International Journal of Applied Research on Information Technology and Computing (IJARITAC), 46-58

Indianjournals.com

Estimation of Web Proxy Server Cache Size using G/G/1 Queuing Model

Riktesh Srivastava

Assistant Professor, Information Systems, Skyline University College,
University City of Sharjah, Sharjah, PO 1797, UAE.
Email ID: rsrivastava@skylineuniversity.com

ABSTRACT

Continuous boost in the number of Internet users has taken an exponential escalation over the years. It is becoming thorny to endow with services to all the Internet users because of infrastructural precincts of WWW. Caching web object at proxy servers has proven to be one of the preeminent alternatives for fast services. Caching web proxy server improves the performance of overall web access. Since the introduction of Web proxy servers, most of the evaluations studies are performed either on Web-replacement algorithms or methodologies of maintaining the data in cache. Only few of studies have been done assuring Web proxy server cache size. This paper describes the methodology of estimating the cache size under high busty traffic situation. An investigational evaluation study applying M/M/1 Queuing Theory methodology is first appraised and then experimented. The study uses a trace-driven simulation framework, real traces containing approximately 10 ^{10} number of user's requests per unit time, and then evaluates the Optimal Cache Size ($CS_{optimal}$) using G/G/1 Queue Analysis. .

Keywords: Web Proxy Server, M/M/1 Queuing Model, G/G/1 Queuing Model, Optimal Cache Size.

1. INTRODUCTION

By all indications, WWW continues its remarkable and seemingly unregulated growth [1]. There is rapid growth in the number of users and hosts [2], number of web servers, application servers and network traffic [3] thereby resulting in increase in network loads and users' response time [4]. The performance of Gateway Servers was studied by [5], [6], [7] using comparative analysis of M/M/1 with M/G/1, G/M/1 and G/G/1 queuing theory. Number of research is being performed to improve the rate of data access and decrease the response time. [8], [9], [10], [11] have carried out the study on workloads of network and server performace under such traffic. [12], [14], [15] have used the mathematical analysis for Internet Servers performance analysis. [15], [16] defined the dynamic buffer management techniques by which the buffer space can be reduced for fast transmission of data. The study by [15] has used the case study of Media servers in order to conduct the experiments. In this paper, the author claims that the most prominent solution to problem is "Web caching". In web caching, the data requested by the users are accessed by web server and stored in the cache memory of web proxy server for further reference. Number of Web replacement algorithms, including FIFO (First In First Out), LRU (Least Recently Used) and MRU (Most Recently Used) were implemented to define the mechanism of data storage, data removal and data retrieval. Surprisingly, not much of the research is done to estimate the Optimal Cache Size ($CS_{optimal}$), under heavy traffic. This paper evaluates the $CS_{optimal}$ by first evaluating the results for M/M/1 Queue Model and then implements the results for G/G/1 Queue Model.

The paper devices the mathematical approach to resolve the size of cache, where λ is the arrival rate of users request at Web Proxy Server and μ defines the departure of accesses resources. Based on the values λ,μ, $CS_{optimal}$ is calculated such that the rate of data access from cache increases. The paper evaluates

the number of 15000 references to cache and later simulates the result for higher rate of arrivals, to the maximum extent of 10^{10} number of requests.

The result of the paper divided as follows: Section 2 delineates the Internet architecture with Proxy Server and then defines the mathematical formulation of ë and ì. Section 3 elaborates the analytical study of M/M/1 Queue model for estimation of cache size mathematically. Section 4 gives the computation results and estimation of Cache size using the regression technique, for higher rate of users' requests to estimate the $CS_{optimal}$ for Web Proxy Server by means of G/G/1 Queuing model. Section 5 concludes the paper.

2. ARCHITECTURE OF INTERNET WITH WEB PROXY SERVER

The architecture of Internet with Web Proxy Server is portrayed in Figure 1, given below:

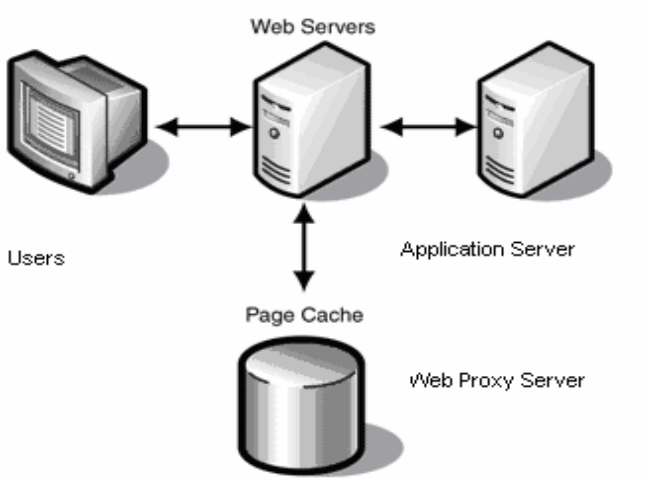

Figure 1: Users request and response from Web Proxy Server

If Figure 1 is further consolidated, it will be condensed to a single request arrival and a single response departure processing system. In this structural design, the resource is first searched in cache memory, if the resource is obtained in Cache, the response is send back to the client, and otherwise, the resource is taken from the application server, stored in cache and response is then send to the requesting client. In this case, when the cache memory is full, web replacement algorithms are applied to place the new resource in cache and obsolete resource is replaced. As Arrival of request and Departure of responses are random in nature, the estimation of memory of cache becomes the issue for Internet Designers. The mission of design should be:

1. No request from user should be overflowed / freezed out.
2. Cache Size should not be abnormally large.
3. There is "n" number of requests from users at a time. The architecture must be such designed that it should provide stable functioning.

These three problems are of prime importance for designing the complete Internet architecture with Web Proxy Server. The Web Proxy Server resembles the working of Queuing theory, where there are number of requests arriving in Web Server and number of resources generated at a single instance of time.

2.1 Mathematical Evaluation of the System

Consider that λ is the summation of users' request at Web Proxy Server per unit time [5],

$$\lambda = \left[req_1 + req_2 + req_3 + ... + req_n / T \right]$$

$$\lambda = 1/T \sum_{i=1}^{n} req_i \quad (1)$$

where,

req_1, req_2, req_3,..... req_n are the number of requests arriving at Web Proxy Server, and, T=Total time in which the request has arrived at Web Proxy Server.

Similarly, if the requested data is found in the cache of web proxy server, the resource is given back to the client. The rate of departure of resources from Web Proxy Server is defined as [5]:

$$\mu = \left[res_1 + res_2 + res_3 + ... + res_n / T \right]$$

$$\mu = 1/T \sum_{i=1}^{n} res_i \quad (2)$$

where,

res_1, res_2, res_3,..... res_n are total number of responses from Web Proxy Server, and, T= Total time in which the resources has departed from Web Proxy Server.

Based on equation (1) and (2), there can be 3 possibilities [6],[7]:

a) If rate of arrival of users request is greater than the departure rate, i.e.,

$$\lambda > \mu \quad (3)$$

This case is called Transient state. If arrival rate of users' request is greater than the departure rate of resources, then, there are chances of more cache miss rather than cache hit, which makes the system unstable. Hence, it is concluded that under no circumstances equation (3) should prevail.

b) When arrival rate of users' request is equal to departure rate of resources from Web Proxy Server, i.e.,

$$\lambda = \mu \quad (4)$$

This is a very typical case, often referred to as Null state. Such phenomenon randomly occurs. This case is typical for academic studies only. Practically, this neither occurs nor is desirable.

c) When arrival rate of users' request is less than departure rate of resources from Web Proxy Server, i.e.,

$$\lambda < \mu \quad (5)$$

This case is called as Ergodic state. If this situation is maintained throughout the system, there were be not system failure and often it results in higher rate of cache hit and less cache miss.

The study is thus made for the estimation of $CS_{optimal}$, which provides all the randomly arriving users' request. In the present study, when a very huge amount of users' request is arriving at the Web Proxy Server for cache reference, and very huge random departure occurs after referring the cache. The problem becomes very complicated to be solved analytically. This problem gets further complicated when the rate of users arrival is huge in number from multiple sources. Had it been arriving from a single source with similar type of requests, the queuing model would have been proximated to*M/M/*1, where first *M* describes the arrival process of users request to be Markovian. Markovian arrival process is nothing but Poisson arrival where inter-arrival distribution is negative exponential. The second*M*, describes the departure process with processing unit 1 (one in number). In case of multiple users' requests arriving at a single instance of time, this assumption does not fit in. Under this condition when arrival of users request or departures of responses, both are considered multi-channel, the appropriate model becomes *G/G/1*. It is worth to start with *M/M/1* queue model have been analytically studied for the estimation of average cache size. However, for multiple requests and responses, the study is carried for G/G/1 model to compute the $CS_{optimal}$. The *M/M/1* model is studied analytically in the next segment.

3. ANALYTICAL STUDY OF M/M/1 QUEUE MODEL FOR ESTIMATION OF CACHE SIZE MATHEMATICALLY

In this section, study is performed to evaluate the Cache Size mathematically using M/M/1 Queue model. Based on the mathematical formulation, $CS_{optimal}$ for high busty traffic using G/G/1 model is evaluated in the next section of the paper.

3.1 Assumptions for Mathematical Formulation of Cache Size

For estimation of "n" number of requests arriving at the Web Proxy Server, certain assumptions have to be made. This can be given as follows:

1) Δt is a very small time, in which only one process can occur, i.e., either arrival of users' requests or departure of resources from Web Proxy Server.
2) The Ergodic state is maintained throughout the study.
3) The state of arrival of users' request is λ and state of departure from Web Proxy Server is μ.

Probability of only one arrival of users' request at Web Proxy Server = $\lambda\Delta t$

Probability of only one departure of resource from Web Proxy Server = $\mu\,\Delta t$

Then, Probability of no arrival of users' request from Web Proxy Server = $1-\ \lambda\Delta t$

and,

Probability of no departure of resource from Web Proxy Server = $1 - \mu\,\Delta t$

Now, consider that there are "n" number of users' request at any time "t". This, will be represented by $P_n(t)$. If the time is increased from "t" to "t+Δt" and at the end of this time, then, this state can be arrived at from the states as given below:

$$P_n(t+\Delta t) = \begin{cases} P_n(t).(1-\lambda\Delta t).(1-\mu\Delta t) \\ \qquad P_{n+1}(t).\mu\Delta t \\ \qquad P_{n-1}(t).\lambda\Delta t \end{cases} \tag{6}$$

or,

$$P_n(t+\Delta t) = P_n(t).(1-\lambda\Delta t)(1-\mu\Delta t) + P_{n-1}(t).\lambda\Delta t + P_{n+1}(t).\mu\Delta t \quad (7)$$

or,

$$\frac{P_0(t+\Delta t) - P_0(t)}{\Delta t} = -\lambda P_n(t) - \mu P_n(t) + \lambda P_{n-1}(t) + \mu P_{n+1}(t) \quad (8)$$

But,

$$\underset{\Delta t \to 0}{Lim}\left\{\frac{P_n(t+\Delta t) - P_n(t)}{\Delta t)}\right\} = \frac{d}{dt}\{P_n(t) = 0\} \text{ for stable condition}$$

Thus, the R.H.S. of equation 8 becomes

$$P_{n-1}(t).\lambda - (\lambda+\mu).P_n(t) + P_{n+1}(t).\mu = 0 \quad (9)$$

To solve, equation 9, we need to consider the initial condition, i.e., there is 0 (nil) presence of users' requests at time (t+Δt). This can be obtained from the states as given under:

$$P_0(t+\Delta t) = P_0(t).(1-\lambda.\Delta t)$$

$$= P_1(t).\mu\Delta t$$

$$= P_0(t)(1-\lambda\Delta t) + P_1(t)\mu\Delta t$$

$$\left(\frac{(P_n(t+\Delta t) - P_n(t)}{\Delta t}\right) = -P_0(t)\lambda + P_1(t).\mu \quad (10)$$

Thus, L.H.S. of equation 10 becomes

$$\underset{\Delta t \to 0}{Lim}\left\{\frac{(P_n(t+\Delta t) - P_n(t)}{\Delta t}\right\}$$

$$\frac{d}{dt}\{P_0(t)\} = 0, \text{ at stable state} \quad (11)$$

Hence, equation 11 becomes

$$P_1(t) = \left(\lambda \big/ \mu\right).P_0(t) \quad (12)$$

From equation 9 and equation 12, following can be easily derived as:

$$\begin{aligned} P_0(t) &= \left(\lambda/\mu\right)^0 .P_0(t) \\ P_1(t) &= \left(\lambda/\mu\right)^1 .P_0(t) \\ P_2(t) &= \left(\lambda/\mu\right)^2 .P_0(t) \\ P_3(t) &= \left(\lambda/\mu\right)^3 .P_0(t) \\ &\vdots \\ &\vdots \\ P_n(t) &= \left(\lambda/\mu\right)^n .P_0(t) \end{aligned} \tag{13}$$

Summation of all the equations in equation 13 is given as under:

$$\sum_{i=0}^{n} P_i(t) = \left\{(\lambda/\mu)^0 + (\lambda/\mu)^1 + (\lambda/\mu)^2 ..... + (\lambda/\mu)^n\right\} P_0(t) \tag{14}$$

Based on limiting condition, when $n \to \infty$, and $\lambda/\mu < 1$, L.H.S. becomes 1 and R.H.S. becomes

$$\left[1/\left(1-\lambda/\mu\right)\right] P_0(t)$$

Thus equation 14 becomes

$$1 = \left[\frac{1}{\left(1-\lambda/\mu\right)}\right] P_0(t) \tag{15}$$

If equation 14 is substituted in equation 13, then

$$P_n(t) = \left(\lambda/\mu\right)^n \left(1-\lambda/\mu\right) \tag{16}$$

Hence, the probability for the presence of "n" data can be computed at any time "t" provided rate of users' request and rate of resources from the Web Proxy Server is known.

3.2 Estimation of Cache Size for M/M/1 Queuing Model

In section 3.1, the probability density function for the existence of "n" data has been derived as:

$$P_n(t) = \left(\lambda/\mu\right)^n \left(1 - \lambda/\mu\right)$$

For variable "n" the average value can be written as:

$$CS = \sum_{n\to\infty}^{N} nP_n(t)$$

Where CS is Cache Size.

$$= \sum_{n\to\infty}^{N} \left(\lambda/\mu\right)^n \left(1 - \lambda/\mu\right)$$

$$= \left(1 - \lambda/\mu\right) \sum_{n\to\infty}^{N} \left(\lambda/\mu\right)^n$$

$$= \left(1 - \lambda/\mu\right) \left\{ \lambda/\mu + 2\left(\lambda/\mu\right)^2 + 3\left(\lambda/\mu\right)^3 + ... \right\}$$

$$= \left(1 - \lambda/\mu\right)\left(\lambda/\mu\right) \left\{ 1 + 2\left(\lambda/\mu\right) + 3\left(\lambda/\mu\right)^2 + ... \right\}$$

$$= \left(1 - \lambda/\mu\right)\left(\lambda/\mu\right) \frac{d}{d\left[\lambda/\mu\right]} \left\{ \left(\lambda/\mu\right) + \left(\lambda/\mu\right)^2 + \left(\lambda/\mu\right)^3 + ... \right\}$$

$$= \left(1 - \lambda/\mu\right)\left(\lambda/\mu\right) \frac{d}{d\left[\lambda/\mu\right]} \left\{ \left(\lambda/\mu\right) \Big/ \left(1 - \lambda/\mu\right) \right\}$$

$$= \left(1 - \lambda/\mu\right)\left(\lambda/\mu\right) \left(\left(\left(1 - \lambda/\mu\right) + \lambda/\mu \right) \Big/ \left(1 - \lambda/\mu\right)^2 \right)$$

$$CS = \frac{\left(\lambda/\mu\right)}{\left(1-\lambda/\mu\right)} \tag{17}$$

The equation 17 measures the Cache size of Web proxy server mathematically for M/M/1 Queue model.

4. COMPUTATIONAL RESULTS AND ESTIMATION OF CACHE SIZE USING G/G/1 QUEUE MODEL IMPLEMENTING REGRESSION TECHNIQUE

The random arrival of users' requests at Web Proxy Server can be assumed either disciplined distribution or it can be assumed undisciplined arrival or departure. Estimation of $CS_{optimal}$ becomes a case of G/G/1 Queue model, as there are multiple requests arriving at Web Proxy Server at a time, and multiple resources are transferred at a time. Initially, the experiment was conducted for λ= 5000, 7500, 10000, 12500, 15000 and for the worst case of cache size is $\mu = \lambda+1$ based becomes μ= 5001, 7501, 10001, 12501, 15001. The average computed Cache Size is given in table 1.

Table 1: Cache Size Computations

S. No.	λ	μ	Cache Size for M/M/1	Cache Size for G/G/1
1	5000	5001	5000	5538
2	7500	7501	7500	8400
3	10000	10001	10000	11190
4	12500	12501	12500	14126
5	15000	15001	15000	17695

The Cache Size computed is for the lower rates of arrivals. It is not possible to have very high arrival rate for computation of models. It is therefore proposed to carryout study for higher values of arrival rates by employing "curve=fitting techniques". Curve-fitting technique, also known as "Regression Techniques" thus, gives the extrapolation for the average queue values at high rate of users' requests.

4.1 Extrapolation of Cache Size Length at High rate of users' request using Regression/Curve Fitting technique

In case of M/M/1 model, the equation is plotted between rate of users' requests and Cache Size, it is then expressed as:

$$CS = \frac{\lambda/\mu}{\left(1-\lambda/\mu\right)} \tag{18}$$

For the worst case of Cache Size [CS], the equation can be derived using:

$$\mu = \lambda + 1 \tag{19}$$

$$CS = \frac{\lambda/\lambda + 1}{1 - \lambda/\lambda + 1}$$

$$CS = \lambda \tag{20}$$

The equation 20 is a linear equation with slope of unity. In case of other models, if we observe the plot, we find that G/G/1 models have curves. The curves are smooth and can be assumed to be a second order polynomial, which is close to first order polynomial formed by model M/M/1.

The equation of the plots can be assumed:

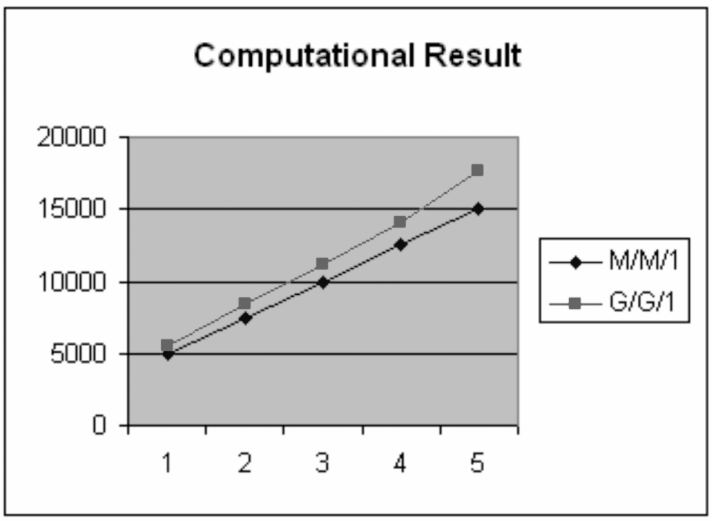

Figure 2: Computation result between M/M/1 and G/G/1 Queue model

$$Cache\ Size\ (CS) = a_1\lambda^2 + a_2\lambda + a_3 \tag{21}$$

This equation can be fitted in all the cases of plots. However, the coefficients in each model will be different. What is required to find out is the values of a_1, a_2 and a_3 for the best fitted curve which has minimal error. For this reason, we have to express error, differentiate it and equate to zero for calculation of coefficients.

4.2 Coefficients of G/G/1 model

It is observed from Figure 2, that the curve of G/G/1 represents a polynomial equation for representing plot of Cache Length w.r.t. rate of arrival of users' requests and Cache Size. To have minimal error in regression, mean square error is made minimal to give good result. Let x_i represents the rate of arrivals for the computation and then the value of Cache Size will be represented by y_i. It is represented as given below:

$$y_i = a_2 x_i^2 + a_1 x_i + a_0 \tag{22}$$

where a_0, a_1 and a_3 are coefficients of polynomial for G/G/1 model.

The equation 22 is valid for larger rates of arrival of users' requests, which includes *CAUSAL EFFECT* . This cannot be employed for lower rates of arrival.

Let "S" represents the error in computation and real values of Cache Size, then, "S", which is square of derivation, is given as:

$$S = \sum \left(y_i - \widehat{y_i^2} \right) = \sum \left(y_i - a_2 x_i^2 - a_1 x_i - a_0 \right)^2 \tag{23}$$

If be differentiate S w.r.t a_0, a_1, a_2 and setting each of these coefficients equal to zero, we get

$$n a_0 + a_1 \sum x_i + a_2 \sum x_i^2 = \sum y_i \tag{24}$$

$$a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 = \sum x_i y_i \tag{25}$$

$$a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 = \sum x_i^2 y_i \tag{26}$$

where "n" represents the degree of polynomials as "n" equations are formed for the summation.

These are three linear equations in three unknowns. These are called *normal equations* for quadratic regression. These may be solved using Gauss Elimination procedure.

Upon calculating, following set of equation were obtained:

$$5a_0 + 50000a_1 + 562500000a_2 = 56949$$
$$50000a_0 + 526500000a_1 + 6875000000000a_2 = 644590000$$
$$526500000a_0 + 6875000000000a_1 + 88828125000000000a_2 = 7918512500000$$

By Gaussian elimination and using back substitution, we obtain:

$a_o = 2.009 \times 10^{-7}$

$a_1 = 0.845028571$

$a_2 = 1.7828571 \times 10^{-5}$

4.3 Optimal Cache Size ($CS_{optimal}$) using G/G/1 Queuing Model

It is to note that computation of Cache size was for average value. The actual Cache size will deviate from average value at every instance. Sometimes, Cache size is lower than average and at other times it is higher average value. The estimation of cache size should be such that cache hit should be higher that than the average value. Thus, it indicates that positive deviations are to be incorporated in the estimation of cache size. Standard deviation, which is positive square root of variance, is required to be added in the

average value of cache size. Standard deviation in many statistical distributions is equal to average value. Table 2 measures the cache size at Web Proxy Server.

Table 2: Average Cach Size

Expected users requests	Average Cach Size	
	M/M/1	G/G/1
10000000	10000000	1791307386
100000000	100000000	178370212857
1000000000	1000000000	17829416028571
1000000000	1000000000	1782865550285710

Using equations, the expected Cache size are computed

$$y_i = a_2 x_i^2 + a_1 x_i + a_0 \qquad for\ \mathrm{G}/G/1\ \text{model}$$

The optimal value for memory size is given as:

$$CS_{optimal} = CS_{av} + Standard\ Deviation\ of\ CS \tag{27}$$

The Standard Deviation is given as:

$$Standard\ Deviation = CS_{av} \tag{28}$$

Hence, the desired cache size to meet the eventualities, the optimal cache size is:

$$CS_{optimal} = 2.CS_{av} \tag{29}$$

The calculated cache size(s) are depicted in table 3.

Table 3: Optimal Cache Size

Expected users requests	Optimal Cach Size ($CS_{optimal}$)	
	M/M/1	G/G/1
10000000	20000000	3582614771
100000000	200000000	356740425714
1000000000	2000000000	35658832057142
10000000000	20000000000	3565731100571420

5. CONCLUSION

The study confirms that the proposed model be such that it should follow the condition $\lambda < \mu$ [ERGODIC CONDITION]. This will guarantee more number of cache hit resulting in stability of the system implementation. The cache size estimated by queuing theory will ensure that there is minimal cache miss by evaluating the optimal size of cache size.

References

1. Pitkow & Recker, *A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns*, Proceedings of the Second International WWW Conference 7 GVU Technical Report: VU-GIT-94-39

2. Grey, M., *Growth of the World-Wide Web*, Available via URL:http://www.mit.edu:8001/afs/sipb/user/mkgray/ht/comprehensive.html, 1994.

3. Merit NIC, *NSFNET Statistics,* 1994. Available via URL: gopher://nic.merit.edu:7043/11/nsfnet/statistics/1994.

4. Viles, C. and French, J, *Availability and Latency of World-Wide Web Information Servers, 1994, University of Virginia Department of Computer Science Technical Report CS-94-36.*

5. Riktesh Srivastava, LK Singh, *Design and Implementation of G/G/1 Queuing Model Algorithm for its Applicability in Internet Gateway Server*, The International Arab Journal of Information Technology, Vol. 5, No. 4, 2008.

6. Riktesh Srivastava, LK Singh, *Memory Estimation of Internet Server using Queuing Theory: Comparative Study between M/G/1, G/M/1 and G/G/1 Queuing Model* , International Journal of Computer and Information Science and Engineering (IJCISE), Volume 1 Number 2, 2007.

7. Riktesh Srivastava, LK Singh, *Estimation of Buffer Size of Internet Gateway Server via G/M/1 Queuing Model*, International Journal of Applied Science, Engineering and Technology, Volume 19, 2007.

8. Barford, P. and Crovella, M. *Generating representative web workloads for network and server performance evaluation, Measurement and Modeling of Computer Systems*, 1998.

9. Menasce, D. and Almeida, V. *Capacity Planning for Web Services: Metrics, Models, and Methods* . Prentice Hall PTR, 2001.

10. Lazowska, E. D. Zahorjan, J. Graham, G. S. and Sevcik, K. C. Eds., *Quantitative system performance: computer system analysis using queueing network models*. Prentice-Hall, Inc., 1984.

11. Abdelzaher, T. F. Lu, C. *Modeling, and Performance Control of Internet Servers, Invited Paper*, *39th IEEEConference on Decision and Control*, Sydney, Australia,December 2000.

12. Jim W. Roberts, *Traffic Theory and the Internet*, IEEE Communications, January 2001.

13. Xiangping Chen, Prasant Mohapatra, *Performance Evaluation of Service Differentiating Internet Servers,* Vol. 51, No. 11, 2002.

14. Steven H. Low, R. Srikant, *A Mathematical Framework for Designing a Low-Loss, Low-Delay Internet,* IEEE transaction on Communications, 2002.

15. Lei Ying, G. E. Dullerud and R. Srikant, *Global Stability of Internet Congestion Controllers with Heterogeneous Delays,* IEEE Transactions on Communications, 2003.

16. Yeon Seung Ryu, Kern Koh, *A Dynamic Buffer Management Technique for Minimizing the Necessary Buffer Space n a Continuous Media Server,* IEEE International Conference on Multimedia Computing and System (ICMCS), 1996.

17. Darrell Anderson, Ken Yocum, Jeff Chase, *A Case for Buffer Servers,* IEEE Seventh Workshop on Hot Topics in Operating Systems, 1999.