Отчет по лабораторной работе №3: Двухсервисная модель с эластичным трафиком

Дисциплина: Основы проектирования сетей и систем телекоммуникаций

Выполнила: Ким Реачна, НПИбд-01-20

Содержание

Список иллюстраций

1 Теоретические сведения

Проаназируем соту сети емкостью C. Пусть пользователи генерируют запросы на передачу данных двух типов. Запросы на передачу данных представляют собой ПП с интенсивностью $\lambda_i, i = 1, 2$. Средняя длина передаваемого файла $\theta_i, i = 1, 2$. Минимальная емкость, необходимая для передачи данных равна $b_i, i = 1, 2$.

Основные обозначения:

- C - пиковая пропускная способность соты;
- $\lambda_i, i = 1, 2$ - интенсивность поступления запросов на передачу данных первого/второго типа [запросов/ед.вр];
- $\theta_i, i = 1, 2$ - длина передаваемого файла первого/второго типа [бит];
- $\rho_i^1, i = 1, 2$ - интенсивность предложенной нагрузки, создаваемой запросами на передачу данных первого/второго типа;
- $a_i, i = 1, 2$ - доля нагрузки,создаваемой запросами на передачу данных первого/второго типа, которая приходится на единицу пропускной способности (безразмерная величина);
- $b_i, i = 1, 2$ - минимальное требование к ресурсам сети, необходимое для передачи данных первого/второго типа;
- $X_i(t), i = 1, 2$ - число обслуживаемых в системе запросов на передачу данных первого/второго типа в момент времени $t, t \geq 0$;
- $X(t) = (X_1(t), (X_2(t))$ - СП, описывающий функционирование системы в момент времени $t, t \geq 0$;
- X - пространство состояний системы;

- $n_i, i = 1, 2$ - число передаваемых в системе блоков данных первого/второго типа;
- $B_i, i = 1, 2$ - множество блокировок запросов на передачу данных первого/второго типа;
- $S_i, i = 1, 2$ - множество приема запросов на передачу данных первого/второго типа.

Схема модели (рис. 1.1):

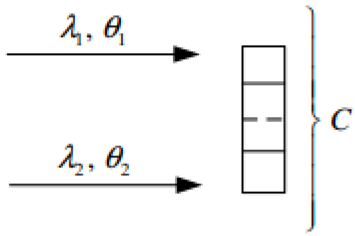

Рис. 1.1: Схема двухсервисной модели с эластичным трафиком

Пространство состояний системы (рис. 1.2):

$$X = \{(n_1, n_2) : n_1 \geq 0, n_2 \geq 0\}. \tag{11.1}$$

Рассмотрим некоторое центральное состояние $(n_1, n_2), (n_1, n_2) \in X$. Построим диаграмму интенсивностей переходов для центрального состояния (рис. 1.2):

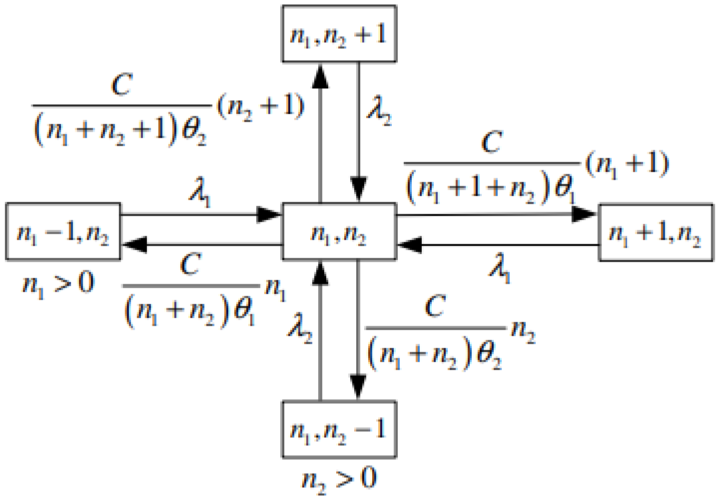

Рис. 1.2: Диаграмма интенсивностей переходов для центрального состояния двухсервисной модели с эластичным трафиком

Пояснения:

- $\frac{C}{n_1+n_2}$ - скорость передачи данных первого/второго типа в состоянии (n_1, n_2);
- $\frac{\theta_1}{\frac{C}{n_1+n_2}} = \frac{\theta_1}{C}(n_1+n_2)$ - среднее время обслуживания запроса на передачу данных первого типа в состоянии (n_1, n_2);
- $\frac{\theta_2}{\frac{C}{n_1+n_2}} = \frac{\theta_2}{C}(n_1+n_2)$ - среднее время обслуживания запроса на передачу данных второго типа в состоянии (n_1, n_2);
- $\frac{C}{\theta_1(n_1+n_2)}$ - интенсивность обслуживания запроса на передачу данных первого типа в состоянии (n_1, n_2);
- $\frac{C}{\theta_2(n_1+n_2)}$ - интенсивность обслуживания запроса на передачу данных второго типа в состоянии (n_1, n_2).

Множество блокировок запросов на передачу данных:

$$B_i = \{\emptyset\}, i = 1, 2 \tag{11.2}$$

Множество приема запросов на передачу данных:

$$S_i = \bar{B}_i = X \backslash B_i = \{0, 1, 2, ...\}, i = 1, 2. \tag{11.3}$$

Система уравнений глобального баланса (СУГБ):

$$\begin{aligned}
&\left(\lambda_1 + \lambda_2 + \frac{C}{(n_1 + n_2)\theta_1} n_1 + \frac{C}{(n_1 + n_2)\theta_2} n_2\right) .p(n_1, n_2) \\
&= \lambda_1 p(n_1 - 1, n_2) U(n_1) + \lambda_2 p(n_1, n_2 - 1).U(n_2) \\
&\quad + \frac{C}{(n_1 + 1 + n_2)\theta_1}(n_1 + 1) p(n_1 + 1, n_2) \\
&\quad + \frac{C}{(n_1 + n_2 + 1)\theta_2}(n_2 + 1) p(n_1, n_2 + 1), (n_1, n_2) \in X.
\end{aligned} \tag{11.4}$$

Чтобы выписать систему уравнений частичного баланса (СУЧБ), проверим критерий Колмогорова. Рассмотрим произвольный замкнутый контур (рис. 1.3):

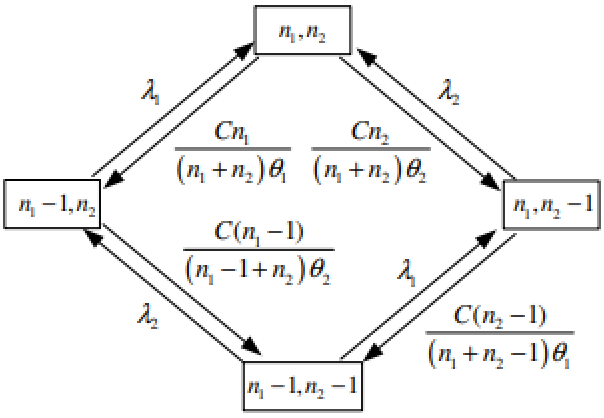

Рис. 1.3: Произвольный замкнутый контур для двухсервисной модели с эластичным трафиком

Рассмотрим произведение интенсивностей переходов:

- по часовой стрелке:

$$\frac{n_2}{n_1 + n_2} \frac{C}{\theta_2} \frac{n_1}{n_1 + n_2 - 1} \frac{C}{\theta_1} \lambda_1 \lambda_2; \tag{1.1}$$

- против часовой стрелки:

$$\frac{n_1}{n_1+n_2}\frac{C}{\theta_1}\frac{n_2}{n_1+n_2-1}\frac{C}{\theta_2}\lambda_1\lambda_2. \tag{1.2}$$

Произведения равны. Критерий выполнен, следовательно СП $(X_1(t), X_2(t))$, описывающий поведение системы является обратимым марковским процессом, СУЧБ существует.

СУЧБ:

$$\begin{cases} p(n_1,n_2)\frac{C}{(n_1+n_2)\theta_1}n_1 = \lambda_1 p(n_1-1,n_2), n_1>0, \\ p(n_1,n_2)\frac{C}{(n_1+n_2)\theta_2}n_2 = \lambda_1 p(n_1,n_2-1), n_2>0, (n_1,n_2)\in X. \end{cases} \tag{11.5}$$

Обозначим $\rho_i = \lambda_i\theta_i, a_i = \frac{\rho_i}{C}, \rho_i < C, i = 1,2$

Стационарное распределение вероятностей состояний системы:

$$p(n_1,n_2) = \frac{a_1^{n_1}}{n_1!}\frac{a_2^{n_2}}{n_2!}(n_1+n_2)!p(0,0), \tag{11.6}$$

где:

$$p(0,0) = \left(\sum_{(n_1,n_2)\in X}(n_1+n_2)!\frac{a_1^{n_1}}{n_1!}\frac{a_2^{n_2}}{n_2!}\right)^{-1} \tag{11.7}$$

Основные вероятностные характеристики модели:

- Вероятность блокировки по времени $E_i, i = 1,2$ запроса на передачу данных первого/второго типа:

$$E_1 = E_2 = 0; \tag{11.8}$$

- Среднее число $\bar{N}_i, i = 1,2$ обслуживаемых в системе запросов на передачу данных первого/второго типа:

$$\bar{N}_i = \lambda_i \frac{\theta_i}{(\theta_1 \lambda_1 + \theta_2 \lambda_2)}, i = 1, 2 \tag{11.9}$$

- Среднее время $T_i, i = 1, 2$ обслуживания запроса на передачу данных первого/второго типа:

$$T_i = \frac{\bar{N}_i}{\lambda_i} \tag{11.10}$$

2 Численный анализ

Для расчета основных вероятностных характеристик модели были взяты следующие параметры:

$$C = 50, \theta_1 = 4, \theta_2 = 7, \lambda_1 = 9, \lambda_2 = 6 \tag{2.1}$$

Код написан на языке Python в Google Colab:

```
[10]
import math
import numpy as np
import matplotlib.pyplot as plt
```

Рис. 2.1: Импортируйте библиотеку

```
[11]
C = 50 # computing unit
lambda1 = 9 # request type-1 arrival intensity [requests/second] unit
theta1 = 4 # type-1 data file size or length [bits]
theta2 = 7 # type-2 data file size or length [bits]
lambda2 = 6 # request of type-2 arrival intensity [requests/second] unit
rho1 = theta1*lambda1
rho2 = theta2*lambda2
a1 = rho1/C
a2 = rho2/C
```

Рис. 2.2: Начальные значения

```
[12]
def p_00():
    sum_n1n2 = 0.0
    for n1 in np.arange(C+1):
        for n2 in np.arange(C+1):
            sum_n1n2 += math.factorial(n1+n2) * (math.pow(a1, n1) / math.factorial(n1))* (math.pow(a2, n2) / math.factorial(n2))
    return 1.0 / sum_n1n2
p00 = p_00()
def p_(n1, n2):
    p = 0
    if n1 == 0 and n2 == 0:
        p = p00
    else:
        p = (math.pow(a1, n1) / math.factorial(n1))*(math.pow(a2, n2) / math.factorial(n2)) * math.factorial(n1 + n2) * p00
    return p
```

Рис. 2.3: Функция рассчета стационарного распределения вероятностей состояний систем $p_{(0,0)}$ и $p_{(n_1,n_2)}$

```
[13]
# print probability that central state is at origine (0, 0)
print("Распределение вероятностей системы в исходном состоянии: p00 = ", p00)

Распределение вероятностей системы в исходном состоянии: p00 =  1.055921997361764e-19
```

Рис. 2.4: Стационарное распределение вероятностей системы в исходном состоянии $p_{(0,0)}$

```python
# print all possible value of p_(n1, n2) in this system
p_n1n2_array = []
print("Распределение вероятностей для каждого состоянии [", end=None)
for n1 in np.arange(C+1):
    for n2 in np.arange(C+1):
        p = p_(n1, n2)
        print("({}, {}) : {}".format(n1, n2, p))
        p_n1n2_array.append(p)
print("]")
print("Сумма всех вероятностей p_(n1, n2) = ", sum(p_n1n2_array))
```

✓ 0.0s

```
Распределение вероятностей для каждого состоянии [
(0, 0) : 1.055921997361764e-19
(0, 1) : 8.869744777838817e-20
(0, 2) : 7.450585613384605e-20
(0, 3) : 6.258491915243068e-20
(0, 4) : 5.2571332088041783e-20
(0, 5) : 4.415991895395509e-20
(0, 6) : 3.709433192132227e-20
(0, 7) : 3.1159238813910707e-20
(0, 8) : 2.6173760603684995e-20
(0, 9) : 2.1985958907095393e-20
(0, 10) : 1.846820548196013e-20
(0, 11) : 1.551329260484651e-20
(0, 12) : 1.3031165788071067e-20
(0, 13) : 1.0946179261979695e-20
(0, 14) : 9.194790580062946e-21
(0, 15) : 7.723624087252873e-21
(0, 16) : 6.487844233292413e-21
(0, 17) : 5.4497891559656265e-21
(0, 18) : 4.5778228910111265e-21
(0, 19) : 3.845371228449346e-21
(0, 20) : 3.2301118318974503e-21
(0, 21) : 2.7132939387938583e-21
(0, 22) : 2.2791669085868412e-21
(0, 23) : 1.9145002032129465e-21
...
(50, 49) : 0.07634236153509992
(50, 50) : 0.12825516737896783
]
Сумма всех вероятностей p_(n1, n2) =  1.0
```

Рис. 2.5: Стационарное распределение вероятностей для каждого состоянии и сумма вероятностей

[28]

```python
mean_N1 = lambda1 * theta1 / (lambda1*theta1+lambda2*theta2)
mean_N2 = lambda2 * theta2 / (lambda1*theta1+lambda2*theta2)

print("Среднее число обслуживаемых запросов 1-ог типа N_1 : ", mean_N1)
print("Среднее число обслуживаемых запросов 2-ог типа N_2 : ", mean_N2)
```

```
Среднее число обслуживаемых запросов 1-ог типа N_1 :  0.46153846153846156
Среднее число обслуживаемых запросов 2-ог типа N_2 :  0.5384615384615384
```

Рис. 2.6: Среднее число обслуживаемых запросов первого/второго типа N_1, N_2

```
[29]
mean_T1 = mean_N1 / lambda1
mean_T2 = mean_N2 / lambda2

print("Среднее время обслуживания запроса 1-ог типа T_1= ", mean_T1)
print("Среднее время обслуживания запроса 1-ог типа T_2 = ", mean_T2)
```

```
Среднее время обслуживания запроса 1-ог типа T_1=  0.05128205128205129
Среднее время обслуживания запроса 1-ог типа T_2 =  0.08974358974358974
```

Рис. 2.7: Среднее время обслуживания запроса первого/второго типа T_1, T_2

```
[33]
lam1_array = [lam for lam in np.arange(1, lambda1 +1)]
T1_array = []
N1_array = []
T2_array = []
N2_array = []
for lambda1 in lam1_array:
    N1 = lambda1 * theta1 / (lambda1*theta1 + lambda2*theta2)
    N2 = lambda2 * theta2 / (lambda1 * theta1 + lambda2 * theta2)
    N1_array.append(N1)
    N2_array.append(N2)
    T1 = N1 / lambda1
    T2 = N2 / lambda2
    T1_array.append(T1)
    T2_array.append(T2)

plt.plot(lam1_array, N1_array, color="blue", label="$N_1$")
plt.plot(lam1_array, N2_array, color="green", label="$N_2$")
plt.title("Среднее число обслуживаемых в зависимости от интенсивности запросов")
plt.xlabel("Интенсивности поступления запросов")
plt.ylabel("Среднее число запросов 1-ог/2-ог типа $N_1$, $N_2$")
plt.legend()
plt.show()
```

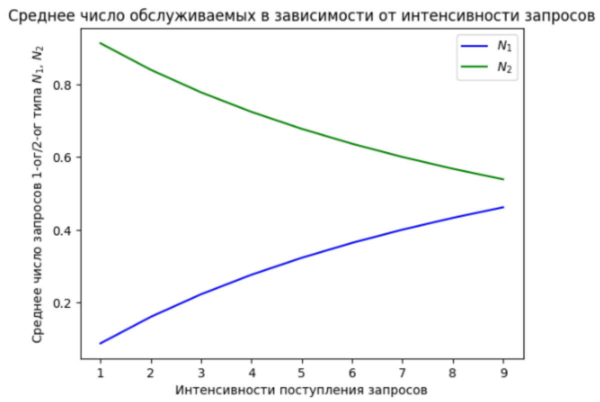

Рис. 2.8: Построение графика зависимости среднего числа обслуженных запросов от интенсивности запросов N_1, N_2

```
[35] plt.plot(lam1_array, T1_array, color="blue", label="$T_1$")
plt.plot(lam1_array, T2_array, color="green", label="$T_2$")
plt.title("Среднее время обслуживания в зависимости от интенсивности запросов")
plt.xlabel("Интенсивности поступления запросов")
plt.ylabel("Среднее время запроса 1-ог/2-ог типа $T_1$, $T_2$")
plt.legend()
plt.show()
```

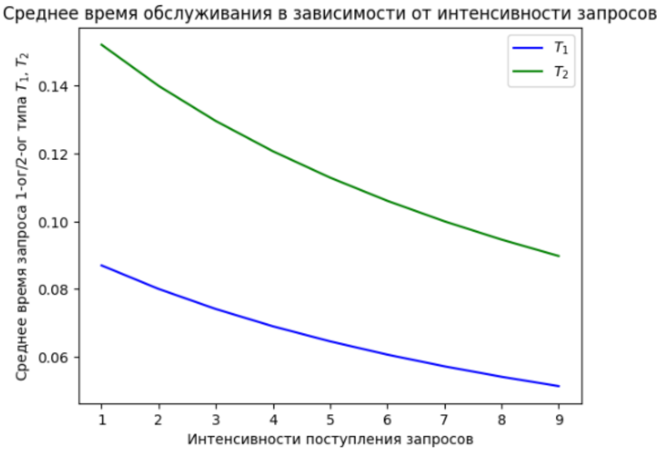

Рис. 2.9: Построение графика зависимости среднего времени обслуживания запросов от интенсивности запросов T_1, T_2