ГУАП

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

доцент, к. т. н.		Е. А. Бакин
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

Исследование алгоритмов множественного доступа

по курсу: Научно-исследовательская работа

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР.	4913М		Л.А. Шубырева
		подпись, дата	инициалы, фамилия

Санкт-Петербург
2020

Цель работы

Целью данной работы является получение навыков моделирования алгоритмов случайного множественного доступа в системах передачи сообщений абонентами:

- введением расписания;
- с вероятностью возникновения коллизий.

Выполнение работы

В данной работе рассматривается элементарная модель системы множественного доступа (Рисунок 1), где под множественным доступом подразумевается использование общего канала связи всеми абонентами. В рассматриваемой модели длительность сообщений от абонентов принято за единицу времени. Время разделено на равные интервалы – слоты, длительность которого равна длительности передаваемого сообщения. Предполагается, что абоненты начинают передачу сообщений только в начале слота. В соответствии с алгоритмом возможны три события.

1) Событие «успех». Если в слоте с индексом k передает только один абонент, считается, что сообщение доставлено успешно. Абонент, успешно передавший сообщение, покидает систему.

2) Событие «конфликт». Если в слоте k передают два и более абонентов, то в этом случае считается, что сообщение не доставлено – произошел конфликт. Абоненты остаются в системе и осуществляют попытки передачи сообщения в следующих слотах.

3) Событие «пусто». В слоте не передает ни один из абонентов.

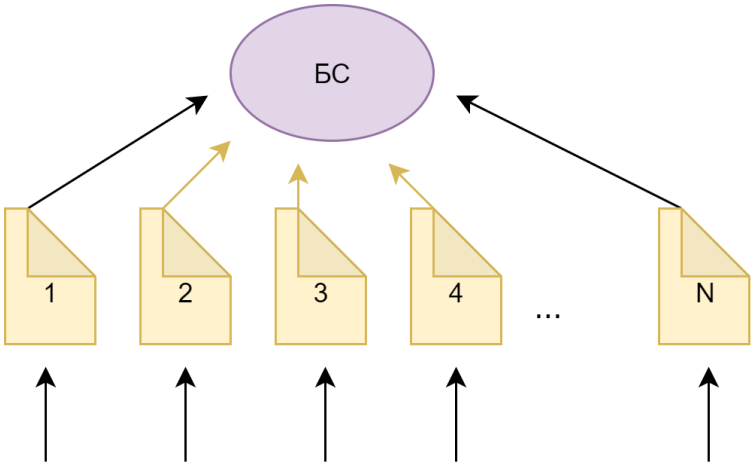

Рисунок 1: Примитивная схема работы БС

Для того, чтобы найти среднее количество сообщений в системе, мы найдем среднее количество сообщений во всех слотах.

При введении расписания каждому абоненту отведен номер слота, в котором он может отправлять сообщения.

Так как число абонентов ограничено, мы можем посчитать среднее количество сообщений одного абонента.

Результат выполнения программы

Из графика, представленного на рисунке 2, можно увидеть, что базовая станция, обрабатывающая заявки каждого абонента в заданном слоте, может работать при небольшом количестве абонентов. Иначе, при увеличении количества абонентов, БС не сможет справляться с потоком заявок, которые будут только накапливаться. Поэтому вероятность отправки сообщения не должна превышать 1/число_абонентов.

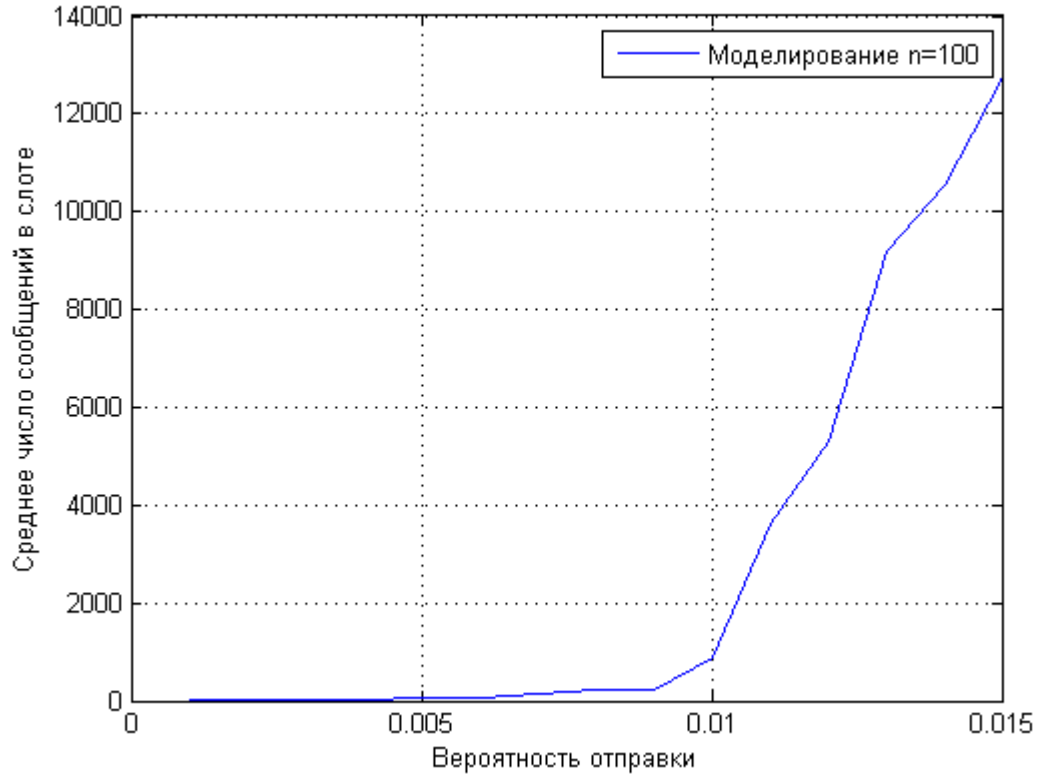

Рисунок 2: График зависимости полученного при моделировании среднего количества ожидающих обслуживания абонентов от интенсивности входного потока.

Во втором случае (Рисунок 3, Рисунок 4) возникновение коллизий невозможно будет избежать при p>= 5/число_абонентов, так как все абоненты отсылают почти в каждом слоте. Оптимальной вероятность можно считать, если она не будет превышать 1/число_абонентов.

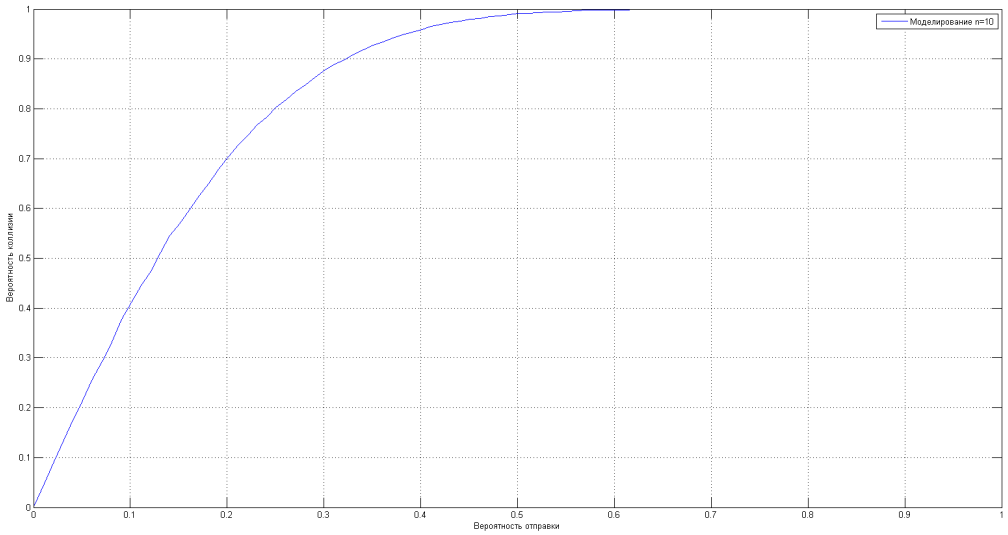

Рисунок 3: Вероятность возникновения коллизии от интенсивности входного потока при 10 абонентах

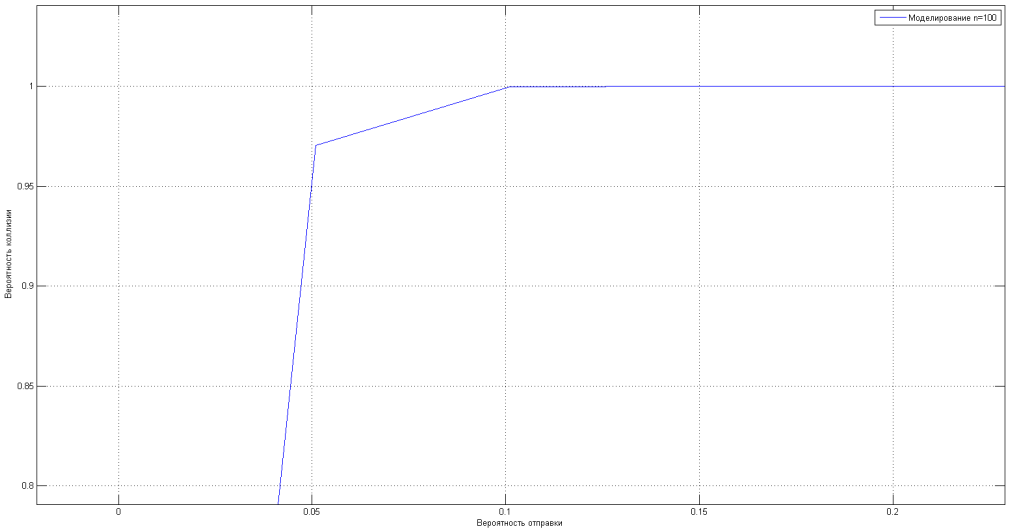

Рисунок 4: Вероятность возникновения коллизии от интенсивности входного потока при 100 абонентах

Вывод

Промоделировали работу элементарной системы, работающей по расписанию и без. Главным недостатком такой системы является то, что при большом количестве абонентов заявки будут бесконечно скапливаться. Также при низкой интенсивности входного потока система будет простаивать, а при большой — заявки будут накапливаться, так как период опроса абонентов большой.

Листинг исходного кода

Код написан на языке Matlab.

```
clear all
clc
n=100; % количество абонентов
p=6; % наш абонент
NumberSlots = 5*10^4; % количество слотов
Q = zeros(1, NumberSlots);% число сообщений в слоте от одного
абонента
%lam = 0.001:0.01:0.99; % вероятность отправки сообщения одним
абонентом
lam=0.001:0.001:0.015;
T = zeros(1,length(lam));% среднее число сообщений в слоте
Kolliz = zeros(1,length(lam)); % 2 part
for i = 1:length(lam)
    Q = zeros(1, NumberSlots);
    Koll = 0;
    Tell = 0;
    SlotFlow = randsrc(1, n, [1, 0; lam(i), 1-lam(i)]);% Все
генерируют в одном слоте
    if sum(SlotFlow) > 1 % количество отправивших в первом
слоте
        Koll = Koll +1; %возникновение коллизии
    end;
    if sum(SlotFlow) == 1
        Tell = Tell +1; % сообщение отправлено без ошибок
    end;
    InterFlow = randsrc(1, NumberSlots, [1, 0; lam(i), 1-
lam(i)]); %распределение случайной величины
    Q(1)=InterFlow(1); % число сообщений в слоте
    for slot=2:NumberSlots
         Q(slot) = Q(slot - 1)+ InterFlow(slot);
        if mod(slot, n) == p
            Q(slot)= Q(slot) -(Q(slot - 1) >0);
        end

        SlotFlow = randsrc(1, n, [1, 0; lam(i), 1-lam(i)]);
         if sum(SlotFlow) > 1
            Koll = Koll +1;
        end;
         if sum(SlotFlow) == 1
            Tell = Tell +1;
        end;
    end
    T(i) = n*sum(Q)/NumberSlots;
    Kolliz(i)= Koll/(Tell + Koll);
end
figure(1);
plot(lam, T,'b');
legend({'Моделирование n=100'});
```

```
grid on
%axis([0,0.015,0,20000])
xlabel('Вероятность отправки');
ylabel('Среднее число сообщений в слоте');

figure(2);
plot(lam, Kolliz,'b');
legend({'Моделирование n=100'});
grid on
xlabel('Вероятность отправки');
ylabel('Вероятность коллизии');
```