Asian Journal of Information Technology 6 (4): 455-461, 2007

Strategic Approach to the Selection of Appropriate Replication Technique to Model a Fault-Tolerant Internet Connectivity

[1]O.O. Adeosun, [2]E.R. Adagunodo, [3]I.A. Adetunde, [4]T.H. Adeosun and [1]E.O. Omidiora
[1]Department of Computer Science and Engineering,
Ladoke Akintola University of Technology, Ogbomoso, Nigeria
[2]Department of Computer Science and Engineering,
Obafemi Awolowo University, Ile-Ife, Nigeria
[3]Department of Applied Mathematics and Computer Science,
Faculty of Applied Sciences, University of Development Studies,
P.O.Box 24 Navrongo (Upper East Region), Ghana
[4]Department of Banking and Finance, Osun State College of Technology,
Esa-Oke, Nigeria

Abstract: The network is a highly dynamic environment where even subtle changes can now have a major, unforeseen impact on application performance and availability causing Internet failures. A system with faults/failures may continue to provide its services (i.e., not fail). Such a system is said to be fault tolerant. This is our focus in this research, to provide Internet services to clients without any interruption even at the presence of faults, through the deployment and redeployment of replicated proxy servers using Markov Reward Model. To achieve this, this study considered different replication techniques that can be used for Internet connectivity and finally suggested the best technique for use.

Key words: Strategic approach, selection, appropriate replication, internet conectivity, technique

INTRODUCTION

Networking is getting tougher. Networks must deliver a growing range of services, from ERP, CRM and e-mail to VoIP and web services applications, each of which has its own idiosyncrasies and requirements. The network itself is constantly changing. The proxy server been the most active and busiest server in Internet connectivity, is a valuable resources that needs to be managed, conserved and shared as effectively as possible. Access to the internet from any particular node is through sever. Our concern is how to make different servers available for different services on Internet system. This will allow continuity of services, if a server is down, others would still be working and only the services provided by the faulty server would be stopped and not the whole activities on the net. If that is done, there should be a dedicated server to coordinate other servers; this is called proxy server. It is acting in this type of arrangement as a gateway for the LAN. All traffic to and fro will be routed through this device. It is of a necessity to monitor and provide adequate protections for this device so that there would not be any interruption of Internet services. The best way to do this is to replicate this device and provide algorithms for the coordination of the replicas to achieve Internet dependability cum availability expected. This type of Internet connectivity design will provide spares in case there is a fault or failure, so that the system can switch to a spare of it thereby keep the Internet services going without any interruption.

Rationale for the study: The growing reliance of individuals, industries and governments on computer and Internet systems provides the motive for malicious attacks and the increased connectivity to the Internet exposes, these systems to more attacks (Murphy and Levidow, 2000). The number of software and hardware errors is increasing due to the growth in size and complexity of systems. Complexity fosters unreliability, not only by making the system less comprehensible, but also as a result of the so-called product law, which states that the probability of a system remaining fault-free over a given

Corresponding Author: I.A. Adetunde, Department of Applied Mathematics and Computer Science, Faculty of Applied Sciences University for Development Studies, P.O. Box 24 Navrongo (Upper East Region), Ghana

period is the product of the individual probabilities of its constituent elements remaining fault-free. Fault-tolerant is the platform on which the axis of this research work is rotating to provide a lasting solution to virtually all the above mentioned problems that are affecting Internet system availability.

MATERIALS AND METHODS

The method used in this research is functional approach or functional-orientated design. The execution of the work is divided into phases. These include an extensive survey of existing model on Internet connectivity with a view to exploring previous works and to attempt designing a fault-tolerant model for the system. An extensive study of Basic Fault-tolerant System (BFS) architecture in order to understand the basic rudiments of the fault-tolerant system design was also done to get our focus right. Finally, a conceptual design of a fault-tolerant Internet connectivity using Markov chain is done. Generation of a quality test plan for the system and performing the test specification by comparing reliability of various replicated techniques approaches.

Different replications approaches

Series: If a system is composed of elements in such a way that the failure of any one element causes a failure of the system, then these elements are considered to be functionally in series (Harper *et al*., 1988). The system in Fig. 1 represents conceptual view of replicated proxy servers in series. For the Internet system to survive any fault or failure, each proxy server in the figure must survive. Let us assume the use of 3 proxy servers in this Internet system connectivity.

This is known as the product rule:

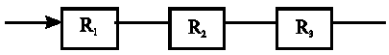

Fig. 1: Series reliability

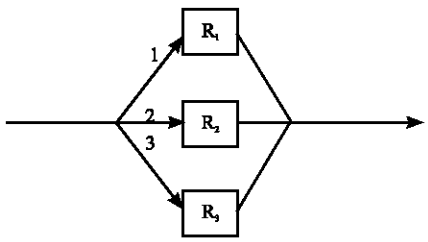

Fig. 2: Parallel reliability

$$R_{System} = \prod_{i=1}^{3} R_i \quad (1)$$

where R_i in (1) is the reliability of the individual proxy server used.

Parallel: Parallel reliability is an illustration of protective redundancy. The system is composed of functionally parallel elements in such a way that if one of the elements fails the parallel unit will continue to do the system function (Harper *et al*., 1988).

The system reliability of Fig. 2 under the assumption of independence of failure of the elements is expressed by:

$$R_{system} = 1-(1-R)^3 \quad (2)$$

Which is the probability that not all the *3* proxy servers have failed. The term (1-R) in (2), known as the unreliability of a proxy server, is the probability that a proxy server will fail. The term $(1-R)^3$ in (2) by the product rule is the probability that all the three proxy servers will fail and one minus that is the probability that not all proxy servers will have failed. Hardware redundancy may be of the fault-masking, self-repair types or a hybrid of these two (Harper *et al*., 1988).

Triple modular: Triple Modular Redundancy (TMR) is also known as the multiple-line voting system (Frison and Wensley, 1982). Svore *et al*. (2006) observe in special applications of modern digital computers, that the canonical method for fault-tolerant computation is Triple Modular Redundancy (TMR). This involves feeding gate inputs copied three times into three gates that fail with probability O(p). The output lines of these faulty gates fan out into three majority voting gates. The majority gates essentially amplify the correct value of the computation so that the fault-tolerant gate fails only if two or more failures occur. Mathematically, the fault-tolerant gate fails with probability $O(p^2)$.

Figure 3 is a TMR proxy server Internet connectivity approach commonly used as fault masking in which the circuitry is triplicated and voted. The voting circuitry can also be triplicated so that individual voter failures can also be corrected by the voting process. A TMR connectivity system fails whenever two modules in a redundant triplet create errors so that the vote is no longer valid. This scheme is generalized to *N*-Modular Redundancy (NMR) where *N* is any odd number of units (McCarthy, 2003).

The TMR reliability is expressed as:

$$R_{System} = [R^3 + 3R^2 (1-R) R_v] \quad (3)$$

This is the product of the reliability R_i (the voter reliability) and the reliability of the idealized TMR system ($R^3 + 3R^2 (1 - R)$) in (3). It is the sum of the probabilities of the two events that, firstly, all three servers survive (R^3) and secondly, at least any two of the proxy servers survive and at most one proxy server fails, $3R^2 (1 - R)$.

Standby replacement: In standby replacement redundancy (Fig. 4) unlike TMR (Fig. 3) only one proxy server is operational at a time. When the active proxy server fails, this event is detected by additional circuitry and a spare unit from a reserve of spares switched-in to replace the failed proxy server, thereby restoring the system to its operational state. The reliability of the system in Fig. 4 is expressed as:

$$R_{system} = 1 - (1-R)^{2+1} = (1-R)^3 \quad (4)$$

which is the probability that not all proxy servers have failed.

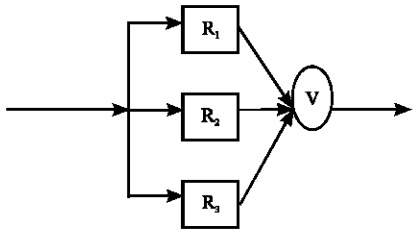

Fig. 3: A triplicated modular redundancy

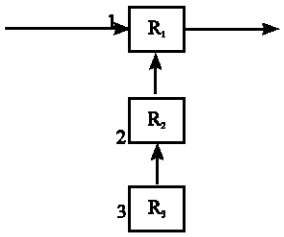

Fig. 4: Standby replacement redundancy

System reliability: For any structure, reliability is a function of the number and organization of the interconnections. The system reliability is under the assumption of independence of failure of the proxy servers. In each of the above approaches, we use 3 replicas (Table 1).

This is the probability that not all the n have failed (Beauquier and Kekkonen-Moneta, 1997). Here, in this study our n is 3, therefore the system reliability under the assumption of independence of failure of the proxy servers is expressed by:

$$R_{System} = 1 - (1-R)^3 \quad (5)$$

This is the probability that not all the three proxy servers have failed. The term (1-R), known as the unreliability of a proxy server, is the probability that a proxy server will fail. The term $(1-R)^3$ by the product rule is the probability that all three proxy servers will fail and one minus that (i.e., $1-(1-R)^3$) is the probability that not all proxy servers will have failed. This is shown in Fig. 5-13.

Modeling the connectivity as a stochastic process: Markovian process is used to model the proposed Internet connectivity as shown in Fig. 14 in other to give room for deployment and redeployment of failed proxy server (s). A Markov chain is a discrete time stochastic process in which each random variable, X_β, depends only upon the previous one, $X_{\beta-1}$ and affects only the subsequent one, $X_{\beta+1}$. The term "chain" suggests the linking of the random proxy server to their immediately adjacent neighbours in sequence as shown in Fig. 15.

Suppose that X_0 represents the proxy server assigned presently and we are interested in X_1, the next proxy server to be assigned. What we want is a probability distribution over the three possible values for X_1. But these probabilities depend on the proxy server on use at a particular point in time. Suppose in Fig. 16, S_1 is the assigned proxy server (i.e., $X_0 = 1$). Then, for its next assignment if there is failure, S_2 will be assigned (i.e., $X_1 = 2$) or S_3 in case S_2 is equally faulty (i.e., $X_1 = 3$). This is shown in Fig. 16.

Table 1: Assumed System Reliabilities for different replication approaches

Single proxy server reliability (%) (Assumed value)	Reliability (%) series $R_{System} = \prod_{i=1}^{3} R_i$	Reliability (%) parallel R_{system} $1-(1-R)^3$	Reliability (%) triple modular $R_{System} = [R^3\ 3R^2$ $(1-R)]\ R_V$	Reliability (%) standby replacement $R_{system} = 1-(1-R)^3$
50.00	12.50	87.50	25.00	87.50
60.00	21.60	93.60	32.40	93.60
70.00	34.30	97.30	39.20	97.30
80.00	51.20	99.20	44.80	99.20
90.00	72.90	99.90	48.60	99.90

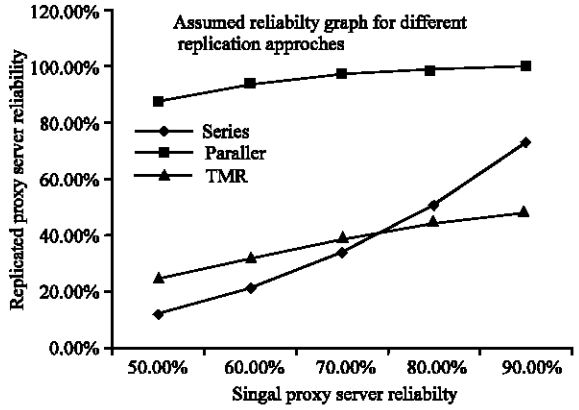

Fig. 5: Graph for different replication techniques

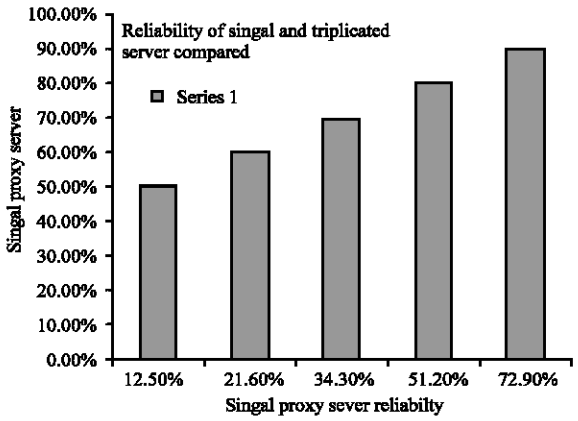

Fig. 6: Graph of single and triplicated proxy servers compared

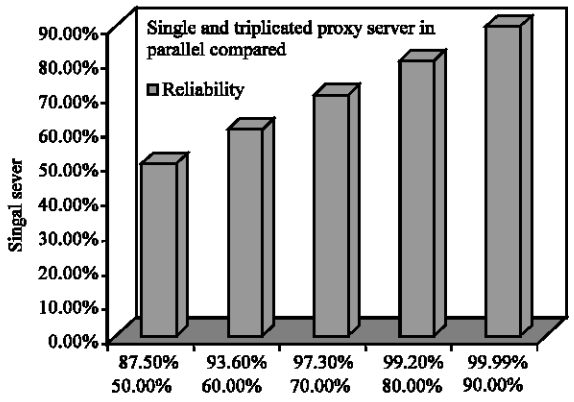

Fig. 7: Single and Triplicated proxy servers in parallel compared

Where there are faults that individual proxy servers can tolerate, we have this scenario: Suppose S_1 is the assigned proxy server (i.e., $X_0 = 1$). Then, for its next assignment if there is fault that can be tolerated, S_1 will be assigned (i.e., $X_1 = 1$), ditto for others as shown in Fig. 16.

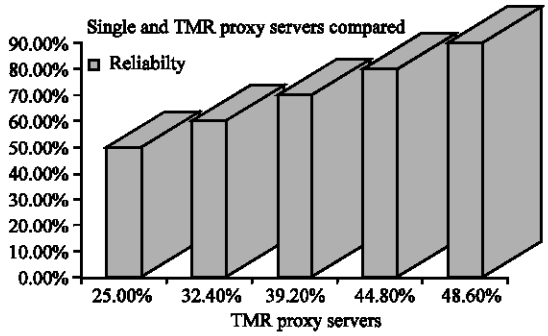

Fig. 8: Single and TMR compared

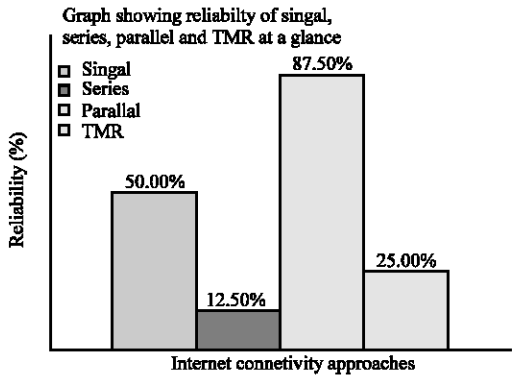

Fig. 9: Reliabilities of single, series, parallel and TMR at a glance

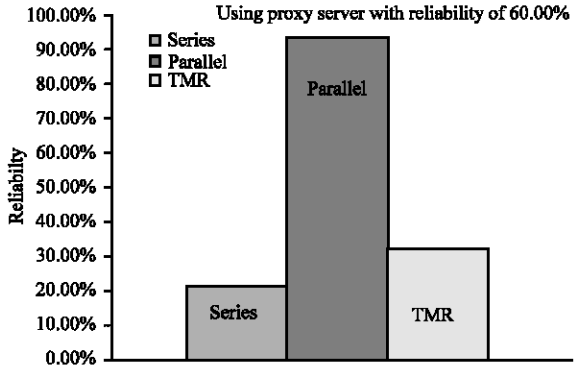

Fig. 10: Implementing replicated proxy servers using proxy server with 60.00% reliability

$$P = \begin{matrix} p_{11} & p_{12} & p_{13} \\ p_{21} & p_{22} & p_{23} \\ p_{31} & p_{32} & p_{33} \end{matrix}$$

The different probabilities can be recorded in matrix form shown below:

The first row of this matrix represents the probabilities that the proxy server assignment will be in this order S_1, S_2, S_3, respectively, if the current assigned is S_1. The second row gives the same

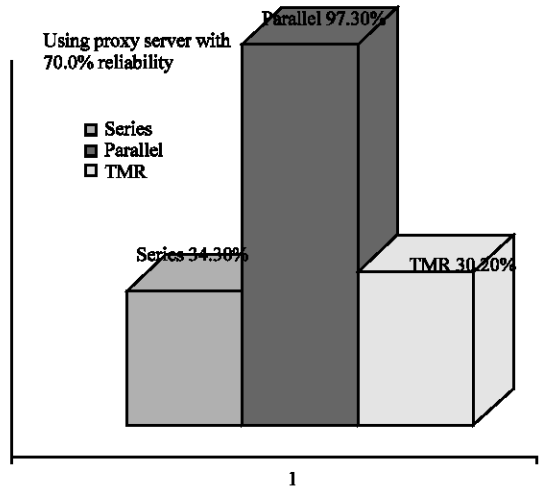

Fig. 11: Implementing replicated proxy servers using proxy server with 70.00% reliability

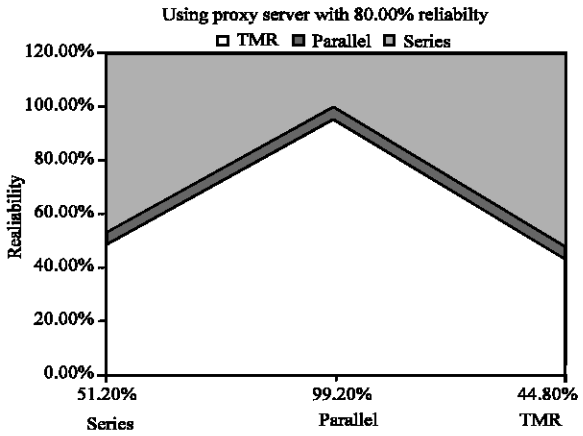

Fig. 12: Implementing replicated proxy servers using proxy server with 80.00% reliability

probabilities if S_2 is first assigned; the third row, if S_3 is first assigned. Each row is, individually, a probability distribution for X_1; the appropriate one to use depends on which value X_0 has. Since each row is a probability distribution (as stated above), it implies that the elements of each row must sum to 1.

The element p_{ij} means the probability that $X_1 = j$ if we know that $X_0 = 1$. This is the conditional probability, $p_{ij} = P\{X_1 = j \mid X_0 = 1\}$. This is a probability of "going" from state 1 to j. The p_{ij} 's are called (one-step) transition probabilities and the matrix P is called the transition matrix.

Algorithm in Fig. 17 can be used to retrieve the requests from all the available proxy servers S_1, S_2, S_3. It can be easily implemented by using a feature of version 1.1 of the Hyper Text Transfer Protocol (HTTP/1.1)(World

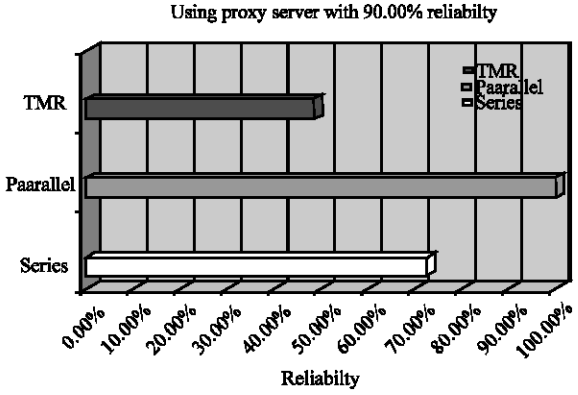

Fig. 13: Implementing replicated proxy servers using proxy server with 90.00% reliability

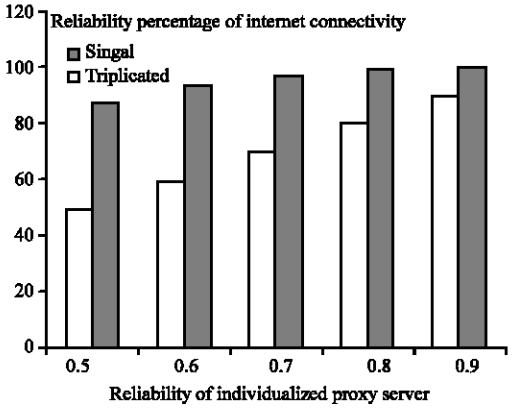

Fig. 14: Reliability percentage chart compared

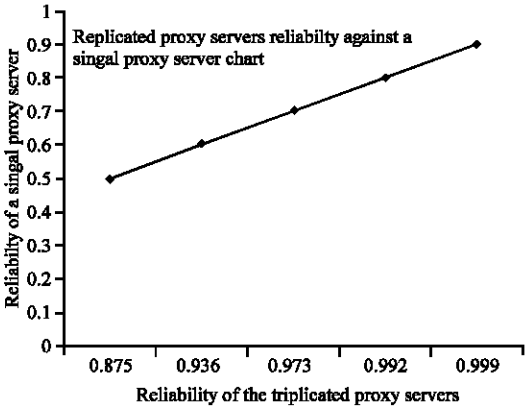

Fig. 15: Non-replicated proxy server and replicated proxy servers compared

Wide Web Consortium), namely the support for Byte-Ranges transfers. HTTP/1.1 compliant *PS*s accept a Range header which can be used to specify which byte

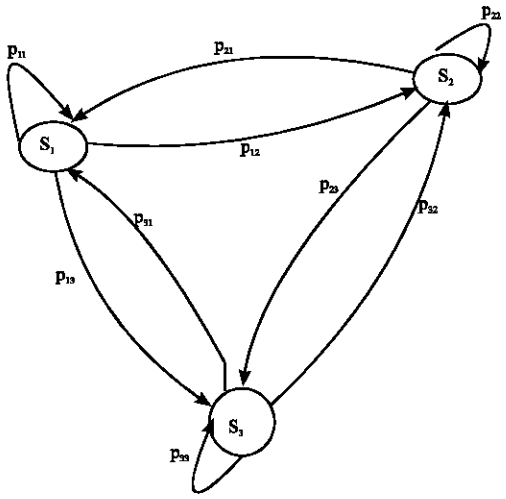

Fig. 16: Transition between the replicated proxy servers

Algorithm for fault-tolerant retrieval of a web page using replicated proxy servers approach

Require: S_1, S_2,S_3 proxy server replicas
Require: 1= K = 3 minimum replies required to reconstruct the page.
Require: *W* Web document to retrieve

Compute R_0, R_1, R_2 {Page request}
C := 0 {Number of requests completed}
A := {S_1, S_2, S_3} {Proxy servers still alive}
for *i* = 0 to 2 do
Asynchronously start request R_β on S_β
end for
while C < K do
Wait for any S_j _, *A* to send part of its request R_j
if Request R_j of S_j completed then b
A := A-S_j {Remove S_j from active servers}
C := C + 1 {Increment completed requests}
W := W ∪ R_j {Update the reconstructed page}
endif
end while

Fig. 17: Algorithm for fault-tolerant retrieval of as Web page

ranges of the specified document are requested. The proxy server encodes the requested fragment into the reply body and returns a status code to the client.

As in (Ghini *et al*., 2001) the program first interrogates the DNS (Domain Name Server) to get the list of IP addresses associated with the domain name of the requested document. After that, it contacts all the proxy server replicas using a HTTP HEAD request. This request is used to check the size of the Web page, whether the requested Web page has been relocated (in such case the new location is contacted) and whether the proxy server replica is down or unresponsive (in such case the replica is not used at all).

Given the number of working proxy server replicas to be 3, the size |W|of the Web page and the user-supplied parameter K, the requests for the proxy servers are first computed. At this point, the client opens 3 asynchronous HTTP conections, one with each proxy server and starts receiving the data. As soon as K replies have been compeleted, all the connections are closed and the page W is reconstructed.

Model of proxy server connections: We use a simple model of the behaviour of client-server connections based on a Markov Reward Model in the analysis of algorithm presented in Fig.17. The analysis is aimed at calculating the Cumulative Distribution Function for the random variable $T_{3,K}$ (W), which denotes the time needed to complete the transfer of a Web document W in time at most t, given 3 proxy server replicas and K sufficient replies to reconstruct W.

Data transfer between a proxy server and the client happen in bursts, that is, each transfer is made of active periods, during which bytes are transferred at a given (fixed) rate Bw, alternating with idle periods, where no data transfer takes place. This model is derived from the packet train model. This is with the assumption that the duration of idle and active periods are independent and exponentially distributed random variables.

The connection between proxy servers S_j, = 1, 2, 3 and the client can be modeled using a two-state continuous time birth-death Markov Chain (MC) as shown in Fig. 16. The underlying continuous-time Markov model $X_j = \{X_j(t), t \geq 0\}$ is defined over the discrete state space{0, 1}. When in state 1 (i.e., X_j (t) = 1), then the connection is active. When in state 0 (i.e., X_j (t) = 0), the connection is idle. The model of a single connection is characterized by three nonnegative parameters: The transition rate μ_j from state 0 to state 1, the transition rate Bw_j from state 1 to state 0 and the transfer rate Bw_j when in state 1. We assume that the 3 client-proxy server network connections are independent.

We compute Pr {$T_{3,K}$ (W) = t}, the probability of downloading the document W in time at most t from 3 proxy servers using algorithm in Fig. 17 with parameter K, as follows:

$$\Pr\{T_{3,K}(W) \le t\} = \sum_{i=k}^{3} \Pr\{\text{i proxy servers replied by time t}\}$$

$$\sum_{i=k}^{3} \sum_{\substack{\Pi \subseteq \{1,2,3\} \\ |\Pi|=i}} \{\text{only PSs } (S_j)\ j \in \Pi \text{ completed by time t}\} \quad (6)$$

We define $O_j(t)$ the total time spent by the network connection from proxy server S_j to the client in state active (i.e., $X_j(t) = 1$) during the interval (0, t).

$$O_j(t) = = X_j(s)ds \quad (7)$$

We let D_j to be the time needed to transfer the whole request R_j from proxy server S_j to the client.

$$D_j = \frac{|W|}{Bw_j} \times \frac{3-K+1}{3} \quad (8)$$

We can now substitute the following in Eq. 8:
Pr {Only PSs $(S_j) j\_$, Ï completed by time t } =

$$\coprod_{j=0}^{3-1} (\Pr\{O_j(t) \ge D_j\} I_{j \in \Pi} + \Pr\{O_j(t) < D_j\} I_{j \in \Pi}) \quad (9)$$

In order to evaluate Pr $\{T_{3,K}(W) = t\}$ it is necessary to compute Pr $\{O_j(t) \ge s\}$ = 1 - Pr $\{O_j(t) < s\}$. This is the operational time distribution of the Markov process X_j over the interval (0, t).

RESULTS

Internet system reliability will be improved upon as shown in Fig. 15 even if the individual proxy servers are having their reliability probability as low as 0.5. Connecting together in parallel, the replicated proxy servers would enhance the system reliability by bring it closer than never before to the targeted system reliability of 99.999%. This would make the Internet system more availabile and dependable even at the presence of fault(s). Also, Internet throughput will be enhanced. The Internet system becomes more stable and its scalability improved.

The proposed work integrates fault-tolerant technology into computer network architectural design using Markovian and Stationarity processes approach. Deployment and Redeployment of replicated proxy server is done on fault-tolerant platform to enhance Internet services stability (Fig. 16).

The whole system accessibility becomes transparent to the clients. The individuals are now served better even at the presence of faults. There is continuity of service even at the notice of fault(s); while the engineers are battling with faulty spares, the viable ones continue to render the required services without clients noticing any faults on the Internet system.

CONCLUSION

It shows clearly that the replicated proxy servers Internet connectivity approach will provide higher and better Internet performance compared to non-replicated proxy server Internet connectivity. Using proxy server replication approach as shown in this work will definitely enhance Internet system available to users even at the presence of failures/faults.

From our analysis above, replication of proxy servers in parallel has the ability to give reliability that is not far from the expected reliability of an Internet system (i.e., 99.999%). If well implemented, Internet system becomes more reliable and self-stabilized by local checking and correction. This is the characteristic of a fault-tolerant system that enhances Internet service performance. The redundancy in this researchwould make the multiple proxy servers more resilient to network failures thereby enhancing Internet stability and promote its scalability.

REFERENCES

Beauquier, J. and S. Kekkonen-Moneta, 1997. Fault-tolerance and self-stabilization: Impossibility results and solutions using self-stabilizing failure detectors. Int. J. Sys. Sci., 28: 1177-1187.

Frison, S.G. and J.H. Wensley, 1982. Interactive Consistency and Its Impact on TMR Systems in Dig. Int. Symp. Fault Tolerant Computing, FTCS, 12: 228-233.

Ghini, V., F. Panzieri and M. Roccetti, 2001. Client-centered load distribution: A mechanism for constructing responsive web services. In Proceeding HICSS'34, Hawaii, IEEE Computer Press.

Harper, R.E., J.H. Lala and J.J. Deyst, 1988. Fault Tolerant Parallel Processors Overview. FTCS, 18: 252-257.

McCarthy, M., 2003. Fault-Tolerant. Tech Target, pp: 13-21.

Murphy, B. and B. Levidow, 2000. Windows 2000 dependability. In: Proceedings of IEEE. International Conference on Dependable Systems and Networks, NY.

Svore, K.M., A.V. Aho, A.W. Cross, I. Chuang and I.L. Markov, 2006. A Layered Software Architecture for Quantum Computing Design Tools, Computer, IEEE. Computer Soc., pp: 74-63.