

Министерство науки и высшего образования РФ
Пензенский государственный университет
Кафедра «Вычислительная техника»

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовому проекту
по дисциплине «Методы моделирования»
на тему: «Имитационное моделирование распределённого приложения для
компьютерной сети»

Выполнил студент группы 22ВВП2:

Перкин П.О.

Приняли:

Зинкин С.А.

Федюнин Р.Н.

ХОР
23.01.25

Пенза 2024

ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет Вычислительной техники

Кафедра "Вычислительная техника"

"УТВЕРЖДАЮ"



Зав. кафедрой ВТ

М. А. Митрохин

« 11 » сентября 2024 г.

ЗАДАНИЕ

на курсовое проектирование по дисциплине

Методы моделирования

Студенту Терещу П.О.
Фамилия И.О.

Группа 22ВВП 2

Тема проекта: Имитационное моделирование распределенного приложения для компьютерной сети

Вариант № 21.

Исходные данные (технические требования) на проектирование:

Заданы распределенный алгоритм и граф межузловых связей в компьютерной сети с коммутацией пакетов сообщений. Требуется построить исполнимую или поведенческую имитационную модель передачи сообщений в данной сети с учетом задержек, образования очередей, скорости передачи пакетов по каналам связи. При выборе технических характеристик необходимо ориентироваться на TCP/IP сети и сети Ethernet.

Объем работы по курсу

1. Расчетная часть

Моделирование на C++, Python, GPSS WORLD и/или PIPE. Требуется построить и отладить приложение или имитационную модель системы коммуникаций с передачей пакетов. Определить задержки в передаче пакетов, загрузку каналов связи. Собрать статистические данные о работе сети

2. Графическая часть

2.1 Структура моделируемой системы

2.2 Алгоритм моделирования

2.3 Результаты моделирования: гистограммы, графики и т.д.

3. Экспериментальная часть

Машинные эксперименты с моделью на C++/#, Python, PIPE, GPSS WORLD. Сбор статистических данных о работе сети.

4. Этапы выполнения проекта

По графику учебного процесса:

1. Модель приложения на C++/#, Python, или Java.
2. Модель алгоритма на языке GPSS WORLD.
3. Модель системы коммуникаций на языке GPSS WORLD.
4. Верификация моделей при помощи PIPE (сети Петри).
5. Оформление записки.

Дата выдачи задания 11 сентября 2024 г.

Дата защиты проекта: по графику учебного про

Руководители _____ /Зинкин С.А., Федюнин Р.Н.

Задание получил "11" сентября 2024 г.

Студент Беркин П.О. / Р

Содержание

Введение	5
1 Постановка задачи.....	6
2 Графическая часть	7
3 Реализация модели на языке Python	8
4 Моделирование модели на языке Python	9
5 Реализация алгоритма на языке GPSS World.....	11
6 Тестирование GPSS-модели конечного автомата.....	13
7 Построение модели с использованием сетей Петри.....	18
8 Моделирование построенной сети Петри.....	20
Заключение.....	22
Список используемой литературы.....	23
Приложение А. Код программы на языке Python.....	24
Приложение Б. Код программы на языке GPSS World	27
Приложение В. Отчёт результата моделирования на GPSS World.....	31
Приложение Г. Моделирование с использованием сетей Петри	35

Введение

Конечный автомат — математическая модель дискретной системы, которая обладает одним входом, одним выходом и в любой момент времени пребывает в одном из возможных состояний. При работе автомата на его вход поступают входные сигналы, и, в зависимости от текущего состояния автомата, на выходе формируются выходные сигналы.

Конечные автоматы (КА) — это математические модели, широко используемые для описания и анализа дискретных систем с конечным числом состояний. Они находят применение в различных областях, включая: программирование и разработку программного обеспечения, обработку последовательностей символов, например, в компиляторах и интерпретаторах, проектирование цифровых схем, моделирование процессов

В рамках данного курсового проекта будут рассмотрены различные методы создания конечных автоматов, а именно: написание программы на языке программирования Python, моделирование с использованием языка GPSS World, а также моделирование с использованием сетей Петри.

1 Постановка задачи

Заданы распределённый алгоритм и граф межузловых связей в компьютерной сети с коммутацией пакетов сообщений. Требуется построить исполнимую или поведенческую имитационную модель передачи сообщений в данной сети с учётом задержек, образования очередей, скорости передачи пакетов по каналам связи. При выборе технических характеристик необходимо ориентироваться на TCP/IPсети и сети Ethernet.

2 Графическая часть

Согласно варианту задания №21 была дана граф-схема конечного автомата (см. рисунок 1).

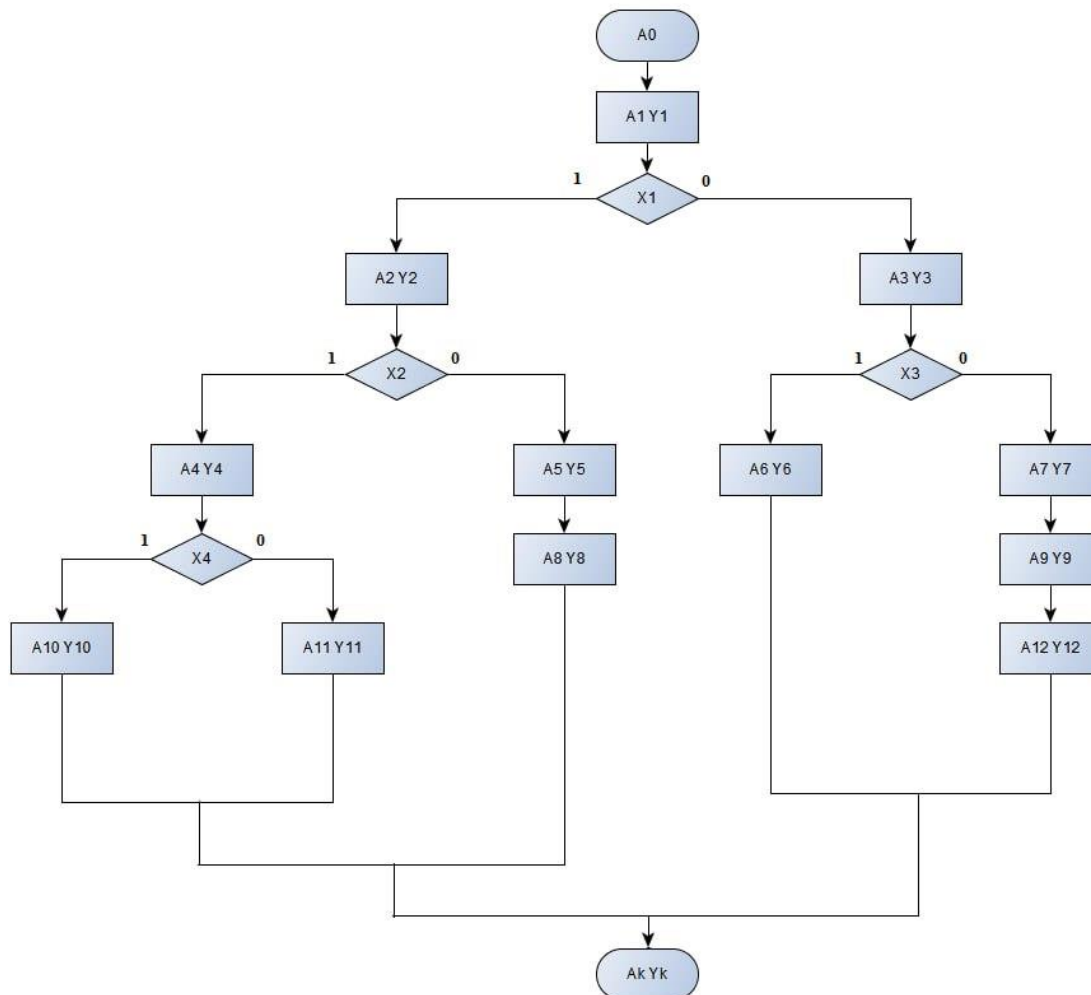


Рисунок 1 – ГСА автомата Мура.

На граф-схеме автомата представлены такие состояния, как: A_0 , A_1 , A_2 , A_3 , A_4 , A_5 , A_6 , A_7 , A_8 , A_9 , A_{10} , A_{11} , A_{12} , A_k . Условия переходов определены, как: X_1 , X_2 , X_3 , X_4 .

Автомат имеет начальное и конечное состояние A_0 и A_k , соответственно. Следующее состояние автомата A_i зависит от текущего состояния автомата, а также при переходе из некоторых состояний в следующее от входного сигнала и соответственно выполнения условия X_i .

3 Реализация модели на языке Python

В соответствие с граф-схемой автомата Мура был реализован распределённый алгоритм для сети на языке Python. Состояния автомата в алгоритме представлены функциями, каждая из которых соответствует определенному состоянию. Например:

```
def a2(signals):  
    signals = signals + 'Y2'  
    a = int(input(f'Вых.Сигналы: {signals}\nСостояние  
автомата: a2\nВходной сигнал X2 равен:'))  
    if a == 0:  
        return a5(signals)  
    elif a == 1:  
        return a4(signals)
```

В качестве параметра в функцию передается массив `signals`, используемый для хранения выходных сигналов. Если на переходе из текущего состояния в следующее присутствует условие, то у пользователя запрашивается входной сигнал, равный «1» или «0», и в зависимости от выполнения условия происходит переход. Если условия нет, то автомат переходит в следующее состояние или завершение программы, если это конечное состояние. Во время выполнения функции массив выходных сигналов заполняется в зависимости от текущего состояния. Программа выводит содержимое массива `signals` на экран каждый раз после перехода в новое состояние.

4 Моделирование модели на языке Python

Реализованную программу необходимо протестировать на разных наборах входных данных для проверки:

- корректного перехода из текущего состояния в следующее в зависимости от входного сигнала и текущего состояния;
- корректного ввода входного сигнала автомата;
- корректного вывода текущего состояния автомата;
- корректного вывода выходных сигналов автомата.

Программа была протестирована на разных наборах данных. Результат моделирования представлен на рисунках 2-3.

```
Вых.Сигналы:  
Состояние автомата: a0  
Вых.Сигналы: Y1  
Состояние автомата: a1  
Входной сигнал X1 равен:  
1  
Вых.Сигналы: Y1Y2  
Состояние автомата: a2  
Входной сигнал X2 равен:  
0  
Вых.Сигналы: Y1Y2Y5  
Состояние автомата: a5  
Вых.Сигналы: Y1Y2Y5Y8  
Состояние автомата: a8  
Вых.Сигналы: Y1Y2Y5Y8Yk  
Состояние автомата: ak
```

Рисунок 2 – Результат первого моделирования, набор выходных сигналов и состояний корректен.

Вых.Сигналы:
Состояние автомата: a0
Вых.Сигналы: Y1
Состояние автомата: a1
Входной сигнал X1 равен:
0
Вых.Сигналы: Y1Y3
Состояние автомата: a3
Входной сигнал X3 равен:
0
Вых.Сигналы: Y1Y3Y7
Состояние автомата: a7
Вых.Сигналы: Y1Y3Y7Y9
Состояние автомата: a9
Вых.Сигналы: Y1Y3Y7Y9Y12
Состояние автомата: a12
Вых.Сигналы: Y1Y3Y7Y9Y12Yk
Состояние автомата: ak

Рисунок 3 – Результат второго моделирования, набор выходных сигналов и состояний корректен.

5 Реализация алгоритма на языке GPSS World

В соответствие с граф-схемой автомата был реализован конечный автомат на языке GPSS. Для этого были использованы блоки, каждый из которых соответствует состоянию автомата. Например:

```
A_1 SEIZE AA1  
    ADVANCE 1  
    RELEASE AA1  
    ASSIGN 1,2  
    TABULATE TAB  
    TRANSFER .5,A_2,A_3
```

Когда транзакт переходит в блок состояния A_1, оно занимает устройство AA1 и освобождает его через 1 единицу времени, после чего присваивает параметру P1 число 2, так как это второе по счету состояние и с вероятностью 50% переходит в состояние A_2 или в состояние A_3.

Для сбора статистики о количестве переходов в каждое состояние и времени прохождения транзактом автомата объявлены две таблицы - TAB и TIME.

```
TAB TABLE P1,0,1,16  
TIME TABLE M1,0,10,20
```

Когда выполняется команда TABULATE TAB, текущий параметр P1 транзакции используется для определения, в какой интервал таблицы TAB записать попадание. Например, если значение параметра P1 будет равняться «1», то попадание запишется в интервал 1-2.

Временной параметр M1 измеряет время пребывания транзакции в автомате. Время фиксируется в момент создания транзакции, а при выполнении команды TABULATE TIME рассчитывается разница между текущим временем и временем создания транзакции, параметр M1 записывается в таблицу в соответствующем диапазоне.

С помощью блока MET1 реализована передача пакета от сервера до клиента. Передача пакета будет осуществляться при переходе из состояния 0 в состояние 1.

```
MET1 GATE NU YZ1
```

```
GATE NU KAN
GATE NU YZ2
TRANSFER SIM, ,MET1
SEIZE YZ1
SEIZE KAN
SEIZE YZ2
ADVANCE 5
RELEASE YZ2
RELEASE KAN
RELEASE YZ1
```

Блок GATE с режимом работы NU проверяет доступность узлов YZ1, YZ2, а также канала KAN. Если окажется, что все три устройства свободны, тогда блок SEIZE захватит устройства YZ1, YZ2, KAN. Иначе будет ожидать, пока не освободятся все три устройства, что достигается за счет блока TRANSFER в режиме работы SIM. После задержки в 5 единиц времени канал и узлы освобождаются при помощи оператора RELEASE и происходит переход в следующее состояние, а именно A_1.

6 Тестирование GPSS-модели конечного автомата

Протестируем полученный алгоритм на языке GPSS World для проверки корректности работы созданной модели дискретного устройства, тестирование включает в себя:

- проверку правильного изменения блока состояния автомата в зависимости от перехода и текущего блока состояния автомата;
- анализ полученного отчета моделирования;
- сбор и оценку статистики количества переходов в каждое состояние и скорости прохождения транзактом автомата.

Тестирование GPSS-модели было проведено несколько раз при количестве транзактов равном: 1000, 50000, 100000. Время задержки между транзактами равно 1000.

Моделирование при количестве транзактов, равном 10000.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	1000011.000	96	17	0

Рисунок 4 – Отчет о количестве использованных ресурсов и времени выполнения моделирования.

FACILITY	ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
AA0	1000	0.001	1.000	1		0	0	0	0	0
YZ1	1000	0.005	5.000	1		0	0	0	0	0
KAN	1000	0.005	5.000	1		0	0	0	0	0
YZ2	1000	0.005	5.000	1		0	0	0	0	0
AA1	1000	0.001	1.000	1		0	0	0	0	0
AA2	534	0.001	1.000	1		0	0	0	0	0
AA5	284	0.000	1.000	1		0	0	0	0	0
AA8	284	0.000	1.000	1		0	0	0	0	0
AAK	1000	0.001	1.000	1		0	0	0	0	0
AA3	466	0.000	1.000	1		0	0	0	0	0
AA6	223	0.000	1.000	1		0	0	0	0	0
AA4	250	0.000	1.000	1		0	0	0	0	0
AA11	133	0.000	1.000	1		0	0	0	0	0
AA10	117	0.000	1.000	1		0	0	0	0	0
AA7	243	0.000	1.000	1		0	0	0	0	0
AA9	243	0.000	1.000	1		0	0	0	0	0
AA12	243	0.000	1.000	1		0	0	0	0	0

Рисунок 5 – Отчет о числе вхождений в устройства.

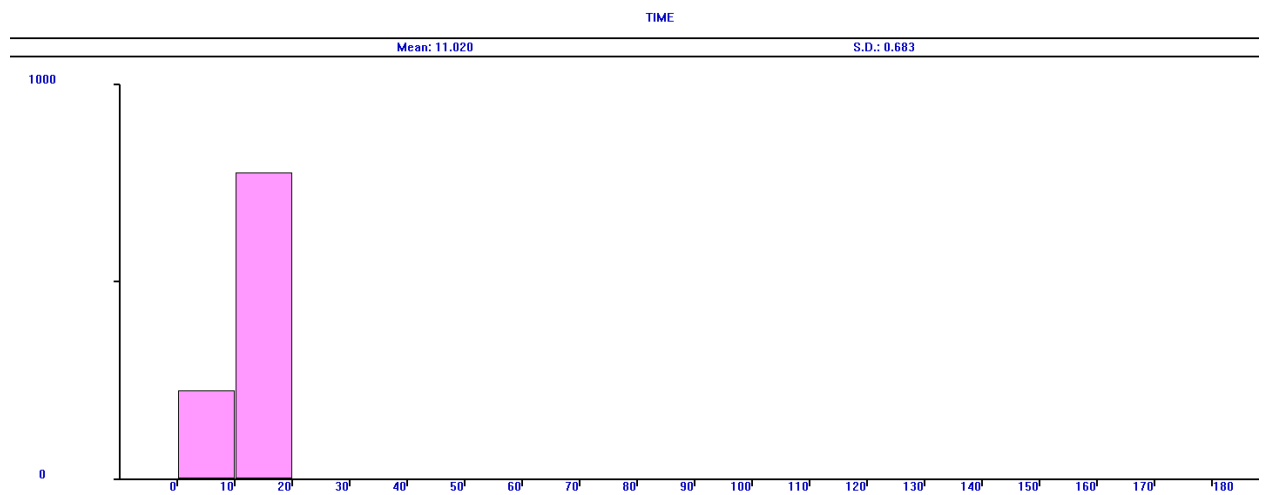


Рисунок 6 – Гистограмма задержек транзактов в автомате.

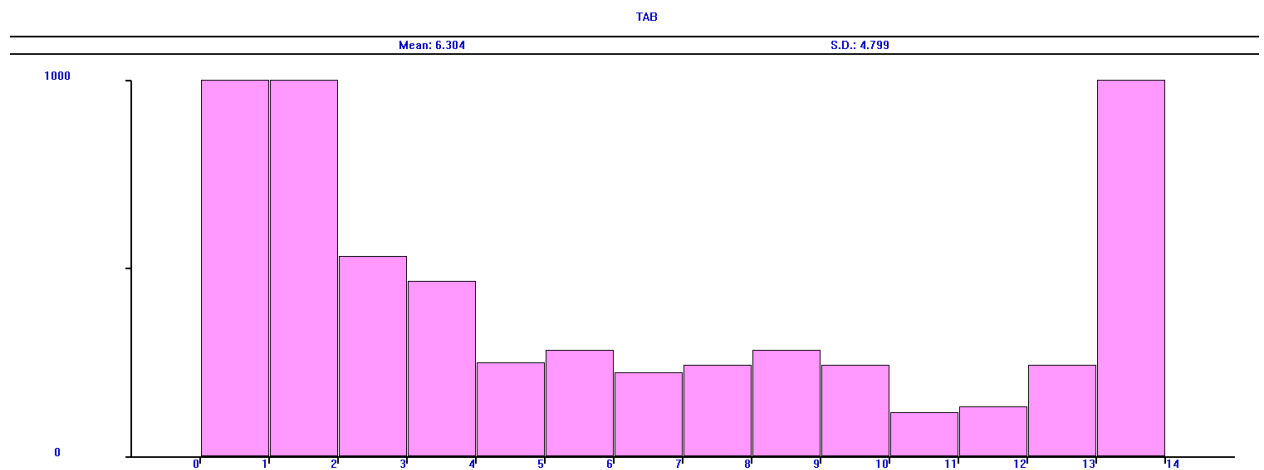


Рисунок 7 – Гистограмма числа вхождений в состояния автомата.

Моделирование при количестве транзактов, равном 50000.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	5000011.000	96	17	0

Рисунок 8 – Отчет о количестве использованных ресурсов и времени выполнения моделирования.

FACILITY	ENTRIES	UTIL.	AVE.	TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
AA0	50000	0.001	1.000	1		0	0	0	0	0
YZ1	50000	0.005	5.000	1		0	0	0	0	0
KAN	50000	0.005	5.000	1		0	0	0	0	0
YZ2	50000	0.005	5.000	1		0	0	0	0	0
AA1	50000	0.001	1.000	1		0	0	0	0	0
AA2	24997	0.000	1.000	1		0	0	0	0	0
AA5	12537	0.000	1.000	1		0	0	0	0	0
AA8	12537	0.000	1.000	1		0	0	0	0	0
AAK	50000	0.001	1.000	1		0	0	0	0	0
AA3	25003	0.001	1.000	1		0	0	0	0	0
AA6	12408	0.000	1.000	1		0	0	0	0	0
AA4	12460	0.000	1.000	1		0	0	0	0	0
AA11	6270	0.000	1.000	1		0	0	0	0	0
AA10	6190	0.000	1.000	1		0	0	0	0	0
AA7	12595	0.000	1.000	1		0	0	0	0	0
AA9	12595	0.000	1.000	1		0	0	0	0	0
AA12	12595	0.000	1.000	1		0	0	0	0	0

Рисунок 9 – Отчет о числе вхождений в устройства.

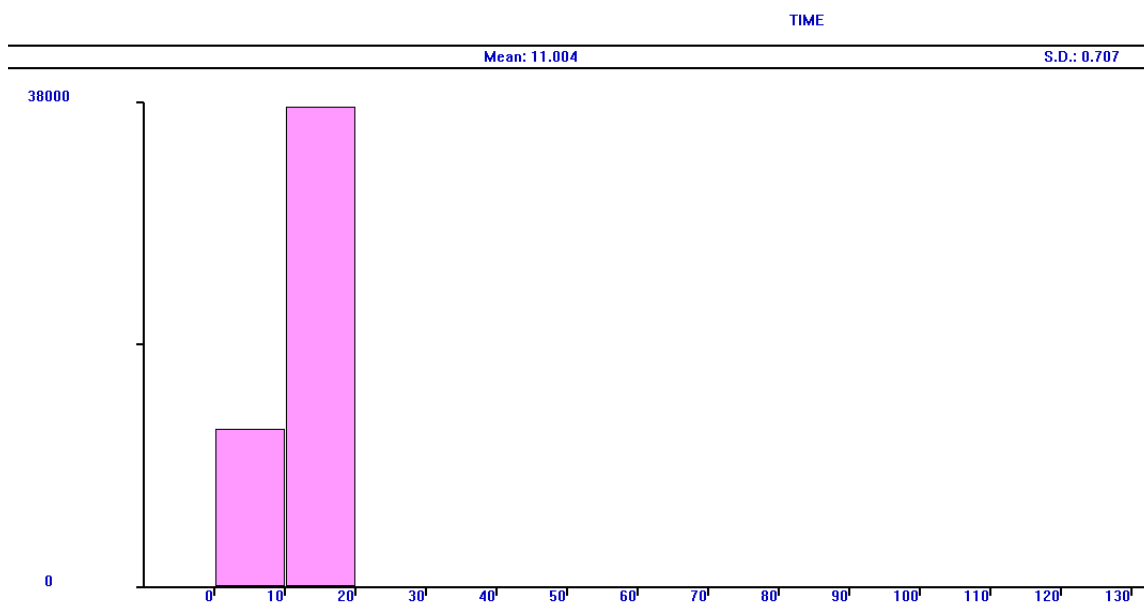


Рисунок 10 – Гистограмма задержек транзактов в автомате.

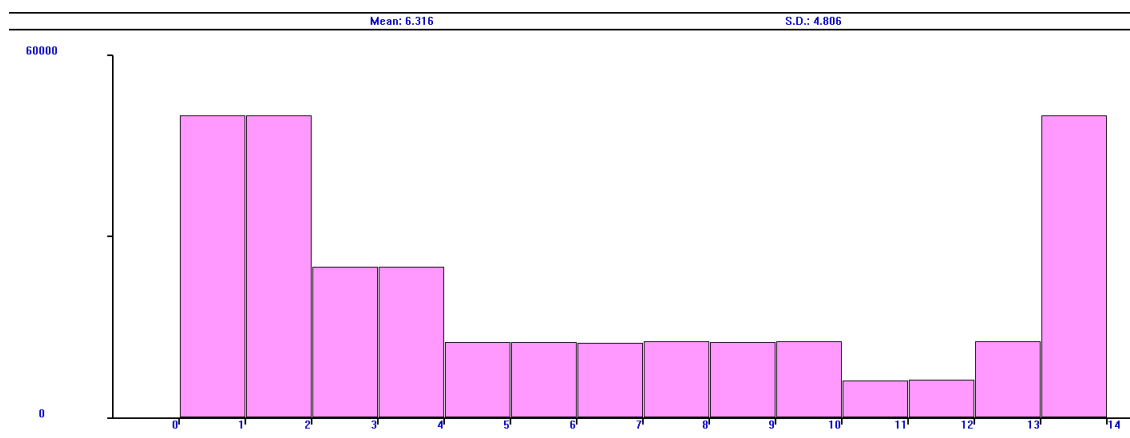


Рисунок 11 – Гистограмма числа вхождений в состояние автомата.

Моделирование при количестве транзактов, равном 100000.

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	100000011.000	96	17	0

Рисунок 12 – Отчет о количестве использованных ресурсов и времени выполнения моделирования.

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
AA0	100000	0.001	1.000	1		0	0	0	0
YZ1	100000	0.005	5.000	1		0	0	0	0
KAN	100000	0.005	5.000	1		0	0	0	0
YZ2	100000	0.005	5.000	1		0	0	0	0
AA1	100000	0.001	1.000	1		0	0	0	0
AA2	50090	0.001	1.000	1		0	0	0	0
AA5	25217	0.000	1.000	1		0	0	0	0
AA8	25217	0.000	1.000	1		0	0	0	0
AAK	100000	0.001	1.000	1		0	0	0	0
AA3	49910	0.000	1.000	1		0	0	0	0
AA6	24768	0.000	1.000	1		0	0	0	0
AA4	24873	0.000	1.000	1		0	0	0	0
AA11	12447	0.000	1.000	1		0	0	0	0
AA10	12426	0.000	1.000	1		0	0	0	0
AA7	25142	0.000	1.000	1		0	0	0	0
AA9	25142	0.000	1.000	1		0	0	0	0
AA12	25142	0.000	1.000	1		0	0	0	0

Рисунок 13 – Отчет о числе вхождений в устройства.

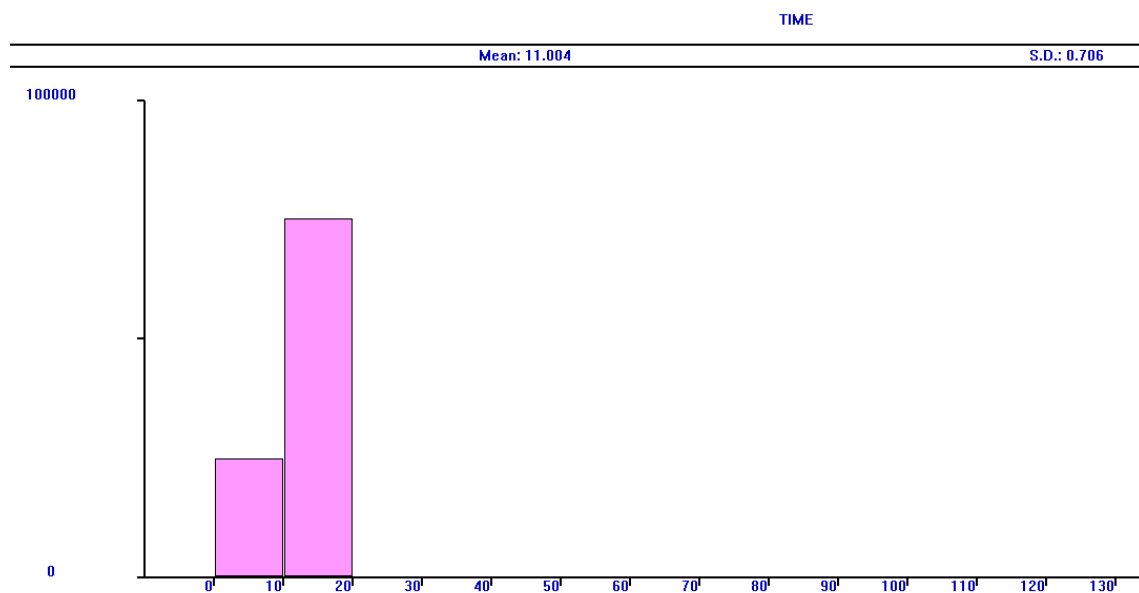


Рисунок 14 – Гистограмма задержек транзактов в автомате.

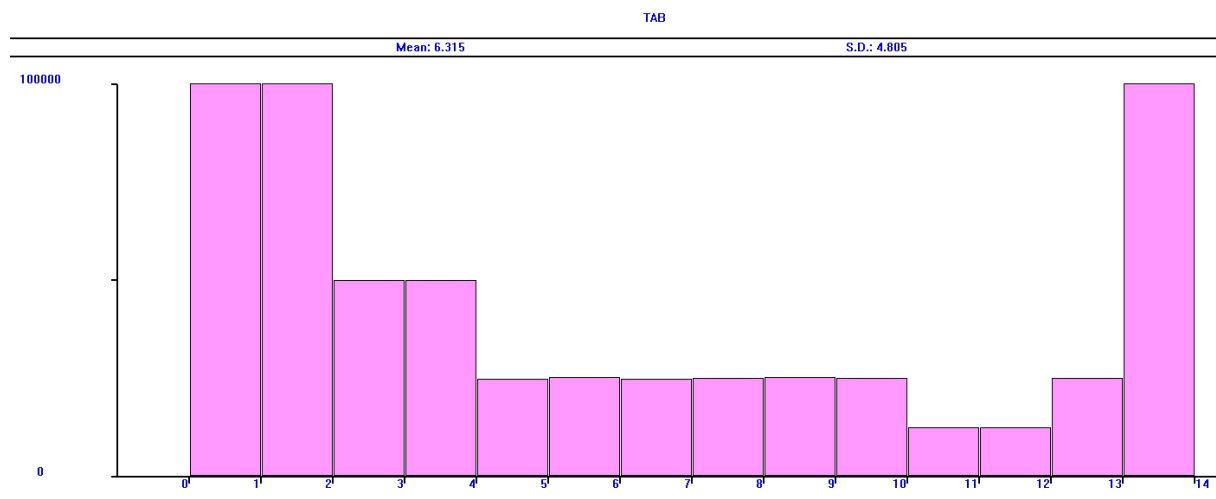


Рисунок 15 – Гистограмма числа вхождений в состояния автомата.

7 Построение модели с использованием сетей Петри

Сети Петри — математический аппарат для моделирования динамических дискретных систем. Впервые описаны Карлом Петри в 1962 году. Сеть Петри состоит из вершин двух типов — позиций и переходов, соединённых между собой дугами. Вершины одного типа не могут быть соединены непосредственно. В позициях могут размещаться метки (маркеры), способные перемещаться по сети.

В соответствие с граф-схемой автомата была построена и протестирована модель в программе PIPE (рисунок 16).

Позиция A_0 является начальным состоянием автомата, поэтому с нее должно начаться прохождение автомата, поставим в эту позицию метку.

С помощью позиций YZ1, KAN и YZ2 реализована передача пакета от сервера до клиента. Передача пакета будет осуществляться при переходе из состояния A_0 в состояние A_1 , дополнительно добавлены позиции P_{21} и P_{22} . Если все четыре позиции YZ1, YZ2, KAN, и P_{21} будут обладать метками, то произойдет захват и освобождение индикационной метки P_{23} , затем метка перейдет к позиции P_{22} и далее в состояние A_1 .

8 Моделирование построенной сети Петри

Моделирование сети Петри заключается в пошаговом отслеживании перехода метки по позициям сети. Во время моделирования метка выбирает случайный путь прохождения сети. На рисунке 17 представлен процесс моделирования полученной сети. В результате моделирования получаем путь, пройденный меткой (рисунок 18).

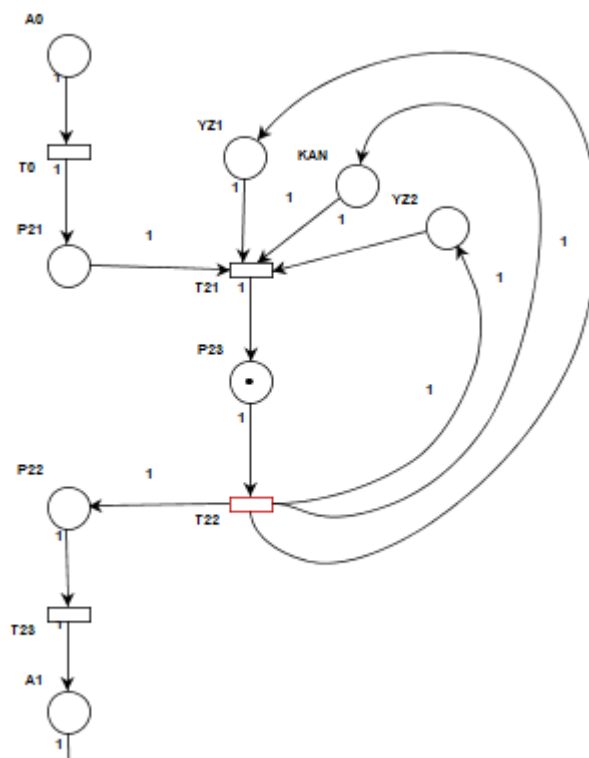


Рисунок 17 – Процесс моделирования сети Петри.

Animation history

Initial Marking

T0

T21

T22

T23

T1

T3

T9

T8

T13

T14

T20

Рисунок 18 – Результат моделирования сети Петри.

По построенной сети Петри сгенерировали граф достижимости сети Петри (рисунок 19).

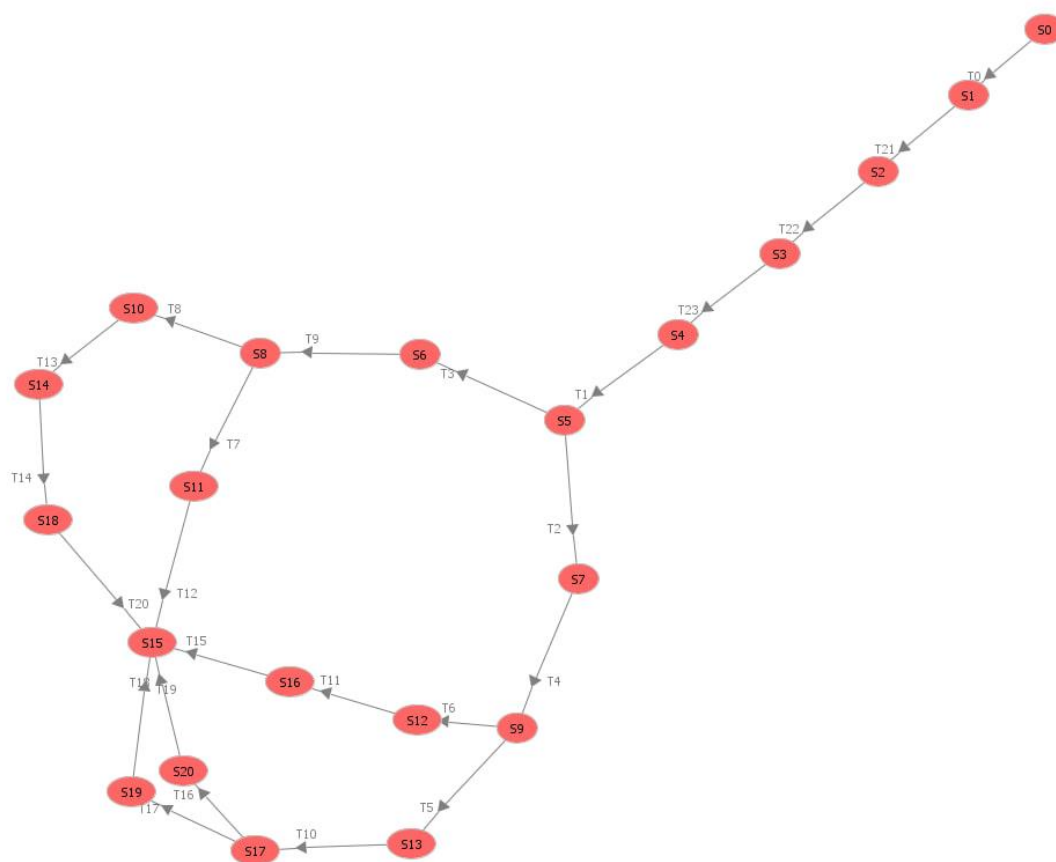


Рисунок 19 – Граф достижимости сети Петри.

Заключение

В ходе выполнения курсового проекта была построена модель распределённого алгоритма для сети по заданной граф-схеме автомата, а также была реализована передача пакета от сервера до клиента. В ходе моделирования и тестирования алгоритма были использованы методы моделирования с помощью языка Python и программ GPSS World и PIPE.

В ходе работы были получены навыки моделирования сетей Петри и моделей GPSS, а также более подробно изучены принципы их реализации и функционирования.

В итоге можно сделать вывод, что такие программные продукты, как GPSS World и PIPE, являются удобными инструментами для моделирования имитационных объектов благодаря интуитивно понятному интерфейсу и предоставляемым возможностям для создания моделей.

Список используемой литературы

1. Питерсон Дж. Теория сетей Петри и моделирование систем. – М.: Мир, 1984, - 264 с.
2. PIPE2: A Tool for the Performance Evaluation of Generalised Stochastic Petri Nets - Nicholas J. Dingle, William J. Knottenbelt, Tamas Suto
3. Марченков, С. С. Конечные автоматы : учебное пособие / С. С. Марченков. — Москва : ФИЗМАТЛИТ, 2008. — 56 с.
4. Боев В. Д. Моделирование систем. Инструментальные средства *GPSS WORLD*. / В. Д. Боев. - СПб.: БХВ-Петербург, 2004. — 368 с.
5. Томашевский В., Жданова Е. Имитационное моделирование в среде GPSS. В. Томашевский, Е. Жданова – М.: Бестселлер, 2003. – 416 с.

Приложение А. Код программы на языке Python

```
def a0(signals):
    signals = signals + ' '
    print(f'Вых.Сигналы: {signals}\nCостояние автомата: a0')
    return a1(signals)
def a1(signals):
    signals = signals + 'Y1'
    a = int(input(f'Вых.Сигналы: {signals}\nCостояние автомата: a1\nВходной сигнал X1 равен:'))
    if a == 0:
        return a3(signals)
    elif a == 1:
        return a2(signals)
def a2(signals):
    signals = signals + 'Y2'
    a = int(input(f'Вых.Сигналы: {signals}\nCостояние автомата: a2\nВходной сигнал X2 равен:'))
    if a == 0:
        return a5(signals)
    elif a == 1:
        return a4(signals)
def a3(signals):
    signals = signals + 'Y3'
    a = int(input(f'Вых.Сигналы: {signals}\nCостояние автомата: a3\nВходной сигнал X3 равен:'))
    if a == 1:
        return a6(signals)
    elif a == 0:
        return a7(signals)
```

```

def a4(signals):
    signals = signals + 'Y4'
    a = int(input(f'Вых.Сигналы: {signals}\nCостояние автомата: a4\nВходной сигнал X4 равен:'))
    if a == 1:
        return a10(signals)
    elif a == 0:
        return a11(signals)
def a5(signals):
    signals = signals + 'Y5'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата: a5')
    return a8(signals)
def a6(signals):
    signals = signals + 'Y6'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата: a6')
    return ak(signals)
def a7(signals):
    signals = signals + 'Y7'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата: a7')
    return a9(signals)
def a8(signals):
    signals = signals + 'Y8'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата: a8')
    return ak(signals)
def a9(signals):
    signals = signals + 'Y9'

```

```

        print(f'Вых.Сигналы: {signals}\nCостояние автомата:
a9')
        return a12(signals)
def a10(signals):
    signals = signals + 'Y10'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата:
a10')
    return ak(signals)
def a11(signals):
    signals = signals + 'Y11'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата:
a11')
    return ak(signals)
def a12(signals):
    signals = signals + 'Y12'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата:
a12')
    return ak(signals)
def ak(signals):
    signals = signals + 'Yk'
    print(f'Вых.Сигналы: {signals}\nCостояние автомата:
ak')
    return 0
str = ''
a0(str)

```

Приложение Б. Код программы на языке GPSS World

```
TAB TABLE P1,0,1,16
TIME TABLE M1,0,10,20
GENERATE 1000
A_0 SEIZE AA0
      ADVANCE 1
      RELEASE AA0
      ASSIGN 1,1
MET1 GATE NU YZ1
      GATE NU KAN
      GATE NU YZ2
      TRANSFER SIM,,MET1
      SEIZE YZ1
      SEIZE KAN
      SEIZE YZ2
      ADVANCE 5
      RELEASE YZ2
      RELEASE KAN
      RELEASE YZ1
      TABULATE TAB
A_1 SEIZE AA1
      ADVANCE 1
      RELEASE AA1
      ASSIGN 1,2
      TABULATE TAB
      TRANSFER .5,A_2,A_3
A_2 SEIZE AA2
      ADVANCE 1
      RELEASE AA2
```

```

        ASSIGN 1,3
        TABULATE TAB
        TRANSFER .5,A_4,A_5
A_3 SEIZE AA3
        ADVANCE 1
        RELEASE AA3
        ASSIGN 1,4
        TABULATE TAB
        TRANSFER .5,A_6,A_7
A_4 SEIZE AA4
        ADVANCE 1
        RELEASE AA4
        ASSIGN 1,5
        TABULATE TAB
        TRANSFER .5,A_10,A_11
A_5 SEIZE AA5
        ADVANCE 1
        RELEASE AA5
        ASSIGN 1,6
        TABULATE TAB
        TRANSFER ,A_8
A_6 SEIZE AA6
        ADVANCE 1
        RELEASE AA6
        ASSIGN 1,7
        TABULATE TAB
        TRANSFER ,A_K
A_7 SEIZE AA7
        ADVANCE 1

```

```
RELEASE AA7
ASSIGN 1,8
TABULATE TAB
TRANSFER ,A_9
A_8 SEIZE AA8
ADVANCE 1
RELEASE AA8
ASSIGN 1,9
TABULATE TAB
TRANSFER ,A_K
A_9 SEIZE AA9
ADVANCE 1
RELEASE AA9
ASSIGN 1,10
TABULATE TAB
TRANSFER ,A_12
A_10 SEIZE AA10
ADVANCE 1
RELEASE AA10
ASSIGN 1,11
TABULATE TAB
TRANSFER ,A_K
A_11 SEIZE AA11
ADVANCE 1
RELEASE AA11
ASSIGN 1,12
TABULATE TAB
TRANSFER ,A_K
A_12 SEIZE AA12
```

```
ADVANCE 1
RELEASE AA12
ASSIGN 1,13
TABULATE TAB
TRANSFER ,A_K
A_K SEIZE AAK
ADVANCE 1
RELEASE AAK
ASSIGN 1,14
TABULATE TAB
TABULATE TIME
TERMINATE 1
```

```
START 50000
```

Приложение В. Отчёт результата моделирования на GPSS World

GPSS World Simulation Report - КурсоваяMet.27.1

Tuesday, December 17, 2024 01:01:58

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	50000011.000	96	17	0

NAME	VALUE
AA0	10002.000
AA1	10006.000
AA10	10015.000
AA11	10014.000
AA12	10018.000
AA2	10007.000
AA3	10011.000
AA4	10013.000
AA5	10008.000
AA6	10012.000
AA7	10016.000
AA8	10009.000
AA9	10017.000
AAK	10010.000
A_0	2.000
A_1	18.000
A_10	72.000
A_11	78.000
A_12	84.000
A_2	24.000
A_3	30.000
A_4	36.000
A_5	42.000
A_6	48.000
A_7	54.000
A_8	60.000
A_9	66.000
A_K	90.000
KAN	10004.000
MET1	6.000
TAB	10000.000
TIME	10001.000
YZ1	10003.000
YZ2	10005.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
A_0	1	GENERATE	50000		0	0
	2	SEIZE	50000		0	0
	3	ADVANCE	50000		0	0
	4	RELEASE	50000		0	0
	5	ASSIGN	50000		0	0
MET1	6	GATE	50000		0	0
	7	GATE	50000		0	0
	8	GATE	50000		0	0
	9	TRANSFER	50000		0	0
	10	SEIZE	50000		0	0
	11	SEIZE	50000		0	0
	12	SEIZE	50000		0	0

A_1	13	ADVANCE	50000	0	0
	14	RELEASE	50000	0	0
	15	RELEASE	50000	0	0
	16	RELEASE	50000	0	0
	17	TABULATE	50000	0	0
A_2	18	SEIZE	50000	0	0
	19	ADVANCE	50000	0	0
	20	RELEASE	50000	0	0
	21	ASSIGN	50000	0	0
	22	TABULATE	50000	0	0
A_3	23	TRANSFER	50000	0	0
	24	SEIZE	24997	0	0
	25	ADVANCE	24997	0	0
	26	RELEASE	24997	0	0
	27	ASSIGN	24997	0	0
A_4	28	TABULATE	24997	0	0
	29	TRANSFER	24997	0	0
	30	SEIZE	25003	0	0
	31	ADVANCE	25003	0	0
	32	RELEASE	25003	0	0
A_5	33	ASSIGN	25003	0	0
	34	TABULATE	25003	0	0
	35	TRANSFER	25003	0	0
	36	SEIZE	12460	0	0
	37	ADVANCE	12460	0	0
A_6	38	RELEASE	12460	0	0
	39	ASSIGN	12460	0	0
	40	TABULATE	12460	0	0
	41	TRANSFER	12460	0	0
	42	SEIZE	12537	0	0
A_7	43	ADVANCE	12537	0	0
	44	RELEASE	12537	0	0
	45	ASSIGN	12537	0	0
	46	TABULATE	12537	0	0
	47	TRANSFER	12537	0	0
A_8	48	SEIZE	12408	0	0
	49	ADVANCE	12408	0	0
	50	RELEASE	12408	0	0
	51	ASSIGN	12408	0	0
	52	TABULATE	12408	0	0
A_9	53	TRANSFER	12408	0	0
	54	SEIZE	12595	0	0
	55	ADVANCE	12595	0	0
	56	RELEASE	12595	0	0
	57	ASSIGN	12595	0	0
A_10	58	TABULATE	12595	0	0
	59	TRANSFER	12595	0	0
	60	SEIZE	12537	0	0
	61	ADVANCE	12537	0	0
	62	RELEASE	12537	0	0
	63	ASSIGN	12537	0	0
	64	TABULATE	12537	0	0
	65	TRANSFER	12537	0	0
	66	SEIZE	12595	0	0
	67	ADVANCE	12595	0	0
	68	RELEASE	12595	0	0
	69	ASSIGN	12595	0	0
	70	TABULATE	12595	0	0
	71	TRANSFER	12595	0	0
	72	SEIZE	6190	0	0
	73	ADVANCE	6190	0	0
	74	RELEASE	6190	0	0
	75	ASSIGN	6190	0	0
	76	TABULATE	6190	0	0

A_11	77	TRANSFER	6190	0	0
	78	SEIZE	6270	0	0
	79	ADVANCE	6270	0	0
	80	RELEASE	6270	0	0
	81	ASSIGN	6270	0	0
A_12	82	TABULATE	6270	0	0
	83	TRANSFER	6270	0	0
	84	SEIZE	12595	0	0
	85	ADVANCE	12595	0	0
	86	RELEASE	12595	0	0
A_K	87	ASSIGN	12595	0	0
	88	TABULATE	12595	0	0
	89	TRANSFER	12595	0	0
	90	SEIZE	50000	0	0
	91	ADVANCE	50000	0	0
	92	RELEASE	50000	0	0
	93	ASSIGN	50000	0	0
	94	TABULATE	50000	0	0
	95	TABULATE	50000	0	0
	96	TERMINATE	50000	0	0

FACILITY DELAY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY
AA0	50000	0.001	1.000	1	0	0	0	0
YZ1	50000	0.005	5.000	1	0	0	0	0
KAN	50000	0.005	5.000	1	0	0	0	0
YZ2	50000	0.005	5.000	1	0	0	0	0
AA1	50000	0.001	1.000	1	0	0	0	0
AA2	24997	0.000	1.000	1	0	0	0	0
AA5	12537	0.000	1.000	1	0	0	0	0
AA8	12537	0.000	1.000	1	0	0	0	0
AAK	50000	0.001	1.000	1	0	0	0	0
AA3	25003	0.001	1.000	1	0	0	0	0
AA6	12408	0.000	1.000	1	0	0	0	0
AA4	12460	0.000	1.000	1	0	0	0	0
AA11	6270	0.000	1.000	1	0	0	0	0
AA10	6190	0.000	1.000	1	0	0	0	0
AA7	12595	0.000	1.000	1	0	0	0	0
AA9	12595	0.000	1.000	1	0	0	0	0
AA12	12595	0.000	1.000	1	0	0	0	0

TABLE CUM. % TAB	MEAN	STD. DEV.	RANGE	RETRY FREQUENCY
	6.316	4.806		0

16.66	0.000	-	1.000	50000
33.31	1.000	-	2.000	50000
41.64	2.000	-	3.000	24997
49.97	3.000	-	4.000	25003
54.12	4.000	-	5.000	12460
58.30	5.000	-	6.000	12537
62.43	6.000	-	7.000	12408
66.63	7.000	-	8.000	12595
70.80	8.000	-	9.000	12537
75.00	9.000	-	10.000	12595
77.06	10.000	-	11.000	6190
79.15	11.000	-	12.000	6270
83.34	12.000	-	13.000	12595
100.00	13.000	-	14.000	50000
TIME	11.004	0.707	0	
24.82	0.000	-	10.000	12408
100.00	10.000	-	20.000	37592

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
50001	0	50001000.000	50001	0	1		

Приложение Г. Моделирование с использованием сетей Петри

