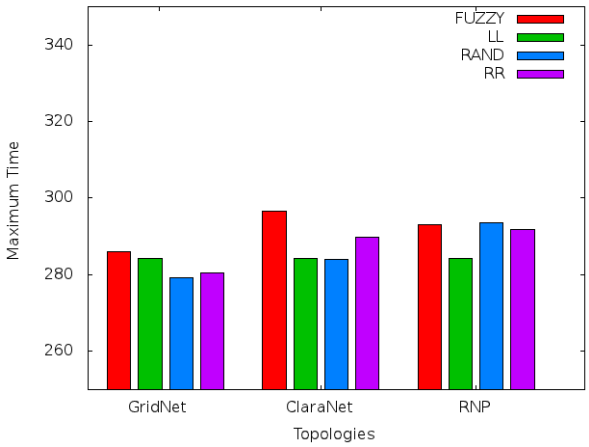

Figure 6. Maximum Response Time

the standard deviation, and with the good results of the minimum and average time to obtain a request enables the deployment of the proposed algorithm in real CDN networks.

VII. CONCLUSION AND FUTURE WORKS

In this paper, we 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, we show 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.

 ISBN: 978-1-61208-245-5

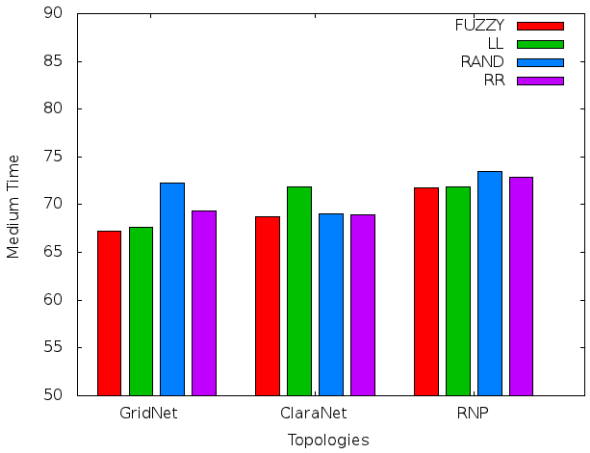

Figure 7. Average Response Time

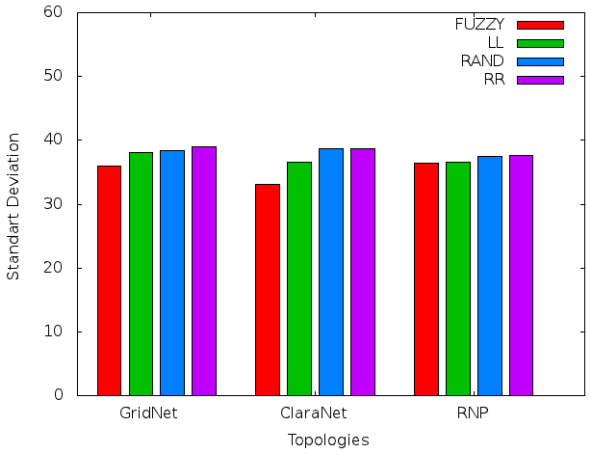

Figure 8. Std Deviation Response Time

Although we know that Fuzzy logic produces a low impact in modern processor's performance, we do not guarantee the algorithm scalability.

As future work, we can analyze another metrics algorithm, for example, the amount of available bandwidth on the link. It should be used a learning technic to choose the replica server in CDN networks, such as neural networks, Bayesian networks or Support Vector Machine (SVM).

REFERENCES

[1] G. Pallis and A. Vakali, "Insight and perspectives for content delivery networks," *Communications of the ACM*, vol. 49, no. 1, pp. 101–106, Jan. 2006.

[2] B. Davison, "A web caching primer," *Internet Computing, IEEE*, vol. 5, no. 4, pp. 38–45, jul/aug 2001.

[3] C. Chen, Y. Ling, M. Pang, W. Chen, S. Cai, Y. Suwa, and O. Altintas, "Scalable request routing with next-neighbor load sharing in multi-server environments," in *Advanced Information Networking and Applications (AINA)*. Taiwan: IEEE, Mar 2005, pp. 441–446.

[4] in *Content Delivery Networks*, ser. Lecture Notes Electrical Engineering, R. Buyya, M. Pathan, and A. Vakali, Eds., 2008, vol. 9.

[5] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu *et al.*, "Advances in network simulation," *Computer*, vol. 33, no. 5, pp. 59–67, 2000.

[6] F. Cece, V. Formicola, F. Oliviero, and S. Romano, "An extended ns-2 for validation of load balancing algorithms in content delivery networks," in *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*. Malaga/Spain: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010, p. 32.

[7] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 9, pp. 1765–1775, 2011.

[8] M. Dahlin, "Interpreting stale load information," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 11, no. 10, pp. 1033–1047, 2000.

[9] J. Chen and S. Liao, "A fuzzy-based decision approach for supporting multimedia content request routing in cdn," in *Parallel and Distributed Processing with Applications (ISPA), 2010 International Symposium on*. IEEE, 2010, pp. 46–51.

[10] S. Manfredi, F. Oliviero, and S. Romano, "A distributed control law for load balancing in content delivery networks," *Networking, IEEE/ACM Transactions on*, vol. PP, no. 99, p. 1, 2012.

[11] Z. Xu and R. Huang, "Performance study of load balancing algorithms in distributed web server systems," *CS213 Parallel and Distributed Processing Project Report*, 2009.

[12] R. Motwani and P. Raghavan, *Randomized algorithms*. Chapman & Hall/CRC, 2010.

[13] M. Dahlin, "Interpreting stale load information," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 11, no. 10, pp. 1033–1047, 2000.

[14] V. Cardellini, M. Colajanni, and P. Yu, "Redirection algorithms for load sharing in distributed web-server systems," in *Distributed Computing Systems, 1999. Proceedings. 19th IEEE International Conference on*. IEEE, 1999, pp. 528–535.

[15] D. Verma, *Content distribution networks*. Wiley, 2002.

[16] M. Arlitt and T. Jin, "A workload characterization study of the 1998 world cup web site," *Network, IEEE*, vol. 14, no. 3, pp. 30–37, 2000.

[17] S. Adler, "The slashdot effect: an analysis of three internet publications," *Linux Gazette*, vol. 38, p. 2, 1999.

 ISBN: 978-1-61208-245-5

[18] S. Sivasubramanian, M. Szymaniak, G. Pierre, and M. Steen, "Replication for web hosting systems," *ACM Computing Surveys (CSUR)*, vol. 36, no. 3, pp. 291–334, 2004.

[19] J. Wang, R. Sharman, and R. Ramesh, "Shared content management in replicated web systems: A design framework using problem decomposition, controlled simulation, and feedback learning," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 38, no. 1, pp. 110–124, 2008.

[20] L. Qiu, V. Padmanabhan, and G. Voelker, "On the placement of web server replicas," in *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3. IEEE, 2001, pp. 1587–1596.

[21] J. Kangasharju, J. Roberts, and K. Ross, "Object replication strategies in content distribution networks," *Computer Communications*, vol. 25, no. 4, pp. 376–383, 2002.

[22] M. Harchol-Balter, B. Schroeder, N. Bansal, and M. Agrawal, "Size-based scheduling to improve web performance," *ACM Transactions on Computer Systems (TOCS)*, vol. 21, no. 2, pp. 207–233, 2003.

[23] D. Olshefski, J. Nieh, and D. Agrawal, "Using certes to infer client response time at the web server," *ACM Transactions on Computer Systems (TOCS)*, vol. 22, no. 1, pp. 49–93, 2004.

[24] B. Huffaker, M. Fomenkov, D. Plummer, D. Moore, and K. Claffy, "Distance metrics in the internet," in *Proc. of IEEE International Telecommunications Symposium (ITS)*, Natal/Brazil, 2002.

[25] L. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[26] L. Wang and J. Mendel, "Generating fuzzy rules by learning from examples," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 22, no. 6, pp. 1414–1427, 1992.

[27] Claranet, "Claranet Limited," Last accessed, Mar 2012. [Online]. Available: http://noc.eu.clara.net/network.jpg

[28] GridNet, "GridNet Inc Network," Last accessed, Mar 2012. [Online]. Available: http://www.nthelp.com/images/gridnet.jpg

[29] RNP, "Rede Nacional de Pesquisa," Last accessed, Mar 2012. [Online]. Available: http://www.rnp.br/backbone

 ISBN: 978-1-61208-245-5