

Министерство образования Республики Беларусь  
Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей  
Кафедра программного обеспечения информационных технологий

Дисциплина: Системный анализ и машинное моделирование

ОТЧЁТ  
по лабораторной работе №3  
на тему

«Построение и исследование аналитической модели дискретно-стохастической системы массового обслуживания»

Выполнил:  
студент гр. 851003  
Вайтусенок И.А

Проверил:  
Мельник Николай Иосифович

Минск 2021

## Задание на лабораторную работу

В соответствии с заданным вариантом структуры СМО построить граф состояний Р-схемы.

Смысл кодировки состояний раскрыть (время до выдачи заявки, число заявок в накопителе и т.д.).

По графу построить аналитическую модель и, решив ее, определить вероятности состояний. Рассчитать теоретическое значение показателей эффективности

$P_{отк}$  – вероятность отказа (вероятность того, что заявка, сгенерированная источником, не будет в конечном итоге обслужена системой);

$P_{бл}$  – вероятность блокировки (вероятность застать источник или канал в состоянии блокировки);

$L_{оч}$  – средняя длина очереди;

$L_c$  – среднее число заявок, находящихся в системе;

$Q$  – относительная пропускная способность (вероятность того, что заявка, сгенерированная источником, будет в конечном итоге обслужена системой);

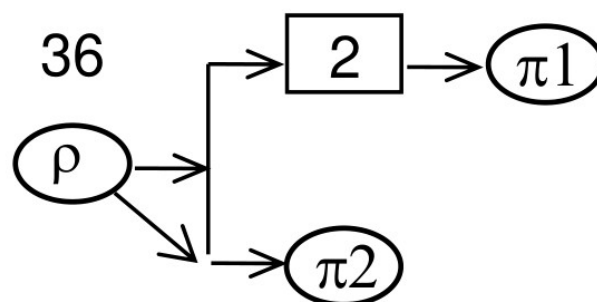
$A$  – абсолютная пропускная способность (среднее число заявок, обслуживаемых системой в единицу времени, т.е. интенсивность потока заявок на выходе системы);

$W_{оч}$  – среднее время пребывания заявки в очереди;

$W_c$  – среднее время пребывания заявки в системе;

$K_{кан}$  – коэффициент загрузки канала (вероятность занятости канала).

Вариант задания №36 представлен следующей Р-схемой:



где  $\rho = 0.3$ ,  $\pi_1 = 0.8$ ,  $\pi_2 = 0.75$

## Кодировки состояний

Код состояния представлен в виде трех последовательно записанных чисел А В С, где:

– А – количество заявок, находящихся в очереди. Может принимать значения 0, 1, 2.

– В – количество заявок, находящихся на обработке в первом канале. Может принимать значения 0 (канал свободен) и 1 (канал занят).

– С – количество заявок, находящихся на обработке во втором канале. Может принимать значения 0 (канал свободен) и 1 (канал занят).

## Аналитическая модель

С моделью построенной графически системы можно ознакомиться на рисунке 1.1. Вершины графа представляют собой состояния, направленные связи (рёбра графа) между вершинами – возможные переходы между состояниями. Вес ребра обозначает вероятность перехода из одного состояния в другое. Где:

$$(1) = (1-\rho) * (1-\pi_1) * (1-\pi_2)$$

$$(2) = \rho * (1-\pi_1) * (1-\pi_2)$$

$$(3) = (1-\rho) * (1-\pi_1) * \pi_2$$

$$(4) = (1-\rho) * \pi_1 * (1-\pi_2)$$

По данному графу была составлена система линейных уравнений для определения вероятностей состояний:

$$\begin{aligned} P_{000} = & \\ & \rho * (1 - \pi_1) * (1 - \pi_2) * P_{011} + \\ & \rho * (1 - \pi_1) * P_{010} + \\ & \rho * (1 - \pi_2) * P_{001} + \\ & \rho * P_{000}; \end{aligned}$$

$$\begin{aligned} P_{010} = & \\ & (1 - \rho) * (1 - \pi_2) * P_{001} + \\ & \rho * (1 - \pi_1) * (1 - \pi_2) * P_{111} + \\ & \rho * (1 - \pi_1) * P_{110} + \\ & \rho * \pi_1 * P_{010} + \\ & (1 - \rho) * (1 - \pi_1) * P_{010} + \\ & \rho * \pi_1 * (1 - \pi_2) * P_{011} + \\ & (1 - \rho) * (1 - \pi_1) * (1 - \pi_2) * P_{011} + \\ & (1 - \rho) * P_{000}; \end{aligned}$$

$$\begin{aligned}
P_{110} &= \\
&(1 - \rho) * \pi_1 * P_{010} + \\
&(1 - \rho) * \pi_1 * (1 - \pi_2) * P_{011} + \\
&\rho * (1 - \pi_1) * (1 - \pi_2) * P_{211} + \\
&\rho * \pi_1 * (1 - \pi_2) * P_{111} + \\
&(1 - \rho) * (1 - \pi_1) * (1 - \pi_2) * P_{111} + \\
&\rho * (1 - \pi_1) * P_{210} + \\
&\rho * \pi_1 * P_{110} + \\
&(1 - \rho) * (1 - \pi_1) * P_{110}; \\
\\
P_{210} &= \\
&(1 - \rho) * \pi_1 * (1 - \pi_2) * P_{111} + \\
&\rho * \pi_1 * (1 - \pi_2) * P_{211} + \\
&(1 - \rho) * (1 - \pi_1) * (1 - \pi_2) * P_{211} + \\
&(1 - \rho) * \pi_1 * P_{110} + \\
&\rho * \pi_1 * P_{210} + \\
&(1 - \rho) * (1 - \pi_1) * P_{210}; \\
\\
P_{211} &= \\
&(1 - \rho) * \pi_1 * P_{210} + \\
&(1 - \rho) * \pi_1 * \pi_2 * P_{111} + \\
&\rho * \pi_1 * \pi_2 * P_{211} + \\
&(1 - \rho) * (1 - \pi_1) * \pi_2 * P_{211} + \\
&(1 - \rho) * \pi_1 * (1 - \pi_2) * P_{211} + \\
&(1 - \rho) * \pi_1 * \pi_2 * P_{211}; \\
\\
P_{111} &= \\
&(1 - \rho) * \pi_1 * \pi_2 * P_{011} + \\
&\rho * \pi_1 * \pi_2 * P_{111} + \\
&(1 - \rho) * (1 - \pi_1) * \pi_2 * P_{111} + \\
&\rho * (1 - \pi_1) * \pi_2 * P_{211}; \\
\\
P_{011} &= \\
&\rho * \pi_1 * \pi_2 * P_{011} + \\
&(1 - \rho) * (1 - \pi_1) * \pi_2 * P_{011} + \\
&\rho * (1 - \pi_1) * \pi_2 * P_{111} + \\
&(1 - \rho) * \pi_2 * P_{001}; \\
\\
P_{001} &= \rho * \pi_2 * P_{001} + \rho * (1 - \pi_1) * \pi_2 * P_{011};
\end{aligned}$$

После решения данной системы уравнений были получены следующие теоретические вероятности состояний:

$$\begin{aligned}
P_{000} &= 0.0006970763 \\
P_{001} &= 0.00018180082 \\
P_{010} &= 0.007122278 \\
P_{011} &= 0.0031309724 \\
P_{110} &= 0.048069507 \\
P_{111} &= 0.04762739 \\
P_{210} &= 0.16564754 \\
P_{211} &= 0.72752345
\end{aligned}$$

На основании этих данных были рассчитаны теоретические показатели эффективности системы:

P<sub>от</sub>: 0.30555984  
 P<sub>bl</sub>: 0  
 L<sub>оч</sub>: 1.8820388  
 L<sub>c</sub> : 3.6596239  
 Q : 0.6944401  
 A : 0.3944401  
 W<sub>оч</sub>: 9.418472  
 W<sub>c</sub> : 9.278022  
 K<sub>1</sub> : 0.9991211  
 K<sub>2</sub> : 0.7784636

### Имитационная модель

В рамках работы была разработана программная модель, симулирующая систему, описанную выше. В результате её работы были получены следующие экспериментальные данные.

RO	0.3	000	0.000826	P_OTK	0.304538
Q_LEN	2	001	0.000166	P_BL	0
PI_1	0.8	010	0.007658	L_QUE	1.880606
PI_2	0.75	011	0.003148	L_C	3.656476
Calculate!		110	0.0491	Q	0.695462
		111	0.046698	A	0.394704
		210	0.165554	W_QUE	9.406134
		211	0.72685	W_C	9.263843
				K_PI1	0.99901
				K_PI2	0.776864

Рисунок 1.2 Имитационная модель

Экспериментальные данные отличаются от полученных ранее теоретических данных незначительно (менее, чем на 0.01).

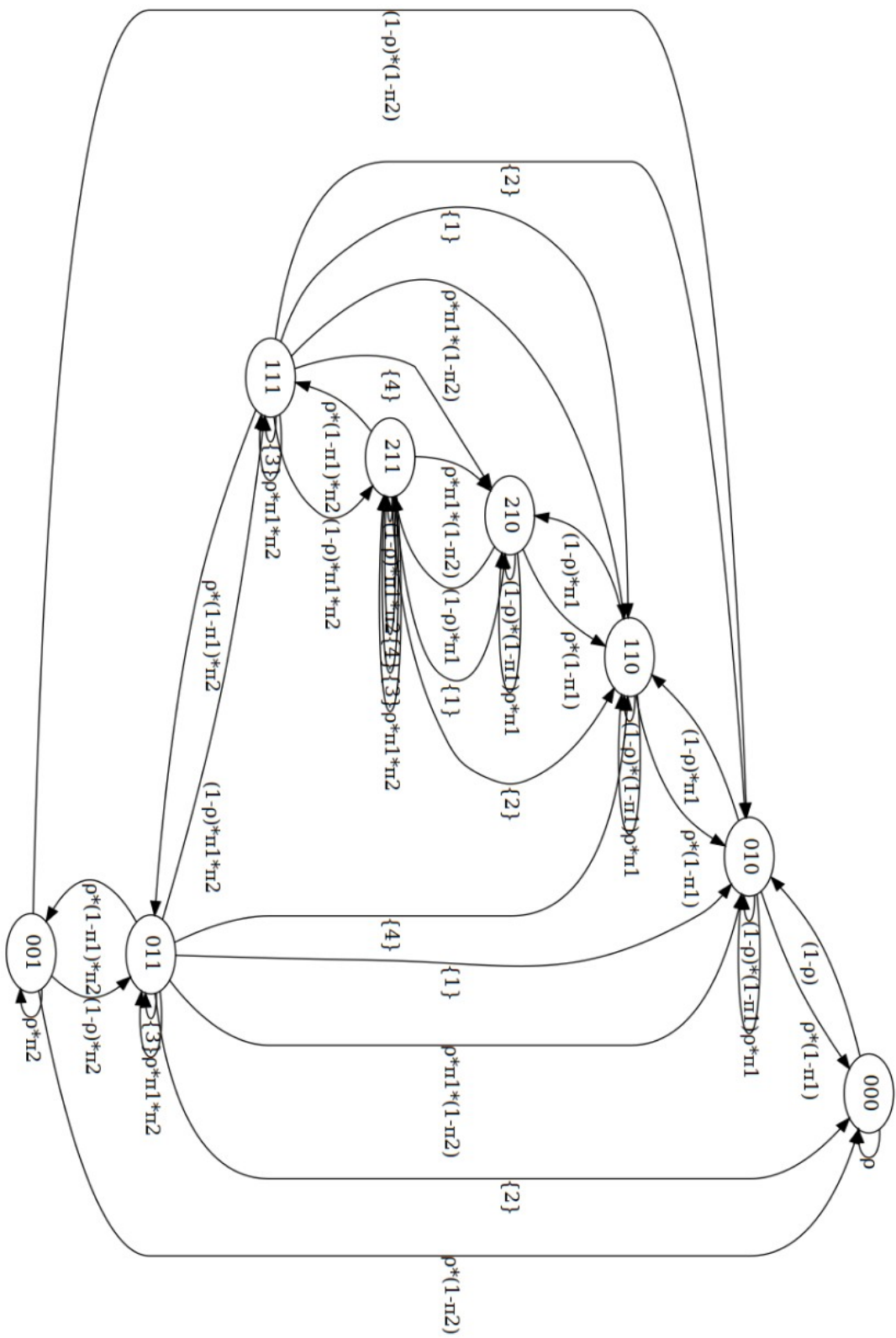


Рисунок 1.1. Граф-схема системы

```

let chance_reject: f32;
let chance_block: f32;
let average_queue_num: f32;
let average_system_num: f32;
let average_queue_time: f32;
let average_system_time: f32;
let relative_throughput: f32;
let average_channel_load_1: f32;
let average_channel_load_2: f32;

let filter_sum = |f: Box<dyn Fn(&State) → _>| {
    states
        .iter()
        .filter(|(state, _)| f(state))
        .map(|x| x.1)
        .sum::<f32>()
};

chance_block = 0.;

chance_reject = filter_sum(Box::new(|state| {
    state[1] == 1 && state[2] == 1 && state[0] == 2
})) * sys.pi1
    * sys.pi2
    * (1. - sys.ro);

let when_1_1 = filter_sum(Box::new(|state| state[1] == 1));
let when_1_2 = filter_sum(Box::new(|state| state[2] == 1));

// Среднее количество за такт
let absolute_throughput_queue = when_1_1 * (1. - sys.pi1);
let absolute_throughput_down = when_1_2 * (1. - sys.pi2);

relative_throughput = 1. - chance_reject;

average_queue_num = filter_sum(Box::new(|state| state[0] == 1))
    + 2. * filter_sum(Box::new(|state| state[0] == 2));
average_system_num = states
    .iter()
    .map(|(state, c)| (state[0] + state[1] + state[2], c))
    .map(|(n, c)| n as f32 * c)
    .sum::<f32>();

average_queue_time = average_queue_num / absolute_throughput_queue;
average_system_time =
    average_system_num / (absolute_throughput_queue + absolute_throughput_down);

average_channel_load_1 = filter_sum(Box::new(|state| state[1] == 1));
average_channel_load_2 = filter_sum(Box::new(|state| state[2] == 1));

```