BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 56, No. 4, 2008

Queueing systems and networks. Models and applications

B. FILIPOWICZ* and J. KWIECIEŃ

Department of Automatics, AGH University of Science and Technology, 30 Mickiewicza Ave., 30-059 Kraków, Poland

Abstract. This article describes queueing systems and queueing networks which are successfully used for performance analysis of different systems such as computer, communications, transportation networks and manufacturing. It incorporates classical Markovian systems with exponential service times and a Poisson arrival process, and queueing systems with individual service. Oscillating queueing systems and queueing systems with Cox and Weibull service time distribution as examples of non-Markovian systems are studied. Jackson's, Kelly's and BCMP networks are also briefly characterized. The model of *Fork-Join* systems applied to parallel processing analysis and the FES approximation making possible of *Fork-Join* analysis is also presented. Various types of blocking representing the systems with limited resources are briefly described. In addition, examples of queueing theory applications are given. The application of closed BCMP networks in the health care area and performance evaluation of the information system is presented. In recent years the application of queueing systems and queueing networks to modelling of human performance arouses researchers' interest. Hence, in this paper an architecture called the Queueing Network-Model Human Processor is presented.

Key words: queueing systems, queueing networks, performance analysis.

1. Introduction

Queueing theory is considered to be a branch of operations research. It constitutes a powerful tool in modelling and performance analysis of many complex systems, such as computer networks, telecommunication systems, call centres, flexible manufacturing systems and service systems. Recently, the queueing theory including queueing systems and networks arouses mathematicians', engineers' and economics interests.

A queueing system consists of inputs, queue and servers as service centres. Generally, it consists of one or more servers for serving customers arriving in some manner and having some service requirements. The customers (the flow of entities) represent users, jobs, transactions or programmes. They arrive at the service facility for service, waiting for service if there is a waiting room, and leave the system after being served. Sometimes customers are lost. The queueing systems are described by distribution of inter-arrival times, distribution of service times, the number of servers, the service discipline and the maximum capacity etc.

The model with multiple systems called a queueing network better represents the real structure than a single system. Depending on the total number of customers the queueing networks can be classified into three categories: open, closed and mixed. Depending on the number of customer classes we have single class networks or multiclass networks. Designing queueing networks we have to specify the queueing and service disciplines, topologies of queueing systems and types of systems. The simplest, Jackson networks are probably the most known and widely applied network model in various fields. Jackson's major contribution was to find a product-form steady-state solution. These networks have o lot of bounds. They are single class networks with exponential systems. These assumptions are not fulfilled in BCMP networks, which have more complicated structure. Application of queueing theory provides methods for the design and study of real systems. In a number of papers an application of queueing theory has been described.

The main aim is to present the review of the most popular queueing systems and networks, mainly with a product-form solution. For these networks the solution for the steady-state probabilities can be presented as a product of factors presenting the state of individual system. In this paper the use of queueing theory is presented in the context of a variety of real systems.

The paper is organized as follows. Section 2 briefly describes the history of queueing theory. Section 3 presents queueing systems with Poisson arrival processes and exponential service times. Section 4 discusses some non-Markovian systems. Section 5 presents the most popular queueing networks. The examples of queueing theory in health care area and to modelling of the information system and human performance are described in Section 6.

2. History

The history of queueing systems and networks goes back to the beginning of XX century. A number of papers and researches have dealt with queueing theory. It is impossible to present all of them, therefore we will mention only the most important – in our opinion – achievements in the queueing theory [1].

A.K. Erlang, a Danish engineer, published his first paper on queueing theory in 1909. The earliest mention of the term "queueing system" appeared in 1951, in the article of D.G. Kendall. In 1953 Kendall published his paper on

*e-mail: filip@ia.agh.edu.pl

the queueing notation. D.R. Cox proposed analysis of non-Markovian process in 1955. In 1957 Jackson had considered an open queueing networks with exponential servers and an exogenous Poisson process. He showed that the steady-state distribution has a product form. In 1958 F. Haight introduced parallel queues. In 1961 J. Little proved a formula with dependency of mean number of jobs in systems (and queue) from mean response time (waiting time). In 1963 Jackson presented queueing networks with arrival process that can depend on the state of the system and closed queueing networks with exponential servers. In 1960's J.F.Ch. Kingman introduced algebra of queues and heavy traffic analysis of queueing systems. W.J. Gordon and G.F. Newell introduced closed queueing networks in 1967. The model of C.E. Skinner occurred in the same year. In 1968 M. Mandelbaum and B. Avi-Itzhak introduced the concept of Fork-Join systems. In 70's researchers were interested in use of queueing theory for computer performance evaluation. In 1973 J.P. Buzen proposed the convolution algorithm to computation of the normalization constant. In 1975 multi-class queueing networks occurred, especially BCMP networks, created by F. Baskett, K.M. Chandy, R.R. Muntz and F.G. Palacios. A special case is network presented by F.P. Kelly, in which jobs belong to different types and have exponential service-time distribution. Each type has a Poisson arrival process and a fixed route through the network. In 1977 decomposition method was introduced by P.J. Courtois. M.I. Reiman described the type of queueing networks called a generalized Jackson network in 1978. In this network job interarrival and service times are not required to be exponentially distributed. For analysis of closed queueing networks, S. Lavenberg and M. Reiser developed the Mean Value Analysis algorithm in 1980. In 1986 S. Fdida introduced representatives networks. In 1991 E. Gelenbe introduced the concept of positive and negative customers and G-networks. In recent years the progress of queueing theory use for software modelling is noticed. Another new important application has also emerged. In 1996 Y. Liu proposed the use of queueing theory in modelling of elementary mental process.

3. The Markovian queueing systems

In this Section we analyze the models with exponential service times, in which the arrival process is Poisson [2–6]. The mean arrival rate (per unit of time) at each system is denoted by λ and it is the reciprocal of the average interarrival time. The parameter of the service time (μ) is expressed as the reciprocal of the mean service time. Traffic intensity ρ is the ratio of arrival λ to service rate μ. From the equilibrium probabilities we can derive expressions for the mean number of customers in the system and the mean time spent in the system.

Kendall's classification of queueing system is a standard notation. It exists in several modifications. According to one of them (proposed by A.M. Lee) the service system is described by the notation *X/Y/m/D/L*, where A indicates the distribution of intervals between arrivals, B denotes the distribution of service duration, m is the number of servers, D is the queueing discipline that represents the way the queue is organized, and L represents the maximum total number of customers in a system (capacity system) [5]. In Markovian systems, the symbol M is used for A and B that denotes exponential distribution and Poisson arrival process.

3.1. The M/M/m/-m loss system. In this system arriving customer is served if at least one server is available. When all servers are occupied the newly arriving customer departs the queueing systems without being served. These customers are lost. The steady-state probability of the system being empty is in the form:

$$\pi_0 = \frac{1}{\sum\limits_{k=0}^{m} \frac{\rho^k}{k!}}. \tag{1}$$

The steady-state probability of k jobs in the system is as follows:

$$\pi_k = \frac{\frac{\rho^k}{k!}}{\sum\limits_{k=0}^{m} \frac{\rho^k}{k!}}. \tag{2}$$

The steady-state probability that the newly arriving customers are lost:

$$\pi_l = \pi_m = \frac{\frac{\rho^m}{m!}}{\sum\limits_{k=0}^{m} \frac{\rho^k}{k!}}. \tag{3}$$

We obtain the mean number of jobs in the system from:

$$\overline{K} = \rho(1 - \pi_l). \tag{4}$$

3.2. The M/M/m/FIFO/∞ system with infinite queueing space. In this system we consider unlimited waiting room and unlimited waiting time. Customers are served in order of arrival (first in, first out). If all servers are busy, the newly arriving customer is waiting in a queue. If not all servers are busy, the waiting time is equal to zero. The condition of system stability is $\lambda < m\mu$ (the condition for ergodicity). The steady-state probability of k jobs in the system is given by:

$$\pi_k = \begin{cases} \pi_0 \dfrac{\rho^k}{k!}, & 0 \le k \le m-1, \quad \rho < m, \\ \pi_0 \dfrac{\rho^k}{m! m^{k-m}}, & k \ge m, \quad \rho < m, \end{cases} \tag{5}$$

with the steady-state probability of no jobs in systems π_0:

$$\pi_0 = \frac{1}{\sum\limits_{k=0}^{m-1} \frac{\rho^k}{k!} + \frac{\rho^m}{(m-1)!(m-\rho)}}. \tag{6}$$

The mean number of jobs:

$$\overline{K} = \rho + \frac{\rho^{m+1}}{(m-\rho)^2 (m-1)!} \pi_0 \tag{7}$$

and the mean queue length is as follows:

$$\overline{Q} = \frac{\rho^{m+1}}{(m-\rho)^2 (m-1)!} \pi_0. \tag{8}$$

Applying Little's law the mean waiting time $\overline{W}$ and the mean response time are given by Eq. (9):

$$\begin{aligned} \overline{W} &= \frac{\overline{Q}}{\lambda}, \\ \overline{T} &= \frac{\overline{K}}{\lambda}. \end{aligned} \tag{9}$$

3.3. The M/M/∞ system with infinite number of servers. In M/M/∞ system new arriving jobs do not have to wait in a queue and they are immediately served. Servers are available for each arriving customer. The steady-state probability of the system being empty can be derived from:

$$\pi_0 = \frac{1}{1 + \sum\limits_{k=1}^{\infty} \left(\frac{\lambda}{\mu}\right)^k \cdot \frac{1}{k!}} = e^{-\frac{\lambda}{\mu}}. \tag{10}$$

The steady-state probability of k jobs in the system is given by:

$$\pi_k = \pi_0 \prod_{i=0}^{k-1} \frac{\lambda}{(i+1)\mu} = \pi_0 \left(\frac{\lambda}{\mu}\right)^k \frac{1}{k!} = \frac{\rho^k}{k!} e^{-\rho}. \tag{11}$$

The expressions for the mean number of jobs in systems and the mean response time are obtained using Eqs. (10) and (11):

$$\begin{aligned} \overline{K} &= \sum_{k=1}^{\infty} k \frac{\rho^k}{k!} e^{-\rho} = e^{-\rho} \sum_{k=1}^{\infty} \rho \frac{\rho^{k-1}}{(k-1)!} = \rho, \\ \overline{T} &= \frac{\overline{K}}{\lambda} = \frac{1}{\mu}. \end{aligned} \tag{12}$$

3.4. The M/M/m/FIFO/m+N system with finite capacity. In this system the maximum number of customers amounts to $m + N$, so there is a limited waiting room (N). If newly arriving customers find more than $m + N$ customers in systems, they are lost. The steady-state probability of no jobs in systems is given by:

$$\begin{aligned} \pi_0 &= \left\{ \sum_{k=0}^{m-1} \frac{\rho^k}{k!} + \frac{\rho^m}{m!} (N+1) \right\}^{-1}, \quad \rho = m, \\ \pi_0 &= \left\{ \sum_{k=0}^{m-1} \frac{\rho^k}{k!} + \frac{\rho^m}{m!} \cdot \frac{1 - \left(\frac{\rho}{m}\right)^{N+1}}{1 - \frac{\rho}{m}} (N+1) \right\}^{-1}, \quad \rho \neq m. \end{aligned} \tag{13}$$

Using Eq. (14) we obtain the steady-state probability that the newly arriving customers are lost:

$$\pi_l = \pi_{m+N} = \frac{\rho^{m+N}}{m^N m!} \cdot \pi_0. \tag{14}$$

The mean number of jobs in system is given by:

$$\begin{aligned} \overline{K} &= \frac{m^m}{m!} \cdot \frac{N(N+1)}{2} \cdot \pi_0 + \rho(1 - \pi_l) = \\ &= \overline{Q} + \rho(1 - \pi_l), \quad \rho = m, \\ \overline{K} &= \frac{\rho^{m+1}}{(m-1)!} \cdot \frac{1 - \left(\frac{\rho}{m}\right)^N \left[N\left(1 - \frac{\rho}{m}\right) + 1\right]}{(m-\rho)^2} \cdot \pi_0 + \\ &+ \rho(1 - \pi_l) = \overline{Q} + \rho(1 - \pi_l), \quad \rho \neq m. \end{aligned} \tag{15}$$

From Eq. (15) the mean response time and mean waiting time by Little's law can be easily derived.

3.5. The M/M/m/FIFO/∞ queueing system with impatient customers. In this system we consider a m-server queueing system with unlimited waiting room with FIFO queueing discipline and limited waiting time in the queue. Each job arriving to the system has its own maximal waiting time T_w. This time is assumed to be with an exponential distribution with parameter δ. If the time which a job would have to wait for accessing a server exceeds T_w, then it departs from the system after time T_w. Note that if $\delta \to 0$, then we have the case of unlimited waiting time. In case of all busy servers and $\delta \to \infty$ we have a system with losses. The stationary probability of no jobs in a system is given by:

$$\pi_0 = \left[\sum_{k=0}^{m} \frac{\rho^k}{k!} + \frac{\rho^m}{m!} \sum_{r=1}^{\infty} \frac{\rho^r}{\prod\limits_{n=1}^{r} \left(m + n\frac{\delta}{\mu}\right)} \right]^{-1}. \tag{16}$$

From Eq. (17) we obtain the probability that jobs will be lost because of exceeding the time limit:

$$\pi_l = \left(\frac{\delta}{\lambda} \cdot \frac{\rho^m}{m!} \cdot \sum_{r=1}^{\infty} \frac{r\rho^r}{\prod\limits_{n=1}^{r} \left(m + n\frac{\delta}{\mu}\right)} \right) \cdot \pi_0. \tag{17}$$

The mean numbers of jobs in queue and in systems are given by:

$$\begin{aligned} \overline{Q} &= \frac{\frac{\rho^m}{m!} \sum\limits_{r=1}^{\infty} \frac{r\rho^r}{\prod\limits_{n=1}^{r} \left(m + n\frac{\delta}{\lambda}\right)}}{\sum\limits_{k=0}^{m} \frac{\rho^k}{k!} + \frac{\rho^m}{m!} \sum\limits_{r=1}^{\infty} \frac{\rho^r}{\prod\limits_{n=1}^{r} \left(m + n\frac{\delta}{\lambda}\right)}}, \\ \overline{K} &= \overline{Q} + \rho(1 - \pi_l). \end{aligned} \tag{18}$$

3.6. The M/M/m/FIFO/N/F closed queueing system with finite population of N jobs. In some cases we have a finite population of jobs. These types of systems are known as closed systems. In model M/M/m/FIFO/N/F we assume a finite population of N customers, so the total number of jobs in a system is no more than N. This system has m servers. If the total number of customers is no more than number of

servers the customers are served without waiting in queue. The steady-state probability of no jobs in systems is given by:

$$\pi_0 = \left[\sum_{i=0}^{m} \frac{N!}{i!(N-i)!}\rho^i + \sum_{j=m+1}^{N} \frac{N!}{m!(N-j)!m^{j-m}}\rho^j \right]^{-1}. \tag{19}$$

The probability of k jobs in the system is obtained from:

$$\begin{aligned} \pi_k &= \frac{N!}{k!(N-k)!}\rho^k \cdot \pi_0, \quad 1 \le k \le m, \\ \pi_k &= \frac{N!}{m!(N-k)!m^{k-m}}\rho^k \cdot \pi_0, \quad m < k \le N. \end{aligned} \tag{20}$$

We may calculate the mean number of jobs in a system and in a queue from the following expressions:

$$\begin{aligned} \overline{K} &= \pi_0 N! \left[\sum_{i=0}^{m} \frac{i}{i!(N-i)!}\rho^i + \sum_{j=m+1}^{N} \frac{j}{m!m^{j-m}(N-j)!}\rho^j \right], \\ \overline{Q} &= \pi_0 N! \sum_{j=m+1}^{N} \frac{j}{m!m^{j-m}(N-j)!}\rho^j. \end{aligned} \tag{21}$$

Each customer arrives with parameter λ, hence the mean response time is given by:

$$\overline{T} = \frac{\overline{K}}{\lambda(N-\overline{K})}. \tag{22}$$

3.7. The queueing systems with individual service. In this chapter a class of queueing systems with individual service studied in Department of Automatics AGH-UST is presented [5, 7]. The servers in these systems can serve at different speeds. Hence we have different traffic intensity of servers. First we analyze the case of infinite queueing space, then the system with finite capacity.

The M/M/m/FIFO/∞ queueing system with individual service and uniform flux of arrivals. The probability of k jobs in the system has the following form:

$$\pi_k = \begin{cases} \pi_0 \dfrac{SK_m^k}{k! \begin{pmatrix} m \\ k \end{pmatrix}^{k-m}}, & 0 \le k < m, \\ \pi_0 \dfrac{SK_m^k}{k! \left(SK_m^{m-1}\right)^{k-m}}, & k \ge m, \end{cases} \tag{23}$$

where: SK_m^k is a sum of k-combination (without repetition) from a set with m-elements $\{\rho_1, ..., \rho_m)$.

The probability π_0 follows from normalization, yields:

$$\begin{aligned} \pi_0 &= \frac{1}{1 + \sum_{k=0}^{m-1} \dfrac{SK_m^k}{k! \begin{pmatrix} m \\ k \end{pmatrix}} + \dfrac{SK_m^m \cdot SK_m^{m-1}}{m!\left(SK_m^{m-1} - SK_m^m\right)}}, \\ &\frac{SK_m^m}{SK_m^{m-1}} < 1. \end{aligned} \tag{24}$$

The mean number of busy servers is given by:

$$\overline{m}_0 = \frac{mSK_m^m}{SK_m^{m-1}} = \frac{m\lambda}{\mu_1 + ... + \mu_m}. \tag{25}$$

The mean length of queue may be found by solving the equation:

$$\begin{aligned} \overline{Q} &= \frac{(SK_m^m)^2}{m!SK_m^{m-1}} \cdot \frac{1}{\left(1 - \dfrac{SK_m^m}{SK_m^{m-1}}\right)^2}\pi_0 \\ &= \frac{SK_m^{m-1}}{m!\left(\dfrac{SK_m^{m-1}}{SK_m^m} - 1\right)^2}\pi_0. \end{aligned} \tag{26}$$

The mean number of customers in the system we can easily obtain from:

$$\overline{K} = \frac{SK_m^{m-1}}{m!\left(\dfrac{SK_m^{m-1}}{SK_m^m} - 1\right)^2}\pi_0 + \frac{mSK_m^m}{SK_m^{m-1}}. \tag{27}$$

Together with Little's law we retrieve formula for the mean waiting time:

$$\overline{W} = \frac{SK_m^{m-1}}{m!\lambda} \cdot \frac{1}{\left(\dfrac{SK_m^{m-1}}{SK_m^m} - 1\right)^2}\pi_0. \tag{28}$$

Similarly, the mean response time is given by:

$$\overline{T} = \frac{1}{\lambda}\left[\frac{SK_m^{m-1}}{m!\left(\dfrac{SK_m^{m-1}}{SK_m^m} - 1\right)^2}\pi_0 + \frac{mSK_m^m}{SK_m^{m-1}} \right]. \tag{29}$$

The M/M/m/FIFO/m+N queueing system with individual service and uniform flux of arrivals. From (30) we get the probability, that there are i jobs in the system:

$$\begin{aligned} \pi_i &= \frac{SK_m^i}{i! \begin{pmatrix} m \\ i \end{pmatrix}}\pi_0, \quad 1 \le i < m, \\ \pi_i &= \frac{(SK_m^m)^{i-m+1}}{m!\left(SK_m^{m-1}\right)^{i-m}}\pi_0, \quad m \le i < m+N. \end{aligned} \tag{30}$$

The probabilities π_0 follows from normalization, yielding:

$$\pi_0 = \left[1 + \sum_{i=1}^{m-1} \frac{SK_m^i}{i! \begin{pmatrix} m \\ i \end{pmatrix}} + \sum_{i=m}^{m+N} \frac{SK_m^m}{m!}(N+1)\right]^{-1}, \quad \frac{SK_m^m}{SK_m^{m-1}} = 1,$$

$$\pi_0 = \left[1 + \sum_{i=1}^{m-1} \frac{SK_m^i}{i! \begin{pmatrix} m \\ i \end{pmatrix}} + \frac{SK_m^m}{m!}\frac{1-q^{N+1}}{1-q}\right]^{-1}, \quad q = \frac{SK_m^m}{SK_m^{m-1}} \neq 1. \tag{31}$$

The probability that a job is lost is an important quantity, obtained from:

$$\pi_l = \pi_{m+N} = \frac{\left(SK_m^m\right)^{N+1}}{m!\left(SK_m^{m-1}\right)^N}\pi_0, \quad \text{for} \quad m \leq i \leq m+N. \tag{32}$$

We can obtain the mean number of busy servers by the following equality:

$$\overline{m}_0 = \frac{mSK_m^m}{SK_m^{m-1}}\left(1 - \frac{\left(SK_m^m\right)^{N+1}}{m!\left(SK_m^{m-1}\right)^N}\pi_0\right). \tag{33}$$

The mean queue length can be obtained from:

$$\overline{Q} = \frac{SK_m^m}{2m!}N(N+1)\pi_0, \quad \text{for} \quad \frac{SK_m^m}{SK_m^{m-1}} = q = 1,$$

$$\overline{Q} = \frac{\left(SK_m^m\right)^2}{m!}SK_m^{m-1}\frac{1-q^N\left[1-N\left(1-q\right)\right]}{\left(SK_m^{m-1}-SK_m^m\right)^2}\pi_0, \quad \text{for} \quad q \neq 1. \tag{34}$$

The formulas for the mean response and waiting time can be found by using Little's law.

4. Non-Markovian systems

In non-Markovian systems we permit either the service time or the input stream intensity to be nonexponentially distributed. For the queueing systems with a Poisson arrival process and general service time distribution (M/G/1/FIFO/∞), we have Pollaczek-Khinchin formula (also known as Kendall formula) to calculate the mean number of jobs and the mean response time in a system:

$$\overline{K} = \lambda E(s) + \frac{\lambda^2 E(s^2)}{2[1-\lambda E(s)]}, \quad \overline{T} = E(s) + \frac{\lambda E(s^2)}{2[1-\lambda E(s)]}, \tag{35}$$

where $E(s)$ and $E(s^2)$ are the first two moments of the service time distribution.

For the G/G/1/FIFO/∞ queueing system and non-Markovian systems with m servers there are approaches, such as the well-known Allen-Cunneen approximation or methods of upper and lower bounds (the Kingman bounds or the Marchal bounds). In the context of queues the approach of Cox can be also useful. Under the assumption that the service time distribution has a rational Laplace transform, a Cox distribution can be used to approximate the general service times. Many approximation methods are mentioned in the literature. Detailed description of non-Markovian systems can be found in [2, 6, 8]. We present only three types of these systems in this chapter.

4.1. The queueing systems with the service zone. The queueing systems with hyperexponential or Cox service time distributions are the examples of the queueing systems with the service zone (the service facility) [2,8]. Only one phase can be occupied by a customer at any time, hence only one customer can be in the service zone. Hyperexponential distribution H_R contains R phases (stages) connected in parallel, each with exponentially distributed times. These phases form the service zone. An arriving customer chooses one phase with given probability. Because, a Cox distribution can be used to obtain performance measures of such systems, here we present the M/Cox$_L$/1/FIFO/∞ system. The Cox$_L$distribution consists of L phases connected in series, each with exponential service time distribution and rate μ_l ($l = 1, \ldots, L$). A job can leave this system after service in lth phase with probability b_l or it can proceed into the next phase with probability a_l (Fig. 1).

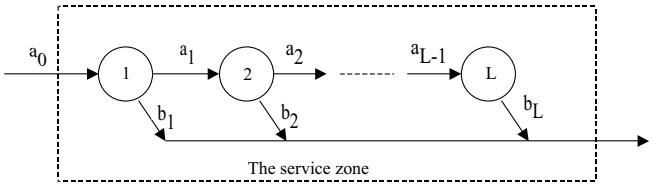

Fig. 1. The service zone with Cox$_L$ distribution

The mean service time in the zone is given as:

$$\overline{T}_o = \sum_{i=1}^{L} \frac{\prod_{j=1}^{l-1} a_j}{\mu_i}. \tag{36}$$

The condition for ergodicity may be written as:

$$\sum_{i=1}^{L} \frac{\prod_{j=1}^{l-1} a_j}{\mu_i} \cdot \lambda < 1. \tag{37}$$

The Laplace transform of the service time density has the following form:

$$F_{Cox_L}(s) = \sum_{l=1}^{L}\left(\left(\prod_{j=1}^{l-1} a_j\right) b_l \prod_{i=1}^{l} \frac{\mu_i}{\mu_i + s}\right). \tag{38}$$

For M/Cox$_2$/1/FIFO/∞ system we obtain directly from Eq. (4.4) the Laplace transform and the first two moments as follows:

$$F_{Cox_2}(s) = b_1\frac{\mu_1}{\mu_1+s} + a_1 b_2 \frac{\mu_1}{\mu_1+s}\frac{\mu_2}{\mu_2+s},$$
$$E(s) = \frac{1}{\mu_1} + \frac{a}{\mu_2}, \tag{39}$$
$$E(s^2) = \frac{2}{\mu_1^2} + a\left(\frac{2}{\mu_1\mu_2} + \frac{2}{\mu_2^2}\right).$$

The mean number of jobs and the mean response time can be found using Eq.(35).

4.2. The queueing systems with getting tired server. In the queueing theory the Weibull distribution can be used to modelling human activity and objects with growing failure rate [9, 10]. This distribution is a well-known example of a heavy tailed distribution. The density function of Weibull distribution with parameters k, v, ε is given by:

$$f(t) = \begin{cases} \dfrac{k}{v-\varepsilon}\left(\dfrac{t-\varepsilon}{v-\varepsilon}\right)^{k-1} e^{-\left(\frac{t-\varepsilon}{v-\varepsilon}\right)^k} & \text{for} \quad t \geq \varepsilon, \\ 0 & \text{for} \quad t < \varepsilon, \end{cases} \tag{40}$$

for $v-\varepsilon > 0$ and $k > 0$.

This implies that for $\varepsilon = 0$ and $k = 1$ we obtain exponential distribution. The Laplace transform of density function has the form:

$$L(s) = \sum_{n=0}^{\infty}(-1)^n \frac{s^n}{n!} v^n \int_0^{\infty} u^{\frac{n}{k}} e^{-u} du = \tag{41}$$
$$= \sum_{n=0}^{\infty}(-1)^n \frac{s^n}{n!} v^n \Gamma\left(\frac{n}{k}+1\right),$$

for

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t} dt, \quad x > 0.$$

Hence the expressions for the first and second moments are given as:

$$M_1 = E(s) = v\Gamma\left(\frac{1}{k}+1\right),$$
$$M_2 = E(s^2) = v^2\Gamma\left(\frac{2}{k}+1\right). \tag{42}$$

These moments are of importance in the analysis of queueing systems. There are a few approaches of the Weibull distribution. Depending on the value of the coefficient of variation, different Cox models are used to approximate the Weibull distributed service time [11, 12]. For the M/W$_k$/1/FIFO/∞ system, the steady-state probabilities may be solved using the method of the imbedded Markov chains for the M/G/1/FIFO/∞ system [6, 8]. For the M/W$_k$/1/FIFO/∞ system the state probability with no jobs is given by:

$$p_0 = 1 - \lambda v\Gamma\left(\frac{1}{k}+1\right). \tag{43}$$

The mean number of jobs and the mean response time can be found by using Pollaczek-Khinchin formula, which may be written as:

$$\overline{K} = \lambda v\Gamma\left(\frac{1}{k}+1\right) + \frac{\lambda^2 v^2 \Gamma\left(\frac{2}{k}+1\right)}{2\left(1-\lambda v\Gamma\left(\frac{1}{k}+1\right)\right)},$$
$$\overline{T} = v\Gamma\left(\frac{1}{k}+1\right) + \frac{\lambda v^2 \Gamma\left(\frac{2}{k}+1\right)}{2\left(1-\lambda v\Gamma\left(\frac{1}{k}+1\right)\right)}, \tag{44}$$

where the condition for ergodicity becomes:

$$\lambda v\Gamma\left(\frac{1}{k}+1\right) < 1.$$

The mean remaining service time of a customer in service given by Eq. (45) for M/W$_k$/1/FIFO/∞ with $k > 1$ is shorter than the mean remaining service time for M/M/1/FIFO/∞ with $1/v$.

$$\overline{T}_d = \frac{v\Gamma\left(\frac{2}{k}+1\right)}{2\Gamma\left(\frac{1}{k}+1\right)}. \tag{45}$$

4.3. The oscillating systems. The oscillating queueing systems solve the continuous running of the server without the growing length of the queue. The goal of the systems is to keep the length of the queue within maximum and minimum threshold bounds ($Q_{\min}$, $Q_{\max}$). In the case of the G/G-G/1/FIFO/∞ queueing system, the service time distribution oscillates between two values dependent on the number of jobs in the queue. At the beginning, a server (service channel) has less efficiency and the length of queue tends to be longer. When this queue reaches a critical level $Q_{\max}$, the server changes its efficiency for a higher one, thus the queue decreases. This higher efficiency is kept to the time point, when the length of this queue reaches satisfactory low value $Q_{\min}$. Then the server works again with less efficiency. The arriving jobs are served with FIFO service strategy. In some situations we have oscillating systems, called the G-G/G/1/FIFO/∞ queueing systems, with the input stream intensity switching and fixed performance of the server. At the beginning the queue length tends to increase. After the length of this queue reaches a critical level $Q_{\max}$, the input stream intensity decreases. When the queue length reaches the level $Q_{\min}$, the input stream intensity is restored to initial value, and so on.

The general model of oscillating system, the G-G/G-G/1/FIFO/∞ queueing system, combines the switching input stream and changing the efficiency of the server (Fig. 2.). In order to analyse of the system operation, let us assume the

G-G/G-G/1/FIFO/∞ system as two G/G/1/FIFO/∞ queueing systems. At the moment $t = 0$, when the queue length is less than $Q_{\max}$, first system starts the operation. When the queue reaches a critical level $Q_{\max}$, this system remains in action as long the served job is being served, then it finishes the operation and the second system, in turn, starts the operation and continues the service of the queue. From this point of time the jobs arrive with another intensity until the moment, when the queue reaches $Q_{\min}$. Then the second system stops the operation and the first system starts the service. The input stream intensity is restored to initial value at the moment, when new jobs arrive after the change of the systems. This switching is known as the first type switching. In case of the second type switching the job service is interrupted, when the queue reaches a critical level $Q_{\max}$.

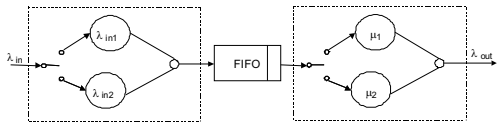

Fig. 2. The G-G/G-G/1/FIFO/∞ queueing model

The articles [13, 14] deal with analysis of oscillating systems.

5. The queueing network

The queueing networks consist of several connected systems. In an open queueing network, customers enter the network from outside, receive service at systems and leave the network. In a closed network the number of customers is constant. If a new customer enters the network exactly when one customer departs, we can model this situation as closed queueing network.

Our main interest is to present the Jackson, Kelly's and BCMP networks. In these networks the steady-state probabilities have product-form solutions, dependent on the first moment of the service time distribution. We assume R customer classes and N systems in queueing network. The rate at which customers of the rth class leave the ith system (known as throughput λ_{ir}) and the mean number of visits e_{ir} can be determined from the routing probabilities p. For an open network we have:

$$\begin{aligned} \lambda_{ir} &= \lambda_{0,ir} + \sum_{j=1}^{N}\sum_{s=1}^{R} \lambda_{js} p_{js,ir}, \\ e_{ir} &= p_{0,ir} + \sum_{j=1}^{N}\sum_{s=1}^{R} e_{js} p_{js,ir}, \end{aligned} \tag{46}$$

and for a closed networks the equation reduces to:

$$\begin{aligned} \lambda_{ir} &= \sum_{j=1}^{N}\sum_{s=1}^{R} \lambda_{js} p_{js,ir}, \\ e_{ir} &= \sum_{j=1}^{N}\sum_{s=1}^{R} e_{js} p_{js,ir}. \end{aligned} \tag{47}$$

5.1. Jackson networks. Jackson network with only one customer class and unlimited overall number of jobs is the simplest form of queueing networks [2, 15]. The model assumes that the external arrival pattern is identified by a Poisson arrival process. All systems have one or more servers with exponential service times. The service rates can depend on the number of customers at the system. In all systems customers are served in order of arrival (FIFO). The systems in the network can be considered as independent M/M/m/FIFO/∞ queueing systems.

According to Jackson's theorem, if all systems hold the ergodicity condition, the solution for the steady-state probabilities has a product-form and can be expressed as the product of the state probabilities of each system:

$$\pi(k) = \prod_{i=1}^{N} \pi_i(k_i), \tag{48}$$

where $\pi_i(k_i)$ is obtained from Eqs. (5) and (6).

5.2. Gordon-Newell networks. These networks, also known as closed Jackson's networks, fulfil the assumptions of Jackson's networks, except one: customer can neither enter nor leave the network. A fixed number of jobs always circulate in this type of queueing network [2, 15].

According to Gordon-Newell theorem, the probability for each network state is given by the following expression:

$$\pi(k) = \frac{1}{G(K)} \prod_{i=1}^{N} F_i(k_i), \tag{49}$$

where $G(K)$ is the normalization constant and $F_i(k_i)$ is the function describing the state probabilities $\pi_i(k_i)$ of ith system.

5.3. Kelly's network. Another case of queueing networks is the network of Kelly with different classes of customers. Each type has a Poisson arrival process and a fixed route in the network. Customers served at each system have exponential service time distribution. Each system may serve several different customer classes. All systems have infinite capacity [15].

5.4. The BCMP queueing networks. The BCMP queueing network is a multi-class network discussed by Baskett, Chandy, Muntz and Palacios. These networks include different class of jobs, different queueing discipline and generally distributed service times. Routes through the network may depend on the job type and the customer can change its class while passing through the network [2, 15, 16].

In this network there are four types of systems:

- Type 1: system with multiple servers, the service times are exponentially distributed and for different customer classes must be identical, the service discipline is FIFO;
- Type 2: system with one server, different customer classes have different general service time distribution with a rational Laplace transform, the service discipline is PS (processor sharing);

– Type 3: system with an ample number of servers (*infinite server*) and the mean service time for job classes can be different, the service times of the customers of different classes must have a rational Laplace transform;
– Type 4: system with one server, different customer classes have different general service time distribution with a rational Laplace transform, the service discipline is LIFO-PR (last in first out with preemptive).

To analyse queueing networks and the determined performance measure we use two BCMP theorems for open and closed networks [2]. Let us assume that we have queueing network with load-independent service and arrival rates, fulfilling the assumption of the BCMP networks.

Theorem BCMP1 for open queueing network: for open queueing network, the steady-state probability of the network can be expressed as the product of state probabilities of the individual systems:

$$\pi(k_1,...,k_N)=\prod_{i=1}^{N}\pi_i(k_i), \tag{50}$$

where

$$\pi_i(k_i)=\begin{cases}(1-\rho_i)\rho_i^{k_i}, & \text{Type 1}(m_i=1),2,4,\\ e^{-\rho_i}\dfrac{\rho_i^{k_i}}{k_i!}, & \text{Type 3},\\ \pi_k, & \text{Type 1}(m_i>1),\end{cases}$$

$$k_i=\sum_{i=1}^{R}k_{ir},\qquad \rho_i=\sum_{r=1}^{R}\rho_{ir},$$

$$\rho_{ir}=\begin{cases}\lambda_r\dfrac{e_{ir}}{m_i\mu_i}, & \text{Type 1}(m_i\geq 1),\\ \lambda_r\dfrac{e_{ir}}{\mu_{ir}}, & \text{Type 2,3,4.}\end{cases}$$

For Type 1 with more than one server π_k is given by:

$$\pi_k=\begin{cases}\dfrac{1}{\sum\limits_{k=0}^{m-1}\dfrac{(m\rho)^k}{k!}+\dfrac{(m\rho)^m}{m!(1-\rho)}}\cdot\dfrac{(m\rho)^k}{k!}, & 0\leq k\leq m,\\ \dfrac{1}{\sum\limits_{k=0}^{m-1}\dfrac{(m\rho)^k}{k!}+\dfrac{(m\rho)^m}{m!(1-\rho)}}\cdot\dfrac{m^m\rho^k}{m!}, & k\geq m,\end{cases} \tag{51}$$

Theorem BCMP2 for closed queueing network: for closed queueing network, the steady-state probability of the network can be expressed as:

$$\pi(\mathbf{S}_1,...,\mathbf{S}_N)=\frac{1}{G(\mathbf{K})}\prod_{i=1}^{N}F_i(\mathbf{S}_i). \tag{52}$$

The normalization constant is given by:

$$G(\mathbf{K})=\sum_{\sum_{i=1}^{N}\mathbf{S}_i=\mathbf{K}}\prod_{i=1}^{N}F_i(\mathbf{S}_i), \tag{53}$$

where:

$$F_i(\mathbf{S}_i)=\begin{cases}k_i!\dfrac{1}{\beta_i(k_i)}\left(\dfrac{1}{\mu_i}\right)^{k_i}\prod\limits_{r=1}^{R}\dfrac{1}{k_{ir}!}e_{ir}^{k_{ir}}, & \text{Type 1},\\ k_i!\prod\limits_{r=1}^{R}\dfrac{1}{k_{ir}!}\left(\dfrac{e_{ir}}{\mu_{ir}}\right)^{k_{ir}}, & \text{Type 2,4},\\ \prod\limits_{r=1}^{R}\dfrac{1}{k_{ir}!}\left(\dfrac{e_{ir}}{\mu_{ir}}\right)^{k_{ir}}, & \text{Type 3}.\end{cases}$$

The function $\beta_i(k_i)$ is given by:

$$\beta_i(k_i)=\begin{cases}k_i! & \text{for}\quad k_i\leq m_i,\\ m_i!m_i^{k_i-m_i} & \text{for}\quad k_i>m_i,\\ 1 & \text{for}\quad m_i=1.\end{cases} \tag{54}$$

A closed multiclass queueing network is much more difficult to solve. In order to compute $G(\mathbf{K})$ we must consider all states of the network. It is a complicated procedure, especially for the large networks. Therefore, there are efficient algorithms to analyse closed queueing networks. The convolution algorithm of Buzen is one of them [2, 3, 17]. The second very important algorithm for product-form networks is the Mean Value Analysis (MVA) [2, 18]. This is an iterative algorithm developed to obtaining the mean values of the performance measures without calculating the normalization constant. The basic equation of MVA relates the mean response time at system with K jobs and the mean number of jobs at this system, if a network has K-1 jobs. Unfortunately, the MVA requires a lot of computation time. In the literature there are also approximation methods based on the MVA, to calculate the performance measures, for example Bard-Schweitzer approximation (BS) used for systems with single server [2] or Self-Correcting Approximation Technique (SCAT) [2, 19].

In other algorithm called the Summation Method (SUM), the mean number of customers of rth class at ith system is a function of throughput of this system λ_{ir} [2]. For the BCMP networks this function has the following expression:

$$f_{ir}(\lambda_{ir})=\overline{K}_{ir}=\begin{cases}\dfrac{\rho_{ir}}{1-\dfrac{K-1}{K}\rho_i}, & \text{Type 1, 2, 4 } (m_i=1),\\ \dfrac{\lambda_{ir}}{\mu_{ir}}, & \text{Type 3},\end{cases} \tag{55}$$

and for Type 1 with multi servers ($m_i>1$):

$$f_{ir}(\lambda_{ir})=\overline{K}_{ir}=m_i\rho_{ir}+\frac{\rho_{ir}}{1-\dfrac{K-m_i-1}{K-m_i}\rho_i}\cdot P_{m_i},$$

with the waiting probability:

$$P_{mi}=\frac{(m_i\rho_i)^{m_i}}{m_i!(1-\rho_i)}\cdot\frac{1}{\sum\limits_{k_i=0}^{m_i-1}\dfrac{(m_i\rho_i)^{k_i}}{k_i!}+\dfrac{(m_i\rho_i)^{m_i}}{m_i!(1-\rho_i)}},$$

where: $\rho_i=\sum\limits_{r=1}^{R}\rho_{ir}$, $K=\sum\limits_{r=1}^{R}K_r$.

By summing over all these functions and with $\lambda_{ir} = \lambda_r e_{ir}$ we obtain the number of r-th job class:

$$\sum_{i=1}^{N} f_{ir}(\lambda_r e_{ir}) = K_r, \quad \text{for} \quad r = 1, ..., R.$$

An algorithm to determine λ_r is described in [2].

5.5. Fork-Join systems. The model of Fork-Join systems (Fig. 3) can be applied to parallel processing analysis [2, 20]. The job (programme) arriving to a fork-join queue splits (at the fork point) into N independent tasks that are simultaneously assigned to N processors. Each task requires a serve. At each processor tasks can belong to the different jobs. When a task completes execution, it will wait at the join queue until all its sibling tasks are served. A join mergers several tasks into a single job. A job is completed and departs the parallel resource after all of its tasks complete execution.

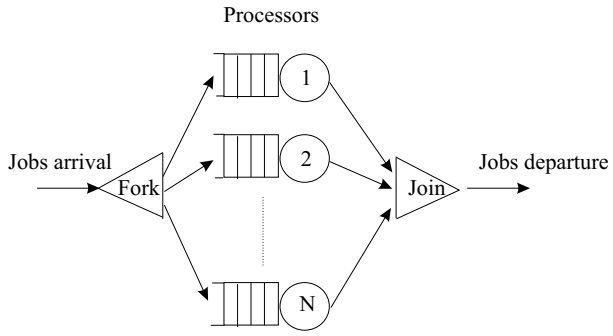

Fig. 3. Fork-Join model

The Fission-Fusion system, in which a job can leave system after N tasks are finished (there are all identical tasks), and the Split-Merge system with no processor queues (a new job is served when all the N tasks are finished) are special cases of basic Fork-Join systems.

It is important to compute performance measures of Fork-Join systems. These models have non-product form solutions but many methods are available for solving such models. The Flow Equivalent Server method (FES) is one of them, which can be described in a few steps [2]:

- A choice of one node and the short-circuit this node by making its mean service time zero and the calculation of throughputs along this short-circuit as a function of the number of jobs in the network.
- The construction of the equivalent network with chosen system and the FES node, the throughput through the short when there are k jobs circulating in that modified network corresponds to the service rate of the FES.
- The computation of the performance measures by any method for product-form network.

Therefore, in order to solve this system we consider the FES node with state-dependent service rate, which replaces the Fork-Join system. The service rates in the FES node are given by analysing the isolated short-circuited subnetwork for every number of tasks. The basic parameters of Fork-Join systems are: the speedup defined as the ratio of the mean response time in the system with N sequential tasks to the mean response time in the *Fork-Join* system with N processors, and synchronization overhead defined as the ratio of the tasks' mean time in the join queue to the mean response time of the *Fork-Join* system [2].

For the two-processor Fork-Join system, with the assumptions of a Poisson arrival process and exponential service time in each processor, the speedup is equal 4/3 and synchronization overhead is equal 3/2. These measures do not depend on the utilization of processors [2]. The two-processor Fork-Join system with service time distribution represented by exponential and Cox distribution was analysed in [21].

5.6. Queueing networks with blocking. In these networks the systems have finite buffer capacity, thus jobs that are routed to these systems may be blocked. Several types of blocking have been defined in the literature in order to model different behaviours of real systems. The most commonly used blocking mechanisms are the blocking after service, the blocking before service and the repetitive service blocking. The blocking after service means that the current system is blocked when a job completing service at this system attempts to enter destination system, which is full. This job is forced to wait in the server of current system until the destination system can be entered. Service at current system will be resumed as soon as a departure occurs from destination system. If several systems are blocked by the same destination system, the unblocking order of the blocked systems must be defined. In blocking before service a job determines its destination system before it starts receiving service at prior system. If destination system is full, the prior system is blocked and the service does not start. When the server at the destination becomes available, the job begins to receive service at prior system. The destination system of a blocked job does not change. Repetitive service blocking refers to the situation, when a job upon completion of its service at current system wishes to enter destination system that is full. This job is looped back into the current queue and it receives a new independent service according to the queue discipline. There are two subcategories that distinguish whether the job, after finishing its repeated service, chooses a new destination system independently of the one that it had selected before: random destination and fixed destination [22, 23].

The product-form solutions have been derived only under special constraints, for different blocking types. General the queueing networks with blocking do not have a product-form solution, but there are approximate analytical methods that have to be applied. Several papers have proposed various approximate methods [22, 23].

6. Area of applications

Queueing theory is commonly used to modelling service centres, to performance evaluation of computer systems, production and flexible manufacturing systems and communication networks. In this chapter, we present three applications.

6.1. Model of chemotherapy unit. As the first example of the queueing theory a chemotherapy unit of an oncological hospital is considered. Chemotherapy is one of the most common treatments for cancer. The type of chemotherapy treatment depends on many factors, mainly on the type and location of the disease. Usually chemotherapy drugs are given to patient in a day unit at the hospital but sometimes chemotherapy means a few days' stay at the hospital. Chemotherapy is usually given as several sessions of the treatment. Each cycle of chemotherapy consists of the treatment and the rest period of a few weeks. The number of cycles depends on how cancer responds to the chemotherapy. Chemotherapy unit can be presented as closed BCMP networks shown in Fig. 4, with known routing probabilities (presented close to arrows). We assume that the class switching is not allowed.

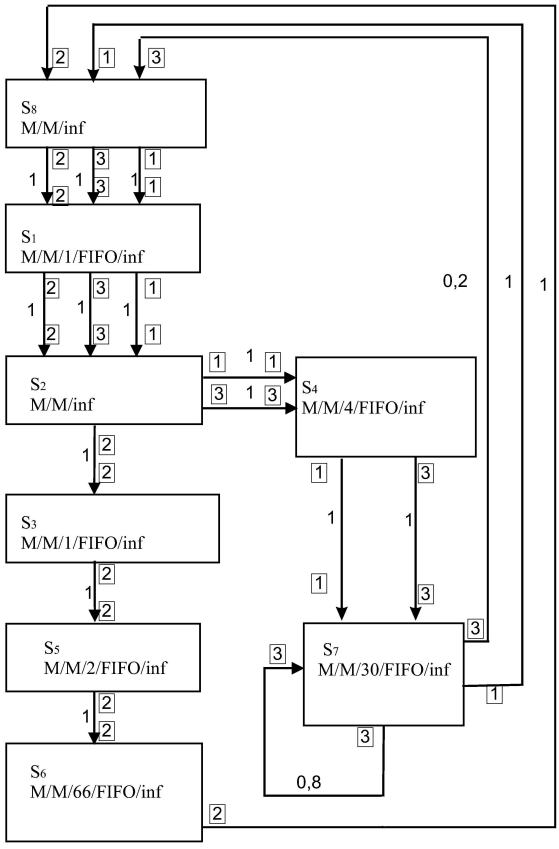

Fig. 4. The queueing network model of chemotherapy unit

We consider only three following classes of patients (marked in rectangles in Fig. 4):

- class "1" – 250 patients of the chemotherapy day unit (having chemotherapy once during one cycle),
- class "2" – 144 patients of a few days' chemotherapy,
- class "3" – 20 patients of the chemotherapy day unit (having chemotherapy five times during one cycle).

The model consists of eight systems, all with exponentially distributed service time:

- nurse's office S_1 (one nurse, $\mu_{11} = \mu_{12} = \mu_{13} = 67$),
- waiting room S_2 ($\mu_{21} = \mu_{22} = \mu_{23} = 8$),
- doctor's office of a few days' chemotherapy patients S_3 (one doctor, $\mu_{32} = 60$),
- doctor's office of a daily chemotherapy patients S_4 (four doctors, $\mu_{41} = \mu_{43} = 8.33$),
- sick-room S_5 (two nurses, $\mu_{52} = 12$),
- a few days' chemotherapy unit S_6 (66 beds, $\mu_{62} = 0.218$),
- the chemotherapy day unit S_7 (30 beds, $\mu_{71} = \mu_{73} = 1$),
- "home-waiting room" S_8 (infinite beds servers, service rate different for the classes of patients: $\mu_{81} = 0.092$, $\mu_{82} = 0.137$, $\mu_{83} = 0.053$): additional node introduced to model the whole cycle of chemotherapy.

Detailed description of chemotherapy unit is presented in [24].

To compute the mean waiting times of patients at each system we can use the summation method and Little's law. In Table 1, we have these paremeters.

Table 1
The mean waiting time in systems (in minutes)

	Class 1	Class 2	Class 3
S_1	28.4	28.4	28.4
S_2	0	0	0
S_3	0	5.9	0
S_4	40.5	0	40.5
S_5	0	38.2	0
S_6	0	44	0
S_7	56.3	0	56.3
S_8	0	0	0

6.2. Performance evaluation of the information systems. An approach for software modelling based on queueing network provides tools in modelling and performance analysis of the information systems, which are of importance in contemporary information society. Performance analysis should be included in the software development process. It can help to identify system bottlenecks and compare design alternatives. The information systems should give quick access to information resources, so the minimization search time (reduction time of delivering information to users) is a main problem.

The information systems existing in LANs can be modelled as the closed queueing networks. In the case of WANs they are modelling as the open queueing networks. The article [25] introduces a network of functional modules i.e. separable sets of software elements. This network is regarded as a queueing networks. The service centres form functional modules, customers correspond to user queries, which are divided into different classes. The queries forms one class if they have similar features, therefore detailed analysis of the algorithms for each query processing is not necessary. The subnetwork of functional modules is responsible for processing of queries from one or several classes. Different modules can process the queries of one class. There is a queue of waiting queries in each module. The network of functional modules has a product-form solution. All parameters can be evaluated by the multiclass BCMP networks.

In [25] a queueing network approach was applied to the evaluation of the response times for different system load conditions for selected information systems such as information searching system about scientific problems WINSWIP, the tourist agencies service system and searching library's catalogue system.

6.3. Modelling of human performance. The application of queueing theory to the modelling of human performance arouses researchers' interest. Therefore, we briefly present a queueing network architecture called Queueing Network-Model Human Processor (QN-MHP), which integrates the queueing network and human cognitive system. It has been proposed by R. Feyen and Y. Liu, and successfully used to modelling behaviour in real time [26–29]. The QN-MHP allows determining the reaction time, that is delays between stimulus presentation and response. Servers perform different procedural functions and represent different brain area. Information traversing the servers is regarded as the customers of queueing network, which could be processed in parallel or in series.

The QN-MHP consists of three subnetworks of queueing servers: perceptual, cognitive and motor. The general structure of QN-MHP is shown in Figure 5.

The first subnetwork includes some servers used to modelling of the visual, auditory and somatosensory systems, e.g. visual processing and location, sound location and processing. The cognitive subnetwork includes servers employed to modelling of a working memory and goal execution functions such as goal prioritization, performance monitoring and procedure selection. Servers included in the working memory contain visuospatial sketchpad, phonological loop and central executor. Subsystems for storage and retrieval are crucial to goal-directed behaviour. The motor subnetwork contains some servers and actuators, such as supplementary motor area, sensorimotor integration, motor sequencing, motor programming, hands, feet and mouth. Each actuator is limited to processing only one information entity at a time. Choosing different routes by customers leads to different processing times and to error occurrences. Traversing entities try to maximize processing speeds during learning processes of queueing network, which are proceeding in two levels: learning process of the individuals systems and self-organization of the queueing network. Long-term procedural memory and long-term declarative memory play important part in learning process regarding speed of motor program retrieving and speed of phonological judgements, visuomotor choices and mental calculations.

Each system of QN-MHP is a single-channel server, processes on entity at a time and services entities according to their joining the queue. All systems follow FIFO queue discipline. Some of them have restricted capacities, a few systems have large capacities to avoid queues. The groups of entities represent different stimuli that can be processed. Some systems could operate concurrently.

The correct modelling of QN-MHP seems to be a promising tool of modelling human behaviour in various situations, including systems disturbances.

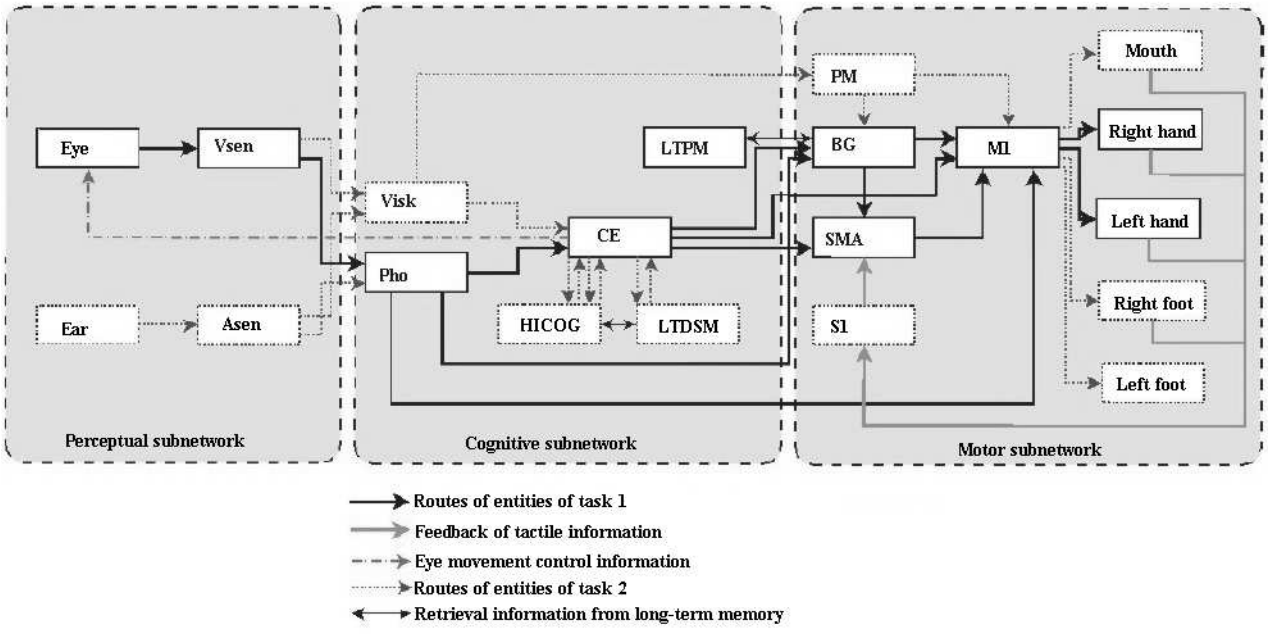

Fig. 5. The structure of the QN-MHP (according to Wu and Liu, 2004); Eye – visual sampling, Vsen – visual sensory memory, Ear – convert sound into neural signals, Asen – auditorial sensory memory, Visk – visuospatial sketchpad, Pho – phonological loop, CE – central executor, HICOG – high-cognitive function, LTDSM – long-term declarative memory, LTPM – long-term procedural memory, PM – select movement, BG – motor program retrieval, SMA – supplementary motor area, S1 – sending sensory information, M1 – primary motor cortex, Mouth, Hands and Feet – actuators

7. Summary

This paper gives a general look at the queueing theory. Presented queueing systems and networks offer a good tool for modelling complex system. They can be used to improve flow of customers, for the evaluation of utilization, throughput and response times. Queueing models are helpful in preparation of optimal decision on structures and service organizations from client and manager viewpoint and in study of methods, which allow calculating of basic characteristics of the service process.

REFERENCES

[1] S. Stidham, "Analysis, design and control of queueing systems", *Operations Research* 50 (1), 197–216 (2002).

[2] G. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi, *Queueing Networks and Markov Chains. Modelling and Performance Evaluation with Computer Science Applications*, John Wiley&Sons, Inc., London, 1998.

[3] T. Czachórski, *Queueing Models in Performance Evaluation of Computer Networks and Systems*, Jacek Skalmierski's Computer Workshop, Gliwice, 1999, (in Polish).

[4] B. Filipowicz, *Stochastic Models in Operation Research: Analysis and Synthesis of the Queueing Systems and Networks*, WNT, Warszawa, 1996, (in Polish).

[5] B. Filipowicz, *Modelling and Optimization of Queueing Systems. Volume 1, Markovian systems*, Kraków, 1999, (in Polish).

[6] L. Kleinrock, *Queueing Systems, Volume 1, Theory*, John Wiley & Sons, New York, 1975.

[7] K. Idzikowska, "Structural optimization of M/Mm/FIFO/m+N queueing system with individual service and flux of arrivals", *ZN AGH Electrotechnics and Electronics* 19 (1), 38-44 (2000), (in Polish).

[8] B. Filipowicz, *Modelling and Optimization of Queueing Systems. Volume 2, Non-Markovian systems*, Kraków, 2000, (in Polish).

[9] A. Rutkowska, *The Models of Queueing Systems and Networks with Weibull Servers*, Ph.D. dissertation, AGH-UST, Kraków, 2001, (in Polish).

[10] W. Weibull, "A statistical distribution function of wide applicability", *J. Appl. Mech.* 18, 293–297 (1951).

[11] H.C. Tijms, *Stochastic Modelling and Analysis: A computational approach*, John Wiley & Sons, London, 1986.

[12] H.C. Tijms, *Stochastic Models, An Algorithmic Approach*, John Wiley & Sons, London, 1994.

[13] A. Chydziński, "The M/G-G/1 oscillating queueing system", *Queueing Systems* 42 (3), 255-268 (2002).

[14] A. Chydziński, "The M-M/G/1-type oscillating systems", *Cybernetics and Systems Analysis* 39 (2), 316-324 (2003).

[15] B. Filipowicz, *Modelling and analysis of queueing networks*, Wydawnictwa AGH, Kraków, 1997, (in Polish).

[16] F. Baskett, K. Chandy K, R. Muntz, and F. Palacios, "Open, closed and mixed networks of queues with different classes of customers", *J. ACM* 22 (2), 248-260 (1975).

[17] J.P. Buzen, "Computational algorithms for closed queueing networks with exponential servers", *Communications of the ACM* 16 (9), 527–531 (1973).

[18] M. Reiser and S. Lavenberg, "Mean value analysis of closed multichain queueing networks", *J. ACM* 27 (2), 313–322 (1980).

[19] D. Neuse and K. Chandy, "SCAT: a heuristic algorithm for queueing network models of computing system", *ACM Sigmetrics Performance Evaluation Review* 10 (1), 59–79 (1981).

[20] A. Duda and T. Czachórski, "Performance evaluation of Fork and Join synchronization Primitives", *Acta Informatica* 24 (5), 525–553 (1987).

[21] B. Filipowicz and J. Kwiecień, "Fork-join systems", *ZN AGH Automatics* 7 (3), 707–716 (2003), (in Polish).

[22] H.G. Perros, *Queueing Networks with Blocking: Exact and Approximate Solutions*, Oxford University Press, New York, 1994.

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

[24] J. Kwiecień, *The Application of Queueing Networks with Multiple Job Classes in Organizational Problems of Health Service*, Ph.D. dissertation, AGH-UST, Kraków, 2004, (in Polish).

[25] A. Zgrzywa, *Queueing Methods in Performance Evaluation of the Information Systems*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1998, (in Polish).

[26] Y. Liu, "Queueing network modelling of elementary mental processes", *Psychological Review* 103 (1), 116–136 (1996).

[27] O. Tsimhoni and Y. Liu, "Steering a driving simulator using the queueing network-model human processor (QNMHP)", *Proc. 2nd Int. Symposium on Human Factors in Driver Assessment, Training and Vehicle Design*, 81–85 (2003).

[28] C. Wu and Y. Liu, "Modelling human transcription typing with queueing network-model human processor (QN-MHP)", *Proc. 47th Annual Meeting of Human Factors and Ergonomics Society* 5, 381–385 (2004).

[29] C. Wu and Y. Liu, "Modelling psychological refractory period (PRP) and practice effect on PRP with queueing networks and reinforcement learning algorithms", *Proc. 6th Int. Conf. on Cognitive Modelling*, 320–325 (2004).