

Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Моделирование

ОТЧЕТ
по лабораторной работе № 3
Вариант № 31

Студенты:

А.С. Сикорин
И.А. Григорик

Проверил:

И.Г. Алексеев

МИНСК 2023

Расчёт варианта

$$05050143 \% 32 = 31$$

Задание 1

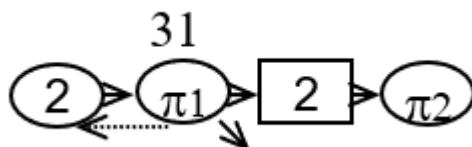


Рисунок 1 – Вариант по заданию

Построим граф состояний, используя следующую кодировку:

i st – время до выдачи заявки $\{1, 2\}$,

k_1 – состояние первого канала $\{0$ – свободен, 1 – занят $\}$,

n – количество заявок в накопителе $n = \{0, 1, 2\}$

k_2 – состояние второго канала $\{0$ – свободен, 1 – занят $\}$.

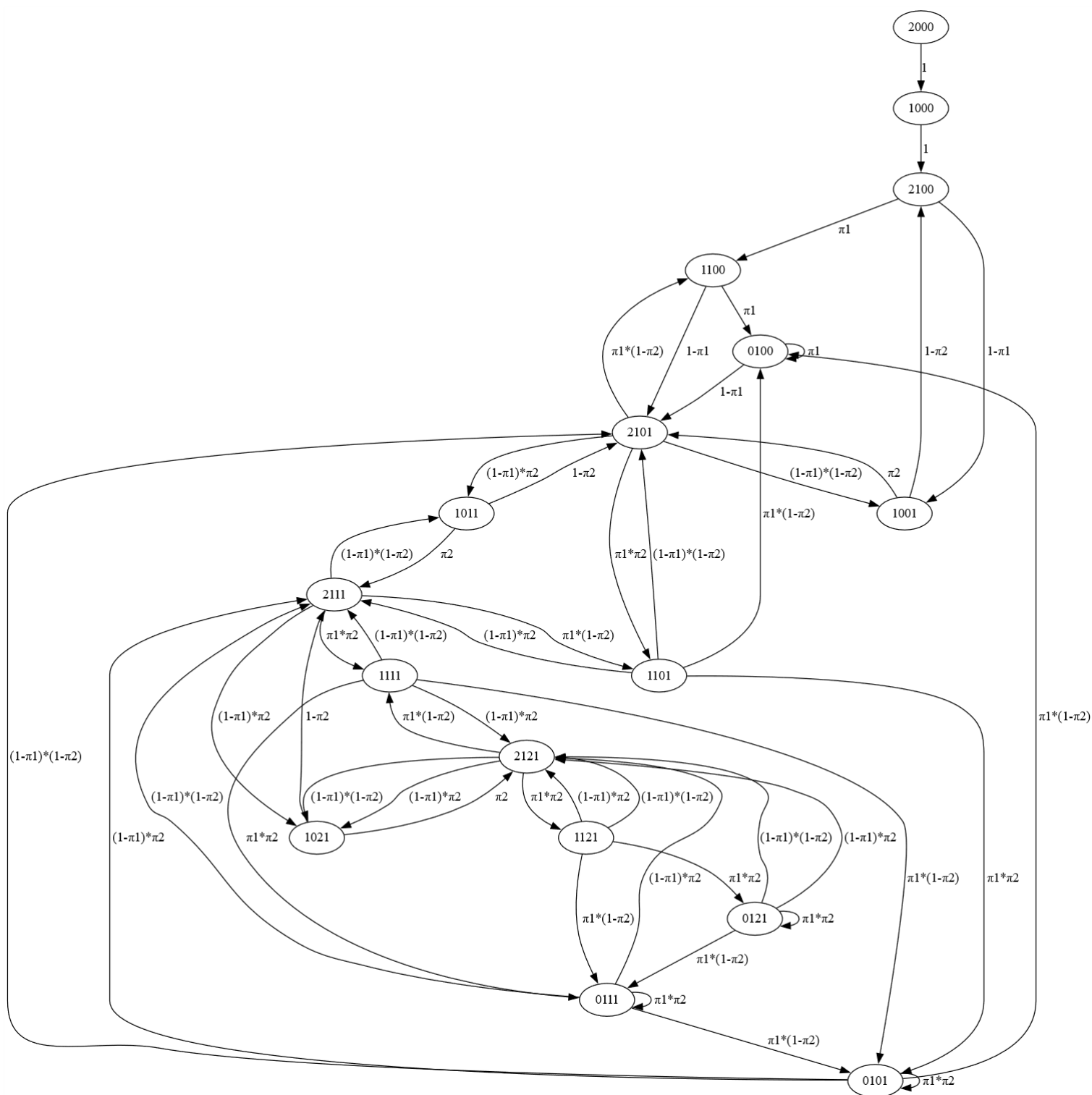


Рисунок 2 – Граф состояний

По графу построим аналитическую модель и, решив ее, определим вероятности состояний.

При $\pi_1=0,4$ $\pi_2=0,5$ получим следующую систему уравнений:

$$P_{2000} = a = 0$$

$$P_{1000} = b = 0$$

$$P_{2100} = c = 0.5g + b$$

$$\begin{aligned}
P_{1100} &= d = 0.4c + 0.2f \\
P_{0100} &= e = 0.4d + 0.4e + 0.2k + 0.2q \\
P_{2101} &= f = 0.6d + 0.6e + 0.5g + 0.3k + 0.5h + 0.3q \\
P_{1001} &= g = 0.3f + 0.6c \\
P_{1011} &= h = 0.3f + 0.3i \\
P_{2111} &= i = 0.5h + 0.3k + 0.3j + 0.5l + 0.3p + 0.3q \\
P_{1111} &= j = 0.2i + 0.2m \\
P_{1101} &= k = 0.2i + 0.2f \\
P_{1021} &= l = 0.3i + 0.3m + 0.3m \\
P_{2121} &= m = 0.3j + 0.5l + 0.3n + 0.3n + 0.3p + 0.3o + 0.3o \\
P_{1121} &= n = 0.2m \\
P_{0121} &= o = 0.2n + 0.2o \\
P_{0111} &= p = 0.2j + 0.2n + 0.2o + 0.2p \\
P_{0101} &= q = 0.2p + 0.2j + 0.2k + 0.2q \\
1 &= a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q
\end{aligned}$$

После решения системы линейных получим:

$$\begin{aligned}
P_{2000} &= 0 \\
P_{1000} &= 0 \\
P_{2100} &= 0.0420846 \\
P_{1100} &= 0.0561128 \\
P_{0100} &= 0.0696589 \\
P_{2101} &= 0.196395 \\
P_{1001} &= 0.0841693 \\
P_{1011} &= 0.0996438 \\
P_{2111} &= 0.135751 \\
P_{1111} &= 0.0405394 \\
P_{1101} &= 0.0664292 \\
P_{1021} &= 0.0808928 \\
P_{2121} &= 0.0669458 \\
P_{1121} &= 0.0133892 \\
P_{0121} &= 0.00334729 \\
P_{0111} &= 0.014319 \\
P_{0101} &= 0.0303219
\end{aligned}$$

Исходя из полученных данных, рассчитаем теоретические значения средней абсолютной пропускной способности (А) и средней длины очереди ($L_{оч}$)

$$A = (1 - 0.5) * (0.841693 + 0.196395 + 0.664292 + 0.0996438 + 0.0303219 + 0.135751 + 0.0405394 + 0.0808928 + 0.014319 + 0.0669458 + 0.133892 + 0.00334729) = 0.5 * 0.83214369 = 0.416071845$$

$$L_{оч} = 1 * P_{1011} + 2 * P_{0121} + 2 * P_{1121} + 1 * P_{2111} + 1 * P_{1111} + 2 * P_{1021} + 1 * P_{0111} + 2 * P_{2121} = 0.0996438 + 0.00669458 + 0.0267784 + 0.135751 + 0.0405394 + 0.1617856 + 0.014319 + 0.1338916 = 0.61940338$$

Задание 2

Построим имитационную модель и исследуем ее.

The screenshot shows the LPR3 simulation software interface. On the left, there are input fields for 'Отказ 1 канала' (0.4) and 'Отказ 2 канала' (0.50). Below these are fields for 'Количество запросов' (5000), 'Количество ошибок' (1102), and 'Счетчик очереди' (3449). On the right, there is a section titled 'Вариант 31' containing a process flow diagram with nodes 2, π_1 , 2, and π_2 . Below the diagram are output fields for 'Абсолютная пропускная способность' (0.42125676), 'Средняя длина очереди' (0.59912924), and 'Количество тактов' (10000). At the bottom is a 'Сгенерировать' button.

Рисунок 4 – Результат работы программы

В итоге статистическое значение искомых характеристик достаточно близко к теоретически рассчитанным.

Исходный код программы:

```
using System;

namespace Queue
{
    public class mod3
    {
        private readonly Random _random = new();
        private Byte requests;
        private Byte c1 { get; set; }
        private Byte buf;
        private Byte c2 { get; set; }
        private bool is_blocked;
        public readonly byte freq = 2;
        private const int buff_size = 1;
        public Double first_channel_failure { get; }
        public Double second_channel_failure { get; }
        public Int32 r_count { get; private set; }
        public Int32 f_count { get; private set; }
        public Int32 queue_count { get; private set; }

        public mod3(Double first, Double second)
        {
            requests = freq;
            c1 = 0;
            buf = 0;
            c2 = 0;

            first_channel_failure = first;
            second_channel_failure = second;

            r_count = 0;
            f_count = 0;
            queue_count = 0;
        }
    }
}
```

```

}

public void generate_next()
{
    if (c2 == 1 && _random.NextDouble() > second_channel_failure)
    {
        c2 = 0;
    }

    if (buf > 0 && c2 == 0)
    {
        buf--;
        c2 = 1;
    }

    if (c1 == 1)
    {
        if (is_blocked && buf < buff_size)
        {
            buf++;
            is_blocked = false;
            c1 = 0;
        }
        else if (_random.NextDouble() > first_channel_failure)
        {
            if (buf == buff_size)
            {
                is_blocked = true;
            }
            else
            {
                c1 = 0;
                buf++;
            }
        }
    }

    if (buf > 0 && c2 == 0)
    {
        buf--;
        c2 = 1;
    }

    if (requests != 1)
    {
        requests--;
    }
    else
    {
        requests = freq;
        r_count++;

        if (c1 == 0)
        {
            c1 = 1;
        }
        else
        {
            f_count++;
        }
    }

    if (buf != 0)
    {
        queue_count++;
    }
}

```

```
        Console.WriteLine($"{requests} {c1} {buf} {c2}");  
    }  
}
```