Proceedings of the 18th IASTED International Conference
MODELLING AND SIMULATION
May 30–June 1, 2007, Montreal, Quebec, Canada
ISBN Hardcopy: 978-0-88986-663-8 / CD: 978-0-88986-664-5

PERFORMANCE ANALYSIS AND SIMULATION OF AN EQUIVALENT MODEL OF SERIAL QUEUES FOR COMPUTER NETWORKS

Chung-Ping Chen* and Ying-Wen Bai**
Graduate Institute of Applied Science and Engineering, Fu Jen Catholic University*
Department of Electronic Engineering, National Taipei University of Technology*
Department of Electronic Engineering, Fu Jen Catholic University**
Taiwan, Republic of China
cpchen@ntut.edu.tw, bai@ee.fju.edu.tw

ABSTRACT
In this paper we propose a serial queues model to estimate the performance of computer networks with a serial stage. First, based on the relationship between the arrival rate and the service rate, which may be equal or not, we have altogether 13 kinds of working regions. For each region or case we deduce the system average response time from the equivalent serial queues model according to the arrival rate and the service rate. The average response time has something to do with arrival rates, service rates and buffer size and the relation between these and with the working regions. For the verification of the serial queues module of the equivalent models, we use the Queueing Network Analysis Tool (QNAT) to simulate a couple of serial queues and to compare model and simulation. The error between model and simulation is 4.18% or 7.06% depending on whether the arrival rate is equal to the service rate or not. Hence the model of the serial queues can predict the average system response time for computer networks with serial multi-stage switches.

KEY WORDS
System Response Time, Equivalent Model, Serial Queues and Equivalent Queues

1. Introduction

As people use the network service to search for information from Web pages, hence the rapid increase in the number of Web servers. Requesting a particular Web page quantity causes an increment in service requests, and this situation will bring about an increment in the system response time. This is because the loading received by the Web page server increases and causes the servicing time for each HTTP connection to increase. The growth of the servicing time influences the system response time directly. If the number of stages of a Web page servicing in the server cluster is decreased, then the system response time is also decreased [1, 2]. These serial stages of computer networks cause the system response time mainly as result of network latency, transmission time, DNS lookup and queueing time as well as by service time. But the network transmission time is usually very hard to predict as the transmission time of a network includes a router, switches and network circuits and thus every kind of process and facility [3].

A typical example is taken from one department on the university campus which provides only one switch or router and already cannot serve the needs of all its teachers and students. Hence the sub-network should increase the number of switches to provide greater service capacity. This, however, generates a serial delay problem from the multiple stages. If we can simplify two stages of a serial queues, then we can obtain an equivalent queue that will help us with a high efficiency of a performance analysis of a queueing network. When the users need to send Web requests to the Web cluster to gather, these requests spread through the Intranet, no matter whether the user uses a PC, a notebook or a PDA for the Web request [4, 5]. The stage 2 switch connects to the stage 1 switch and gives the user access by means of parallel tree structures. Each sub-tree can be a serial stage as shown in Fig. 1 [6].

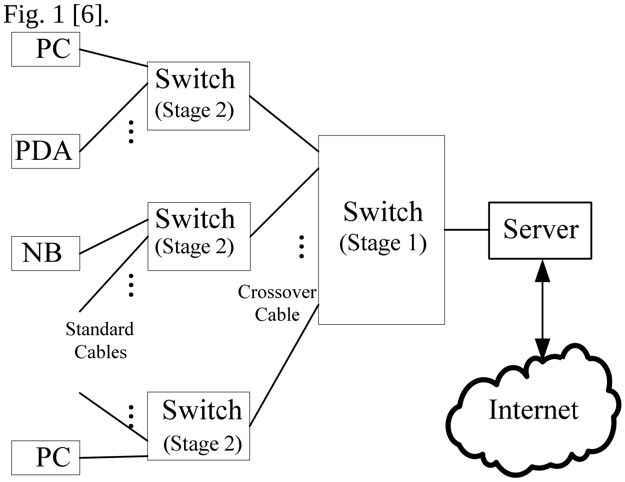


Fig. 1 Network connection from multi-stage switches

In this paper we use QNAT to verify the proposed serial model. We use the equivalent equation of the serial queue for a sub-tree, no matter whether the service rate is equal, greater than or less than the arrival rate, because the serial

queue can have various working regions. For each working region we propose the equivalent model for analyzing the system response time of two serial queues.
The organization of this paper is as follows: In Section 2 the sub-tree of the network system is represented by a couple of serial stages by the M/M/1 queue model. In Section 3 we propose the equations for different cases of serial queues in the different working regions. In Section 4 we use QNAT to simulate serial queues and then compare the performance to verify the equivalent equations. In Section 5 we draw our conclusions according to the results of analysis and simulation.

2. The Serial Queues Representing the Sub-tree of Computer Networks

In a real Internet system, if all personal computers (PCs) want to make use of a network connection, several devices are needed, for example a network card, a modem, a repeater, a router and a switch. Then the PC can deliver the message to the Web server through several stages of a serial connection to the Internet. We assume that messages from the Internet enter into multi-stage switches to reach the Web server. In the sub-tree of the network the number of switches and serial connections is "*n*", and the service rate can be represented as μ_1 , μ_2 ,…, μ_n respectively in each switch. We use an equivalent serial queue model to represent a typical sub-tree as shown in Figure 2.

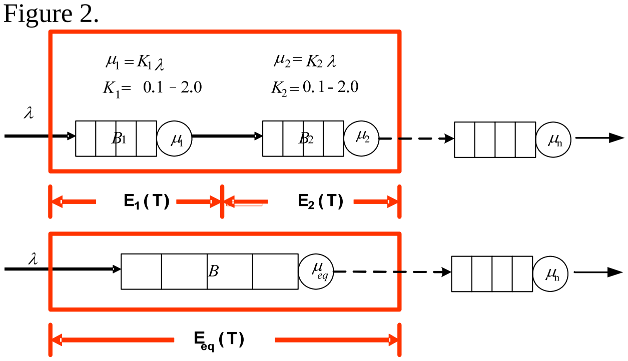


Fig. 2 The equivalent model of serial queues for a sub-tree of networks

Table 1 The definitions and units of the system parameters of the queueing model

Symbol	*Description*	*Unit*
λ	Web request rate	Requests/second
μ_n	The *n*'th server service rate	Requests/second
μ_{eq}	Service rate of an equivalent queue	Requests/second
B_n	The *n*'th server buffer	Number
$E_n(t)$	The *n*'th average response time	Second
$E_{eq}(t)$	Average response time of an equivalent queue	Second
K_n	The ratio of the *n*'th service rate with respect to the arrival rate	μ_n / λ

Table 1 shows the definitions of the system parameters of the queueing model.
According to Fig. 2 we try to simplify two serial queues into a single equivalent queue with an equivalent service rate to provide us with an easy way to analyze a complicated network. We also check the accuracy of the equivalent equations [7, 8]. While analyzing the sub-tree represented by the equivalent serial queue, we first assume the following:
(1) There is only one job class in the network.
(2) The overall number of jobs in the network is unlimited.
(3) Each node in the network has Poisson arrivals from outside.
(4) A job can leave the network from any node.
(5) All service times are exponentially distributed.
(6) The service discipline at all nodes is first come first served (FCFS).
(7) If the request is blocked by the network, the node transfers the request to another node.
The arrival rate of the first queue is λ and passes through the first queue whose size is B_1 and which has the service rate μ_1. The arrival rate then passes through the second queue whose size is B_2 and which has the service rate μ_2. The request leaves the sub-tree after passing through the second queue. There is no rejection and no feedback. We can, therefore, try to simplify these two serial queues into a single queue under the above-mentioned conditions. The equivalent serial queue model is shown in Fig. 2.

3. The Derivation of the Equivalent Equations for the Serial Queues

When the service rate is greater than the arrival rate, the system is at low utilization, and the response time of the system is $E_n(t) = \dfrac{1}{\mu_n - \lambda}$ [1]. When the service rate is equal to the arrival rate, the response time is $E_n(t) = \dfrac{B_n}{2\mu_n}$. When the service rate is less than the arrival rate, the response time is $E_n(t) = \dfrac{B_n}{\mu_n}$. We assume $\mu_1 = K_1\lambda$, $\mu_2 = K_2\lambda$ when K_1 >1, K_2 >1, K_1 =1, K_2 =1, K_1 <1, K_2 <1, K_1 = K_2 and B_1 = B_2 , so the total combination of the working regions of two serial queues can consist of 13 cases as shown in Fig. 3.
There are several working regions, here called cases, of the equivalent serial queue. In case 1 the service rate of the first queue is less than the arrival rate, which is a high blocking rate. In other words, the service rate of the second queue is greater than the arrival rate, which is a low blocking rate. If the bottleneck is at the first queue, then the system response time is, to a great part, spent in the first queue. In this case the response time of the first queue is related to its buffer size. If the buffer size is

larger, the response time is also longer. The equivalent equations for each case are as follows.

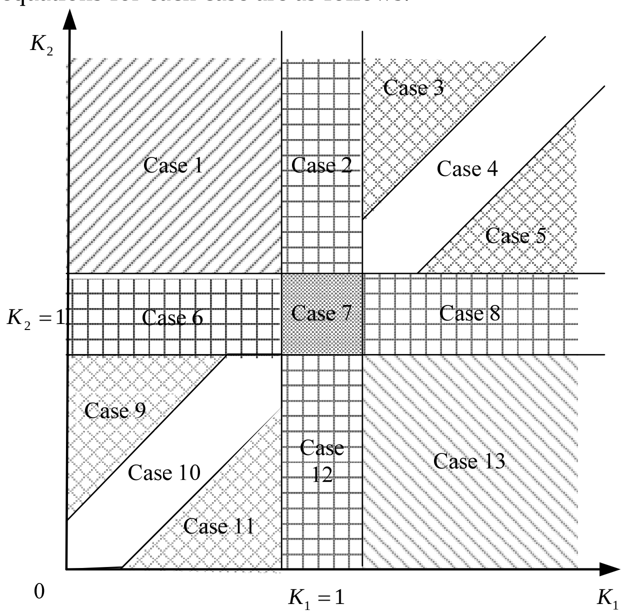


Fig. 3 The13 regions or cases of the equivalent serial queue

Case 1. $K_1 < 1$, $K_2 > 1$, then $\mu_1 < \lambda$, $\mu_2 > \lambda$, $\mu_1 < \mu_2$

The system response time of the first queue is as shown in Eq. (1),

$$E_1(t) = \frac{B_1}{\mu_1} \tag{1}$$

The system response time of the second queue is as shown in Eq. (2).

$$E_2(t) = \frac{1}{\mu_2 - \mu_1} \tag{2}$$

Because the bottleneck is at the first queue, the system response time of the equivalent serial queue is as shown in Eq. (1), $E_{eq}(t) = \frac{B_1}{\mu_{eq}}$ (3)

We assume this is a linear system, so we can add the average response times of the two serial queues:

$$E_1(t) + E_2(t) = E_{eq}(t) \tag{4}$$

We plug Eq. (1), (2) and (3) into Eq. (4) and obtain

$$\frac{B_1}{\mu_1} + \frac{1}{\mu_2 - \mu_1} = \frac{B_1}{\mu_{eq}} \tag{5}$$

From Eq. (5) we obtain the equivalent service rate of the equivalent serial queue as shown in Eq. (6).

$$\mu_{eq} = \frac{\mu_1 B_1(\mu_2 - \mu_1)}{\mu_1 + B_1(\mu_2 - \mu_1)} = \frac{K_1 B_1 (K_2 - K_1)}{K_1 + B_1(K_2 - K_1)}\lambda \tag{6}$$

By examining Eq. (6) we learn that $\mu_{eq} = f(K_1, K_2, B_1, \lambda)$

We plug Eq. (6) into Eq. (3) and obtain the average response time as shown in Eq. (7).

$$E_{eq} = \frac{\mu_1 + B_1(\mu_2 - \mu_1)}{\mu_1(\mu_2 - \mu_1)} \tag{7}$$

Case 2. $K_1 = 1$, $K_2 > 1$, then $\mu_1 = \lambda$, $\mu_2 > \lambda$, $\mu_1 < \mu_2$

The system response time of the first queue is as shown in Eq. (8).

$$E_1(t) = \frac{B_1}{2\mu_1} \tag{8}$$

The system response time of the second queue is as shown in Eq. (9).

$$E_2(t) = \frac{1}{\mu_2 - \mu_1} = \frac{1}{\mu_2 - \lambda} \tag{9}$$

Because the bottleneck is at the first queue, the system response time of the equivalent serial queue is as shown in Eq. (10). $E_{eq}(t) = \frac{B_1}{2\mu_{eq}}$ (10)

We assume that this is a linear system, so we can add the average response time of the two serial queues:

$$E_1(t) + E_2(t) = E_{eq}(t) \tag{11}$$

We plug Eq. (8), (9) and (10) into Eq. (11) and obtain

$$\frac{B_1}{2\mu_1} + \frac{1}{\mu_2 - \lambda} = \frac{B_1}{2\mu_{eq}} \tag{12}$$

From Eq. (12), we obtain the equivalent service rate of the equivalent serial queue as shown in Eq. (13).

$$\mu_{eq} = \frac{\mu_1 B_1(\mu_2 - \lambda)}{2\mu_1 + B_1(\mu_2 - \lambda)} = \frac{K_1 B_1(K_2 - 1)}{2K_1 + B_1(K_2 - 1)}\lambda \tag{13}$$

By examining Eq. (6), we learn that $\mu_{eq} = f(K_1, K_2, B_1, \lambda)$

We plug Eq. (13) into Eq. (10) and obtain the average response time as shown in Eq. (14).

$$E_{eq} = \frac{2\mu_1 + B_1(\mu_2 - \lambda)}{2\mu_1(\mu_2 - \lambda)} \tag{14}$$

Case 3. $K_1 > 1$, $K_2 > 1$, $K_1 < K_2$ then $\mu_1 > \lambda$, $\mu_2 > \lambda$, $\mu_1 < \mu_2$.

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (15) and (16).

$$\mu_{eq} = \frac{\mu_1\mu_2 - \lambda^2}{\mu_1 + \mu_2 - 2\lambda} = \frac{K_1 K_2 - 1}{K_1 + K_2 - 2}\lambda \tag{15}$$

$$E_{eq} = \frac{\mu_1 + \mu_2 - 2\lambda}{\mu_1\mu_2 - \lambda(\mu_1 + \mu_2) + \lambda^2} \tag{16}$$

Case 4. $K_1 > 1$, $K_2 > 1$, $K_1 = K_2$ then $\mu_1 > \lambda$, $\mu_2 > \lambda$, $\mu_1 = \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (17) and (18).

$$\mu_{eq} = \frac{\mu_1 + \lambda}{2} = \frac{\mu_2 + \lambda}{2} = \frac{K_1 + 1}{2}\lambda = \frac{K_2 + 1}{2}\lambda \tag{17}$$

$$E_{eq} = \frac{2}{\mu_1 - \lambda} = \frac{2}{\mu_2 - \lambda} \tag{18}$$

Case 5. $K_1 > 1$, $K_2 > 1$, $K_1 > K_2$ then $\mu_1 > \lambda$, $\mu_2 > \lambda$, $\mu_1 > \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (19) and (20).

$$\mu_{eq} = \frac{\mu_1\mu_2 - \lambda^2}{\mu_1 + \mu_2 - 2\lambda} = \frac{K_1K_2 - 1}{K_1 + K_2 - 2}\lambda \tag{19}$$

$$E_{eq} = \frac{\mu_1 + \mu_2 - 2\lambda}{\mu_1\mu_2 - \lambda(\mu_1 + \mu_2) + \lambda^2} \tag{20}$$

Case 6. $K_1 < 1$, $K_2 = 1$, then $\mu_1 < \lambda$, $\mu_2 = \lambda$, $\mu_1 < \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (21) and (22).

$$\mu_{eq} = \frac{\mu_1 B_1(\mu_2 - \mu_1)}{\mu_1 + B_1(\mu_2 - \mu_1)} = \frac{K_1B_1(K_2 - K_1)}{K_1 + B_1(K_2 - K_1)}\lambda \tag{21}$$

$$E_{eq} = \frac{\mu_1 + B_1(\mu_2 - \mu_1)}{\mu_1(\mu_2 - \mu_1)} \tag{22}$$

Case 7. $K_1 = 1$, $K_2 = 1$ then $\mu_1 = \lambda$, $\mu_2 = \lambda$, $\mu_1 = \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (23) and (24).

$$\mu_{eq} = \mu_1 = \mu_2 = \lambda \tag{23}$$

$$E_{eq} = \frac{B_1 + B_2}{2\mu_1} = \frac{B_1 + B_2}{2\mu_2} = \frac{B_1 + B_2}{2\lambda} \tag{24}$$

Case 8. $K_1 > 1$, $K_2 = 1$ then $\mu_1 > \lambda$, $\mu_2 = \lambda$, $\mu_1 > \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (25) and (26).

$$\mu_{eq} = \frac{(\mu_1 - \lambda)B_2\mu_2}{(\mu_1 - \lambda)B_2 + 2\mu_2} = \frac{(K_1 - 1)K_2B_2}{(K_1 - 1)B_2 + 2K_2}\lambda \tag{25}$$

$$E_{eq} = \frac{(\mu_1 - \lambda)B_2 + 2\mu_2}{2(\mu_1 - \lambda)\mu_2} \tag{26}$$

Case 9. $K_1 < 1$, $K_2 < 1$, $K_1 < K_2$ then $\mu_1 < \lambda$, $\mu_2 < \lambda$, $\mu_1 < \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (27) and (28).

$$\mu_{eq} = \frac{B_1\mu_1(\mu_2 - \mu_1)}{\mu_1 + B_1(\mu_2 - \mu_1)} = \frac{B_1K_1(K_2 - K_1)}{K_1 + B_1(K_2 - K_1)}\lambda \tag{27}$$

$$E_{eq} = \frac{\mu_1 + B_1(\mu_2 - \mu_1)}{\mu_1(\mu_2 - \mu_1)} \tag{28}$$

Case 10. $K_1 < 1$, $K_2 < 1$, $K_1 = K_2$ then $\mu_1 < \lambda$, $\mu_2 < \lambda$, $\mu_1 = \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (29) and (30).

$$\mu_{eq} = \mu_1 = \mu_2 = K_1\lambda = K_2\lambda \tag{29}$$

$$E_{eq} = \frac{(B_1 + B_2)}{2\mu_1} = \frac{(B_1 + B_2)}{2\mu_2} \tag{30}$$

Case 11. $K_1 < 1$, $K_2 < 1$, $K_1 > K_2$ then $\mu_1 < \lambda$, $\mu_2 < \lambda$, $\mu_1 > \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (31) and (32).

$$\mu_{eq} = \frac{\mu_1\mu_2(B_1 + B_2)}{\mu_1B_2 + \mu_2B_1} = \frac{K_1K_2(B_1 + B_2)}{K_1B_2 + K_2B_1}\lambda \tag{31}$$

$$E_{eq} = \frac{\mu_1B_2 + \mu_2B_1}{\mu_1\mu_2} \tag{32}$$

Case 12. $K_1 = 1$, $K_2 < 1$, then $\mu_1 = \lambda$, $\mu_2 < \lambda$, $\mu_1 > \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (33) and (34).

$$\mu_{eq} = \frac{2\mu_1\mu_2B_2}{2\mu_1B_2 + \mu_2B_1} = \frac{2K_1K_2B_2}{2K_1B_2 + K_2B_1}\lambda \tag{33}$$

$$E_{eq} = \frac{2\mu_1B_2 + \mu_2B_1}{2\mu_1\mu_2} \tag{34}$$

Case 13. $K_1 > 1$, $K_2 < 1$ then $\mu_1 > \lambda$, $\mu_2 < \lambda$, $\mu_1 > \mu_2$

By using a similar procedure we obtain the equivalent service rate and the system response time as shown in Eq. (35) and (36).

$$\mu_{eq} = \frac{(\mu_1 - \lambda)(B_1 + B_2)\mu_2}{(\mu_1 - \lambda)B_2 + \mu_2B_1} = \frac{K_2(B_1 + B_2)(K_1 - 1)}{K_2B_1 + B_2(K_1 - 1)}\lambda \tag{35}$$

$$E_{eq} = \frac{\mu_2B_1 + B_2(\mu_1 - \lambda)}{\mu_2(\mu_1 - \lambda)} \tag{36}$$

Fig. 3 shows the service rates of 13 cases or working regions of the equivalent serial queue with the equations representing the equivalent service rate and the system response time.

4. The Simulation of Two Serial Queues

We have chosen to use the simulation tool QNAT of the computer network to verify our analysis. This software has been announced at international conferences [9, 10] and has been many times applied in academic research [1]. We use QNAT to simulate two serial queues and a single equivalent queue and compare their system response times. In the first step we simulate the two serial queues with the arrival rate set at 100 requests/sec and the service rate of the first queue at from 10 requests/sec to 200 requests/sec. The service rates of the second queue are from 10 requests/sec to 200 requests/sec with an increment step of 10 requests/sec. The buffer size can be established as INFTY. We simulate and compare the two system response times from the two serial queues with the single equivalent queue. In the second step we use the service rates of the single equivalent queue as shown in Eqs. (6), (13), (20) and (22) for a single equivalent queue. In the third step the service rates are used for the single queue and find the system average response time. In the fourth step we compare the system average response time with an error between the two serial queues and the single equivalent queue.

For example: When there are two switches in a serial network system, the arrival rate is 100 requests/sec and the service rate of the first switch 50 requests/sec, and the service rate of the second switch is 200 requests/sec. In the first step we choose the arrival rate for 100 requests/sec with respect to the first service rate of 50 requests/sec (K_1 = 0.5), and with respect to the second service rate of 200 requests/sec (K_2 = 2.0). By using QNAT we obtain the system average response time of 2480.558 ms, which is as shown in Fig. 5. In the second step, according to Eq. (7) $E_{eq} = \frac{\mu_1 + B_1(\mu_2 - \mu_1)}{\mu_1(\mu_2 - \mu_1)}$, we find that the buffer size is 124027.6. In the third step, by using Eq. (6) $\mu_{eq} = \frac{\mu_1 B_1(\mu_2 - \mu_1)}{\mu_1 + B_1(\mu_2 - \mu_1)} = \frac{K_1 B_1(K_2 - K_1)}{K_1 + B_1(K_2 - K_1)}\lambda$ with the buffer size 124027 we obtain the equivalent queue service of 49.999 requests/sec. In the fourth step, by using the service rate 49.999 requests/sec, we use QNAT to simulate the system average response time of 2472.610 ms, which is as shown in Fig. 6. In the fifth step, using Eq. (37) in which "*M*" and "*C*" represent the system response times of the two serial queues and the single equivalent queue respectively, we compute the difference between them.

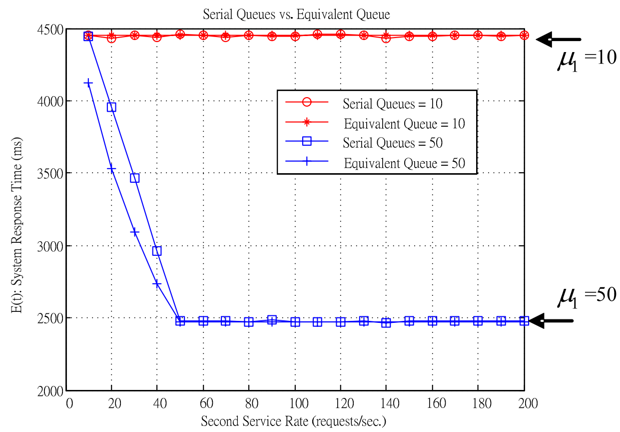
$$Error = \frac{|M - C|}{C} \tag{37}$$

The system response times of the two serial queues and the single equivalent queue are 2480.557 ms and 2472.610 ms respectively. In Eq. (37), |*M*-*C*| represents the difference of the system response time between the two serial queues and the single equivalent queue, and Eq. (38) shows the error ratio to be about 0.321%.

$$Error = \frac{|2480557 - 2472610|}{2472610} \times 100\% = 0.321\% \tag{38}$$

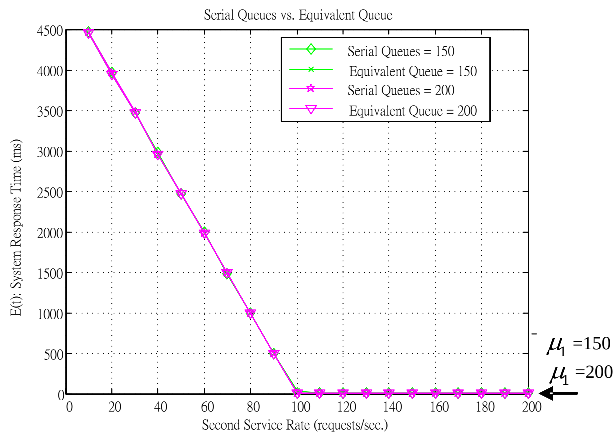
We also use QNAT to simulate a single equivalent queue based on the above-mentioned data which are plugged into QNAT and obtain the average response time.

The simulation results are shown in Fig. 4 and 5. There are two cases shown in Fig. 4. When $\mu_1 < \lambda$ we learn that blocking occurs at a higher region, and that the average response time is about 2500 ms higher. There are two cases shown in Fig. 5. When $\mu_1 > \lambda$ and μ_2 >100 requests/sec, we learn that blocking occurs at a lower region and that the average response time is lower. When μ_2<100 requests/sec, we learn that blocking occurs at a higher region, and thus that the average response time is higher. Overall, the average error margin is about 7.06%.



The unit of μ_1 is requests/sec

Fig. 4 The response time comparison between the two serial queues and the single equivalent queue using QNAT



The unit of μ_1 is requests/sec

Fig. 5 The response time comparison between the two serial queues and the single equivalent queue using QNAT

The difference between the simulation results of the two serial queues and the single equivalent queue narrows to become very small, as shown in Fig. 5. When the blocking probability is very high, the average system response time is also very large. If we want to improve the average response time for a single equivalent queue by using Eq. (37), we have currently three methods. First, we raise the service rates. Second, we reduce the arrival rates. Third, we reduce the buffer size.

When the two service rates of serial queues are unequal to the arrival rate, we can see that the average response time of the single equivalent queue is very close to that of the two serial queues. But at K_1 =1 the model shows more error.

When the service rate is unequal to the arrival rate, the average system response times of the two serial queues and the single equivalent queue are very close. When the blocking probability at the first queue is higher, the single equivalent queue and the two serial queues show a great difference, as shown in Fig. 6 and 7.

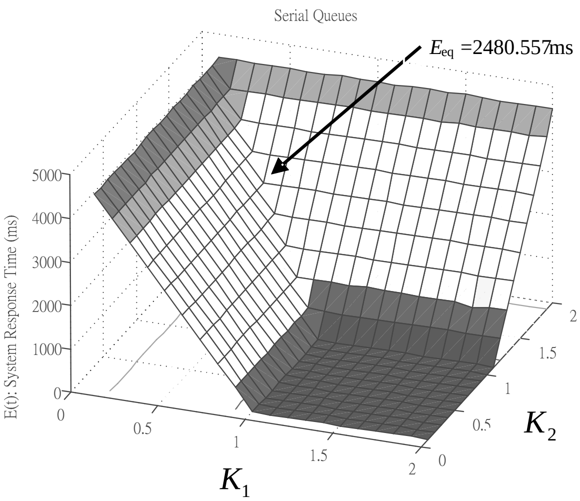


Fig. 6 The average response time of the two serial queues by using QNAT

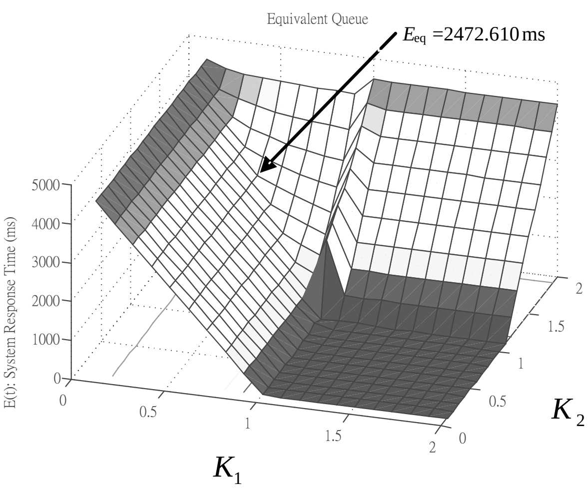


Fig. 7 The average response time of the single equivalent queue using QNAT

When the arrival rate is greater than the service rate, the buffer size will affect the response time of the common system. When the buffer is larger, it can accept a larger arrival. If the service rate no longer increases, as a result the average system response time increases, thus reducing the buffer size can improve average response time.

5. Conclusion

In this paper we propose and compare the difference between the two serial queues and the equivalent queue representing the serial multi-stage switches in a computer network. The simulation results show that the single equivalent queue provides a system response time which is accurate enough to simplify the computation of the average response time of the serial stages. Since the errors are small, below 0.11%, we can use easy equations to compute the system response time for the two serial queues. Furthermore we can observe the system response time of a network tree with serial stages. In the future we will continue to investigate the equivalent methods in parallel and equivalent queues with feedback. By using these equivalent methods we simplify the complicated queuing network into a couple of major queues; this is an efficient way to estimate the overall performance with respect to the parameters of the queuing network.

References

[1] Ying-Wen Bai and Yu-Nien Yang, An Analysis and Measurement of the Equivalent Model of Serial Queues for a Load Balancer and a Web Server of a Web Cluster with a Low Rejection Rate, *IEEE International Conference and Workshop on the Engineering of Computer Based Systems Conf. Proc*., pp.485-486, 2006.

[2] Harvey M. Deitel, Paul J. Deitel, B. Du Waldt, and L. K. Trees, *Web Services: A Technical Introduction*, Prentice Hall, First Edition, 2003.

[3] Danniel A. Menasce Virgilio A. F. Almeida, *Capacity Planning for Web Services, Metric, Models, and Methods*, Prentice-Hall, 2002.

[4] Cardellini V., Colajanni M., Yu, Dynamic load balancing on Web-server systems, *IEEE Internet Computin*g, Vol. *3,* Issue *2*, pp. 28-39, 1999.

[5] Yakoh T, Synchronized TDMA in multi hop network to stabilize transmission time, *IECON / Industrial Electronics Society*, Vol. 4, pp. 2903 - 2908, 2002.

[6] Fineberg S. A., Pedretti K. T., Analysis of 100 Mb/s Ethernet for the Whitney Commodity Computing Testbed, *Frontiers of Massively Parallel Computation*, pp. 276 - 283, 1999.

[7] Thomas G. Robertazzi, *Computer Networks And Systems - Queueing Theory and Performance Evaluation*, Springer-Verlag, New York, Second Edition, 2000.

[8] Gunter Bolch, Stefan Greiner, Hermann de Meer, Kishor S. Trivedi, *Queueing Networks and Markov Chains*, Wiley-Interscience, 1998.

[9] D. Manjunath, D. M. Bhaskar, Hema Tahilramani, Sanjay K. Bose, M. N. Umesh, QNAT: A graphical tool for the analysis of queueing networks, *IEEE/Global Connectivity in Energy, Computer, Communication and Control Conf. Proc*., pp. 320-323, 1998.

[10] H. T. Kaur, D. Manjunath, S. K. Bose, The Queueing Network Analysis Tool (QNAT), *IEEE/International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems Conf. Proc*., pp. 341-347, 2000.