МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»

Кафедра телекоммуникационных систем и вычислительных средств
(ТС и ВС)

Отчет по лабораторной работе №5
по дисциплине
Теория массового обслуживания

по теме:
МАРКОВСКИЕ ЦЕПИ. ИССЛЕДОВАНИЕ ЭРГОДИЧЕСКИХ СВОЙСТВ.

Студент:
Группа ИА-331 *Я.А Гмыря*

Предподаватель:
Преподаватель *А.В Андреев*

Новосибирск 2025 г.

СОДЕРЖАНИЕ

ЦЕЛЬ И ЗАДАЧИ

Цель: Исследовать свойства конечной дискретной, однородной цепи Маркова. Оценить параметры распределения числа коммутаций пакетов в сети.

Задание к лабораторной работе

Задание к лабораторной работе

1. Открыть рабочий скрипт MATLAB, созданный в лабораторной работе №1.3.
2. С помощью функции **MarkovTrajectory(P, N, s)**, реализованной выше, вычислить и отобразить в виде графика траекторию движения пакета по сети.
3. Задать точность, с которой будут производиться расчёты — **ε**.
4. Запрограммировать и рассчитать следующие величины:
   - Вероятность пребывания пакета в узле j после m коммутаций, при условии, что пакет поступил в сеть через узел i.
   - Вероятность первого перехода пакета в узел j из узла i после m коммутаций.
   - Длину кратчайшего пути перехода пакета в узел j из узла i.
   - Математическое ожидание длины пути перехода пакета в узел j из узла i.
   - Дисперсию длины пути перехода пакета в узел j из узла i (формулы 5.1 – 5.5).

Примечание: вычисления необходимо производить до тех пор, пока значения вероятностей первого перехода из узла i в узел j после m

33

коммутаций не станут меньше либо равны заданной точности ε ($\varepsilon \le 10^{-5}$).

Рисунок 1 — Задание для лабораторной работы

5. Построить зависимости:
    - Вероятности пребывания пакета в узлах от их номеров.
    - Вероятности первого перехода пакета в узел j из узла i после m коммутаций от номеров узлов.
    - Длины кратчайших путей переходов от номеров узлов.
    - Математические ожидания и дисперсии от номеров узлов.
6. Оформить полученные данные в виде рабочего листа MATLAB.
7. Сохранить файл в папке «Мои документы\ОТМО\», имя файла задать следующим образом: <Группа>.<Фамилия>.<№ лабораторной работы>.
8. Сдать и защитить работу.

Контрольные вопросы

1. Определение цепи Маркова.
2. Классификация цепей Маркова.
3. Свойства цепей Маркова.
4. Состояния цепи Маркова.
5. Дискретные и непрерывные цепи Маркова.
6. Вероятности перехода за m шагов.
7. Длины кратчайших путей перехода пакета.
8. Алгоритм функции вычисления траектории цепи Маркова.
9. Общие понятия о вычислительных сетях с коммутацией пакетов, и методах маршрутизации в них.

Рисунок 2 — Задание для лабораторной работы

ТЕОРИЯ

Основные сведения

Обозначения и расчетные формулы

L – количество узлов в сети с коммутацией пакетов (число состояний цепи Маркова)

$P_{i,j}$, i, j = $\overline{0, L}$, L - маршрутная матрица (матрица перехода однородной цепи Маркова); $P_{i,j}$ - вероятность передачи пакета узлом i в узел j (из лабораторной работы №4.3).

$P_{i,j}^{(m)}$, i, j = $\overline{0, L}$, m > 0 – вероятность пребывания пакета в узле j после m коммутаций при условии, что пакет поступил в сеть через узел j:

$$P^{(m)} = P^m. \tag{5.1}$$

$f_{i,j}^{(m)}$, i, j = $\overline{0, L}$, m > 0 – вероятность первого перехода пакета в узел j из узла i после m коммутаций:

$$f_{i,j}^{(m)} = P_{i,j}^{(m)} \times \prod_{q=1}^{m-1} 1 - P_{i,j}^{(q)}. \tag{5.2}$$

$M_{i,j}$, i, j = $\overline{0, L}$, m > 0 – длина кратчайшего пути перехода пакета в узел j из узла i:

$$M_{i,j} = \min_{f_{i,j}^{(m)} > 0} m\,. \tag{5.3}$$

$\overline{\mathrm{M}}_{i,j}$, i, j = $\overline{0, L}$, m > 0 – математическое ожидание длины пути перехода пакета в узел j из узла i:

$$\overline{\mathrm{M}}_{i,j} = \sum_{m=1}^{\infty} m \times f_{i,j}^{(m)}. \tag{5.4}$$

$D_{i,j}$, i, j = $\overline{0, L}$, m > 0 – дисперсия длины пути перехода пакета в узел j из узла i:

$$\overline{\mathrm{D}}_{i,j} = \sum_{m=1}^{\infty} m^2 \times f_{i,j}^{(m)} - \overline{\mathrm{M}}_{i,j}^2. \tag{5.5}$$

Рисунок 3 — Теория для лабораторной работы

Функция для расчета траектории движения пакета по сети

32

```
function E = MarkovTrajectory(P, N, s)
    % Функция для расчета траектории движения пакета по сети
    % P - матрица переходов
    % N - количество шагов
    % s - начальное состояние
    % Возвращает E - массив состояний пакета на каждом шаге

    % Инициализация
    E = zeros(1, N + 1); % Хранение траектории
    E(1) = s;            % Начальное состояние
    S = size(P, 1);      % Количество состояний в матрице P

    % Цикл по шагам
    for i = 1:N
        r = rand(); % Генерация случайного числа от 0 до 1
        cumulativeProb = 0; % Кумулятивная вероятность

        % Поиск следующего состояния
        for j = 1:S
            cumulativeProb = cumulativeProb + P(E(i), j); % Обновляем сумму вероятностей
            if r < cumulativeProb
                E(i + 1) = j; % Переход в состояние j
                break;
            end
        end
    end
```

Рисунок 4 — Теория для лабораторной работы

ХОД РАБОТЫ

Моделирование цепи Маркова. Визуализация переходов

Для лабораторной работы используется матрица переходов из лабораторной работы №1. Вот так выглядит цепь Маркова для такой матрицы переходов:

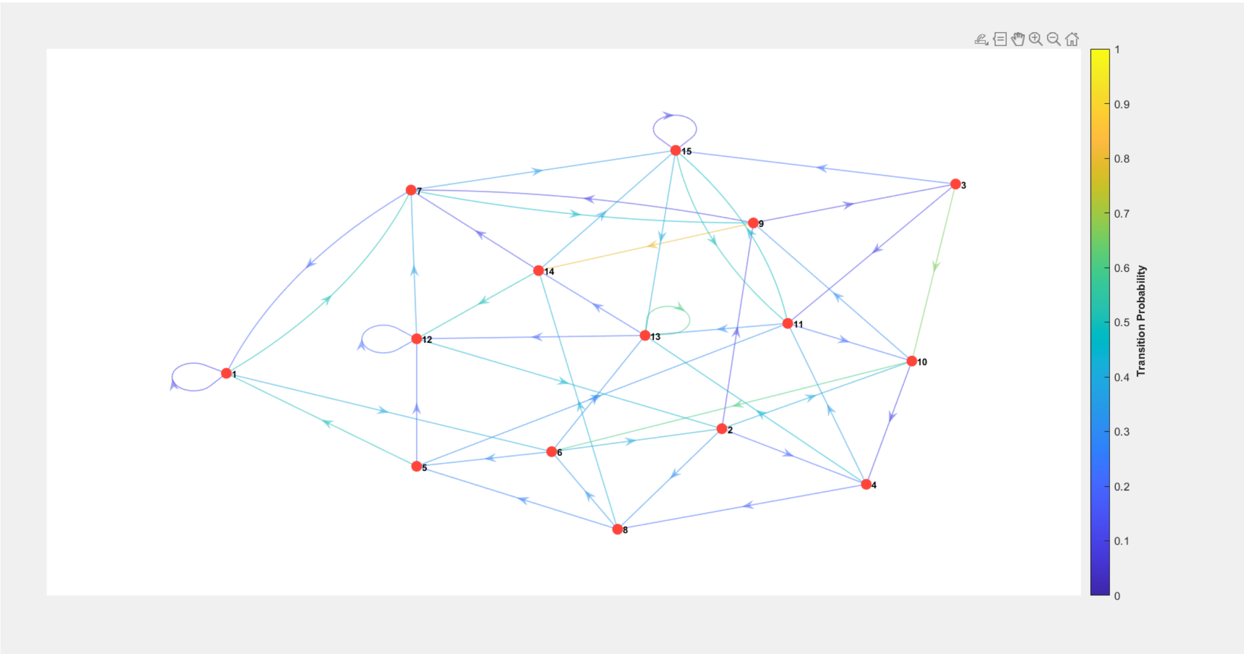

Рисунок 5 — Цепь Маркова. L=15

С помощью функции, данной в теории, промоделируем цепь Макрова, т.е посомтрим, как будут осуществляться переходы между состояниями. Для этого зададим функцию:

```
function E = MarkovTrajectory(P, N, s)
        E = zeros(1, N + 1);
        E(1) = s;
        S = size(P, 1);

        for i = 1:N
        r = rand();
        cumulativeProb = 0;

        for j = 1:S
            cumulativeProb = cumulativeProb + P(E(i), j);
            if r < cumulativeProb
```

```
                E(i + 1) = j;
                break;
            end
        end
    end
end
```

А потом вызовем ее в коде и получим вектор, содержащий состояния цепи Маркова в каждый момент времени, и построим график переходов.

```
%modeling markov chain
N = 400;
start = 1;
E = MarkovTrajectory(P, N, start);

%create and show plot
t = 0 : 1 : N;

figure;
plot(t, E);
```

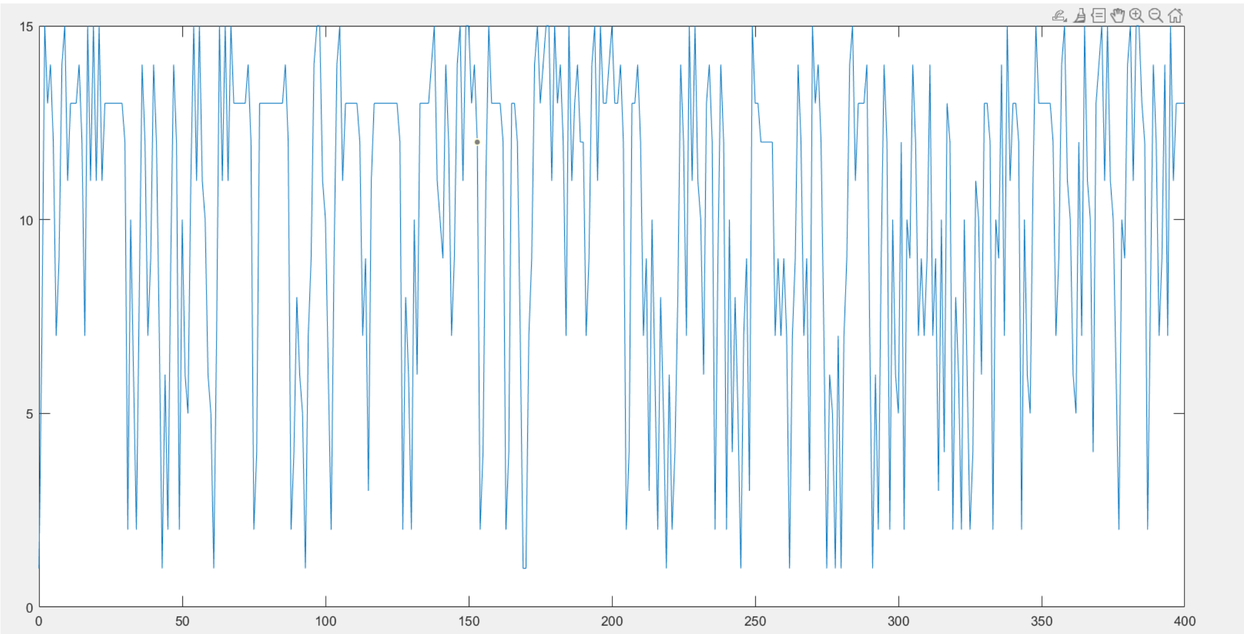

Рисунок 6 — График переходов между состояними в цепи Маркова

Можем наблюдать случайный процесс.

Вычислим вероятность пребывания пакета в узле j после m коммутаций, при условии, что пакет поступил в сеть через узел i

Для этого необходимо возвести матрицу переходов P в степень m и взять элемент P_{ij}. Почему матрица возводится в степень? Дело в том, что матрица умноженная саму на себя дает все комбинации переходов. Если просчитывать все комбинации отдельно, то в конечном счете получим матричное умножение P саму на себя.

Реализация функции:

```
function P_ij = func_1(P, m, i, j)
    Pm = P^m;
    P_ij = Pm(i, j);
end
```

Найдем вероятность перехода из состояния 2 в состояние 4 за 2 шага и выведем результат:

```
P_ij = func_1(P, 2, 2, 4);
disp(P_ij);
```

В результате получим 0.04. Если анализировать граф, то можно заметить, что такая ситуация может быть только при переходах 2->10->4. Вероятность перейти из 2 в 10 составляет 0.4, а перейти из 10 в 4 - 0.1. 0.4 * 0.1 = 0.04. Значит, все верно.

Вычислим вероятность первого перехода пакета в узел j из узла i после m коммутаций

Иными словами, m-1 шагов мы не попадали в узел j, а на шаге m попали. Т.е нам нужно на каждом m-1 шаге перемножать вероятность НЕ попасть в нужный узел с вероятностью попасть в узел на шаге m.

Реализация:

```
function res = func_2(P, m, i, j)
    k = 0;
    cum_mul = 1;
    P_tmp = P;
    while k < m - 1
```

```
        cum_mul = cum_mul * (1 - P_tmp(i,j));
        P_tmp = P_tmp * P;
        k = k + 1;
    end
    if m > 1
         P_tmp = P_tmp * P;
    end

    res = P_tmp(i,j) * cum_mul;
end
```

Найдем вероятность первого перехода из состояния 2 в состояние 4 за 4 шага и выведем результат:

```
P_m2 = func_2(P, 4, 2, 4);
disp(P_m2);
```

Получаем 0.142

Вычислим длину кратчайшего пути перехода пакета в узел j из узла i

Логично, что кратчайший путь - минимальное m, при котором $P_{ij}! = 0$. То есть необходимо перемножать матрицу на себя до тех пор, пока не найдется такое m. Если в графе одна компонента сильной связности (нет изолированной точки), то такое значение обязательно существует.

Реализация:

```
function res = min_len(P, i, j)
    m = 1;
    while P(i,j) == 0
        P = P * P;
        m = m + 1;
    end
    res = m;
end
```

Рассчитаем минимальный путь из узла 2 в узел 13:

```
min_l = min_len(P, 2, 13);
disp(min_l);
```

В результате получаем 2, что логично, ведь есть маршрут 2->4->13.

Вычислим математическое ожидание длины пути перехода пакета в узел j из узла i

Для вычисление мат.ожидания необходимо на каждом шаге коммутации умножать m (длину пути) на вероятность первого пребывания в узле P_{ij}. По сути это стандартное определение мат.ожидания. В идеале вычисления нужно производить бесконечно, чтобы получить максимально точный результат. Я выставил максимальное число переходов равным 500. Этого вполне хватает, чтобы мат.ожидание сошлось к какому-то значению.

Реализация:

```
function res = m(P, i, j)
    m = 1;
    max_m = 500;
    res = 0;
    while m <= max_m
        f = func_2(P, m, i, j);
        res = res + m * f;
        m = m + 1;
    end
end
```

Вычислим мат.ожидание для пути 2->13:

```
E_m = m(P, 2, 13);
disp(E_m);
```

Получаем значение 6.8, т.е нам понадобится в среднем 7 ходов, чтобы попасть из узла 2 в узел 13 при минимальной длине пути равной 2.

Вычислим дисперсию длины пути перехода пакета в узел j из узла i

Для вычисления необходимо просуммировать шаг коммутации в квадрате, умноженный вероятность первого пребывания пакета в узле. Потом от этой суммы нужно отнять мат.ожидание длины пути в квадрате. По сути тоже стандарное определение дисперсии. Максимальное значение m выставил равным 500.

Реализация:

```
function res = D(P, i, j)
    m = 1;
    max_m = 500;
    res = 0;
    while m <= max_m
        f = func_2(P, m, i, j);
        res = res + m^2 * f;
        m = m + 1;
    end

    res = res - M(P, i, j)^2;
end
```

Вычислим дисперсию для пути из узла 2 в узел 13:

```
D_E = D(P, 2, 13);
disp(D_E);
```

Получим 15.7. Это значит, что длина маршрута может отклоняться от средней длины на $+-\sqrt{15.7} \approx 3.9$ шагов.

Графики

Построим графики зависимости вычисленных характеристик от номеров узлов.

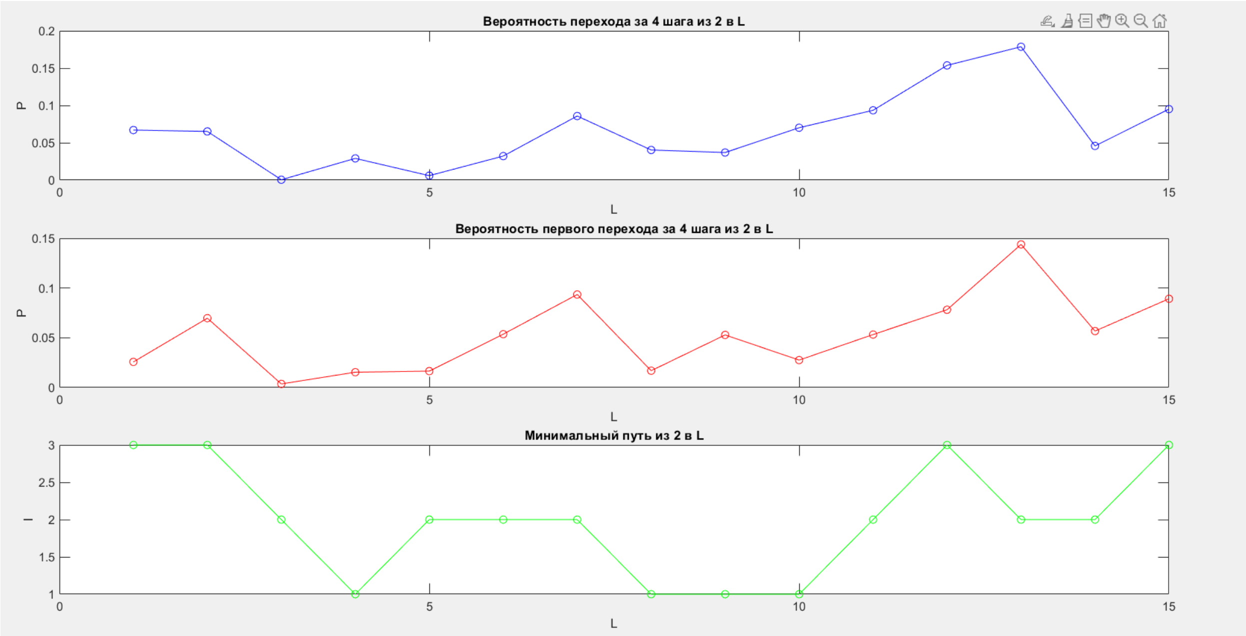

Рисунок 7 — При i = 2, j = [1,16]

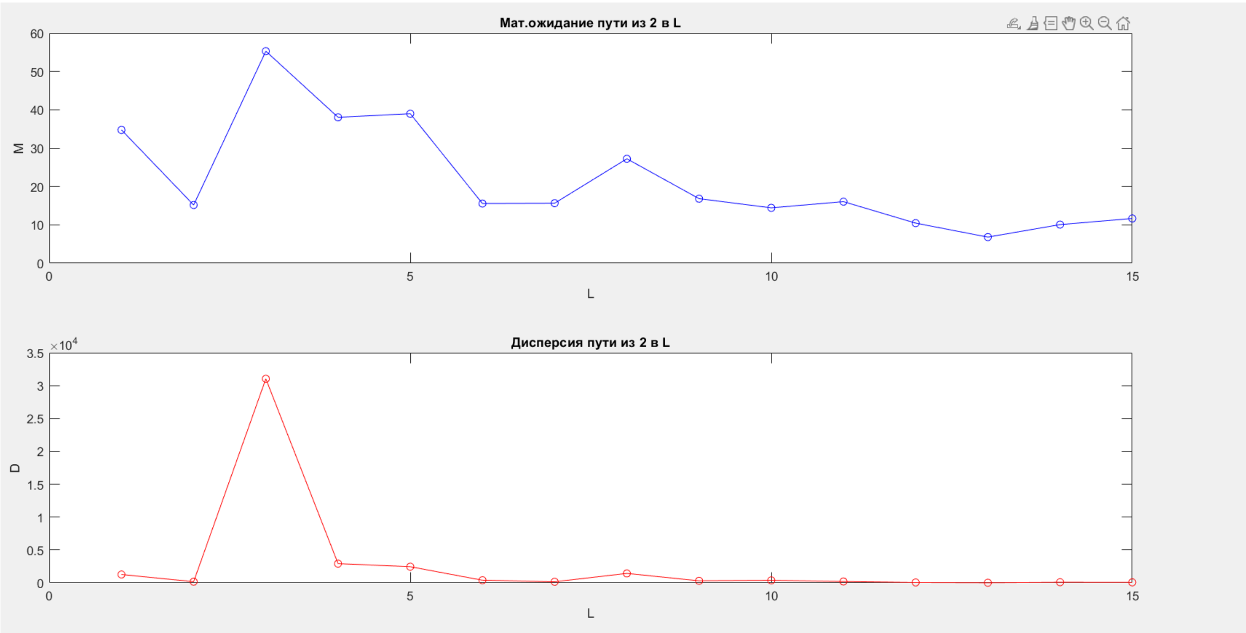

Рисунок 8 — При i = 2, j = [1,16]

Можем заметить, что какой-то взаимосвязи между номерами узлов и характеристиками нет. Характеристики ведут себя случайно. Влияние оказывает скорее сама матрица переходов и кол-во переходов m (в функциях, где она нужна). Может, я неудачно подобрал номера узлов? Построим графики при i = 13 и переменных j:

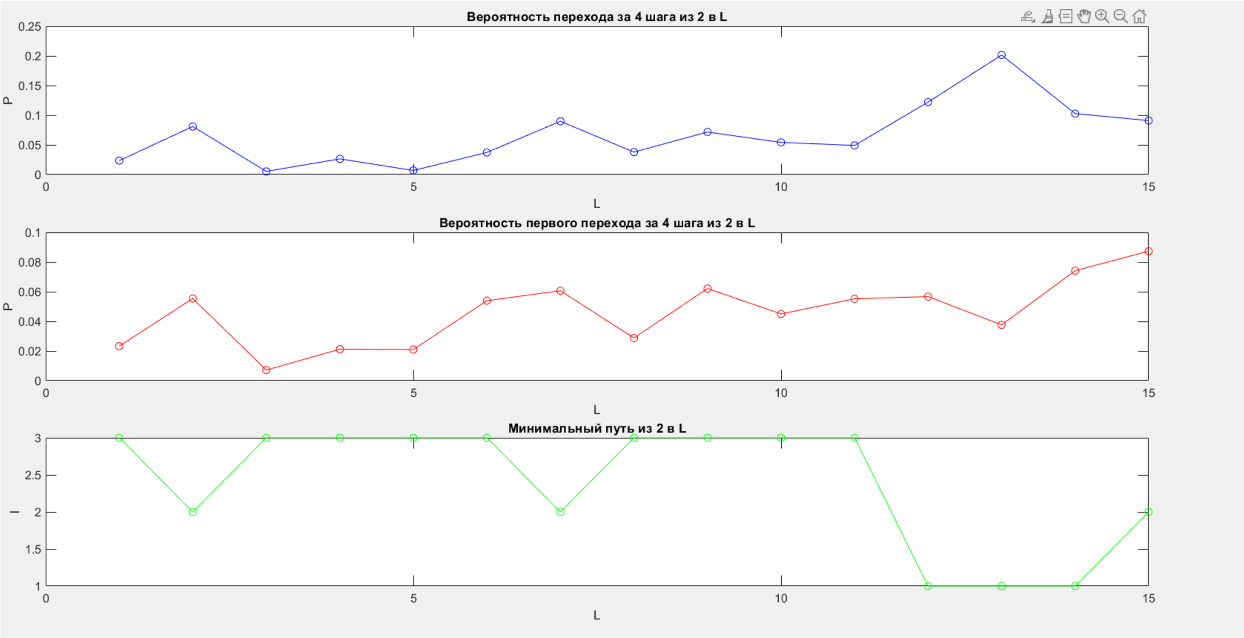

Рисунок 9 — При i = 13, j = [1,16]

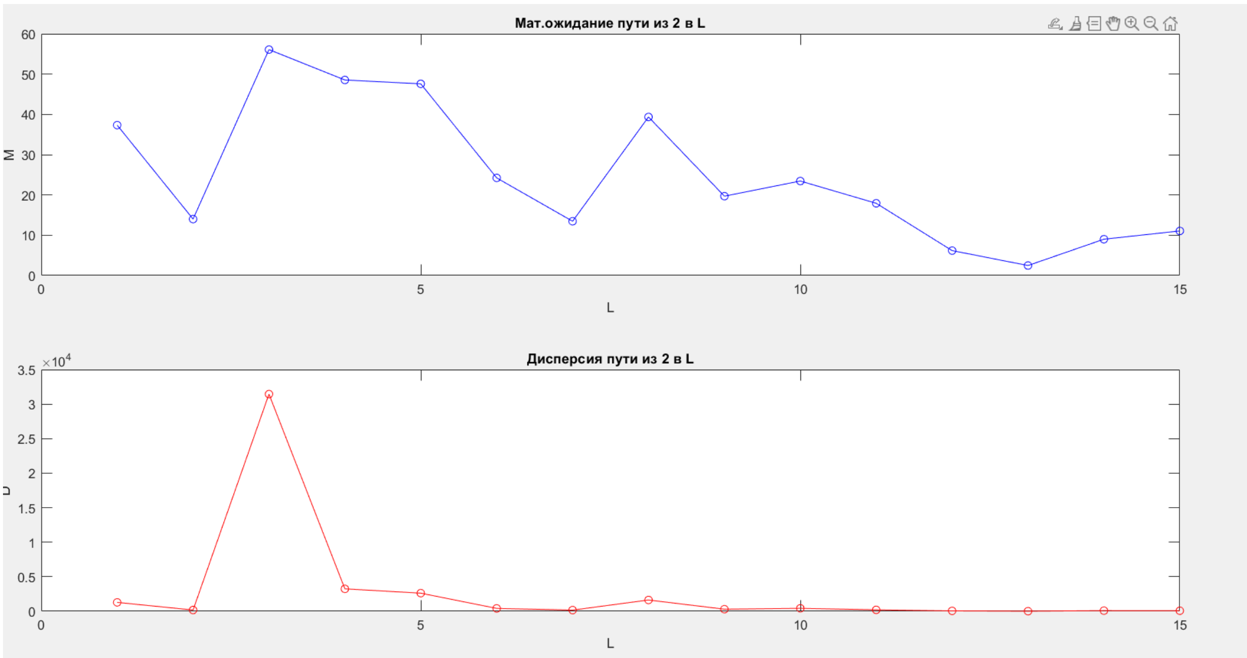

Рисунок 10 — При i = 13, j = [1,16]

Видим, что зависимости тоже нет, но ведут себя графики очень похоже. Это значит, что нет разницы, в какой точке пакет входит в сеть и в какую точку его необходимо доставить. На это влияют другие характеристики.

Контрольные вопросы

1. Определение цепи Маркова.
2. Классификация цепей Маркова.
3. Свойства цепей Маркова.
4. Состояния цепи Маркова.
5. Дискретные и непрерывные цепи Маркова.
6. Вероятности перехода за m шагов.
7. Длины кратчайших путей перехода пакета.
8. Алгоритм функции вычисления траектории цепи Маркова.
9. Общие понятия о вычислительных сетях с коммутацией пакетов, и методах маршрутизации в них.

Рисунок 11 — Контрольные вопросы к лабораторной работе

1. 1.Цепь Маркова – последовательность случайных событий с конечным или счётным числом исходов, характеризующаяся тем свойством, что, говоря нестрого, при фиксированном настоящем будущее независимо от прошлого.
2. 2. Цепи Маркова бывают однородные, т.е при шагах коммутации матрица переходов не изменяется (не зависит от шагов и времени). Также цепи Маркова бывают стационарные, т.е если цепь задана каким-то распределением, то это распределение остается неизменным. Неприводимая цепь Маркова - цепь Маркова, в которой из любого состояние можно попасть в любое другое за конечное число шагов. Эргодичная цепь Маркова - цепь Маркова, которая сходится к какому-то распределению.
3. 3. Состояния цепи Маркова - позиции, в которых может находиться система. Также это вершины графа.
4. 4. Дискретные цепи Маркова определены только в дискретные моменты времени (k = 0,1,2...). Непрерывные цепи Маркова определены в любой момент времени из области определения. Они обладают разными параметрами.
5. 5. Вероятность перехода за m шагов коммутации - это та вероятность, с которой система окажется в состоянии N за m шагов. На каждом шаге коммутации матрица перемножается сама на себя, та-

ким образом просчитываются все возможные комбинации (переходы).

6. 6. Длины кратчайших путей перехода пакета - минимальное число переходов для перехода из состояния i в состояние j. Вычисляется путем подбора такого m (шагов коммутации) при котором P_{ij} != 0.
7. 7. Алгоритм функции вычисления траектории цепи Маркова заключается в том, что в каждый дискретный момент времени вычисляется случайное число на отрезка [0;1] по равномерному распределнию и потом нужно подобрать такое следующее состояние системы k, при котором сгенерированное число будет меньше, чем вероятность перехода в состояние k. Это симулирует случайность при переходах.
8. 8. Сеть с коммутацией пакетов - сеть, в которой данные передаются не потоком, а пакетами, содержащими полезную нагрузку и служебную информацию. Основные методы маршрутизации RIP, OSPF, BGP, MPLS, Static Route.

ВЫВОД

В ходе работы я изучил свойства цепи Маркова путем моделирования переходов на языке matlab.