International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:3, 2009

Analysis of Linked in Series Servers with Blocking, Priority Feedback Service and Threshold Policy

Walenty Oniszczuk

***Abstract*—**The use of buffer thresholds, blocking and adequate service strategies are well-known techniques for computer networks traffic congestion control. This motivates the study of series queues with blocking, feedback (service under Head of Line (HoL) priority discipline) and finite capacity buffers with thresholds. In this paper, the external traffic is modelled using the Poisson process and the service times have been modelled using the exponential distribution. We consider a three-station network with two finite buffers, for which a set of thresholds (*tm1* and *tm2*) is defined. This computer network behaves as follows. A task, which finishes its service at station *B*, gets sent back to station *A* for re-processing with probability σ. When the number of tasks in the second buffer exceeds a threshold *tm2* and the number of task in the first buffer is less than *tm1,* the fed back task is served under HoL priority discipline. In opposite case, for fed backed tasks, "no two priority services in succession" procedure (preventing a possible overflow in the first buffer) is applied. Using an open Markovian queuing schema with blocking, priority feedback service and thresholds, a closed form cost-effective analytical solution is obtained. The model of servers linked in series is very accurate. It is derived directly from a two-dimensional state graph and a set of steady-state equations, followed by calculations of main measures of effectiveness. Consequently, efficient expressions of the low computational cost are determined. Based on numerical experiments and collected results we conclude that the proposed model with blocking, feedback and thresholds can provide accurate performance estimates of linked in series networks.

***Keywords*—**Blocking, Congestion control, Feedback, Markov chains, Performance evaluation, Threshold-base networks.

I. Introduction

FINITE buffer queues with thresholds, blocking and service priorities are of great importance towards the effective computer network traffic congestion control. Congestion occurs when many users compete for the network resources and the resources are inadequate for the requests. The use of thresholds and blocking for controlling congestion in computer network buffers is well known and often used. Congestion control based on thresholds and blocking is aimed to control the traffic-causing overload and so to satisfy the Quality of Service (QoS) requirements of the different classes of traffic.

The aim of this paper is to build up an analytical model of computer networks with blocking, feedback priority service and threshold policies; seeking to obtain high network utilization, acceptable delay time, and some degree of fairness among users. It is difficult to compute the queue length, waiting time and other queuing features of this network, since this is a two-queue network and both are finite capacity queues with thresholds. Throughput studies, and efficiency studies of the network with blocking, thresholds and feedback using this model, are important for real-life applications. In the past couple of years, there has been a strong demand for computer networks that can provide adequate quality of service (QoS) among users. Such demand initiated a need for a solution where the servers play an active role in congestion control. A series of threshold policies and congestion control procedures were proposed to control queue lengths and to promote fairness among task generating sources [1, 2, 8, 9, 10, 12, 17, 21, 24, 25]. Most computer networks are connection oriented, also known as linked in series. There are many blocking models that can be used to provide insight into the performance of those networks. Blocking models, if they can be solved efficiently, are often used in network planning and dimensioning. Due to obvious resource constrains, realistic models have finite capacity buffers, where the queue length cannot exceed its arbitrary maximum capacity. When the queue length reaches its capacity, the buffer and the server are said to be full (blocking factors). Queuing network models (QNMs) with finite capacity queues and blocking provide powerful and practical tools for performance evaluation and predication of discrete flow systems as computer systems and networks. As a consequence, cost-effective numerical techniques and analytic approximations are needed for study of complex queuing systems. Time priority mechanisms such as Head of Line (HoL), take into account that some services may tolerate longer delays than others and deal with the order with which tasks are transmitted [14]. The traditional analyses of the ONMs with blocking are based on the Markov Chain approach [3, 4, 7, 11, 16]. In recent years, extensive research in this field produced many results that are well explained in the literature. An excellent study may be found in the well-known series of books by Balsamo [2] and Perros [21]. In addition, many interesting theories and models appeared in a

Manuscript received April 7, 2008.
This work was supported by the Bialystok University of Technology under Grant W/WI/5/09.
Walenty Oniszczuk is with the Faculty of Computer Science, Bialystok University of Technology, 15-351 Bialystok, Wiejska str. 45, Poland (phone +48-85-746-90-03, e-mail: walenty@wi.pb.bialystok.pl).

variety of journals and at worldwide conferences in the field of computer science, traffic engineering and communication engineering [6, 13, 19, 20, 23].

Despite all the research done so far, there are still many important and interesting models to be studied. One such model is finite capacity queues under various blocking mechanisms and synchronization constraints, such as those involving feedback service or priority scheduling. In this kind of model, a task with a fixed probability can return to the previous node immediately after its service at the current node. Although feedback queues have already been extensively studied in literature see [12, 15, 18], series queues with priority feedback are more complex object for research than the queues without feedback. The introduction of multiple-thresholds can give rise to inter-dependency between the thresholds setting, which are also dependent on the service discipline used. Because of these complex inter-dependencies, a suitable analytical model is necessary to understand the resulting interactions. The main aim of this paper is to formulate such a model with multiple queue thresholds and examine the queuing behaviour under a priority service discipline for feed backed tasks.

The rest of the paper is organized as follows: Section 2 presents and explains the analytical model. Section 3, analyzes a three-node network with blocking, priority feedback and thresholds. Procedures for calculating the performance measures and quality of service (QoS) parameters are presented in Section 4. Numerical results obtained using our solution techniques are given in Section 5. Section 6 finally concludes the paper.

II. Model description

The general model description is:

- The arrival process from source station is Poisson.
- Each station consists of a single server and queue.
- Two stations provide service that is exponentially distributed.
- Scheduling disciplines are FCFS and HOL.
- All queues have finite capacity *m1* and *m2* with thresholds *tm1* and *tm2*.

Fig. 1 presents a simplified three-station (source, station *A* and station *B)* description of the proposed model. Tasks arrive from the source at station *A* according to the Poisson process with rate λ and they are processed in a FIFO manner. The service received by station A is as follows. The task first receives an exponentially distributed service with rate μ_1^A. After service completion at station *A*, the task proceeds to station *B* (exponentially distributed service with rate μ^B). Once it finishes at station *B*, it gets sent back to station *A* for re-processing with probability σ (exponentially distributed service with rate μ_2^A - according to HoL priority discipline). Once finished, each re-processed task departs from the network. We are also assuming that tasks after being processed in the station *B* leave the network with *1- σ* probability.

A feed backed task is served at station *A* according to a non-preemptive priority scheme (HoL). It is served independently of all other events if the number of tasks in the second buffer exceeds a threshold *tm2* and the number of task in the first buffer is less than *tm1*. If the number of tasks in second buffer is less than *tm2* and the number of tasks in the first buffer exceeds *tm1*, another procedure is applied. This "no two priority services in succession" procedure prevents a possible overflow in the first buffer.

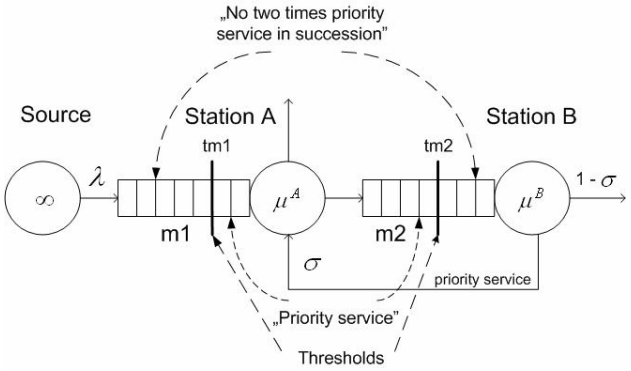

Fig.1. Illustration of the three-station network model with blocking, priority feedback service and thresholds.

The HoL priority service means that a task cannot get into service at station *A* (it waits at station *B* – blocking factor) until the task currently in service is completed. The successive service times at both stations are assumed to be mutually independent and they are independent of the state of the network.

In such networks, another one blocking factor may occurs when a station reaches its maximum capacity, which, in turn, may momentarily stop the traffic of all incoming tasks to that station. Let us say that between station *A* and station *B,* there is a waiting buffer with finite capacity *m2*. When the buffer fills up completely, the accumulation of new tasks from the first station is temporarily suspended. Similarly, if the first buffer (with capacity *m1*) in front of the first station gets full, then that source station is momentarily blocked. This is a classical mechanism for controlling the intensity of an arriving task streams from a source station. According to our model specification (see Fig. 1), when either of the buffers is full, any task upon completion of service at the source station or at station *A,* is forced to wait at its station. The task flows from the source station to station *A* or from the first station-to-station *B* depending on service process in stations *A* or *B* respectively. Physically, blocked tasks stay on the source station or on station *A*, but the nature of the service process in stations *A* and *B* allows us to treat them as located in additional places in the buffers and belonging to stations *A* or *B* (see Fig. 2 and Fig. 3). If the server *A* is busy, any task which needs a repeated service in this station, after having its service completed in the station *B*, is forced to wait in this station (blocking factor). The nature of the service process in this case depends only of the service rates in station *A*. Similarly; it allows one to treat this task as located in the additional place in the buffer *A*.

Deadlocks can occur in a multistage network with feedback and blocking - as depicted above. For example, assume that station *A* is blocked by station *B* (the second buffer is full). In

such situation, it is possible that a task in station *B*, upon its completion may get send back to station *A*, which in turn, will cause a deadlock. In this paper, we assume that deadlocks are detected and resolved instantaneously without any delay, simply by exchanging the blocked tasks.

III. PROPOSED SOLUTION (QUEUEING MODEL)

Markov processes constitute the fundamental theory underlying the concept of queuing systems and provide very flexible, powerful, and efficient means for the description and analysis of dynamic computer network properties. Each queuing system can, in principle, be mapped onto an instance of a Markov process and then mathematically evaluated in terms of this process. The linked in series network model described in Section 2 is a multistage queuing system with recycling that also allows blocking. Service at each station (see Fig. 1) is provided by a single exponential server. External arrivals (tasks) from the source station join the first station in a Poisson fashion at the rate λ. We assume that each successive service at both station and the inter-arrival times are independent of each other. Under such assumptions the queuing network we are describing can be represented by a continuous-time Markov chain, in which the underlying Markov process can analyze the stationary and transient behavior of the network. We consider this network in its stationary conditions. As such, the queuing network model reaches a steady-state condition and the underlying Markov chain has a stationary state distribution. If each queue has finite capacity, the underlying process yields finite state space. The solution of the Markov chain representation may then be computed and the desired performance characteristics, such as queue length distribution, utilizations, and throughputs, obtained directly from the stationary probability vector. In addition, features such as blocking, priority feedback service, thresholds, may be incorporated into a Markov chain representation – although the effect of doing so will increase the size of the state space.

In theory, any Markov model can be solved numerically. In particular, solution algorithm for Markov queuing networks with blocking, priority feedback service and thresholds is a five-step procedure:

1. Definition of the series network state space (choosing a state space representation).
2. Enumerating all the transitions that can possible occur among the states.
3. Definition of the transition rate matrix $\boldsymbol{Q}$ that describes the network evaluation (generating the transition rate).
4. Solution of linear system of the global balance equations to derive the stationary state distribution vector (computing appropriate probability vector).
5. Computation from the probability vector of the average performance indices.

The state of the queuing network with blocking, priority feedback service and thresholds (see Fig. 2 and Fig. 3) can be described by random variables (*i,j,k*), where *i* indicates the number of tasks at the first station, *j* indicates the number of tasks at second server and *k* represents the state of each server. Here, the index *k* may have the following values: *0, 1, 2, 3, 4*.

TABLE I
THE INDEX K VALUE

Index k:	Task Description:
0	Idle network
1	Regular task service for $i \le m1+1$ and source blocking for $i = m1+2$
2	Priority task service for $i \le m1+1$ and source blocking for $i = m1+2$
3	Blocking one node and regular task service at the other one
4	Blocking one node and priority task service at the other one

Based on an analysis the state space diagrams, the process of constructing the steady-state equations in the Markov model can be divided into several independent steps which describe some similar, repeatable schemas (see Fig. 2 and Fig. 3). These steady-state equations are:

$$\lambda \cdot p_{0,0,0} = \mu^B (1-\sigma) \cdot p_{0,1,1} + \mu_2^A \cdot p_{1,0,2} \quad \text{for } i=0, j=0, k=0$$

$$(\lambda + \mu^B \sigma + \mu^B (1-\sigma)) \cdot p_{0,j,1} = \mu_1^A \cdot p_{1,j-1,1} + \mu_2^A \cdot p_{1,j,2} + \mu^B(1-\sigma) \cdot p_{0,j+1,1} \quad \text{for } i=0,\ j=1, \dots, m2,\ k=1;\ \text{if } m2>0$$

$$(\lambda + \mu^B \sigma + \mu^B (1-\sigma)) \cdot p_{0,m2+1,1} = \mu_1^A \cdot p_{1,m2,1} + \mu_2^A \cdot p_{1,m2+1,2} + \mu^B(1-\sigma) \cdot p_{0,m2+2,3} \quad \text{for } i=0,\ j=m2+1,\ k=1$$

$$(\lambda + \mu_1^A) \cdot p_{1,0,1} = \lambda \cdot p_{0,0,0} + \mu^B(1-\sigma) \cdot p_{1,1,1} + \mu_2^A \cdot p_{2,0,2} \quad \text{for } i=1,\ j=0,\ k=1$$

$$(\lambda + \mu_1^A) \cdot p_{i,0,1} = \lambda \cdot p_{i-1,0,1} + \mu^B(1-\sigma) \cdot p_{i,1,1} + \mu_2^A \cdot p_{i+1,0,2} \quad \text{for } i=2, \dots, m1+1,\ j=0,\ k=1;\ \text{if } m1>0$$

$$(\lambda + \mu_2^A) \cdot p_{1,0,2} = \mu^B \sigma \cdot p_{0,1,1} + \mu^B(1-\sigma) \cdot p_{1,1,2} + \mu_2^A \cdot p_{1,1,4} \quad \text{for } i=1,\ j=0,\ k=2$$

$$(\lambda + \mu_2^A) \cdot p_{i,0,2} = \lambda \cdot p_{i-1,0,2} + \mu^B(1-\sigma) \cdot p_{i,1,2} + \mu_2^A \cdot p_{i,1,4} \quad \text{for } i=2, \dots, tm1+1,\ j=0,\ k=2;\ \text{if } tm1 \ge 1$$

$$(\lambda + \mu_2^A) \cdot p_{i,0,2} = \lambda \cdot p_{i-1,0,2} + \mu^B(1-\sigma) \cdot p_{i,1,2} \quad \text{for } i = tm1+2, \dots, m1+1,\ j = 0,\ k = 2;\ \text{if } tm1 < m1$$

$$(\lambda + \mu^B(1-\sigma) + \mu_1^A + \mu^B \sigma) \cdot p_{i,j,1} = \lambda \cdot p_{i-1,j,1} + \mu_1^A \cdot p_{i+1,j-1,1} + \mu^B(1-\sigma) \cdot p_{i,j+1,1} + \mu_2^A \cdot p_{i+1,j,2} \quad \text{for } i=1, \dots, m1+1,\ j=1, \dots, m2,\ k=1;\ \text{if } m2 > 0$$

$$(\lambda + \mu^B(1-\sigma) + \mu_1^A + \mu^B \sigma) \cdot p_{i,m2+1,1} = \lambda \cdot p_{i-1,m2+1,1} + \mu_1^A \cdot p_{i+1,m2,1} + \mu^B(1-\sigma) \cdot p_{i,m2+2,3} + \mu_2^A \cdot p_{i+1,m2+1,2} \quad \text{for } i=1, \dots, m1+1,\ j=m2+1,\ k=1$$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B \sigma) \cdot p_{1,j,2} = \mu_1^A \cdot p_{1,j,3} + \mu^B(1-\sigma) \cdot p_{1,j+1,2} + \mu^B \sigma \cdot p_{0,j+1,1} + \mu_2^A \cdot p_{1,j+1,4} \quad \text{for } i=1,\ j=1, \dots, m2,\ k=2;\ \text{if } m2 > 0$$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B \sigma) \cdot p_{i,j,2} = \lambda \cdot p_{i-1,j,2} + \mu_1^A \cdot p_{i,j,3} + \mu^B(1-\sigma) \cdot p_{i,j+1,2} + \mu_2^A \cdot p_{i,j+1,4} \quad \text{for } i=2, \dots, tm1+1,\ j=1, \dots, m2,\ k=2;\ \text{if } tm1 \ge 1 \ \&\ m2 > 0$$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B \sigma) \cdot p_{i,j,2} = \lambda \cdot p_{i-1,j,2} + \mu_1^A \cdot p_{i,j,3} + \mu^B(1-\sigma) \cdot p_{i,j+1,2} \quad \text{for } i=tm1+2, \dots, m1+1,\ j=1, \dots, tm2+1,\ k = 2;\ \text{if } tm1 < m1 \ \&\ tm2 < m2 \tag{1}$$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B \sigma) \cdot p_{i,j,2} = \lambda \cdot p_{i-1,j,2} + \mu_1^A \cdot p_{i,j,3} + \mu^B(1-\sigma) \cdot p_{i,j+1,2} \quad \text{for } i=tm1+2, \dots, m1+1,\ j=1, \dots, m2,\ k=2;\ \text{if } tm1 < m1 \ \&\ tm2 = m2 \ \&\ m2 > 0$$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B \sigma) \cdot p_{i,j,2} = \lambda \cdot p_{i-1,j,2} + \mu_1^A \cdot p_{i,j,3} + \mu^B(1-\sigma) \cdot p_{i,j+1,2} + \mu_2^A \cdot p_{i,j+1,4} \quad \text{for } i=tm1+2, \dots, m1+1,\ j=tm2+2, \dots, m2,\ k=2;\ \text{if } tm1 < m1 \ \&\ tm2 < m2-1$$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B \sigma) \cdot p_{i,m2+1,2} = \lambda \cdot p_{i-1,m2+1,2} + \mu_1^A \cdot p_{i,m2+1,3} + \mu^B \sigma \cdot p_{i-1,m2+2,3}$$

for $i=2, \dots, m1+1, j=m2+1, k=2$; if $m1 > 0$

$$(\lambda + \mu^B(1-\sigma) + \mu_2^A + \mu^B\sigma) \cdot p_{1,m2+1,2} = \mu_1^A \cdot p_{1,m2+1,3} + \mu^B\sigma \cdot p_{0,m2+2,3}$$

for $i=1, j=m2+1, k=2$

For states with blocking the equations are:

$$(\lambda + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{0,m2+2,3} = \mu_1^A \cdot p_{1,m2+1,1}$$

for $i=0, j=m2+2, k=3$

$$(\lambda + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{i,m2+2,3} = \lambda \cdot p_{i-1,m2+2,3} + \mu_1^A \cdot p_{i+1,m2+1,1}$$

for $i=1, \dots, m1, j=m2+2, k=3$; if $m1 > 0$

$$(\mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+1,m2+2,3} = \lambda \cdot p_{m1,m2+2,3} + \mu_1^A \cdot p_{m1+2,m2+1,1}$$

for $i=m1+1, j=m2+2, k=3$

$$\mu_1^A \cdot p_{m1+2,0,1} = \lambda \cdot p_{m1+1,0,1} + \mu^B(1-\sigma) \cdot p_{m1+2,1,1}$$

for $i=m1+2, j=0, k=1$

$$(\mu_1^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,j,1} = \lambda \cdot p_{m1+1,j,1} + \mu^B(1-\sigma) \cdot p_{m1+2,j+1,1}$$

for $i=m1+2, j=1, \dots, m2, k=1$; if $m2 > 0$

$$(\mu_1^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,m2+1,1} = \lambda \cdot p_{m1+1,m2+1,1}$$

for $i=m1+2, j=m2+1, k=1$

$$\mu_2^A \cdot p_{m1+2,0,2} = \lambda \cdot p_{m1+1,0,2} + \mu^B(1-\sigma) \cdot p_{m1+2,1,2}$$

for $i=m1+2, j=0, k=2$; if $tm1 < m1$ or $m1=0$

$$\mu_2^A \cdot p_{m1+2,0,2} = \lambda \cdot p_{m1+1,0,2} + \mu^B(1-\sigma) \cdot p_{m1+2,1,2} + \mu_2^A \cdot p_{m1+2,1,4} \quad (2)$$

for $i=m1+2, j=0, k=2$; if $tm1 = m1$ and $m1 > 0$

$$(\mu_2^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,j,2} = \lambda \cdot p_{m1+1,j,2} + \mu^B(1-\sigma) \cdot p_{m1+2,j+1,2} + \mu_1^A \cdot p_{m1+2,j,3}$$

for $i=m1+2, j=1, \dots, tm2+1, k=2$; if $tm1< m1$ & $tm2<m2-1$

$$(\mu_2^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,j,2} = \lambda \cdot p_{m1+1,j,2} + \mu^B(1-\sigma) \cdot p_{m1+2,j+1,2} + \mu_1^A \cdot p_{m1+2,j,3}$$

for $i=m1+2, j=1, \dots, m2, k=2$; if $tm1<m1$ & $tm2=m2$ & $m2>0$

$$(\mu_2^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,j,2} = \lambda \cdot p_{m1+1,j,2} + \mu^B(1-\sigma) \cdot p_{m1+2,j+1,2} + \mu_1^A \cdot p_{m1+2,j,3} + \mu_2^A \cdot p_{m1+2,j+1,4}$$

for $i=m1+2, j=tm2+2, \dots, m2, k=2$; if $tm1<m1$ & $tm2 = m2-2$

$$(\mu_2^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,j,2} = \lambda \cdot p_{m1+1,j,2} + \mu^B(1-\sigma) \cdot p_{m1+2,j+1,2} + \mu_1^A \cdot p_{m1+2,j,3} + \mu_2^A \cdot p_{m1+2,j+1,4}$$

for $i=m1+2, j=1, \dots, m2, k=2$; if $tm1= m1$ & $tm2 > 0$

$$(\mu_2^A + \mu^B(1-\sigma) + \mu^B\sigma) \cdot p_{m1+2,m2+1,2} = \lambda \cdot p_{m1+1,m2+1,2} + \mu^B\sigma \cdot p_{m1+1,m2+2,3} + \mu_1^A \cdot p_{m1+2,m2+1,3}$$

for $i=m1+2, j=m2+1, k=2$

$$(\lambda + \mu_1^A) \cdot p_{1,j,3} = \mu^B\sigma \cdot p_{1,j,1} + \mu_2^A \cdot p_{2,j,4}$$

for $i=1, j=1, \dots, tm2+1, k=3$; if $tm1=0$

$$(\lambda + \mu_1^A) \cdot p_{1,j,3} = \mu^B\sigma \cdot p_{1,j,1}$$

for $i=1, j = tm2+2, \dots, m2+1, k=3$; if $tm1=0$ & $tm2 < m2$

$$(\lambda + \mu_1^A) \cdot p_{1,j,3} = \mu^B\sigma \cdot p_{1,j,1}$$

for $i=1, j=1, \dots, m2+1, k=3$; if $tm1 > 0$

$$(\lambda + \mu_1^A) \cdot p_{i,j,3} = \mu^B\sigma \cdot p_{i,j,1} + \lambda \cdot p_{i-1,j,3}$$

for $i=2, \dots, tm1+1, j=1, \dots, m2+1, k=3$; if $tm1 \geq 1$

$$(\lambda + \mu_1^A) \cdot p_{i,j,3} = \mu^B\sigma \cdot p_{i,j,1} + \lambda \cdot p_{i-1,j,3} + \mu_2^A \cdot p_{i+1,j,4}$$

for $i=tm1+2, \dots, m1+1, j=1, \dots, tm2+1, k=3$; if $tm1 < m1$

$$(\lambda + \mu_1^A) \cdot p_{i,j,3} = \mu^B\sigma \cdot p_{i,j,1} + \lambda \cdot p_{i-1,j,3}$$

for $i=tm1+2, \dots, m1+1, j=tm2+2, \dots, m2+1, k=3$; if $tm1 < m1$ & $tm2 < m2$

$$\mu_1^A \cdot p_{m1+2,j,3} = \mu^B\sigma \cdot p_{m1+2,j,1} + \lambda \cdot p_{m1+1,j,3}$$

for $i=m1+2, j=1, \dots, m2+1, k=3$

$$(\lambda + \mu_2^A) \cdot p_{1,j,4} = \mu^B\sigma \cdot p_{1,j,2}$$

for $i=1, j=1, \dots, m2+1, k=4$

$$(\lambda + \mu_2^A) \cdot p_{i,j,4} = \mu^B\sigma \cdot p_{i,j,2} + \lambda \cdot p_{i-1,j,4}$$

for $i=2, \dots, m1+1, j=1, \dots, m2+1, k=4$

$$\mu_2^A \cdot p_{m1+2,j,4} = \mu^B\sigma \cdot p_{m1+2,j,2} + \lambda \cdot p_{m1+1,j,4}$$

for $i=m1+2, j=1, \dots, m2+1, k=4$

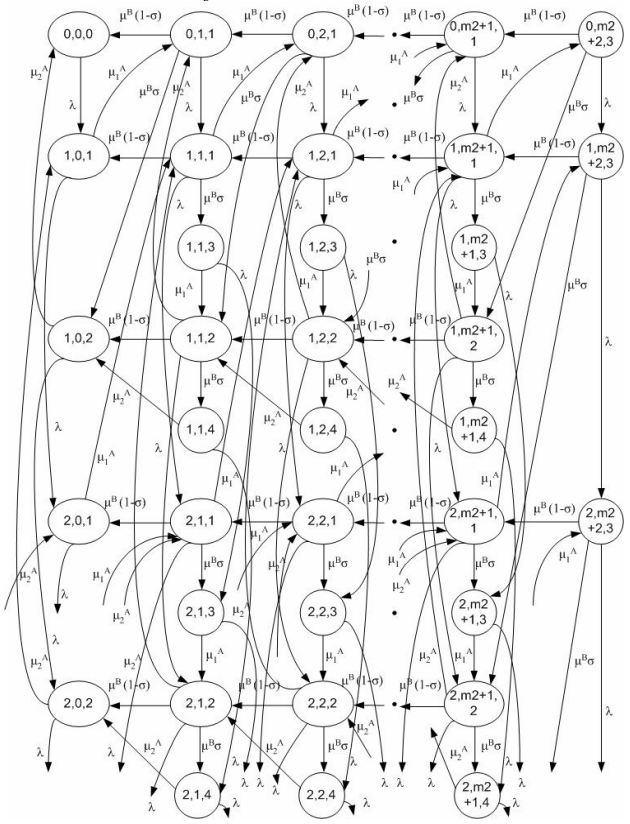

Fig. 2. Two-dimensional network state diagram (first part)

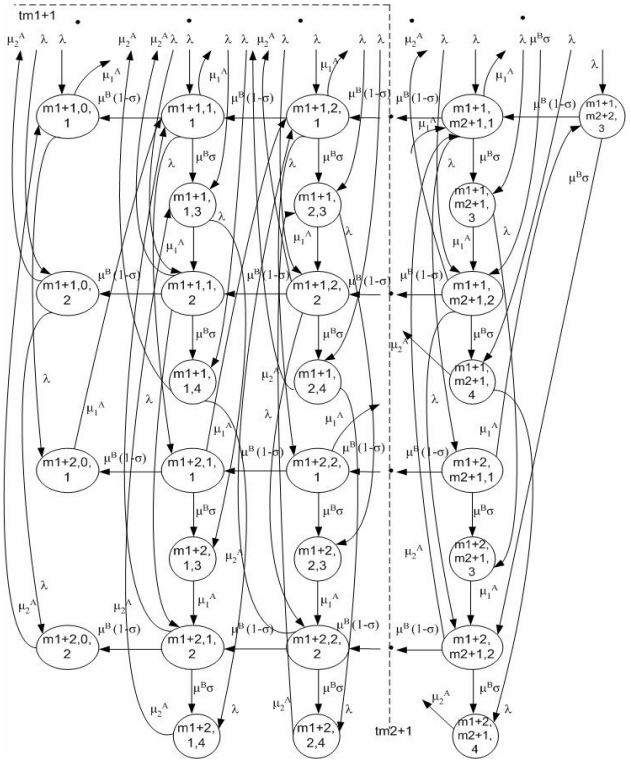

Fig. 3. Two-dimensional network state diagram (second part)

Here, a linked in series network with blocking is formulated as a Markov process and the stationary probability vector can be obtained using numerical methods for linear systems of equations. The process of solving the set of equations given by (1) and (2) with common algorithms is not a trivial case, because part of the graph has an irregular shape. There are many methods for the solution of a system of linear algebraic equations but some of these are restricted to certain regular structures of the parameter matrix. In this paper, a method is used, in which the whole graph column by column is sequentially re-numbered in order to solve this problem (to get a standard finite-state one-dimensional Markov chain). This operation is necessary for solving a set of linear equations in MATLAB based on the well-known MATLAB efficient sparse storage schemas and efficient sparsity-preserving algorithms. The state of any Markov chain may be represented as an integer-valued row vector adopted by the above-mentioned MATLAB algorithms. After this operation, new set of linear equations can be solved using classical numerical methods, based on algorithms typical for sparse and diagonal matrices (for example – numerical experiments in MATLAB). The generation of the rate matrix $\boldsymbol{Q}$ can now be accomplished by going through the list of states and generating all the feasible transitions out of each state and the associated rate of transition. For this kind of Markov process in a steady state, we simply have [2, 5, 22]:

$$x\boldsymbol{Q} = 0 \tag{3}$$

Where $\boldsymbol{x}$ is the stationary probability vector whose k-th element x_k is the steady-state probability that the system is in state k. Vector $\boldsymbol{x}$ can be obtained from (3) and the normalizing condition $\sum_{all\ \ states} x_k = 1$, using equation-solving techniques.

In the next step, calculated state probabilities are assigned to each state shown on the two-dimensional state graphs.

IV. Performance Measures (An Integral Approach)

The different types of queuing systems are analyzed mathematically to determine performance measure from the description of the system. Since most queuing systems have stochastic elements, these measures are often random variables and their probability distributions, or at the very least their expected values desired to be found. Normally, however, we are content with the results in the steady state. The system is said to be in steady state when all transient behaviour has ended, and the values of the performance measures are independent of time. The solution of the Markov chain representation may then be computed. The desired performance characteristics, such as queue length distribution, utilizations, and throughputs can be also obtained directly from the stationary probability distribution vector.

The most important performance measures are:

1. Idle probability p_{idle}:

$$p_{idle} = p_{0,0,0} \tag{4}$$

2. Station A blocking probability p_{blA}:

$$p_{blA} = \sum_{i=0}^{m1+1} p_{i,m2+2,3} \tag{5}$$

3. Source station blocking probability p_{blS}:

$$p_{blS} = \sum_{j=0}^{m2+1} (p_{m1+2,j,1} + p_{m1+2,j,2}) + \sum_{j=1}^{m2+1} (p_{m1+2,j,3} + p_{m1+2,j,4}) + p_{m1+1,m2+2,3} \tag{6}$$

4. Both stations (source and station A) simultaneous blocking probability p_{blAS}:

$$p_{blAS} = p_{m1+1,m2+2,3} \tag{7}$$

5. Station B blocking probability p_{blB}:

$$p_{blB} = \sum_{i=1}^{m1+2} \sum_{j=1}^{m2+1} (p_{i,j,3} + p_{i,j,4}) \tag{8}$$

6. Both stations (source and station B) simultaneous blocking probability p_{blBS}:

$$p_{blBS} = \sum_{j=1}^{m2+1} (p_{m1+2,j,3} + p_{m1+2,j,4}) \tag{9}$$

7. The average number of blocked tasks in station A:

$$n_{blA} = \sum_{i=0}^{m1+1} (1 \cdot p_{i,m2+2,3}) \tag{10}$$

8. The average number of active (non-blocked) tasks in station A:

$$l_A = \sum_{i=1}^{m1+2} \sum_{j=0}^{m2+1} \sum_{k=1}^{2} (1 \cdot p_{i,j,k}) + \sum_{i=1}^{m1+2} \sum_{j=1}^{m2+1} \sum_{k=3}^{4} (1 \cdot p_{i,j,k}) \tag{11}$$

9. The average number of tasks in the first buffer v_A:

$$v_A = \sum_{i=2}^{m1+1} \sum_{j=0}^{m2+1} \sum_{k=1}^{2} (i-1) \cdot p_{i,j,k} + \sum_{i=2}^{m1+1} \sum_{j=1}^{m2+1} \sum_{k=3}^{4} (i-1) \cdot p_{i,j,k} + \sum_{j=0}^{m2+1} \sum_{k=1}^{2} (m1 \cdot p_{m1+2,j,k}) + \sum_{j=1}^{m2+1} \sum_{k=3}^{4} (m1 \cdot p_{m1+2,j,k}) + \sum_{i=1}^{m1} (i \cdot p_{i,m2+2,3}) + m1 \cdot p_{m1+1,m2+2,3} \tag{12}$$

10. The average number of tasks in station A:

$$n_A = \sum_{i=1}^{m1+1} \sum_{j=0}^{m2+1} \sum_{k=1}^{2} (i \cdot p_{i,j,k}) + \sum_{i=1}^{m1+1} \sum_{j=1}^{m2+1} \sum_{k=3}^{4} (i \cdot p_{i,j,k}) + \sum_{j=0}^{m2+1} \sum_{k=1}^{2} (m1+1) \cdot p_{m1+2,j,k} + \sum_{j=1}^{m2+1} \sum_{k=3}^{4} (m1+1) \cdot p_{m1+2,j,k} + \sum_{i=0}^{m1} (i+1) \cdot p_{i,m2+2,3} + (m1+1) \cdot p_{m1+1,m2+2,3} \tag{13}$$

11. The average number of blocked tasks in source station:

$$n_{blS} = \sum_{j=0}^{m2+1}(1\cdot p_{m1+2,j,1} + 1\cdot p_{m1+2,j,2}) + \sum_{j=1}^{m2+1}(1\cdot p_{m1+2,j,3} + 1\cdot p_{m1+2,j,4}) + 1\cdot p_{m1+1,m2+2,3} \quad (14)$$

12. The average number of simultaneous blocked tasks in both stations (source and station *A*) n_{blAS}:

$$n_{blAS} = 2\cdot p_{m1+1,m2+2,3} \quad (15)$$

13. The average number of tasks blocked in station *B*:

$$n_{blB} = \sum_{i=1}^{m1+2}\sum_{j=1}^{m2+1}(1\cdot p_{i,j,3} + 1\cdot p_{i,j,4}) \quad (16)$$

14. The average number of active (non-blocked) tasks in station *B*:

$$l_B = \sum_{i=1}^{m1+2}\sum_{j=1}^{m2+1}\sum_{k=1}^{2}(1\cdot p_{i,j,k}) + \sum_{j=1}^{m2+1}(1\cdot p_{0,j,1}) + \sum_{i=0}^{m1+1}(1\cdot p_{i,m2+2,3}) \quad (17)$$

15. The average number of tasks in the second buffer v_B:

$$v_B = \sum_{j=2}^{m2+1}(j-1)\cdot p_{0,j,1} + \sum_{i=1}^{m1+2}\sum_{j=2}^{m2+1}\sum_{k=1}^{4}(j-1)\cdot p_{i,j,k} + \sum_{i=0}^{m1+1}(m2\cdot p_{i,m2+2,3}) \quad (18)$$

16. The average number of tasks in station *B*:

$$n_B = \sum_{j=1}^{m2+1} j\cdot p_{0,j,1} + \sum_{i=1}^{m1+2}\sum_{j=1}^{m2+1}\sum_{k=1}^{4}(j\cdot p_{i,j,k}) + \sum_{i=0}^{m1+1}(m2+1)\cdot p_{i,m2+2,3} \quad (19)$$

17. The average number of simultaneous blocked tasks in both stations (source and station *B*) n_{blBS}:

$$n_{blBS} = \sum_{j=1}^{m2+1} 2\cdot(p_{m1+2,j,3} + p_{m1+2,j,4}) \quad (20)$$

18. The mean blocking time in station *A*:

$$t_{blA} = \frac{n_{blA}}{\mu^B} = \frac{1}{\mu^B}\cdot\sum_{i=0}^{m1+1}(1\cdot p_{i,m2+2,3}) \quad (21)$$

19. The mean blocking time in station *B*:

$$t_{blB} = \frac{n^1_{blB}}{\mu^A_1} + \frac{n^2_{blB}}{\mu^A_2} = \frac{1}{\mu^A_1}\cdot\sum_{i=1}^{m1+2}\sum_{j=1}^{m2+1}(1\cdot p_{i,j,3}) + \frac{1}{\mu^A_2}\cdot\sum_{i=1}^{m1+2}\sum_{j=1}^{m2+1}(1\cdot p_{i,j,4}) \quad (22)$$

20. The mean blocking time in source station:

$$t_{blS} = \frac{n^1_{blS}}{\mu^A_1} + \frac{n^2_{blS}}{\mu^A_2} + \frac{n^3_{blS}}{\mu^B} = \frac{1}{\mu^A_1}\cdot[\sum_{j=0}^{m2+1}(1\cdot p_{m1+2,j,1}) + \sum_{j=1}^{m2+1}(1\cdot p_{m1+2,j,3})] + \frac{1}{\mu^A_2}\cdot[\sum_{j=0}^{m2+1}(1\cdot p_{m1+2,j,2}) + \sum_{j=1}^{m2+1}(1\cdot p_{m1+2,j,4})] + \frac{1}{\mu^B}\cdot p_{m1+1,m2+2,3} \quad (23)$$

21. The simultaneous mean blocking time in both stations (source and station *A*):

$$t_{blAS} = \frac{n_{blAS}}{2}\cdot\frac{1}{\mu^B} \quad (24)$$

22. The simultaneous mean blocking time in both stations (source and station *B*):

$$t_{blBS} = \frac{1}{\mu^A_1}\cdot\sum_{j=1}^{m2+1}1\cdot p_{m1+2,j,3} + \frac{1}{\mu^A_2}\cdot\sum_{j=1}^{m2+1}1\cdot p_{m1+2,j,4} \quad (25)$$

23. The mean waiting time in the buffer *A*:

$$w_A = \frac{(v_{A1}+v_{A3})}{\mu^A_1} + \frac{(v_{A2}+v_{A4})}{\mu^A_2} = \frac{1}{\mu^A_1}\cdot[\sum_{i=2}^{m1+1}\sum_{j=0}^{m2+1}(i-1)\cdot p_{i,j,1} + \sum_{j=0}^{m2+1} m1\cdot p_{m1+2,j,1} + \sum_{i=2}^{m1+1}\sum_{j=1}^{m2+1}(i-1)\cdot p_{i,j,3} + \sum_{i=1}^{m1}(i\cdot p_{i,m2+2,3}) + \sum_{j=1}^{m2+1} m1\cdot p_{m1+2,j,3} + m1\cdot p_{m1+1,m2+2,3}] + \frac{1}{\mu^A_2}\cdot[\sum_{i=2}^{m1+1}\sum_{j=0}^{m2+1}(i-1)\cdot p_{i,j,2} + \sum_{j=0}^{m2+1} m1\cdot p_{m1+2,j,2} + \sum_{i=2}^{m1+1}\sum_{j=1}^{m2+1}(i-1)\cdot p_{i,j,4} + \sum_{j=1}^{m2+1} m1\cdot p_{m1+2,j,4}] \quad (26)$$

24. The mean response time in station *A*:

$$q_A = \frac{1}{\mu^A_1} + \frac{1}{\mu^A_2}\cdot\sigma + t_{blA} + w_A \quad (27)$$

25. The mean waiting time in the buffer *B*:

$$w_B = \frac{(v_{B1}+v_{B2})}{\mu^B} + \frac{v_{B3}}{\mu^A_1} + \frac{v_{B4}}{\mu^A_2} = \frac{1}{\mu^B}\cdot[\sum_{j=2}^{m2+1}(j-1)\cdot p_{0,j,1} + \sum_{i=1}^{m1+2}\sum_{j=2}^{m2+1}((j-1)\cdot(p_{i,j,1}+p_{i,j,2})) + \sum_{i=0}^{m1+1}(m2\cdot p_{i,m2+2,3})] + \frac{1}{\mu^A_1}\cdot\sum_{i=1}^{m1+2}\sum_{j=2}^{m2+1}(j-1)\cdot p_{i,j,3} + \frac{1}{\mu^A_2}\cdot\sum_{i=1}^{m1+2}\sum_{j=2}^{m2+1}(j-1)\cdot p_{i,j,4} \quad (28)$$

26. The mean response time in station B:

$$q_B = w_B + \frac{1}{\mu^B} + t_{blB} \tag{29}$$

27. The average network throughput (sojourn) time:

$$t_{thr} = \frac{1}{\lambda} + t_{blS} + q_A + q_B \tag{30}$$

28. The effective input stream rate (intensity):

$$\lambda_1 = \frac{1}{\frac{1}{\lambda} + t_{blS}} \tag{31}$$

29. Station A utilization ρ_A:

$$\rho_A = l_A + n_{blA} \tag{32}$$

30. Station B utilization ρ_B:

$$\rho_B = l_B + n_{blB} \tag{33}$$

V. Numerical Results

To demonstrate our analysis procedures of a three-station network with blocking, priority feedback service and thresholds proposed in Section 3, we have performed numerous calculations. The first group of calculations was realized for many parameters combinations by varying the feedback probability σ within a range from *0.0* to *1.0* and by varying both threshold values within a range from *0* to *10* for *tm1*, plus within a range from *10* to *0* for *tm2* (*tm1+tm2* = const). The inter-arrival rate λ from the source station to station A is chosen as equal to *2.0*. The service rates in station A and station B are equal to $\mu_1^A = 4.0$, $\mu_2^A = 1.0$, $\mu^B = 3.0$. Based on such parameters, the following results were obtained and presented in Fig. 4 and Fig. 5. Figs. 4-5 depict the Quality of Service (QoS) parameters as a function of the thresholds policy and of the feedback probability value. In the two figures, the buffers size is taken as *m1 = 12* and *m2* = 10.

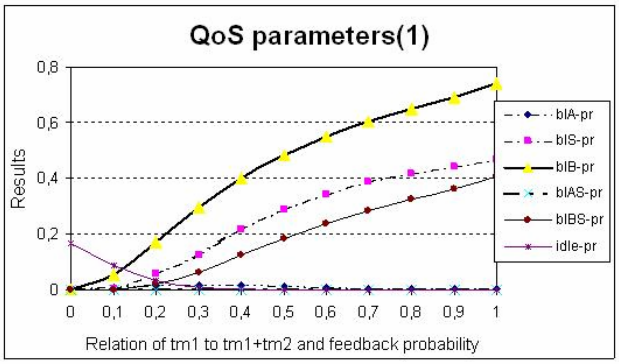

Fig. 4. Graphs of QoS parameters, where, *blA-pr* is the station A blocking probability, *blS-pr* is the source station blocking probability, *blB-pr* is the station B blocking probability, *blAS-pr* is the simultaneous blocking probability of the source station and station A, *blBS-pr* is the simultaneous blocking probability of the source station and station B and *idle-pr* is idle network probability.

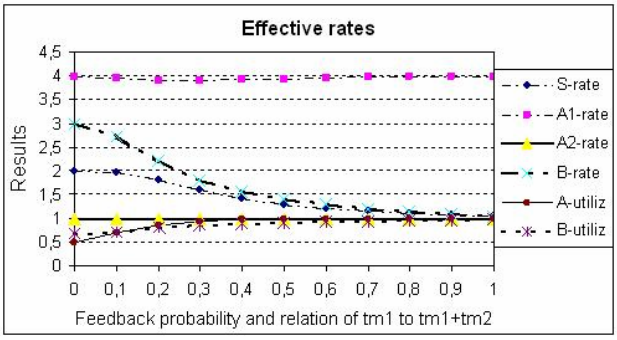

Fig. 5. Graphs of QoS parameters, where *S-rate* is the effective input rate, *A1-rate* is the effective service rate of regular tasks in station A (blocking factor), *A2-rate* is the service rate of priority tasks in station A (no blocking), *B-rate* is the effective service rate in station B (blocking factor), *A-utiliz* and *B-utiliz* are servers utilization coefficients.

For the second group of experiments the following parameters were chosen: the service rates in station A and station B are equal to $\mu_1^A = 2.0$, $\mu_2^A = 2.0$, $\mu^B = 1.6$. The inter-arrival rate λ from the source station to station A is *3.0*. The feedback probability σ is equal to *0.6*. Buffer capacities are: *m1 = 2* with threshold *tm1 = 1* and *m2* is changed within the range from *0* to *16* (with step 2) with threshold *tm2* equal to *m2/2*. For this model the following results were obtained and the majority of them are presented in Table 1 and Fig. 6.

TABLE II
The Measures of Effectiveness

$m2$	w_A	w_B	t_{blA}	t_{blS}	t_{thr}	t_{blB}	t_{blAS}	t_{blBS}	λ_1
0	0.96	0.00	0.15	0.36	3.36	0.11	0.11	0.08	1.42
2	0.94	1.06	0.18	0.37	4.45	0.12	0.13	0.10	1.41
4	0.92	1.68	0.09	0.33	4.94	0.14	0.06	0.11	1.49
6	0.87	1.87	0.05	0.29	5.01	0.16	0.03	0.11	1.59
8	0.85	2.21	0.03	0.28	5.32	0.17	0.02	0.11	1.62
10	0.85	2.56	0.02	0.27	5.63	0.17	0.01	0.12	1.64
12	0.84	2.91	0.01	0.27	5.98	0.18	0.01	0.12	1.65
14	0.84	3.29	0.01	0.26	6.35	0.18	0.00	0.12	1.66
16	0.84	3.70	0.00	0.26	6.76	0.18	0.00	0.12	1.66

Given parameters: $1/\lambda = 0.333$, $1/\mu_1^A = 0.500$, $1/\mu_2^A = 0.500$, $1/\mu^B = 0.625$, *m1 = 2, m2 (var) = 0 - 16, tm1 = 1, tm2 (var) = 0 – 8,* and $\sigma = 0.60$.

The results of the experiment clearly show that the effect of the blocking, feedback phenomena and threshold policy must be taken into account when analyzing performance of a computer network. As noted above, feedback probability σ, blocking factor and threshold policy considerably change the performance measures in such networks. Figs. 4-6 illustrate dependencies of QoS parameters and effective input and service rates on the feedback probability and buffer threshold policy.

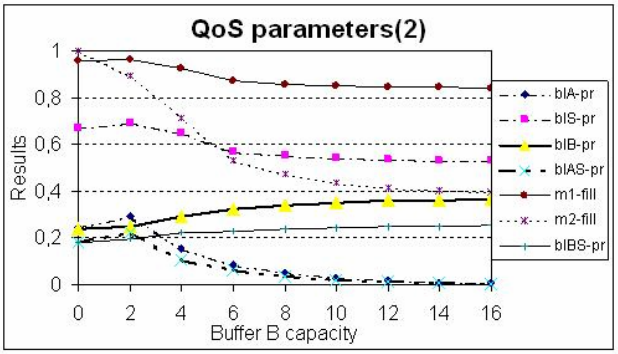

Fig. 6. Graphs of QoS parameters, where, *blA-pr* is the station *A* blocking probability, *blS-pr* is the source station blocking probability, *blB-pr* is the station *B* blocking probability, *blAS-pr* is the simultaneous blocking probability of the source station and station *A, m1-fill* and *m2-fill* are buffers filling coefficients and *blBS-pr* is the simultaneous blocking probability of the source station and station *B*.

VI. Conclusion

In this paper, we investigated the problem of analytical (mathematical) modeling and calculation of the stationary state probabilities for a multistage network with recycling task, blocking and threshold policy. We have developed an analytical, queuing-base model for the blocking characteristics in a series computer network. In particular, we modeled the threshold-based buffers filling control algorithm. Tasks blocking probabilities and some other fundamental performance characteristics of such network were derived, followed by numerical examples. The results confirm importance of a special treatment for the models with blocking, with HOL feedback service, and threshold policy in finite capacity buffers, which justifies this research. The results can be used for capacity planning and performance evaluation of real-time computer networks where blocking, feedback and thresholds are present. Moreover, this proposal is useful in designing buffer sizes or channel capacities for a given blocking probability requirement constraint.

References

[1] I. Awan, "Analysis of multiple-threshold queues for congestion control of heterogeneous traffic streams", *Simulation Modelling Practice and Theory*, vol. 14, pp. 712-724, 2006.

[2] S. Balsamo, V. de Nito Persone, R. Onvural, "*Analysis of Queueing Networks with Blocking*", Boston: Kluwer Academic Publishers, 2001.

[3] S. Balsamo, V. de Nito Persone, P. Inverardi, A review on queueing network models with finite capacity queues for software architectures performance predication, *Performance Evaluation*, vol. 51, no. 2-4, pp. 269-288, 2003.

[4] A. Badrah, T. Chachórski, J. Domańska, J.-M. Fourneau, F. Quessette, "Performance evaluation of multistage interconnection networks with blocking – discrete and continuous time Markov models", *Archiwum Informatyki Teoretycznej i Stosowanej,* vol. 14, no. 2, pp. 145-162, 2002.

[5] G. Bolch, S. Greiner, H. de Meer, K.S. Trivedi, *Queueing Networks and Markov Chains. Modeling and Performance Evaluation with Computer Science Applications*, New York: John Wiley, 1998.

[6] A. Bose, X. Jiang, B. Lui, G. Li, "Analysis of manufacturing blocking systems with Network Calculus", *Performance Evaluation*, vol. 63, pp. 1216-1234, 2006.

[7] R.J. Boucherie, N.M. van Dijk, "On the arrival theorem for product form queueing networks with blocking", *Performance Evaluation*, vol. 29, no. 3, pp. 155-176, 1997.

[8] S-T. Cheng, Ch-M. Chen, I-R. Chen, "Performance evaluation of an admission control algorithm: dynamic threshold with negotiation", *Performance Evaluation*, vol. 52, pp. 1-13, 2003.

[9] B.D. Choi, S.H. Choi, B. Kim, D.K. Sung, "Analysis of priority queueing systems based on thresholds and its application to signalling system no. 7 with congestion control", *Computer Networks*, vol. 32, pp. 149-170, 2000.

[10] A.W. Eckford, F.R. Kschischang, S. Pasupathy, "On Designing Good LDPC Codes for Markov Channels", *IEEE Transactions on Information Theory*, vol. 53, no.1, pp. 5-21, 2007.

[11] A. Economou, D. Fakinos, "Product form stationary distributions for queueing networks with blocking and rerouting", *Queueing Systems*, vol. 30, no. 3/4, pp. 251-260, 1998.

[12] A. Gomez-Corral, M.E. Martos, "Performance of two-stage tandem queues with blocking: The impact of several flows of signals", *Performance Evaluation*, vol. 63, pp. 910-938, 2006.

[13] U.C. Gupta, S.K. Samanta, R.K. Sharma, M.L. Chaudhry, "Discrete-time single-server finite-buffer under discrete Markovian arrival process with vacations", *Performance Evaluation*, vol. 64, pp. 1-19, 2007.

[14] T. Katayama, K. Kobayashi, "Analysis of a nonpreemptive priority queue with exponential timer and server vacations", *Performance Evaluation*, vol. 64, pp. 495-506, 2007.

[15] C.S. Kim et al. "The BMAP/*G*/1-> ·/PH/1/*M* tandem queue with feedback and losses", *Performance Evaluation*, vol. 64, pp. 802-818, 2007.

[16] D. Kouvatsos, I.U. Awan, R.J. Fretwell, G. Dimakopoulos, "A cost-effective approximation for SRD traffic in arbitrary multi-buffered networks", *Computer Networks*, vol. 34, pp. 97-113, 2000.

[17] J.C.S. Lui, L. Golubchik, "Stochastic complement analysis of multi-server threshold queues with hysteresis", *Performance Evaluation*, vol. 35, pp. 19-48, 1999.

[18] R.D. van der Mei, B.M.M. Gijsen, N., in't Veld, J.L. van den Berg, "Response times in a two-node queueing network with feedback", *Performance Evaluation*, vol. 49, pp. 99-110, 2002.

[19] Oniszczuk W. Analysis of an Open Linked Series Three-Station Network with Blocking, in *Advances in Information Processing and Protection*, J. Pejaś, K. Saeed Eds., New York: Springer Science+Business Media, LLC, 2007, pp. 419-429.

[20] R. Onvural, "Survey of closed queuing networks with blocking", *Computer Survey*, vol. 22, no.2, pp. 83-121, 1990.

[21] Perros H.G. *Queuing Networks with Blocking. Exact and Approximate Solution*, New York: Oxford University Press, 1994.

[22] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, New Jersey: Princeton University Press, 1994.

[23] T. Tolio, S.B. Gershwin, "Throughput estimation in cyclic queueing networks with blocking", *Annals of Operations Research*, vol. 79, pp. 207-229, 1998.

[24] E. Xu, A.S. Alfa, "A vacation model for the non-saturated Readers and Writers system with a threshold policy", *Performance Evaluation*, vol. 50, pp. 233-244, 2002.

[25] H. Zhang, Z-P. Jiang, Y. Fan, S. Panwar, "Optimization based flow control with improved performance", *Communications in Information and Systems*, vol.4, no. 3, pp. 235-252, 2004.

Walenty Oniszczuk received his M.Sc. and Ph.D. degrees in Computer Science in 1976 and 1980, respectively. Since 1980 he has been employed at University of Technology in Bialystok (Poland), Faculty of Computer Science.

His main research interests include discrete-state queuing models with priority scheduling, queuing models with blocking, analytical investigation of large computer systems and computer networks, concepts and theory of simulation plus simulation languages, traffic control, application of Hidden Markov Models, etc. He is a member of the Society for Modelling and Simulation International (SCS).