САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПЕТРА ВЕЛИКОГО

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

**РАСЧЁТНОЕ ЗАДАНИЕ**

**«Лабораторная работа 4»**

по дисциплине «Системный анализ и принятие решений»

Выполнил:

студент гр. 5130901/10101

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Тучков Д.А.

(подпись)

Преподаватель:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сиднев А.Г.

(подпись)

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Санкт-Петербург

2024

12. Исследование маршрутизатора глобальной компьютерной сети, обеспечивающего обмен данными по протоколу. Входящие пакеты записываются в буферную память маршрутизатора, обрабатываются процессором и передаются по каналу в соответствующем направлении. Копия принятого пакета хранится в буферной памяти до тех пор, пока не будет получена квитанция о безошибочной доставке пакета адресату. Отсутствие квитанции-подтверждения в течение времени time-out вынуждает маршрутизатор к повторной передаче пакета по тому же каналу в направлении адресата. При отсутствии места в буферной памяти входящие в маршрутизатор пакеты получают отказ.

Требуется построить модель маршрутизатора в форме сети массового обслуживания. При этом следует иметь в виду, что поступивший в маршрутизатор пакет занимает буферную память в течение следующих интервалов времени:

1) ввод в маршрутизатор и запись в буфер со скоростью канала связи,

2) ожидание в очереди на обработку в процессоре,

3) обработка в процессоре,

4) ожидание освобождения канала в направлении выхода из маршрутизатора,

5) передача из маршрутизатора по выходному каналу,

6) ожидание квитанции об удачной доставке с возможным повторением п. 5 (по истечении интервала time-out).

Каждая из отмеченных временных задержек пакета в маршрутизаторе обеспечивается ожиданием и обслуживанием в соответствующем узле сети массового обслуживания. Условия задачи приводят к разомкнутой сети с блокировками, которая может быть заменена стандартной замкнутой сетью массового обслуживания, характеризуемой тем же марковским процессом перехода из состояния в состояние [47].

Определить следующие характеристики маршрутизатора:

а) вероятность отказа в приёме пакета в буферную память,

б) среднее время пребывания пакета в буферной памяти,

в) среднее число пакетов в маршрутизаторе,

г) среднюю интенсивность потока пакетов, занимающих буферную память маршрутизатора,

д) зависимость указанных показателей от размера буферной памяти маршрутизатора.

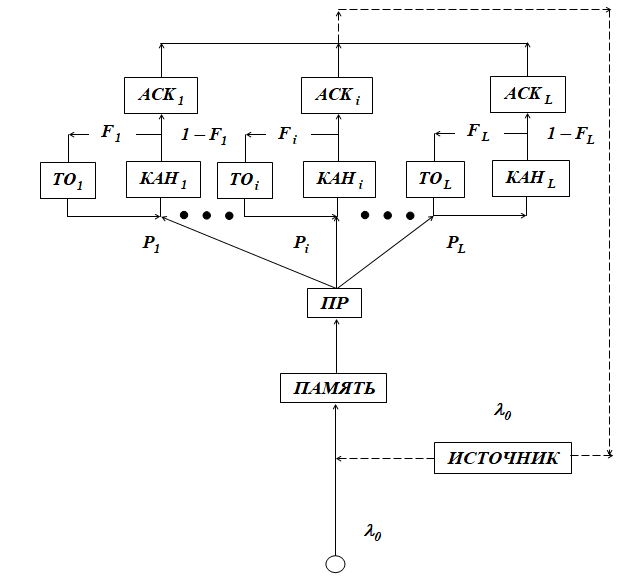
Варианты заданий приведены в табл. 12.20.

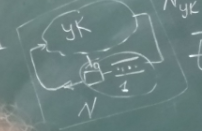
*Примечание.* Предполагается равновероятная передача пакета по любому из **** каналов, среднее время time-out принимается равным , время успешной доставки квитанции равно . Все временные задержки в маршрутизаторе, в том числе интервал time-out, и время доставки квитанции считаются случайными величинами, распределёнными по показательному закону. Входной поток пакетов — простейший.

Таблица 12.20

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№** задания | Интенсив-  ность  поступления  пакетов | Размер  буфера | Число  каналов | Интенсивность  передачи  пакетов  по каналу | Интенсив-  ность  обработки  пакета  в процессоре | Вероятность  неуспешной  передачи  пакета  по каналу |
| 5 | 9 | 25 | 9 | 0,8 | 15 | 0,1 |

Как сказано в задании мы переведем разомкнутую сеть к замкнутой

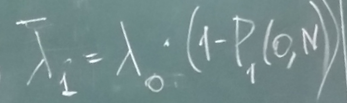




УК – узел коммутации

N=25 (размер буфера) в данном случае - число заявок в сети

Интенсивность поступления пакетов в закрытой системе:



****=9 (интенсивность поступления пакета с источника в разомкнутой системе)

P(0,N) - вероятность отсутствия пакетов на источнике

L = 9 –число каналов

= - вероятность передачи пакета по каналу I (Исходит из условия задачи) Для i= 1,2….,L

**** = = = 2,5 (Среднее время time out)

TD = = = 0,125

F = 0,1 (вероятность неуспешной передачи пакета по каналу)

Описание узлов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Источник | Память | Процессор | Канал передачи(КАНi) | ACKi | TOi |
| μ |  |  | μ\_пр | μ\_пер | 1/TU | 1/TO |
| ω | 1 | 1 | 1 |  |  |  |
| Число каналов(m) | 1 | N | 1 | 1 | N | N |

Программа

N = 25;

nodes = 30;

P = zeros(N+1, N+1); % создание матрицы из нулей

P(0+1,0+1) = 1; %P(0,0)

l0 = 9;

mu\_pam = l0;

mu\_per = 0.8; %интенсивность передачи пакетов по каналу

mu\_pr = 15;% интенсивность обработки пакета в процессоре

L=9; %Число каналов по условию задачи

mu = zeros(1,nodes);

TO = 2/mu\_per; %среднее время тайм оут

L\_TO = 1/TO; % интенсивность

TU = 0.1 / mu\_per; % время успешной доставки квитанции

L\_TU = 1/TU;% интенсивность

mu = [l0, mu\_pr,mu\_pam,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO, ];

m = [1,N,1,1,1,1,1,1,1,1,1,1,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N]; %число каналов

%расчет w

P\_k = (1/L) \* ones(1, L);

F = 0.1 \* ones(1, L);

w\_l = zeros(1, L);

w\_m = zeros(1, L);

w\_r = zeros(1, L);

w\_k0 = 1;

w\_l0 = 1;

w\_m0 = 1;

for i = 1:L

w\_l(i) = P\_k(i) / (1-F(i));

w\_m(i) = P\_k(i);

w\_r(i) = F(i) \* P\_k(i) / (1-F(i));

end

w = zeros(1,nodes);

w = [w\_k0,w\_l0,w\_m0, w\_l, w\_m, w\_r];

w\_sum = sum(w);

P\_0\_N = zeros(1,nodes);

%расчет по шагам

for i = 1:nodes

mu\_i = zeros(1,N);

%расчет mu

for n = 1:N

if n < m(i)

mu\_i(n) = n \* mu(i);

else

mu\_i(n) = m(i) \* mu(i);

end

end

t=zeros(1,N);

l1 = zeros(1,N);

for r = 1:N

%шаг 1

for n = 1:r

t(r) = t(r) + n/mu\_i(n) \* P(n-1+1, r-1+1);

end

%шаг 2

l1(r)= r / (w\_sum \* t(r) / w(1));

%шаг 3

sum\_P = 0;

for n = 1:r

%неизвестно какое w писать

P(n+1, r+1) = w(i) \* l1(r) \* P(n-1+1, r-1+1)/(w(1) \* mu\_i(n));

sum\_P = sum\_P + P(n+1,r+1);

end

P(0+1,r+1) = 1-sum\_P;

end

P\_0\_N(i) = P(0+1,N+1);

end

disp(P\_0\_N); % Вывод результата

n1 = 0;

for ni = 0:N

n1 = n1 + ni \* P(ni+1, N + 1);

end

l1 = l0\*(1-P\_0\_N(1));

N\_uk = N-n1;

t = N\_uk / (l0 \* (1-P\_0\_N(1)));

disp("a)вероятность отказа в приёме пакета в буферную память");

fprintf("P(0,N) =%.2f\n", P\_0\_N(1))

disp("б)среднее время пребывания пакета в буферной памяти");

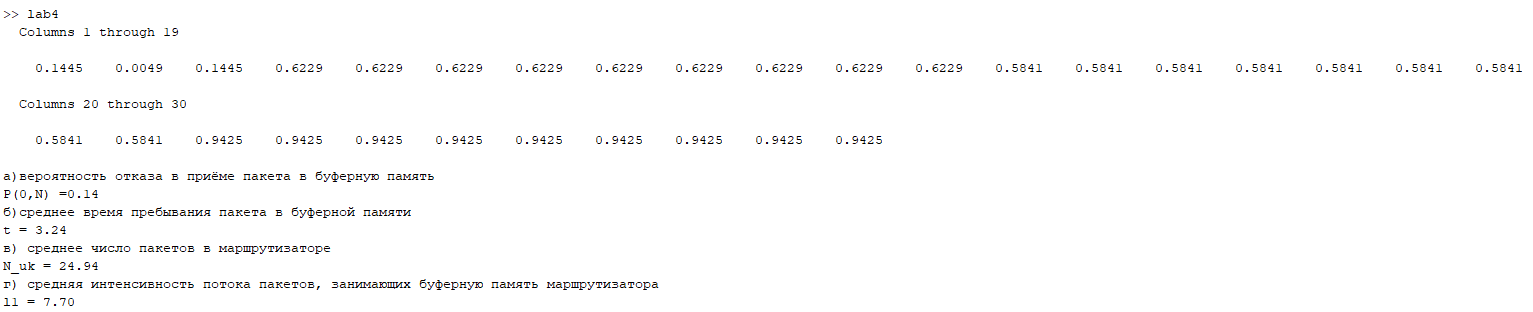
fprintf("t = %.2f\n", t)

disp("в) среднее число пакетов в маршрутизаторе");

fprintf("N\_uk = %.2f\n", N\_uk)

disp("г) средняя интенсивность потока пакетов, занимающих буферную память маршрутизатора");

fprintf("l1 = %.2f\n", l1)



Результаты

А) 0.14

Б) 3.24

В) 24.94

Г) 7.70

Графики

д) зависимость указанных показателей от размера буферной памяти маршрутизатора.

max\_bs = 50;

n1 = zeros(1,max\_bs);

l1\_res = zeros(1,max\_bs);

N\_uk = zeros(1,max\_bs);

t\_res = zeros(1,max\_bs);

P\_0\_N\_N = zeros(1,max\_bs);

P\_array = cell(1, nodes);

for N = 1:max\_bs

nodes = 30;

P = zeros(N+1, N+1); % создание матрицы из нулей

P(0+1,0+1) = 1; %P(0,0)

l0 = 9;

mu\_pam = l0;

mu\_per = 0.8; %интенсивность передачи пакетов по каналу

mu\_pr = 15;% интенсивность обработки пакета в процессоре

L=9; %Число каналов по условию задачи

mu = zeros(1,nodes);

TO = 2/mu\_per; %среднее время тайм оут

L\_TO = 1/TO; % интенсивность

TU = 0.1 / mu\_per; % время успешной доставки квитанции

L\_TU = 1/TU;% интенсивность

mu = [l0, mu\_pr,mu\_pam,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,mu\_per,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TU,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO,L\_TO, ];

m = [1,N,1,1,1,1,1,1,1,1,1,1,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N,N]; %число каналов

%расчет w

P\_k = (1/L) \* ones(1, L);

F = 0.1 \* ones(1, L);

w\_l = zeros(1, L);

w\_m = zeros(1, L);

w\_r = zeros(1, L);

w\_k0 = 1;

w\_l0 = 1;

w\_m0 = 1;

for i = 1:L

w\_l(i) = P\_k(i) / (1-F(i));

w\_m(i) = P\_k(i);

w\_r(i) = F(i) \* P\_k(i) / (1-F(i));

end

w = zeros(1,nodes);

w = [w\_k0,w\_l0,w\_m0, w\_l, w\_m, w\_r];

w\_sum = sum(w);

P\_0\_N = zeros(1,nodes);

%расчет по шагам

for i = 1:nodes

mu\_i = zeros(1,N);

%расчет mu

for n = 1:N

if n < m(i)

mu\_i(n) = n \* mu(i);

else

mu\_i(n) = m(i) \* mu(i);

end

end

t=zeros(1,N);

l1 = zeros(1,N);

for r = 1:N

%шаг 1

for n = 1:r

t(r) = t(r) + n/mu\_i(n) \* P(n-1+1, r-1+1);

end

%шаг 2

l1(r)= r / (w\_sum \* t(r) / w(1));

%шаг 3

sum\_P = 0;

for n = 1:r

%неизвестно какое w писать

P(n+1, r+1) = w(i) \* l1(r) \* P(n-1+1, r-1+1)/(w(1) \* mu\_i(n));

sum\_P = sum\_P + P(n+1,r+1);

end

P(0+1,r+1) = 1-sum\_P;

end

P\_0\_N(i) = P(0+1,N+1);

P\_array{i} = P;

end

%disp(P\_0\_N); % Вывод результата

P\_0\_N\_N(N) = P\_0\_N(1);

n1(N) = 0;

for ni = 0:N

n1(N) = n1(N) + ni \* P\_array{1}(ni+1, N + 1);

end

l1\_res(N) = l0\*(1-P\_0\_N(1));

N\_uk(N) = N-n1(N);

t\_res(N) = N\_uk(N) / (l0 \* (1-P\_0\_N(1)));

end

N = 1:max\_bs; % Значения N от 1 до 50

% Построение первого графика в первом окне

figure; % Создание нового окна

plot(N, P\_0\_N\_N, 'LineWidth', 2);

xlabel('N');

ylabel('P(0,N)');

title('График P(0,N)');

grid on;

% Построение второго графика во втором окне

figure; % Создание нового окна

plot(N, t\_res, 'LineWidth', 2);

xlabel('N');

ylabel('t');

title('График t');

grid on;

% Построение первого графика в первом окне

figure; % Создание нового окна

plot(N, N\_uk, 'LineWidth', 2);

xlabel('N');

ylabel('N\_uk');

title('График N\_uk');

grid on;

% Построение второго графика во втором окне

figure; % Создание нового окна

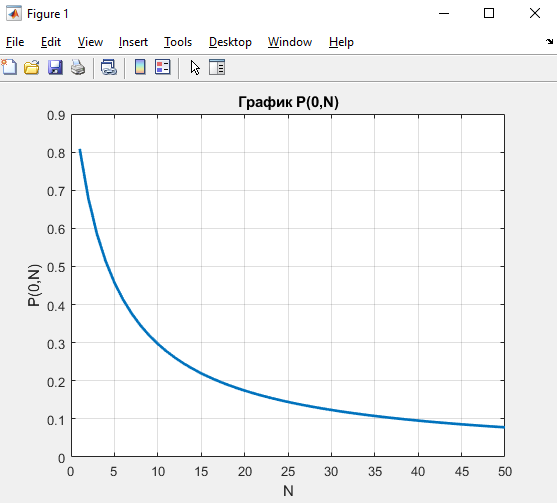
plot(N, l1\_res, 'LineWidth', 2);

xlabel('N');

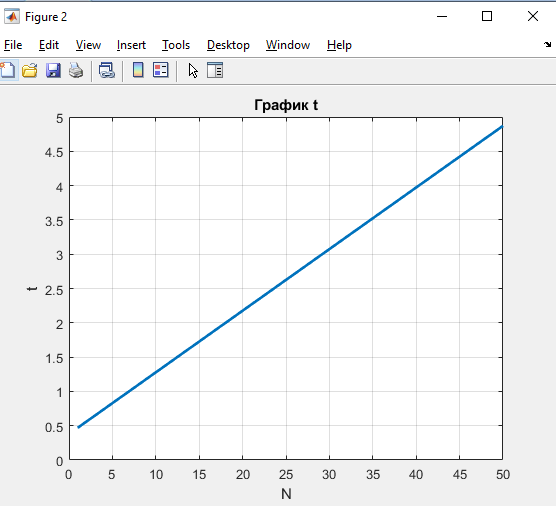
ylabel('l1');

title('График l1');

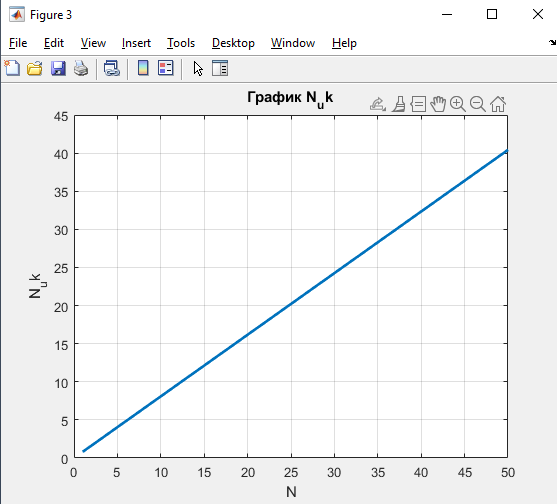
grid on;

а) вероятность отказа в приёме пакета в буферную память,  


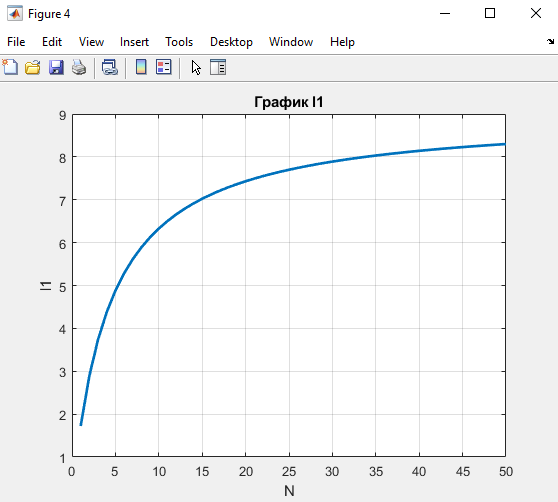
б) среднее время пребывания пакета в буферной памяти



в) среднее число пакетов в маршрутизаторе,



г) средняя интенсивность потока пакетов, занимающих буферную память маршрутизатора



Вывод

В ходе расчетной работы была реализована на Matlab рекурсивная процедура для многоканальной однородной замкнутой сети СМО. Благодаря которой были вычислены маргинальные распределения числа заявок в узлах однородной замкнутой сети. Что помогло в расчетах показателей требуемых в задаче.