International Journal of Applied Mathematics

Volume 26 No. 2 2013, 233-240
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)
doi: http://dx.doi.org/10.12732/ijam.v26i2.9

FINITE CAPACITY QUEUE WITH MULTIPLE POISSON ARRIVALS AND GENERALLY DISTRIBUTED SERVICE TIMES

Angel Vassilev Nikolov

Department of Mathematics and Computer Science
National University of Lesotho
Roma 180, LESOTHO

Abstract: The present article explores a queuing system with multiple inputs, single server, different service rates, and limited size of the buffer. The system parameters are crucial for the performance of numerous applications. We develop an analytical model of such a system and obtain the following results: steady-state probabilities of the system and system throughput.

AMS Subject Classification: 90B22
Key Words: queuing, Markov process, Kendal notations, supplementary variable

1. Introduction

The main topic of this article is the study of behaviour in equilibrium of queue with Poisson arrivals and single server. Such queues have a wide range of applications, see for example [1], [2], [5], [9], [10], [12]. Adopting the queuing theory to estimate the network traffic becomes an important way of network performance prediction, analysis and estimation. In [10] two queuing models (M/M/1/k and M/M/2/k in Kendal notations) have been applied to determine the forecast way for stable congestion rate (blocking probability) of the network traffic. The processes of data coding, decoding, and sending to the higher layer are covered by a single server. In [2] a queuing network model is constructed

Received: March 30, 2013

to study the dynamics of installing a Proxy Cache Server (PCS), that deals with a more realistic case of the presence of external visits to the remote Web servers with limited buffers. The underlying structure is a mixed queuing network with exponential arrival streams and exponential servers. Web server with processor sharing discipline is considered in [5]. The arrival process of HTTP requests is assumed to be Poissonian and the number of requests that can be served is limited. Such a system can be viewed as a network with one node. The average of the service time requirement and the limit of the number of requests being served are model parameters. The parameters are estimated by maximizing the log-likelihood function of the measured average response time. The exponential growth of in the Web traffic has led to unacceptable response times and unavailability of services, thereby driving away the customers, see [12]. Many companies are trying to address this problem using multiple Web servers with a front-end load balancer. Centralized and distributed load balancing models are developed for three routing policies. The average response time and the rejection rate for centralized model are derived first, then analysis is extended to the distributed load balancing that minimizes the average response time. Three node queuing network is chosen to describe the operation of an asynchronous Web server event handler, demultiplexer, and completion handler [9]. The demultiplexing node receives requests from two input streams, namely, read requests and write requests. The following performance metrics for each request type are determined: expected throughput, expected busy handlers, expected queue length, and expected probability of request loss. Examples of queues with single server can be found in air defence systems, see [1]. Each air defence center contains many operational consoles. Over a period of time the operator is presented with a sequence of situations that he is required to examine for problems, and that he must resolve by taking the appropriate action. The basic functions carried on at these positions are air surveillance, identification, and weapons directions.

Both simulation and analytical methods are used to describe the behaviour of queues and to derive the performance metrics. In [9] simulation model of the network is presented employing CSIM language. The analytical models rely mostly on Markov chains. The process can be imbedded at the points of departure or arrival, [4]. An M/M/1/k system is examined using this approach and equations of steady state probabilities are set up. A different approach is taken in [8]: A two-moments approximation schema is developed for the probability distribution of M/G/1/k system and extended to the analysis of M/G/1/k networks. The system studied in [1] has multiple exponential arrivals, and different exponentially distributed service times. An infinite buffer is assumed and the

performance characteristics are derived from Markov process equations.

In this paper we drop the assumption of infinite buffer due to the simple fact that it does not exist in real-life situations. Finite buffer causes the system throughput to decrease because some requests are rejected and lost. The probability of loss or rejection is often called blocking probability and it is important performance measure. We consider multiple exponential arrivals and independent generally distributed service times and derive the steady-state probabilities.

2. Definition of the Model and Description

Our mathematical model is as follows: m distinct types of requests arrive randomly and independently at the server where all are processed with service times that are random and independent. The service discipline is: First Come First Served (FCFS). Inter-arrival times for a request of type j $(j = 1, ..., m)$ are exponentially distributed with parameter λ_j. The service times follow general distribution. The system can be in one of the following states: 1) S_0^0 -no request in queue and none in service, 2) $S_n^j - n$ requests in queue and type j in service, 3) $S_k^j - k$ requests in queue and type j in service and the buffer is full. In the latter state S_k^j the system cannot accept more jobs and all arriving requests are rejected, i.e. the system is blocked for new arrivals. Acceptance of new requests resumes after completion of the job in the server.

Under the assumption of generally distributed repair time the process is not Markovian, so that a supplementary variable x, elapsed time in service, is added to obtain suitable equations, see [3].

Throughout this paper we use the following notations:

$P_n^j(x) = P$ [in the equilibrium state n requests in queue, type j in service and the elapsed service time lies between x and $x+dx$] $(n = 1, ..., k;\ j = 1, ..., m)$;

$P_0^0 = P$ [no requests in queue and none in service], steady-state probability;

$P_n^j = \int_0^\infty P_n^j(x)dx$ steady-state probability;

$P_n = \sum_{j=1}^{m} P_n^j = P$ [n requests in queue], steady-state probability;

$P_k = P_b$ blocking probability;

$F_j(x)$ c.d.f. of the service time of type j;

$f_j(x)$ p.d.f. of the service time of type j;

$h_j(x) = \frac{f_j(x)}{1-F_j(x)}$ service rate for type j;

$\lambda = \sum_{i=1}^{m} \lambda_i$;

$\delta_{m,n}$ the Kronecker delta's Laplace operator;
$\bar{g}(s)$ Laplace transform of $g(x)$;
$g^{(n)}(x)$ the n^{th} derivative of $g(x)$; $g^{(0)}(x) = g(x)$.

3. Analysis of the Model

Having in view the nature of the system, we obtain the following set of integro-differential equations:

$$P_0^0\lambda = \sum\nolimits_{j=1}^{m} \int_0^{\infty} P_0^j(x)\, h_j(x)\, dx \tag{1}$$

$$\left[\frac{d}{dx} + \lambda + h_j(x)\right] P_0^j(x) = 0 \tag{2}$$

$$\left[\frac{d}{dx} + \lambda + h_j(x)\right] P_n^j(x) = P_{n-1}^j(x)\,\lambda \tag{3}$$

for $n = 1, ..., k-1$; $j = 1, ..., m$,

$$\left[\frac{d}{dx} + h_j(x)\right] P_k^j(x) = P_{k-1}^j(x)\ \lambda. \tag{4}$$

We have the following boundary and initial conditions:

$$P_0^j(0) = P_0^0\lambda_j + \frac{\lambda_j}{\lambda} \sum\nolimits_{j=1}^{m} \int_0^{\infty} P_1^j(x)\, h_j(x)\, dx \tag{5}$$

$$P_n^j(0) = (1 - \delta_{n,k)})\frac{\lambda_j}{\lambda} \sum\nolimits_{j=1}^{m} \int_0^{\infty} P_{n+1}^j(x)\, h_j(x)\, dx \tag{6}$$

for $n = 1, ..., k$; $j = 1, ..., m$,

$$P_0^0 + \sum\nolimits_{n=0}^{k} \sum\nolimits_{j=1}^{m} P_n^j = 0. \tag{7}$$

The multiplicand $\frac{\lambda_j}{\lambda}$ in (5), (6) is the probability that type j demands initiation of service or type j is next in line, see [1].

We divide (2)-(4) by $1 - F_j(x)$ and denote $u_n^j(x) = \frac{P_n^j(x)}{1-F_j(x)}$ for $n = 0, ..., k$ and $j = 1, ..., m$. Then taking into account that $f_j(x) = (1 - F_j(x)^{(1)}$, we have from (2)-(4) after some manipulations,

$$\left[\frac{d}{dx} + \lambda\right] u_0^j(x) = 0 \tag{8}$$

$$\left[\frac{d}{dx}+\lambda\right]u_n^j(x)=u_{n-1}^j(x)\,\lambda \tag{9}$$

for $n=1,...,k-1;\ j=1,...,m$,

$$\frac{d}{dx}P_k^j(x)=u_{k-1}^j(x)\,\lambda. \tag{10}$$

By using the Laplace transform, the above equations are transformed as follows:

$$[s+\lambda]\,u_0^j(s)=0, \tag{11}$$

$$[s+\lambda]\,u_n^j(s)=u_{n-1}^j(s)\,\lambda \tag{12}$$

for $n=1,...,k-1;\ j=1,...,m$,

$$su_k^j(s)=u_{k-1}^j(s)\ \lambda. \tag{13}$$

From (11) we get

$$u_0^j(s)=\frac{u_0^j(0)}{s+\lambda}. \tag{14}$$

We use induction on n to prove the following relationship

$$u_n^j(s)=\sum_{i=0}^{n}\frac{\lambda^{n-i}P_i^j(0)}{(s+\lambda)^{n-i+1}}. \tag{15}$$

Proof. After substitution of (15) into (14) for $u_{n+1}^j(s)$ and some simplifications, we have

$$u_{n+1}^j(s)=\sum_{i=0}^{n+1}\frac{\lambda^{n-i+1}\,P_i^j(0)}{(s+\lambda)^{n-i+2}},$$

and this completes the proof.

For $u_k^j(s)$ we determine from (13) and (15)

$$u_k^j(s)=\frac{\sum\limits_{i=0}^{k-1}\frac{\lambda^{k-1-i}P_i^j(0)}{(s+\lambda)^{k-i}}}{s}. \tag{16}$$

The inverse Laplace transform of (14) and (15) yields

$$u_0^j(x)=P_0^j(0)\,e^{-\lambda x}, \tag{17}$$

$$u_n^j(x)=\sum_{i=0}^{n}\frac{\lambda^{n-i}x^{n-i}P_i^j(0)\,e^{-\lambda x}}{(n-i)!} \tag{18}$$

and

$$u_k^j(x) = \int \sum_{i=0}^{k-1} \frac{\lambda^{k-1-i} x^{k-1-i} P_i^j(0)\, e^{-\lambda x}}{(k-1-i)!}\, dx. \tag{19}$$

In order to compute the integrals in (19) we derive the following formula:

$$\int x^n e^{-\lambda x} dx = \frac{e^{-\lambda x}}{\lambda^{n+1}} \sum_{i=0}^{n} \frac{\lambda^i x^i n!}{i!}. \tag{20}$$

The proof of this is done by repeatedly applying the expression

$$\int x^l e^{-\lambda x} dx = \frac{x^l e^{-\lambda x}}{-\lambda} + \frac{l}{\lambda} \int x^{l-1} e^{-\lambda x} dx \frac{e^{-x}}{\lambda^{l+1}} \quad \text{to} \quad \int x^n e^{-\lambda x} dx.$$

After substitution of (20) into (19) we express $u_k^j(x)$:

$$u_k^j(x) = \sum_{i=0}^{k-1} \frac{e^{-\lambda x} P_i^j(0)}{\lambda} \sum_{l=0}^{k-1-i} \frac{\lambda^l x^l}{l!}. \tag{21}$$

Then, the steady-state probabilities in terms of the system parameters are:

$$P_0^0 = \frac{\sum_{j=1}^{m} P_0^j(0)\, \bar{f}_j(\lambda)}{\lambda}, \tag{22}$$

$$P_0^j = \frac{P_0^j(0)\, [1 - \bar{f}_j(\lambda)]}{\lambda}, \tag{23}$$

$$P_n^j = \sum_{i=0}^{n} P_i^j(0) \left[\frac{1}{\lambda} - \frac{(-1)^{n-i} \lambda^{n-i} \bar{F}_j^{(n-i)}(\lambda)}{(n-i)!}\right], \tag{24}$$

$$P_k^j = \sum_{i=0}^{k-1} \frac{P_i^j(0)}{\lambda} \sum_{l=0}^{k-1-i} \left[\frac{1}{\lambda} - \frac{(-1)^l \lambda^l \bar{F}_j^{(l)}(\lambda)}{l!}\right]. \tag{25}$$

Software tools like *Mathematica* [11] can be used to compute the derivatives in (21)-(25). To determine the coefficients $u_n^j(0)$ we obtain from (5) and (17):

$$P_0^j(0) = P_0^0 \lambda_j + \frac{\lambda_j}{\lambda} \sum_{j=1}^{m} \left[-\lambda P_0^j \bar{f}_j^{(1)}(\lambda) + P_1^j(0)\, \bar{f}_j(\lambda)\right]. \tag{26}$$

And from (6) and (18),

$$P_n^j(0) = \frac{\lambda_j}{\lambda} \sum_{j=1}^{m} \sum_{i=0}^{n} P_i^j(0) \frac{(-1)^{n-i} \lambda^{n-i} \bar{f}_j^{(n-i)}(\lambda)}{(n-i)!}. \tag{27}$$

The equations (7), (22), (26) and (27) form a set of simultaneous equations from which the unknowns $P_n^j(0)$ can be computed. Since modern computers have a capability to solve tens of thousands and even hundreds of thousands of linear equations, producing solutions of these equations is not a challenging task, see e.g. [6], [7].

The throughput H is diminishing due to the rejection of requests:

$$H = \lambda(1 - P_b). \tag{28}$$

4. Concluding Remarks

This paper presents an analysis of a single-server limited in size queuing system for m different types of customers having independent Poisson arrivals and generally distributed service times. The approach eliminates some constraints of the known related analyses. We obtain the steady-state probabilities thus creating opportunities for computation of the system output parameters. Although we start with fairly sophisticated set of integro-differential equations, the output of the model is a set of linear equations from which the steady-state probabilities can be determined. The ease of obtaining performance measures in a meaningless time and without much computational effort makes very feasible the incorporation of the model as a design tool for Web servers, networks, etc.

References

[1] C.J. Ancher Jr., A.V. Gafarian, Queuing with multiple Poisson inputs and exponential service times, *Operations Research*, **9**, No 3 (1961), 321-327.

[2] T. Berczes, J. Sztrick, Performance modelling of proxy cache servers, *Journal of Universal Computer Science*, **12**, No 9 (2006), 1139-1153.

[3] U.N. Bhat, *Introduction to Queuing Theory: Modelling and Analysis in Applications*, Springer, 2008.

[4] S.K. Bose, *Introduction to Queuing Systems*, Kluwer/Plenum Publishers, 2001.

[5] J. Cao, M. Anderson, C. Nyberg, M. Kihl, Web server performance modelling using an M/G/1K* PS queue, *Proc. 10th Intern. Conference on Telecommunications, ICT 2003*, **2** (2003), 1501-1506.

[6] M.T. Jones, P.E. Plassmann, Solution of large, sparse systems of linear equations in massively parallel applications, *Proc. of Supercomputing' 92* (1992), 551-560.

[7] `http://m4ri.sagemath.org/performance.html`

[8] J. Macgregor Smith, Properties and performance modelling of finite buffer M/G/1/K networks, *Computers and Operations Research* (2010), 1-15.

[9] U. Praphamontripong, S. Gokhale, A. Gokhale, J. Gray, Performance analysis of an asynchronous Web server, *Computer Software and Applications Conference*, *COMPSAC'06 30th Annual International*, **2** (2006), 22-28.

[10] S.S. Ray, P. Sahoo, Monitoring of network traffic based on queuing theory, *ARPN J. of Science and Technology*, **1**, No 1 (2011), 1-10.

[11] Wolfram, *Mathematica* 8.0.

[12] Z. Zhang, W. Fan, Web server load balancing: A queuing analysis, *European J. of Operational Research*, **186** (2008), 681-693.