**Bohdan Pustovyi**
Computer Engineering Department
Odessa National Academy of Food Technologies
E-mail: b.pustoviy@gmail.com

# Automation of analytical model construction for intellectual superstructure in next generation networks

**Keywords:** intellectual services, Markov model, analytical model, ngn, intellectual superstructure, centralized control principle, decentralized control principle.

## Introduction

Currently, the number of services rendered by telecommunications networks has considerably grown. Intellectual services (IS) are specially distinguished in this list.

This is due to popularity of these services among the users, and, consequently, their profitability for operators; thus, the issues of IS efficiency control assessment are getting more urgent. The issues related to study of assessment methods of telecommunications services efficiency control are dealt with in works of [2, 4] and other scientists.

It should be noted that IS efficiency control issues are still not adequately investigated. According to ITU [3] recommendations, to determine efficiency of IS granting control, technical indices of the network operation should be calculated – the time of requisition for IS stay in the network, the probability of requisition deadlocking, the number of requisitions waiting for serving. To calculate these indices, analytical models of IS control systems are used.

Nearly all models of discrete systems with stochastic functioning are developed based on **Queuing Systems** (QS) the processes in which are accidental ones, in many cases the Markov or somehow associated with the Markov processes. That is why mathematical apparatus of the Markov processes theory can be used while solving tasks of queuing theory.

The use of the Markov processes is especially efficient and successful at QS studies and queuing networks (QN) with storing devices of limited capacity [1].

## The use of Markov process for ISCCP representation

At the current stage of NGN development intellectual superstructure with centralized principle (ISCCP) of intellectual services control is used.

ISCCP can be represented as QS of type *M*/*M*/1/*r* [2] based on the use of the Markov processes theory framework. To provide QS example (fig. 1) we will give the system description:

1. The system contains one serving device (SD) and is a *one-channel*.
2. The flow of requisitions entering the system, is homogenious.
3. Though, requisitions of some classes exist, nevertheless, we will still assume that $\lambda$ and $\mu$ are similar for them.
4. The duration of requisitions serving in the device is an accidental value.
5. A storage device for requisitions has *limited capacity*.

Assumption [2]: The duration of requisitions serving in the device is divided exponentially with *intensity* $\mu=1/b$, where *b* – an average duration of requisitions serving in the device.

1. Buffering discipline – *with losses*: a requisition that entered the system and found the storage device full, is lost.
2. Nonpreemptive service priority order – due to entry order based on 'First In First Out' rule (FIFO).

QS with limited capacity storage device always has steady mode, since the length of a waiting line would not grow infinitely, even at big load values.

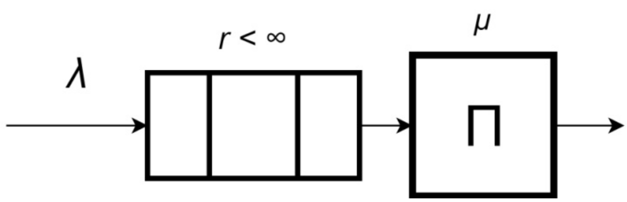


**Fig. 1.** QS with a limited capacity storage device

The use of ISCCP can cause a number of problems in highly loaded systems. Intellectual superstructure with decentralized control principle (ISDCP) can be a solution to all these problems. ISDCP can be viewed as several Queuing Systems connected with each other, that is, as a queuing network (QN).

## The use of the Markov process for ISDCP representation

A combination of final numeral D of serving nodes in which requisitions circulate, thus transferring from one node to another, is a queuing network. ISDCP is an open-loop network. An open-loop network is such an open network to which requisitions come from the environment and are directed to the environment after being served in the network. Requisitions from one server output can come to inputs of others.

We will call requisitions ingress flow a requisitions flow that come to input of a specific server from the environment (from the program commutator), that is, not from output of any other server. In general case the number of ingress flows in the control system corresponds to a number of the servers used.

Description of ISDCP as QN [1]:

ISDCP is an open-loop exponential QN with D nodes that correspond to ISDCP servers.

1. QN nodes are *one-channel.*
2. Storage devices in nodes have a limited capacity $r_i$. Let's define $r_i = M$, where $i = \overline{1, D}$.
3. The ingress flow of requisitions is not *homogenious:* N classes of requisitions come to the system with intensities $\lambda_{ij}$, where $i = \overline{1, D}$ – servers number, $j = \overline{1, N}$ – requisitions class.
4. Buffering discipline in the nodes – with requisitions losses, if storing devices are full.
5. Nonpreemptive service priority order – *with relative priorities*: the lesser the N value, the higher the requisitions priority. A top priority requisition is selected each time from the storage device for serving. In this case, during arrival of top priority requisition to the system the serving is not interrupted.
6. The set matrixes $Q^j = \left\| q_{ik}^j \right\|$ of probabilities of a requisition transmission from the current server i to other servers k, or serving by the current server, where $i, k = \overline{1, D}$ – the servers number for requisitions classes $j = \overline{1, N}$. When $i \neq k$ $q^j$ corresponds to probability of transmission, when $i = k$ $q^j$ corresponds to the probability of serving by the current server.

Assumptions and suppositions.

The duration of requisitions serving in QN nodes is divided exponentially with serving intensities: $\mu = 1/b_{ij}$, where $i = \overline{1, D}$ – servers number, $j = \overline{1, N}$ – the requisitions class. $b_{ij}$ – average duration of the requisitions serving of j class on $i$ – server. In open-loop QN stationary mode exists at any mode, since there cannot be infinite queues in the network nodes.

For the purpose of a more detailed analysis, we will use the Markov processes theory framework. This work suggests that ISDCP should be represented in the following way:

1. ISDCP – open-loop exponential queuing network (QN). In fig. 2 ISDCP is represented with two one-channel servers (fig. 2).
2. Storage devices in both servers have a limited capacity. In example (fig. 2) the storage devices capacity is assumed as: $r_1$=$r_2$=1.
3. Nonpreemptive service priority order – with relative priority. First class requisitions have a higher priority than another class priority.
4. Buffering discipline - nonpreemptive. A requisition that entered the system and found the storage device full, is lost.

Assumptions:

Requisitions of two classes with intensities $\lambda_{11}$, $\lambda_{12}$, $\lambda_{21}$, $\lambda_{22}$ enter from the environment.

1. The duration of requisitions serving in QN nodes is divided *exponentially* with serving intensities: $\mu_{11}=1/b_{11}$, $\mu_{12}=1/b_{12}$, $\mu_{21}=1/b_{21}$, $\mu_{22}=1/b_{22}$, where $b_{11}$, $b_{12}$, $b_{21}$, $b_{22}$ – average durations of serving.
2. The probability of serving a requisition of j-class by i-server will be defined as $q_{ij}$. Then requisitions after serving in server 1 with probabilities $q_{11}$, $q_{12}$ are directed to server 2 with probabilities (1– $q_{11}$), (1– $q_{12}$) leaving QN. Requisitions after serving in server 2 with probabilities $q_{21}, q_{22}$ are directed to server 1 with probabilities (1– $q_{21}$), (1– $q_{22}$) leaving QN.
3. Since requisitions can be lost in the network, open-loop QN is non-linear, that is, intensities of requisitions flows that come to QN nodes, are not associated with each other by linear dependence, and cannot be calculated by solving a system of linear algebraic equations.

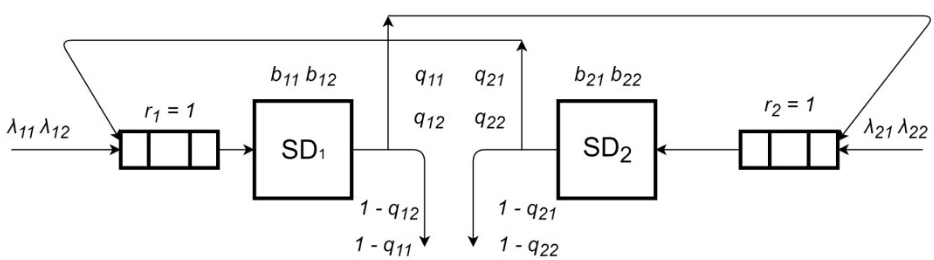


**Fig. 2.** The ISDCP with two servers is represented by QN

## The Formation of Analytical Model

Markov processes states should be coded in order to define the stationary probability of states. This can achieved in the following way: (П,Ч).

For example, (fig. 2) Markov processes coding is performed in the following way: П = {0, 1, 2} – the state of a serving device, which is set by a requisition class, which is being served ('0' – the device is free; '1' or '2' – a requisition of class 1 or 2 respectively, is being served in the device).

The state of the storage device can be represented in the following way: Ч= {0, 1, 2, 11, 12, 22}, where '0' – means the absence of requisitions in the storage device; '1' – the presence of only one requisition of class 1 in the storage device; '2' – the presence of requisition of class 2 in the storage device; '11' – the presence of two class 1 requisitions in the storage device1; '22' – the presence of two class 2 requisitions in the storage device and '12' – the presence of one class 1 requisition and and one class 2 requisition in the storage device.

State '12' does not differentiate in which order these requisitions came to the system which is determined by availability of the relative priority between them – irrespective of the moment the requisitions came for serving, class 1 requisition will always be chosen.

In case of serving without priority, when requisitions of different classes are chosen for serving based on the order of their arrival, one more storage device state should be introduced – '21', which means that class 2 requisition came to the system earlier than class 1 requisition, whereas state '12' means that class 1 requisition came earlier to the system.

Then, the Markov process can occur in one of the following states at any moment:

$E_0$ : $k = 0$ – the system contains no requisition;

$E_1$ : $k = 1$ – the system contains 1 requisition being served in the device;

$E_2$ : $k = 2$ – the system contains 2 requisitions: one – being served in the device and the other one – waiting for in the storage device;

$E_{r+1}$ : $k = r + 1$ – the system contains $(r + 1)$ requisitions: one – being served in the device and $r$ – in the storage device.

The marked graph of random probability transitions is represented in fig.3 [2].

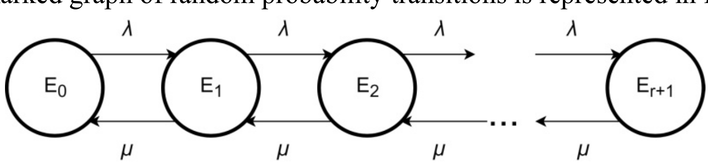


**Fig. 1.** The graph of Markov process transition

One and the same occurrence can take place in the system at the same time only once:

- the arrival of a requisition with intensity *λ*, that corresponds to increase in the requisitions by one in the system and the random probability transition to a state with number exceeding by one;
- completion of serving a requisition in a device with intensity *μ* that corresponds to lessening in the number of requisitions in the system and random probability transition to a state with number which is less by one.

Based on the defined states of the Markov process, the system of balance equations is generated (whose number equals to the number of states), and whose solving provides possibility for the Markov process probabilities defining and further – for obtaining technical indices of the network operation.

## Assessment of the balance equations system

It should be noted that labor intensity of solving the balance equations system grows significantly at growing the number of classes of requisitions for IS and at growing the waiting line. Figures 4 and 5 show the change of the system states (that is, the size of the equations system) subject to the number of classes of requisitions for IS, and to the waiting line. As we can see, the abrupt increase in the number of states of the equations system disables the use of manual calculations.

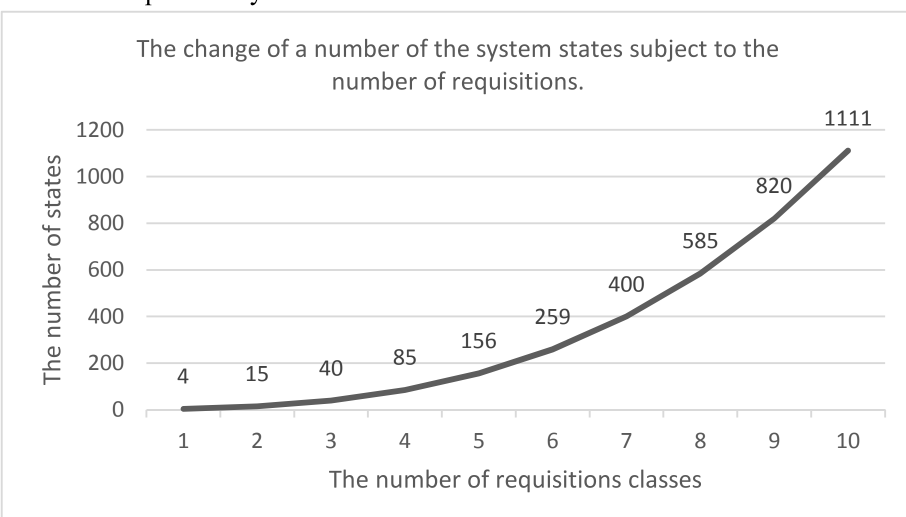


**Fig. 2.** The diagram of the change in the number of possible states of the system at the change of requisitions classes number in the system with 1 SD and the waiting line for serving of 2 requisitions

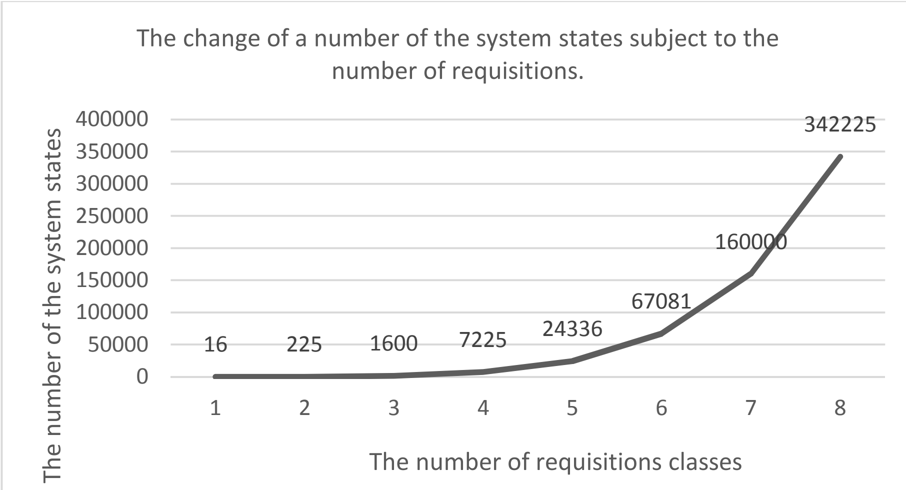


**Fig. 3.** The diagram of the change in the number of possible states of the system at the change of requisitions classes number in the system with 2 SD and the waiting line for serving of 2 requisitions

## Software description

In this work the software is proposed which is designed for automation of the system probable states calculation, the construction of the Markov process transition graph and generation of equations system to determine stationary probabilities. SW is designed for the use both in ISCCP (QS), and ISDCP (QN).

The SW written in the JavaScript scripting language with use of the open library React JS which allows its launching in any browser. The maximal number of states for calculation of QS (QN) depends on a computer technical characteristics.

The following assumptions were accepted when developing this SW:

- the SD can serve only one class of requisitions at a time;
- the queue for serving is built as per FIFO discipline;
- the system functions without requisitions losses.

The input data of the developed SW is:

- SD number;
- the maximal length of serving waiting line;
- the number of requisitions classes and their priority.

In fig. 6,7,8 SW interface is shown. The SW performs the following functions:

- ensures data input for calculation;
- performs coding of all possible states of the Markov process for the system chosen;
- constructs and models the Markov process transition graph;
- the SW generates the equations system for determining stationary probabilities based on the generated transition graph;
- generates expressions for technical indices calculation based on found values of stationary probabilities of an accidental process states:
  - average number of requisitions in the queue;
  - average number of requisitions in the system;
  - probability of requisitions loss;
  - allows obtained results download to a text file.

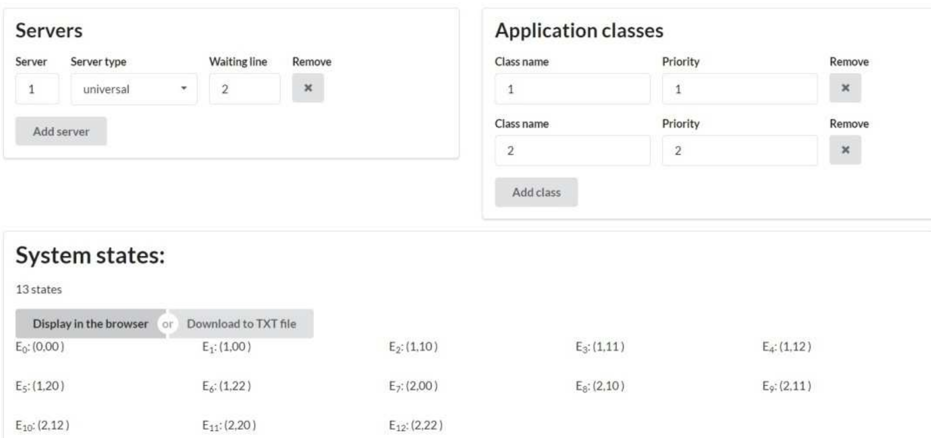


**Fig. 4.** Form for input data introduction in SW

Based on the introduced input data, the SW codes all possible states of the Markov process for a set system. The coding occurs in the following way:

$E_n = (П_1Ч_1 \,/\, П_2Ч_2 \,/\, ... \,/\, П_iЧ_i/../П_NЧ_N)$, where: (1)

- $E_n$ – the Markov process state, $n$ – the state number;
- $П_i = \overline{0,\ m}$ – the state of a serving device, in which only one requisition of a specific class can be served at a time ('0' - the device is free, 'm' – 'm' class requisition is served), 'i' – the device number, 'N' – SD number;

- $\text{Ч}_i = \overline{\underbrace{0, m \ldots m}_{k}}$ – the state of the serving queue ('0' – the queue is empty, 'm' – 'm' class requisition in queue for being served), '$k$' - the number of places in queue for being served.

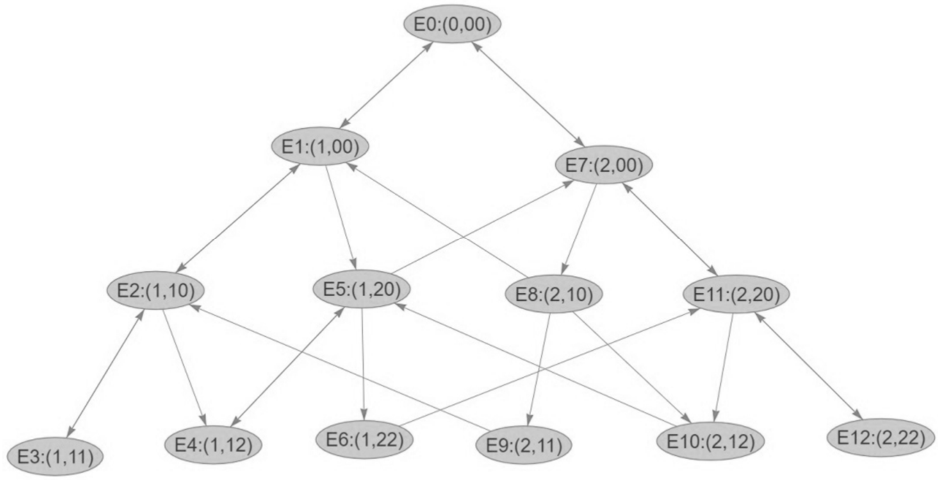


**Fig. 5.** The marked Markov process transition graph

Using encoded states, the SW constructs the Markov processes marked graph. In fig.7 the given example of the Markov process transitions graph in the system with one SD with maximal waiting line of two requisitions with different priority.

**Subject to transitions graph we'll generate the equations system to define the values of stationary probabilities:**

Display in the browser or Download to TXT file

$p_0*\lambda_1+p_0*\lambda_2=p_1*\mu_1+p_7*\mu_2$

$p_1*\mu_1+p_1*\lambda_1+p_1*\lambda_2=p_0*\lambda_1+p_2*\mu_1+p_5*\mu_1$

$p_2*\mu_1+p_2*\lambda_1+p_2*\lambda_2=p_1*\lambda_1+p_3*\mu_1+p_4*\mu_1$

$p_3*\mu_1=p_2*\lambda_1$

$p_4*\mu_1+p_4*\mu_1=p_5*\lambda_1$

$p_5*\mu_1+p_5*\lambda_1+p_5*\lambda_2=p_4*\mu_1+p_6*\mu_1+p_7*\lambda_1$

$p_6*\mu_1=p_{11}*\lambda_1$

$p_7*\mu_2+p_7*\lambda_1+p_7*\lambda_2=p_0*\lambda_2+p_8*\mu_2+p_{11}*\mu_2$

$p_8*\mu_2=p_1*\lambda_2+p_9*\mu_2+p_{10}*\mu_2$

$p_9*\mu_2=p_2*\lambda_2$

$p_{10}*\mu_2+p_{10}*\mu_2=p_5*\lambda_2$

$p_{11}*\lambda_1+p_{11}*\mu_2+p_{11}*\lambda_2=p_7*\lambda_2+p_{10}*\mu_2+p_{12}*\mu_2$

$p_{12}*\mu_2=p_{11}*\lambda_2$

**Fig. 6.** Equations system of the Markov process stationary probabilities

The SW generates the equations system (2) for determining stationary probabilities based on the transitions graph:

$$\begin{cases} p0 \cdot \lambda1 + p0 \cdot \lambda2 = p1 \cdot \mu1 + p7 \cdot \mu2 \\ p1 \cdot \mu1 + p1 \cdot \lambda1 + p1 \cdot \lambda2 = p0 \cdot \lambda1 + p2 \cdot \mu1 + p5 \cdot \mu1 \\ p2 \cdot \mu1 + p2 \cdot \lambda1 + p2 \cdot \lambda2 = p1 \cdot \lambda1 + p3 \cdot \mu1 + p4 \cdot \mu1 \\ p3 \cdot \mu1 = p2 \cdot \lambda1 \\ p4 \cdot \mu1 + p4 \cdot \mu1 = p5 \cdot \lambda1 \\ p5 \cdot \mu1 + p5 \cdot \lambda1 + p5 \cdot \lambda2 = p4 \cdot \mu1 + p6 \cdot \mu1 + p7 \cdot \lambda1 \\ p6 \cdot \mu1 = p11 \cdot \lambda1 \\ p7 \cdot \mu2 + p7 \cdot \lambda1 + p7 \cdot \lambda2 = p0 \cdot \lambda2 + p8 \cdot \mu2 + p11 \cdot \mu2 \\ p8 \cdot \mu2 = p1 \cdot \lambda2 + p9 \cdot \mu2 + p10 \cdot \mu2 \\ p9 \cdot \mu2 = p2 \cdot \lambda2 \\ p10 \cdot \mu2 + p10 \cdot \mu2 = p5 \cdot \lambda2 \\ p11 \cdot \lambda1 + p11 \cdot \mu2 + p11 \cdot \lambda2 = p7 \cdot \lambda2 + p10 \cdot \mu2 + p12 \cdot \mu2 \\ p12 \cdot \mu2 = p11 \cdot \lambda2 \\ \sum_{k=0}^{12} p_k = 1 \end{cases} \quad (2)$$

where $\lambda$ – the intensity of the requisitions arrival, $\mu$ – the intensity of the requisitions serving.

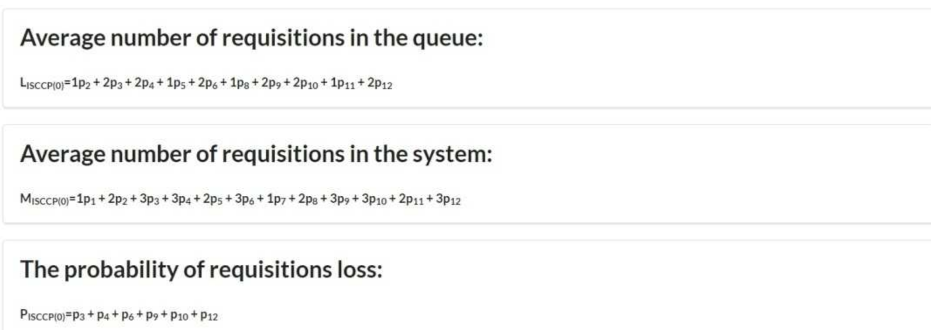


**Fig. 9.** The display of the Markov process transitions graph construction results

For the proposed QS (fig. 7) the SW generates expressions for technical indices calculation [1] abased on calculated values of stationary probabilities of the accidental process.

Average number of requisitions in the queue:

$$L=1p_2 + 2p_3 + 2p_4 + 1p_5 + 2p_6 + 1p_8 + 2p_9 + 2p_{10} + 1p_{11} + 2p_{12} \quad (3)$$

Average number of requisitions in the system:

$$M=1p_1 + 2p_2 + 3p_3 + 3p_4 + 2p_5 + 3p_6 + 1p_7 + 2p_8 + 3p_9 + 3p_{10} + 2p_{11} + 3p_{12} \quad (4)$$

The probability of requisitions loss:

$$P= p_3 + p_4 + p_6 + p_9 + p_{10} + p_{12} \quad (5)$$

This example demonstrates expressions for calculation of technical indices y for QS where there is only one control device, though, the SW allows calculation of indices i for QN with any number of servers.

## Conclusions

Automation of the Markov processes balance equation generation reduces labor intensity of calculations significantly which allows the calculation of IC control systems technical indices with any number of servers. Besides, the use of the developed automation system will allow selection of control principle and the structure of NGN intellectual superstructure as early as the network design stage.

## Literature

1. Aliev T.I. The basics of discrete systems modelling.– St-Pb:SPbGU ITMO, 2009. – 363 p.
2. Knyazeva N.O. Intellectual services control in the next generation networks:monograph /N.O.Knyazeva , S.V. Shestoopalov – Odesa: Bondarenko M.O., 2017. – 268 p.
3. 'International Telecommunication Union (ITU)', official Internet-agency. — Access mode: http://www.itu.int (accessed date March 20, 2015).
4. Berkman L.N. Invariance of converged networks control systems in emergency modes / Berkman L.N., L.O. Komarova, I. A. Boyko// Scientific notes of the Ukrainan Scientific-Research Communications Institute.. - 2014. - No. 1. - P. 11-15. - Access mode: http://nbuv.gov.ua/UJRN/Nzundiz_2014_1_4.

## Abstract

The software is suggested for automation of analytical model construction for intellectual superstructure in next generation networks. The software is designed for automation of the system probable states calculation, the construction of the Markov process transition graph and generation of equations system to determine stationary probabilities. An example of analytical model construction is represented for intellectual superstructure with centralized control principle.

## Streszczenie

Zaproponowano oprogramowanie do zautomatyzowania analitycznego modelu budowy intelektualnej nadbudowy w sieciach następnej generacji. Oprogramowanie jest przeznaczone do automatyzacji obliczania stanów prawdopodobieństwa systemu, konstrukcji wykresu przejścia procesu Markowa i generowania układu równań w celu ustalenia stacjonarnych prawdopodobieństw. Przykład konstrukcji modelu analitycznego jest reprezentowany dla nadbudowy intelektualnej z zasadą scentralizowanej kontroli.

**Słowa kluczowe:** usługi intelektualne, model Markowa, model analityczny, nadbudowa intelektualna, zasada scentralizowanej kontroli, zasada zdecentralizowanej kontroli.