

Fuzzy Logic Based Quality of Service Broker Model: Simulation in Simulink

A.O. Ajayi, G.A. Aderounmu, E.A. Olajubu and S.A. Bello

Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria

Abstract: Today, increasingly complex control problems abound, for which mathematical modelling representations may be difficult to obtain. This study relied on the reported strength of the fuzzy logic technique in controlling highly nonlinear systems to improve the user perceived Quality of Service (QoS) of the web application. The fuzzy logic based QoS broker model was designed, modelled, and simulated using MATLAB 7.0; its performance was evaluated using two QoS parameters: Response time and throughput. The simulation results revealed an increase in throughput as well as a reduction in response times perceived by the clients.

Key words: Bandwidth, inference engine, response time, throughput, traffic

INTRODUCTION

Web applications such as e-commerce, are usually accessed via the web over a network. Workload pattern is a major challenge to the network administrators administering the performance of the infrastructures supporting these applications. Quality of service (QoS) provides better service to selected traffic over various technologies to maintain a good level of user-perceived QoS even when the network conditions are unpredictably changing, and it is usually expressed by performance metrics such as response time, throughput, predictability, and consistent perceptual quality. Though, QoS provisioning in the Internet has been a topic of active research in the last few years, and due to financial and technical reasons, many of the proposed solutions are not commonly employed in practice (Romano *et al.*, 2004). It is generally recognized that poor performance perceived by the buyers can be a major impediment of the e-commerce success (Nielsen, 1999). Thus, a scalable and intelligent QoS brokering model that provides different levels of QoS to sessions from different customers is required. However, many of the parameters to be considered are numerous and building the required mathematical model may be difficult, hence in this study, a simulation technique in Simulink environment that applied the fuzzy inference engine to effectively optimize the web server QoS provisioning and other resources is proposed. Specifically, an intelligent QoS broker is simulated to improve the user perceived response time by providing different levels of QoS to clients sessions based on the limited bandwidth between clients and Internet Service Provider (ISP), and the clients computing system characteristics using the MATLAB software, the leading simulator software specialized for science and engineering applications.

LITERATURE REVIEW

Existing works on the simulation of web QoS provisioning abound, a review of few of these follow. An admission control algorithm for web servers was designed in (Cherkasova and Phaal, 2002), which consists of the clients, the queue, the web server, and a session generator. The simulated model using CSim software ignored the dynamic contents and the different classes of clients. Menasce and Bennani (2003) reported a model that self-adjusts configuration parameters of a web server to meet QoS requirements using C/C++ language. The model however, did not consider different classes of requests. Teixeira *et al.* (2004) proposed a provisioning technique, simulated using Simpack for differentiated services in web servers, which considered only the static content. A QoS differentiation technique on e-commerce servers, evaluated using simulation was also discussed in (Zhou *et al.*, 2004), where the server is modelled as a single abstract resource, with an exponential service time. However, the model ignored real demand of requests on different resources.

Web server QoS models, simulated in OPNET, that used a single queue and scheduling rules to differentiate QoS for classes of requests with different priorities were presented in (Ye *et al.*, 2005). The processing on the server is modelled using two finite state machines, with service time of requests following the Pareto distribution. This model assumed static contents in the web server, which is unacceptable today. The major problem with these existing schemes is that they focused on controlling the web server response time and not the response time perceived by the end users. Other relevant factors that need consideration in creating a highly valued customer experience include limited bandwidth and clients computing characteristics (processor speed, memory size,

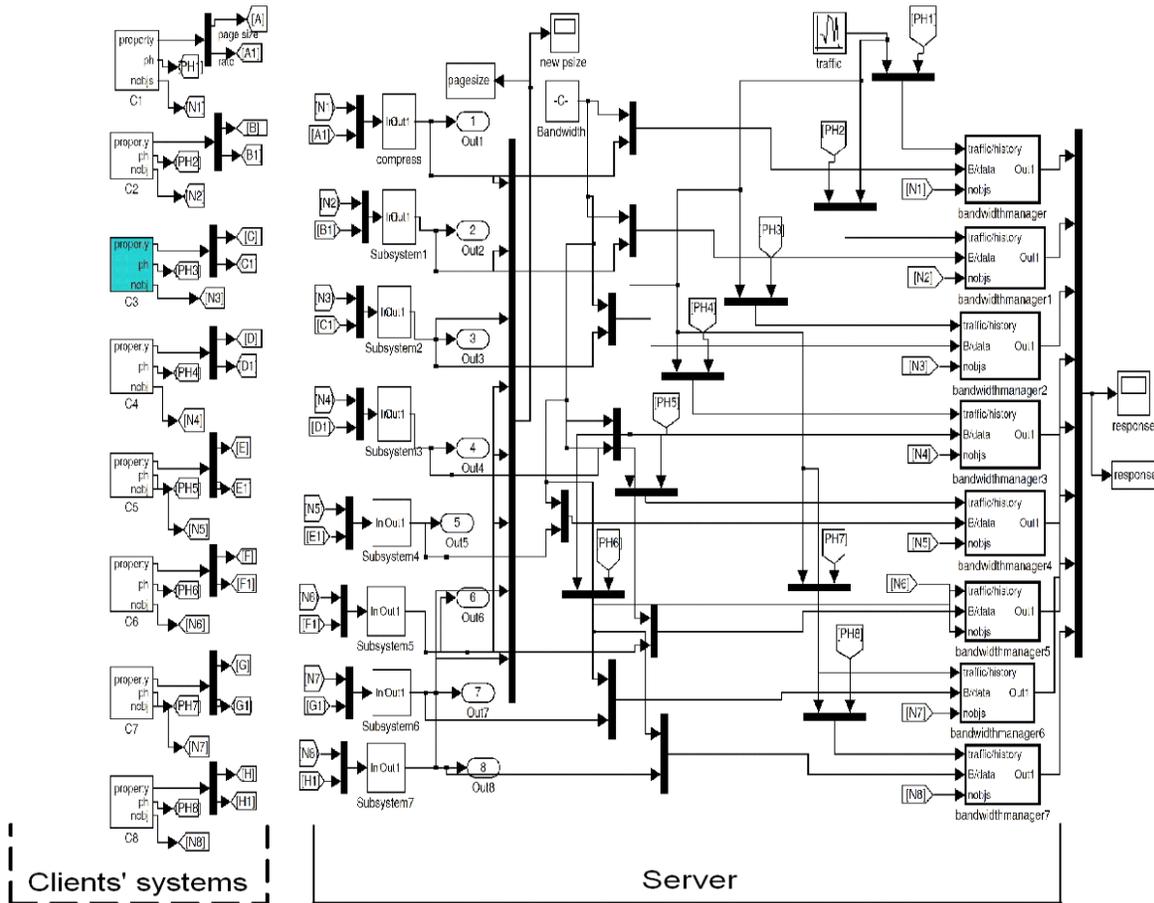


Fig. 1: Top level view of the simulation environment

+screen resolution, and so on) that may cause intolerable and unpleasant response time experienced by the users. A scalable fuzzy logic based QoS brokering model is presented to provide different levels of QoS to clients sessions in order to reduce the poor performance perceived by users.

PROPOSED QOS BROKER

The proposed intelligent QoS provisioning framework consists of the following modules:

Fuzzy inference QoS broker (FIQB): This module allots bandwidth to clients by classifying them into classes (high, medium, or low) using three input parameters, namely, client’s performance measured over a given time period, request type (browsing, searching, payment, checkout, order), and traffic state (light, normal, or heavy). The output variable is the bandwidth. These parameters (input and output) are made up of linguistic variables and fuzzy sets, while the control action is described by a set of fuzzy conditional rules such as:

If traffic is light and performance is normal and request is low then bandwidth is normal

Membership function is used to evaluate the degree to which an input belongs to a fuzzy set, and after a careful analysis of the data, the triangular and trapezoidal membership functions were selected.

Clients performance in relation to their interaction with the website is computed using (1), adapted from (Ajayi *et al.*, 2010). Where PF is the ratio of frequency, PV is the ratio of volume, nP is the number of times a purchase is made, nV is the number of visits by client, cp_i is the accumulated volume of purchase over a given period, and ts_j is the total sales made by the e-commerce

$$PF = nP/nV \tag{1}$$

$$PV = \left(\frac{\sum_i^n cp_i}{\sum_i^n ts_j} \right) * 100$$

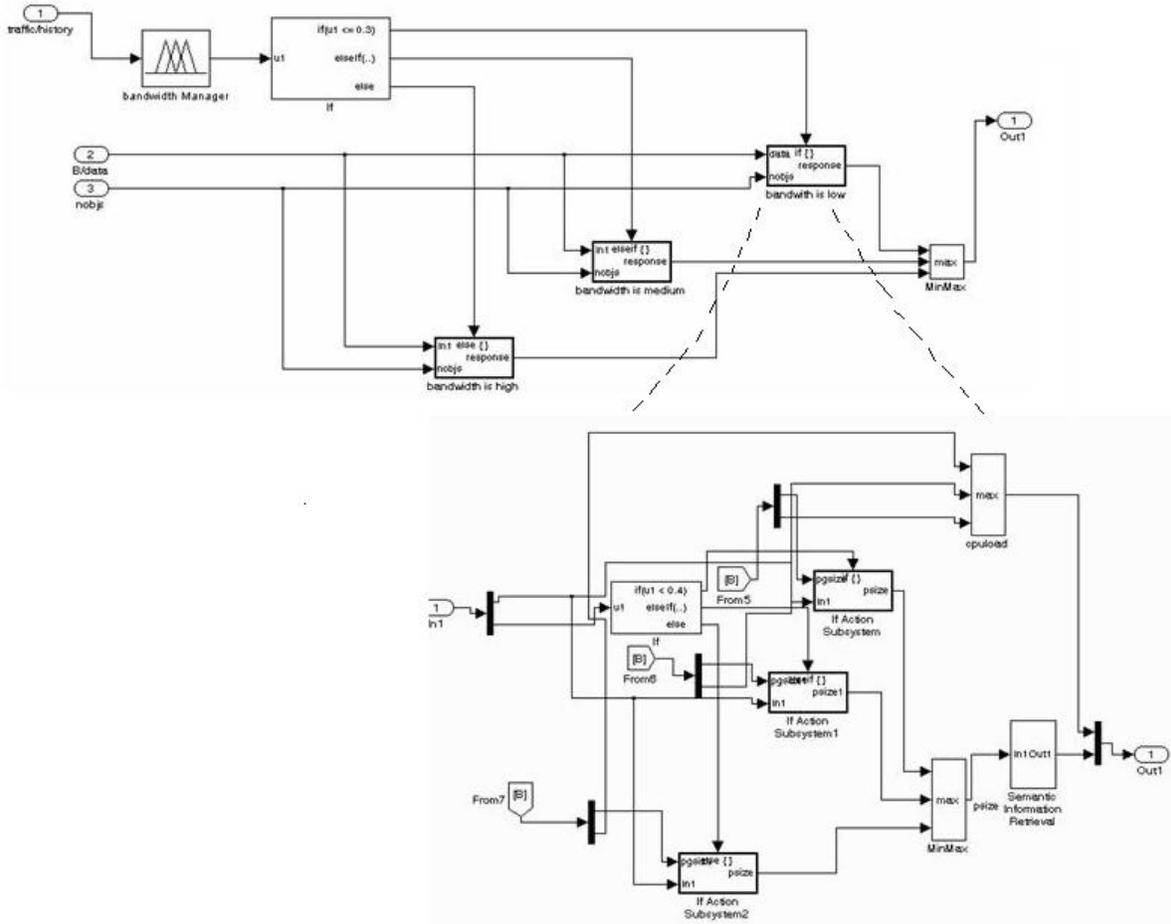


Fig. 2: A sub-module that classifies requests to classes

The FIQB places requests in any of the three classes, and implements the Class Based Queuing (Floyd and Jacobson, 1995) scheduling principle to service the requests.

Fuzzy Semantic Information Retrieval (FSIR): This module is composed of representations from database contents and user queries in conjunction to a ranking algorithm. Since more than 70% of the functions performed by e-commerce clients are products searching (Menasce *et al.*, 2000), this module ensures that exact information required is extracted, thus reducing the size of information transmitted. Semantic ambiguity is handled using fuzzy logic concepts as well, in which objects' features are divided into two sets $C(C_1 \dots C_n)$ and $F(F_1 \dots F_k)$, to represent the category and the attribute of objects in the database. Also, a query vector $Q(C_q, F_q)$ is formulated, where C_q is the query object's category vector and F_q is its feature vector. A ranking algorithm combines the category and feature distances between the query and targets in the database to produce the function $D(S, Q)$,

which is the total distance of the confidence values of all the considered features sorted in ascending order. The total distance function (2) is adapted from (Huang and Tsai, 2005).

$$D(S, Q) = W_1 * d_1(C, C) + W_2 * d_2(F, F_q) \quad (2)$$

where, $d_1(C, C_q)$ and $d_2(F, F_q)$ are defined as:

$$d_2(C, C_q) = \left[\sum_{i=1}^z (C[i] - C_q[i])^2 \right]^{1/2},$$

$$d_2(F, F_q) = \left[\sum_{i=1}^a r * \left[\sum_{j=1}^p F_i[j] - F_{qi}[j] \right]^2 \right]^{1/2}$$

and d_1 is the category distance, d_2 is the feature distance, p is the number of attributes in the i^{th} feature, and

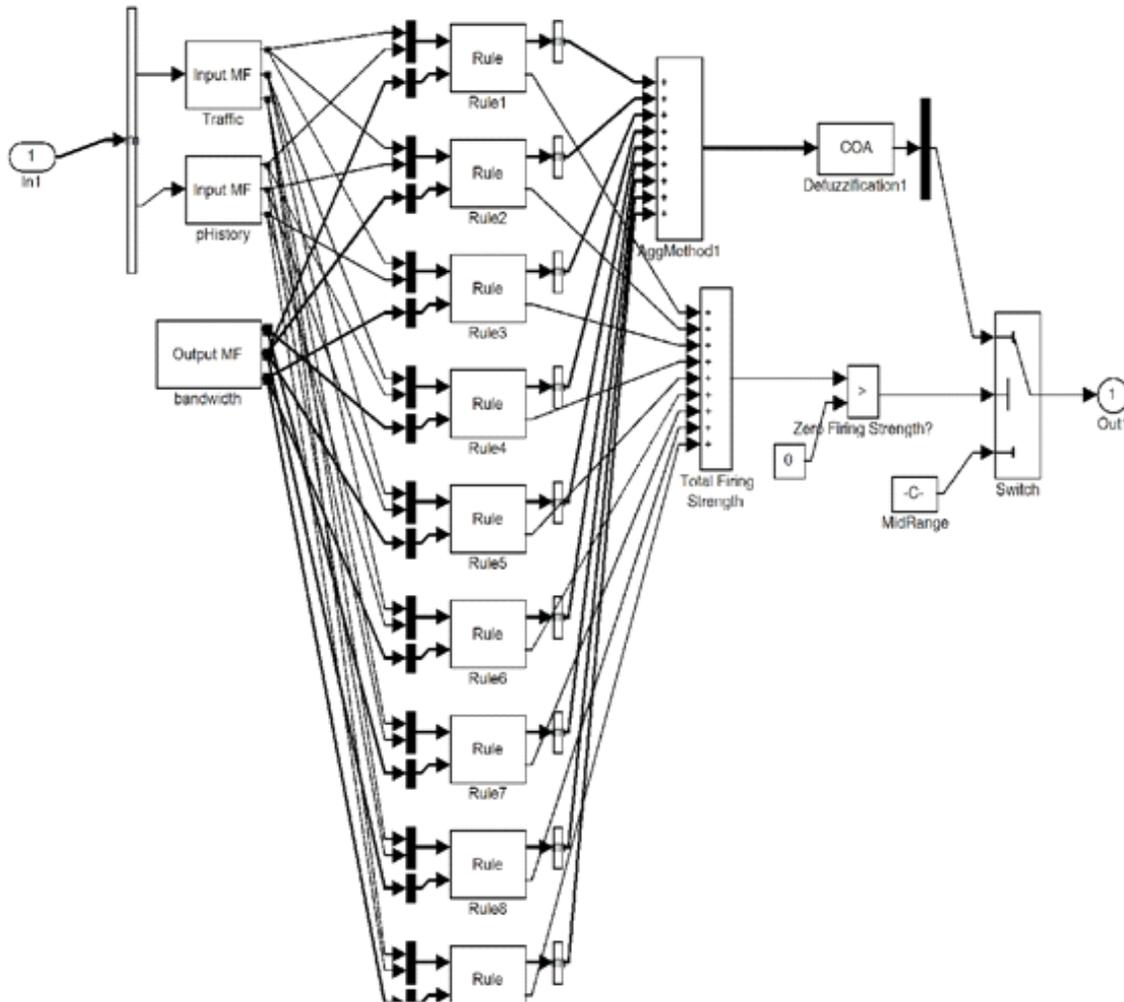


Fig. 3: Fuzzy inference engine that allots bandwidth to users requests

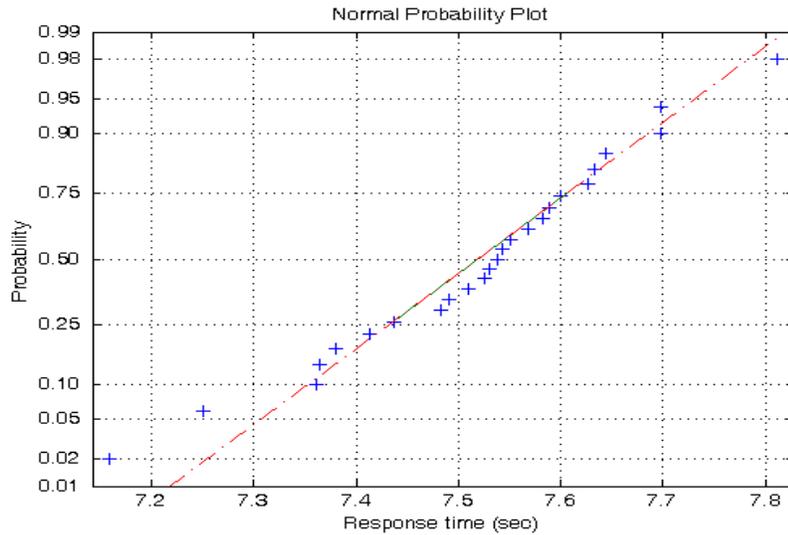
$(w_1 + w_2) = 1$. The weights w_1 and w_2 are the significant levels for d_1 and d_2 respectively, n is the number of category, m represents the feature and r_i is the probability of each option of the i^{th} feature.

Client system performance evaluator: This module is a fuzzy inference engine that ranks the client resources (processor speed, memory size, screen resolution, and so on.) based on their present condition to ensure that appropriate data is delivered by the server to the client to reduce the processing delay at the later. Information on system performance is made available to the server to determine the details of the multimedia data that is appropriate for the client system.

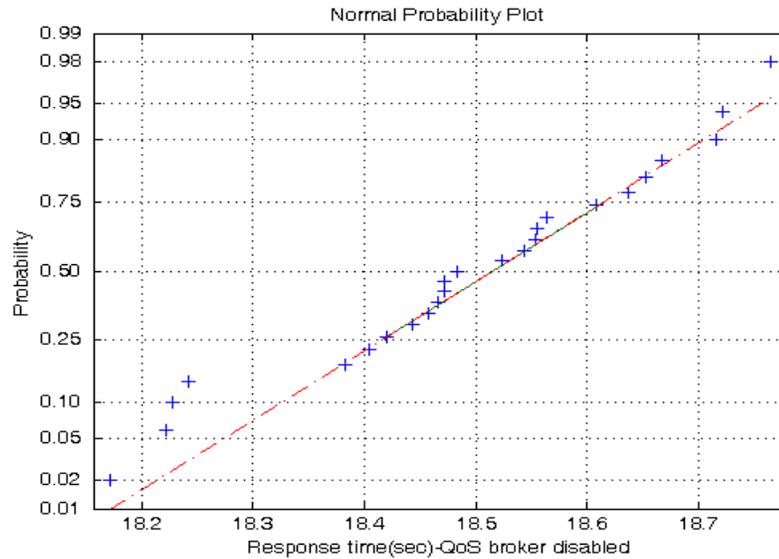
SIMULATION AND RESULTS

The MATLAB software supports the design of communication networks, devices, controls, and

applications with great ease and flexibility. In addition, the excellent GUI and block module provided by Simulink allow users to easily build, simulate, and test models through standard and custom block libraries. Furthermore, since the MATLAB contain Fuzzy Logic Toolbox, the software turn into a powerful intelligent systems simulation and analysis tool. The simulated experiment carried out in Simulink, an integral part of the MATLAB software, compared two scenarios: an environment when the proposed QoS broker is enabled, and when the proposed QoS broker is disabled. The two scenarios were designed to approximate the typical e-commerce operations (browsing, product search, and payment) performed on a typical web server. The search operation followed a Pareto distribution with shape parameter 65536 and scale parameters of 1.8 and 1.2, respectively. Thus, average sizes of returned pages are between about 15 Kb to about 40 Kb. The service rates for clients' requests in the three classes (high, medium,



(a) proposed QoS broker enabled



(b) proposed QoS broker disabled

Fig. 4: Response time experienced with proposed QoS broker enabled and disabled

low) are 50, 30 and 20% respectively, and the total bandwidth link was fixed at 1544000 bits/sec. The simulation was conducted on a Pentium 2.8 GHz CPU and 512 MB of RAM running on the Windows XP operating system. Figure 1 depicts the high level view of the Simulink model built hierarchically using the top-down approach. Figure 2 and 3 are few examples of sub modules of the model. Figure 2 for instance, depicts a fuzzy inference sub module that classifies clients' requests into classes (high, medium, and low) using the following parameters such as user computing system's

rating, client's performance rating, traffic condition, type of e-commerce operations (browsing, payment, shopping, etc.) while Fig. 3 is the fuzzy inference engine module that computes data rate which is allotted to clients. Eight end-user systems (C1-C8), as shown in Fig. 1 competed for bandwidth resource while interacting with the server. In terms of system's working condition, the systems were configured as follows: C1, C5, C6, C7, and C8 were rated poor; C2 was rated as good; and C3 and C4 were rated average. Clients using systems C1, C3, and C4 to interact with the web server, had request type configured to

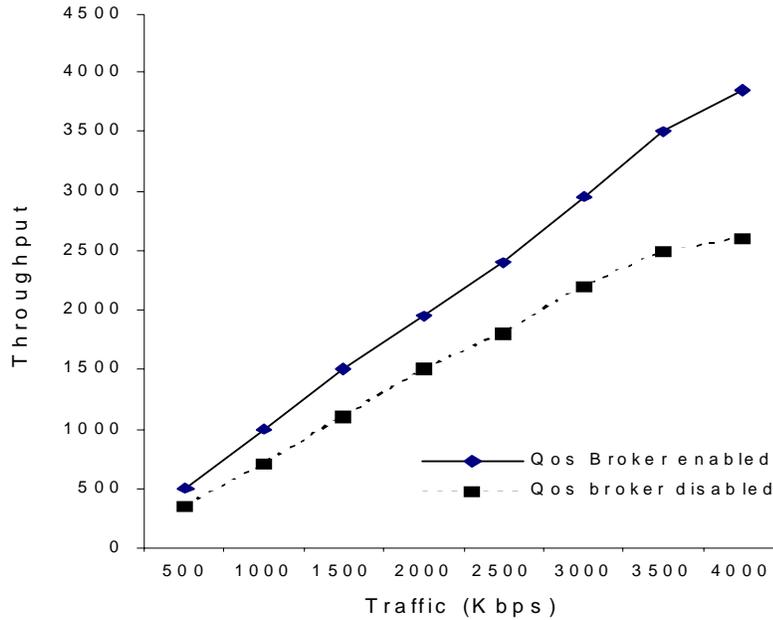


Fig. 5. System throughput

payment (high), while clients using systems C2, C5, C8 had request type configured to product searching (medium), and clients using systems C6 and C7 had request type configured to browsing (low). Similarly, to provide different levels of QoS to clients based on their purchase history, clients using systems C1 and C2 were configured to have good performance history (heavy buyers), clients using systems C4 and C8 had average performance history, while clients using systems C3, C5, C6 and C7 had poor performance history (occasional buyers). Each simulation result is an average of 100 runs.

The QoS provisioning ability of the fuzzy logic QoS broker was measured using response time and throughput as performance parameters. Fig. 4a and b is a normal plot showing the response times experienced by clients when the intelligent QoS is enabled and when it is disabled. As shown in Fig. 4a, 90% of clients experienced response times less than 8s when the intelligent QoS was enabled while 90% of the clients experienced response time around 18s when it was disabled (Fig. 4b). Figure 5 depicts the throughput of the system, a throughput of 98% was achieved when the intelligent QoS broker was enabled compared to 72% when it was disabled.

CONCLUSION

The goal of any QoS mechanism is to maintain a good level of user-perceived QoS even when the network conditions are constantly changing. Limited bandwidth and poor state of client systems can be attributed to the high response times perceived by online buyers, and as

confirmed in (Olshefski and Nieh, 2006), previous QoS approaches focused on controlling only the web server response time of individual URL requests. This study formulated a fuzzy logic based QoS broker that considers the limited bandwidth between end user and ISP; and the characteristics of the client's device (processor speed, processor load, screen resolution, and so on) to improve response times perceived by online buyers. In addition, the model manages network and computing resources optimally to reduce traffic on the network.

ACKNOWLEDGMENT

The authors are grateful to the Obafemi Awolowo University, Ile-Ife, for providing the enabling environment to carry out the research.

REFERENCES

- Ajayi, A.O., G.A. Aderounmu, H.A. Soriyan and A. David, 2010. An intelligent quality of service brokering model for e-commerce. *Expert Sys. Appl.*, 37(1): 816-823.
- Cherkasova, L. and P. Phaal, 2002. Session-based admission control: A mechanism for peak load management of web sites. *IEEE Trans. Comp.*, 51: 669-685.
- Floyd, S. and V. Jacobson, 1995. Link-sharing and resource management models for packet networks. *IEEE/ACM Trans. Network.*, 3(4): 365-386.

- Huang, Y.P. and T. Tsai, 2005. A fuzzy semantic approach to retrieving bird information using handheld devices. *IEEE Intell. Sys.*, 20(1): 16-23.
- Menasce, D. and M. Bennani, 2003. On the Use of Performance Models to Design Self-Managing systems. *Proceedings of the Computer Measurement Group Conference*, Dallas, TX, December, pp: 7-12.
- Menasce, D., R. Fonseca, V. Almeida and M. Mendes, 2000. Resource management policies for E-Commerce servers. *ACM Perf. E.R. Si.*, 27(4): 27-35.
- Nielsen, J., 1999. Why People Shop on theWeb. Retrieved from: <http://www.useit.com/alertbox/990207.html>.
- Olshefski, D. and J. Nieh, 2006. Understanding the Management of Client Perceived Response Time. *Proceedings of the ACM SIGMETRICS Conference on Measurements and Modelling of computer systems*, Saint Malo, France, June 26-30, pp: 240-251.
- Romano, P., F. Quaglia and B. Ciciani, 2004. A Protocol for Improved user Perceived QoS in Web Transactional Applications. *Proceedings of the 3rd IEEE International Symposium on Network Computing and Applications*, Boston, MA, Aug 30-Sep 1, pp: 69-76.
- Teixeira, M., M. Santana and R. Santana, 2004. Using Adaptive Priority Scheduling for Service Differentiation in QoS-aware Web Servers. *Proceedings of the 23rd IEEE International Conference on Performance Computing and Communications*, Arizona, April 15-17, pp: 279-285.
- Ye, N., E.S. Gel, X. Li, T. Farley and Y.C. Lai, 2005. Web server QoS models: Applying scheduling rules from production planning. *Comput. Oper. Res.*, 32: 1147-1164.
- Zhou, X., J. Wei and C. Xu, 2004. Modelling and analysis of 2D service differentiation on e-commerce servers. *Proceedings of the 24th International Conference on Distributed Computing Systems*, Tokyo, Mar 23-26, pp:740-747