Отчет по лабораторной работе №4: Неполнодоступная двухсервисная модель Эрланга с резервированием для второго типа и различными интенсивностями обслуживания. Первыми заполняются зарезервированные приборы

Дисциплина: Основы проектирования сетей и систем телекоммуникаций

Выполнила: Ким Реачна, НПИбд-01-20

Содержание

Список иллюстраций

1 Теоретические сведения

Рассмотрим звено сети емкостью . Пусть пользователям сети предоставляются услуги двух типов. Запросы на предоставление услуг представляют собой ПП с интенсивностями λ_1, λ_2. Среднее время обслуживания запросов каждого типа μ_1^{-1}, μ_2^{-1} соответственно. Рассмотрим случай равенства μ. Часть пропускной способности соты зарезервирована для обслуживания запросов на предоставление услуги 2-го типа. Оставшаяся часть пропускной способности является полнодоступной для запросов на предоставление услуг обоих типов. Предположим, что сначала заполняются зарезервированные приборы.

Основные обозначения:

- C - общее число приборов;
- g - число полнодоступных приборов;
- $C - g$ -число полнодоступных приборов;
- λ_1, λ_2 - интенсивность поступления заявок $1, 2$-го типа [заявок/ед.вр.];
- μ^{-1} - среднее время обслуживания заявки $1, 2$-го типа [ед.вр.];
- ρ_1, ρ_2 - интенсивность предложенной нагрузки, создаваемой заявками $1, 2$-го типа;
- $X(t)$ - число заявок, обслуживаемых в системе в момент времени $t, t \geq 0$
- X - пространство состояний системы;
- n - число обслуживаемых в системе заявок;
- B_1, B_2 - множество блокировки заявок $1, 2$-го типа;
- S_1, S_2 - множество приема заявок $1, 2$-го типа.

Схема модели (рис. 1.1):

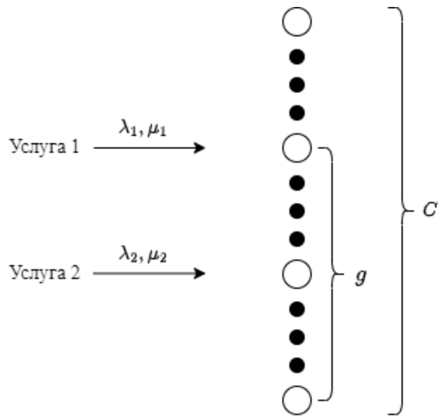

Рис. 1.1: Схема неполнодоступная двухсервисная модель Эрланга с различными интенсивностями обслуживания

Пространство состояний системы (рис. 1.2):

$$X = \{(n_1, n_2) : n_1 = \overline{0, g}, n_2 = \overline{0, C}, n_1 + n_2 \leq C\} \tag{4.1}$$

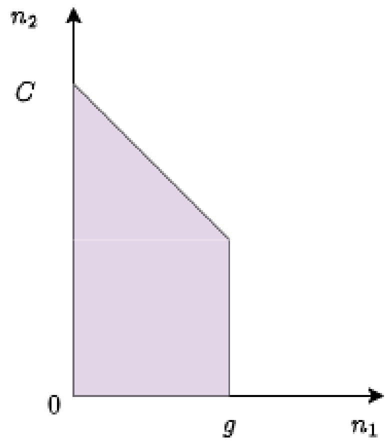

Рис. 1.2: Пространство состояний неполнодоступной двухсервиснаой модель Эрланга с различными интенсивностями обслуживания в общем виде

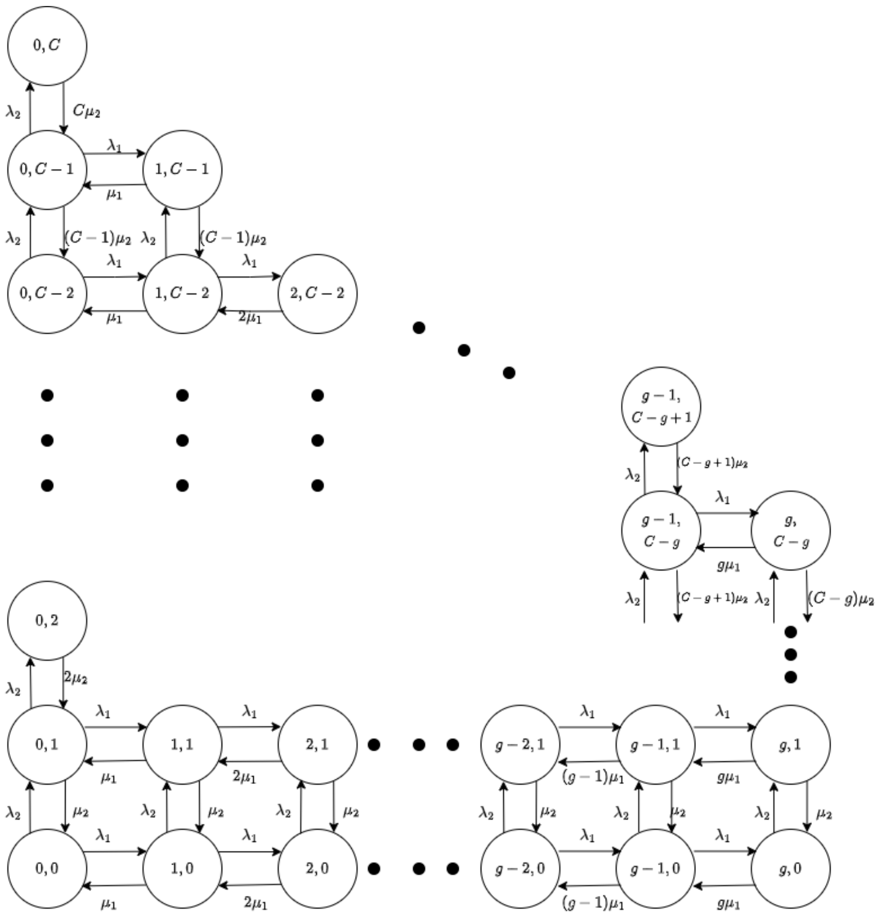

Рис. 1.3: Диаграмма интенсивностей переходов для неполнодоступной двухсервиснаой модель Эрланга с различными интенсивностями обслуживания

Множество блокировок запросов на предоставление услуги i -типа, $i = 1, 2$:

$$B_1 = \{(n_1, n_2) \in X : n_1 = g \vee n_1 + n_2 = C\} \tag{4.2}$$

$$B_2 = \{(n_1, n_2) \in X : n_1 + n_2 = C\} \tag{4.3}$$

Множество приема запросов на предоставление услуги i -типа, $i = 1, 2$:

$$S_1 = \{(n_1, n_2) \in X : n_1 < g \vee n_1 + n_2 < C\} \tag{4.4}$$

$$S_2 = \{(n_1, n_2) \in X : n_1 + n_2 < C\} \tag{4.5}$$

СУГБ:

$$\begin{cases} 1)\, p(n_1,n_2)(\lambda_1+\lambda_2+(n_1+n_2)\mu) \\ \quad = p(n_1-1,n_2)\lambda_1 + p(n_1,n_2-1)\lambda_2 \\ \qquad + p(n_1+1,n_2)(n_1+1)\mu + p(n_1,n_2+1)(n_2+1)\mu, \\ \quad n_1 = \overline{1,g-1}, n_2 = \overline{1,C-1} \\ 2)\, p(n_1,0)(\lambda_1+\lambda_2+n_1\mu) \\ \quad = p(n_1-1,0)\lambda_1 + p(n_1+1,0)(n_1+1)\mu_1 + p(n_{1,1}\mu_2), \\ \qquad n_1 = \overline{1,g-1}; \\ 3)\, p(0,n_2)(\lambda_1+\lambda_2+n_2\mu) \\ \quad = p(0,n_2-1)\lambda_2 + p(1,n_2)\mu + p(0,n_2+1)(n_2+1)\mu, \\ \qquad n_2 = \overline{1,C-1} \\ 4)\, p(n_1,C-n_1)(n_1\mu+(C-n_1)\mu) \\ \quad = p(n_1-1,C-n_1)\lambda_1 + p(n_1,C-n_1-1)\lambda_2, \\ \qquad n_1 = \overline{1,g} \\ 5)\, p(0,0)(\lambda_1,\lambda_2) = p(1,0)\mu + p(0,1)\mu \\ 6)\, p(0,C)C\mu = \lambda_2 p(0,C-1) \\ 7)\, p(g,0)(\mu+\lambda_2) \\ \quad = p(g-1,0)\lambda_1 + p(g,1)\mu \\ 8)\, p(g,n_2)(\lambda_2+g\mu+n_2\mu) \\ \quad = p(g,n_2+1)(n_2+1)\mu + p(g-1,n_2)\lambda_1 + p(g,n_2-1)\lambda_2 \\ \qquad n_2 = \overline{0,c-g-1} \end{cases} \tag{4.6}$$

СУЛБ:

$$\begin{cases} p(n_1, n_2)n_1\mu_1 = p(n_1 - 1, n_2)\lambda_1, n_1 > 0, \\ p(n_1, n_2)n_2\mu_2 = p(n_1, n_2 - 1)\lambda_2, n_2 > 0, (n_1, n_2) \in X \end{cases} \tag{4.7}$$

Обозначим

$$\rho_1 = \frac{\lambda_i}{\mu_i}, i = 1, 2 \tag{1.1}$$

Стационарное распределение вероятностей состояний системы:

$$p(n_1, n_2) = p(0, 0) \cdot \frac{\rho_2^{n_2}}{n_2!} \cdot \frac{\rho_1^{n_1}}{n_1!} \tag{4.8}$$

где:

$$p(0, 0) = \left(\sum_{(n_1, n_2) \in X} \frac{\rho_2^{n_2}}{n_2!} \cdot \frac{\rho_1^{n_1}}{n_1!} \right)^{-1} \tag{4.9}$$

Основные вероятностные характеристики модели:

- Вероятность блокировки заявок на предоставление услуги 1-ог типа:

$$B_1 = \sum_{(n_1, n_2) \in B_1} p(n_1, n_2) \tag{4.11}$$

- Вероятность блокировки заявок на предоставление услуги 2-ог типа:

$$B_2 = \sum_{(n_1, n_2) \in B_2} p(n_1, n_2) \tag{4.12}$$

- Среднее число $\bar{N}$ обслуживаемых в системе запросов

$$\bar{N} = \sum_{(n_1, n_2) \in X} (n_1 + n_2) p(n_1, n_2) \tag{4.13}$$

2 Численный анализ

Для расчета основных вероятностных характеристик модели были взяты следующие параметры:

$$C = 4, g = 2, \lambda_1 = \overline{1, 20}, \lambda_2 = 1, \mu_1 = 2, \mu_2 = 1 \tag{2.1}$$

Диаграмма интенсивностей переходов имеет следующий вид (рис. 2.1):

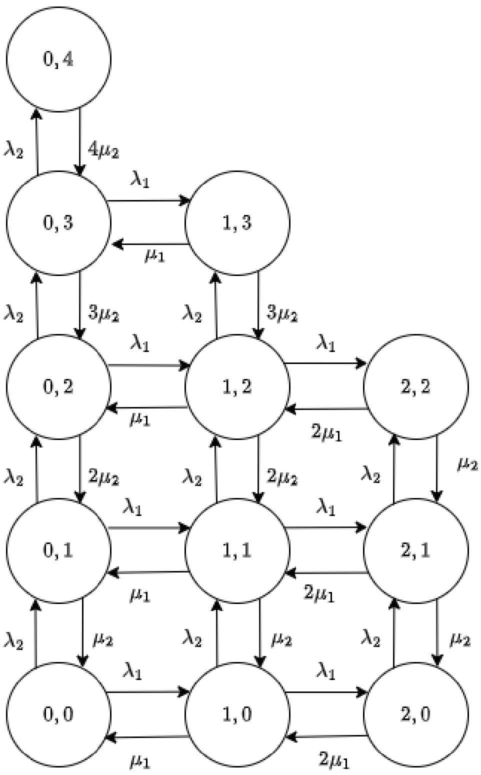

Рис. 2.1: Диаграмма интенсивностей переходов

Постранство состояний:

$$X = \{(n_1, n_2) : n_1 = \overline{0,2}, n_2 = \overline{0,4}, n_1 + n_2 \leq C\}, |X| = 12 \tag{2.2}$$

Множества блокировок запросов B_1, B_2:

$$\begin{gathered} B_1 = \{(n_1, n_2) \in X : n_1 = 2 \vee n_1 + n_2 = 4\}, \\ B_1 = \{(0,4), (1,3), (2,2), (2,1), (2,0)\} \end{gathered} \tag{2.3}$$

$$\begin{gathered} B_2 = \{(n_1, n_2) \in X : n_1 + n_2 = 4\}, \\ B_2 = \{(0,4), (1,3), (2,2)\} \end{gathered} \tag{2.4}$$

Множества приема запросов S_1, S_2:

$$\begin{gathered} S_1 = \{(n_1, n_2) \in X : n_1 < 2 \vee n_1 + n_2 < 4\}, \\ S_1 = \{(0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2)\} \end{gathered} \tag{2.5}$$

$$\begin{gathered} S_2 = \{(n_1, n_2) \in X : n_1 + n_2 < 4\}, \\ S_2 = \{(0,0), (0,1), (0,2), (0,3), (1,0), (1,1), (1,2), (2,0), (2,1)\} \end{gathered} \tag{2.6}$$

Код написан на языке Python в Google Colab:

[15]

```python
import math
import numpy as np
import matplotlib.pyplot as plt
```

Рис. 2.2: Импортируйте библиотеку

[16]

```python
C = 4
g = 2
lambda1 = 20
lambda2 = 1
mu1 = 2
mu2 = 1
rho1 = lambda1/mu1
rho2 = lambda2/mu2
```

Рис. 2.3: Начальные значения

[17]

```python
def compute_probabilities(C, rho1, rho2, g):
    p = np.zeros((g+1, C+1))

    for n1 in range(g+1):
        for n2 in range(C+1):
            if n1 + n2 <= C:
                p[0][0] += (rho1 ** n1) * (rho2 ** n2) / (math.factorial(n1) * math.factorial(n2))

    p[0][0] = 1 / p[0][0]

    for n1 in range(g+1):
        for n2 in range(C+1):
            if n1 == n2 == 0:
                continue
            if n1 + n2 <= C:
                p[n1][n2] = p[0][0] * (rho1 ** n1) * (rho2 ** n2) / (math.factorial(n1) * math.factorial(n2))
    return p
```

Рис. 2.4: Функция рассчета стационарного распределения вероятностей состояний систем $p_{(0,0)}$ и $p_{(n_1,n_2)}$

```
[18] p = compute_probabilities(C, rho1, rho2, g)
     print(f"Распределение вероятностей системы в исходном состоянии p(0,0) = {p[0][0]}")
```

```
Распределение вероятностей системы в исходном состоянии p(0,0) = 0.006477732793522267
```

Рис. 2.5: Стационарное распределение вероятностей системы в исходном состоянии $p_{(0,0)}$

```
[19] print(f"Распределение вероятностей для каждого состаянии:")
     for n1 in range(g+1):
         for n2 in range(C+1):
             print(f"p({n1},{n2}) = {p[n1][n2]}")
     total_p = np.sum(p)
     print(f"Сумма всех вероятностей: {total_p}")
```

```
Распределение вероятностей для каждого состаянии:
p(0,0) = 0.006477732793522267
p(0,1) = 0.006477732793522267
p(0,2) = 0.0032388663967611335
p(0,3) = 0.0010796221322537112
p(0,4) = 0.0002699055330634278
p(1,0) = 0.06477732793522267
p(1,1) = 0.06477732793522267
p(1,2) = 0.032388663967611336
p(1,3) = 0.010796221322537112
p(1,4) = 0.0
p(2,0) = 0.32388663967611336
p(2,1) = 0.32388663967611336
p(2,2) = 0.16194331983805668
p(2,3) = 0.0
p(2,4) = 0.0
Сумма всех вероятностей: 1.0
```

Рис. 2.6: Стационарное распределение вероятностей для каждого состоянии и сумма вероятностей

[20]
```
lambda1_values = np.arange(1, lambda1 + 1)
B1_values = []
B2_values = []
N_values = []

for lambda1 in lambda1_values:
    rho1 = lambda1 / mu1
    rho2 = lambda2 / mu2
    p = compute_probabilities(C, rho1, rho2, g)

    B1 = 0
    B2 = 0
    N = 0

    for n1 in range(g+1):
        for n2 in range(C+1):
            if n1 == g or n1 + n2 == C:
                B1 += p[n1][n2]
            if n1 + n2 == C:
                B2 += p[n1][n2]
            if n1 + n2 <= C:
                N += (n1 + n2) * p[n1][n2]

    B1_values.append(B1)
    B2_values.append(B2)
    N_values.append(N)

    if lambda1 == 20:
        print(f"Вероятность блокировки заявок на услуги 1-ог типа B1: {B1}")
        print(f"Вероятность блокировки заявок на услуги 2-ог типа B2: {B2}")
        print(f"Среднее число N обслуживаемых запросов N: {N}")
```

```
Вероятность блокировки заявок на услуги 1-ог типа B1: 0.820782726045884
Вероятность блокировки заявок на услуги 2-ог типа B2: 0.17300944669365723
Среднее число N обслуживаемых запросов N: 2.6191632928475035
```

Рис. 2.7: Вероятность блокировки заявок на услуги 1,2-ог типа B_1, B_2 и Среднее число $\bar{N}$

```
[21] plt.plot(lambda1_values, B1_values, label="$B_1$", color='red')
plt.plot(lambda1_values, B2_values, label="$B_2$", color='blue')
plt.title("Вероятность блокировки в зависимости от интенсивности запросов")
plt.xlabel("Интенсивности поступления запросов")
plt.ylabel("Вероятность блокировки")
plt.legend()
plt.show()
```

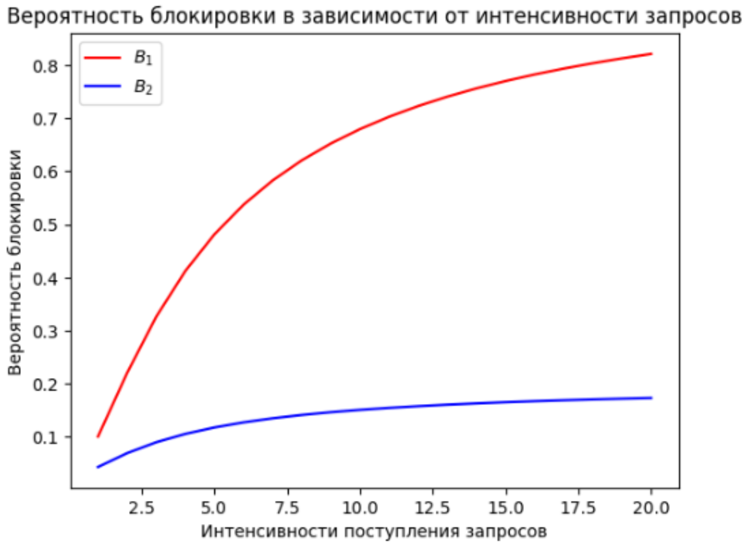

Рис. 2.8: Построение графика зависимости вероятности блокировки обслуженных запросов от интенсивности запросов для услуги $1, 2$ типа

```
[22] plt.plot(lambda1_values, N_values, label="$N$", color='blue')
     plt.title("Среднее число N обслуживаемых в зависимости от интенсивности запросов")
     plt.xlabel("Интенсивности поступления запросов")
     plt.ylabel("Среднее число запросов")
     plt.legend()
     plt.show()
```

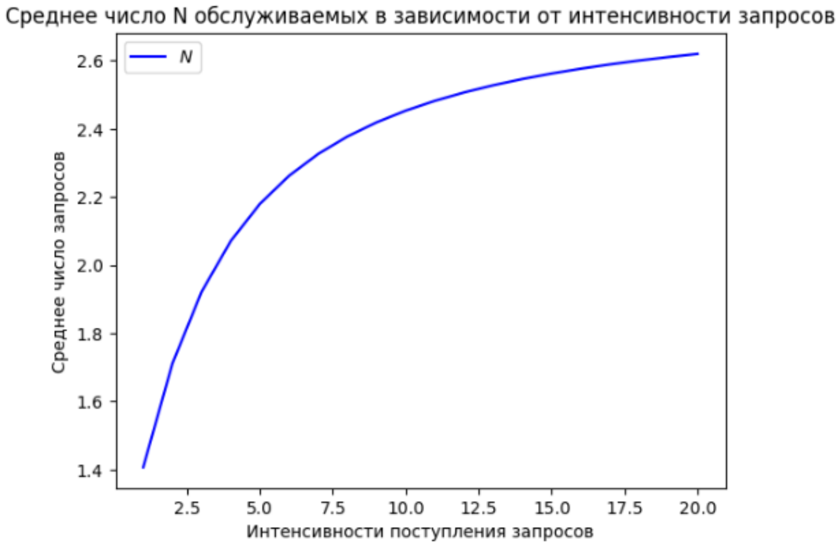

Рис. 2.9: Построение графика зависимости среднего числа обслуживаемых запросов от интенсивности поступления