Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computers and Electrical Engineering 31 (2005) 460–467

Computers and Electrical Engineering

www.elsevier.com/locate/compeleceng

State diagrams and steady-state balance equations for open queuing network models

Vidhyacharan Bhaskar *

Département Génie des systèmes d'information et de Télécommunication at the Université de Technologie de Troyes, France, 12 Rue Marie Curie, 10010 Troyes Cedex, France

Received 3 March 2005; received in revised form 14 August 2005; accepted 5 September 2005
Available online 4 November 2005

Abstract

In this paper, the state diagrams and steady-state balance equations for two kinds of open queuing network models are presented. The first model comprises a network of single queues with single servers, while the second model comprises multiple servers for single queues. State diagrams are drawn for $(2,3)$ queuing networks with (i) single servers and (ii) multiple servers. Steady-state balance equations are derived from the state diagrams. The paper provides a method to solve open queuing networks by analyzing the stochastic process involved in the transition of states in a continuous time Markov chain which represents the state diagram of a queuing system.

Keywords: State diagrams; Steady-state balance equations; Open queuing networks; Traffic intensity; Synchronization processor; Execution processor

1. Introduction

Queuing networks have been successfully used in performance modelling of computer and communication systems [1,2]. They are especially useful for resource contention and queuing

* Tel.: +33 325 71 8428; fax: +33 325 71 5699.
E-mail address: vidhyacharan.bhaskar@utt.fr

0045-7906/$ - see front matter
doi:10.1016/j.compeleceng.2005.09.001

for service. Queuing models for computer system performance analysis have a service center for the CPU(s), a service center for each I/O channel, and possibly others [3].

In some systems, several CPUs and I/Os may be required. For such systems, a queuing network containing several queues and servers will be necessary. A job requesting service waits in a queue if it finds that all servers are busy. Once a job is serviced by a server, the next job waiting in the queue is selected for service. Jobs get serviced in a first-in-first-out (FIFO) fashion as they wait in a queue. After completion of service at one service center, a job may leave the queuing network or may move to another service center for further service [3].

Queuing network models are broadly categorized into (i) Open queuing network, and (ii) Closed queuing network models. Open queuing network models are used when the number of job arrivals is small. Closed queuing network models are used when the number of job arrivals is large.

Open queuing network models are characterized by one or more job arrivals corresponding to one or more job departures from the network. In closed queuing network models, jobs neither depart nor enter the network.

The performance measures of the (m,n) open queuing network models shown in this paper are derived in [4]. This paper extends the work in [4] by discussing the state diagrams and steady-state balance equations for the open queuing network models.

Section 2 discusses the block diagram of an open queuing network model with single servers; and describes the state diagrams and steady-state balance equations of $(2,3)$ queuing network with single servers. Section 3 discusses the block diagram of an open queuing network model with multiple servers; and describes the state diagrams and steady-state balance equations of $(2,3)$ queuing network with multiple servers. Finally, Section 4 presents the conclusions.

2. Open queuing network model with single servers

In this section, an open queuing network of single queues with single servers is presented. State diagrams are drawn and steady-state balance equations are derived for $(2,3)$ queuing networks with single servers.

2.1. System description

Consider the (m,n) queuing model shown in Fig. 1. The model contains m synchronization processors (SPs) and n execution processors (EPs). Each SP and EP is analogous to a server with a single queue. Each queue in the network of m queues is served by an SP_i and is denoted as Q_{SP_i} $(1 \leqslant i \leqslant m)$, and each queue in the network of n queues served by an EP_j is denoted as Q_{EP_j}, $(1 \leqslant j \leqslant n)$.

The model shown in Fig. 1 is an open network of $(m+n)$ queues. Each queue is $M/M/1$ in nature. The first M in the $M/M/1$ notation denotes that the arrivals to the queue are memoryless. The arrivals are random in nature and form a Poisson process at a constant arrival rate. The second M denotes that the service times are memoryless. The service times have an exponential distribution. The queuing discipline is first-come-first-served (FCFS) [3]. Each server has a queue of unlimited capacity.

If a "customer" denotes a job arriving into a computer system, then the server represents the computer system. Since most computer systems consist of a set of interacting resources, and hence