

СЕВЕРО-КАВКАЗСКИЙ ФИЛИАЛ  
ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ  
ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО  
УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  
«Московский технический университет связи и информатики»

Кафедра «Информационно вычислительная техника»

А.А. Нерсисянц

## **Моделирование**

Учебное пособие по дисциплине

«Моделирование»

Направление: бакалавриат – Информатика и вычислитель-  
ная техника (09.03.01)

Ростов-на-Дону  
2019

**Нерсисянц А.А. Моделирование**

Учебное пособие по дисциплине:

Моделирование

Направление: бакалавриат –

Информатика и вычислительная техника (09.03.01)

Учебное пособие по дисциплине «Моделирование» предназначено для изучения студентами очной и заочной форм обучения, основных положений по способам и приёмам анализа систем и сетей связи с помощью имитационного моделирования. Пособие содержит пять частей:

- общие принципы имитационного моделирования систем;
- место имитационного моделирования при анализе ИК-систем;
- краткие сведения по структуре языка *GPSS*;
- практические занятия по изучению способов моделирования систем и сетей связи;
- лабораторные занятия:
  - Исследование маршрутизатора сети коммутации пакетов методами имитационного и аналитического моделирования;
  - Оптимизация структурных параметров в сетях с коммутацией пакетов.

Рецензент: доцент кафедры ИТСС - к.т.н. Борисов Б.П.

Рассмотрено и одобрено на заседании кафедры ИВТ

Протокол №11 от 26.08.19

Ростов-на-Дону: СКФ МТУСИ, 2019 – 126 с.:

Отпечатано в Полиграфическом центре СКФ МТУСИ, Серафимовича, 62.

## СОДЕРЖАНИЕ

Введение.....	6
1 Общие принципы имитационного моделирования систем.....	10
1.1 Основные определения.....	10
1.2 Применение имитационного моделирования.....	11
1.3 Виды имитационного моделирования.....	12
1.4 Пример построения имитационной модели.....	13
1.5 Преимущества имитационного моделирования.....	16
2 Место имитационного моделирования при анализе ТК-систем.....	18
2.1 Виды моделирования.....	18
2.2 Задачи анализа и синтеза, при проектировании и исследо- вании систем и сетей связи.....	20
2.3 Реализация случайных воздействий в моделях СиСС.....	22
2.4 Механизм системного времени.....	28
2.4.1 Общие замечания.....	28
2.4.2 Механизм системного времени с шагом до события.....	29
2.4.3 Механизм системного времени с фиксированным ша- гом .....	30
2.5 Моделирование методом вложенной марковской цепи.....	33
2.6 Методы планирования эксперимента.....	34
2.6.1 Стратегическое планирование.....	34
2.6.2 Тактическое планирование.....	36
3 Краткие сведения по структуре языка GPSS.....	39
3.1 Модель как система массового обслуживания.....	39
3.2 Программирование в системе GPSS World.....	42
3.2.1 Общецелевая система моделирования GPSS World .....	42
3.2.2 Основные элементы и свойства.....	44

3.2.3	Классы языка GPSS.....	48
3.2.4	Блоки GPSS.....	51
3.2.5	Движение транзактов во времени.....	51
3.2.6	Таймер модельного времени.....	52
3.2.7	Управляющие операторы GPSS.....	53
3.3	Краткое описание языка GPSS .....	54
3.3.1	Структура оператора.....	54
3.3.2	Стандартные числовые атрибуты.....	54
3.3.3	Примеры наиболее часто встречающихся операторов.....	56
3.3.4	Изменение маршрутов транзактов.....	60
3.3.5	Способы задания экспоненциального закона распре- деления.....	61
3.3.6	Оператор изменения параметров транзактов.....	62
3.3.7	Вычислительный оператор variable.....	63
3.3.8	Оператор присвоения.....	63
3.3.9	Представление гистограмм.....	63
3.3.10	Удаление транзактов.....	64
4	Практические занятия по изучению способов моделирования.....	65
4.1	Замечания по применению системы моделирования GPSS World.....	65
4.1.1	Запуск моделирующей программы.....	65
4.1.2	Доступ к результатам моделирования.....	66
4.1.3	Внесение изменений в параметры моделируемой сис- темы.....	66
4.2	Практические занятия.....	67
5	Лабораторные занятия.....	96
	Лабораторная работа № 1.Исследование маршрутизатора сети ком- мутации пакетов методами имитационного и аналитического моде- лирования.....	96

Приложение А. Критерии согласия для проверки гипотез.....	114
Приложение Б. Основные операторы (блоки) языка GPSS.....	117
Приложение В. Примеры использования операторов GPSS.....	120
Приложение Г. Список стандартных числовых атрибутов.....	123
Список использованных источников.....	126

## ВВЕДЕНИЕ

Настоящее учебное пособие предназначено для проведения практических (лабораторных) занятий при изучении дисциплины «Мультисервисные сети связи». Занятия проводятся с применением специализированного языка высокого уровня *General Purpose Simulation System* – общецелевая система моделирования (*GPSS*) в усовершенствованной версии *GPSS World 5.2.2*.

Пособие содержит теоретическую часть и 14 не связанных друг с другом практических занятий, построенных по нарастающей сложности. Темы занятий направлены на изучение вероятностных процессов в отдельных элементах систем и сетей связи. При этом, в большинстве занятий проводится сопоставление методов имитационного и аналитического моделирования. Такой подход вполне оправдан, так как при исследовании современных телекоммуникационных систем широко применяются именно эти два метода, объединённые общим названием – методы математического моделирования. Естественно, что в данных занятиях основное внимание уделяется методу имитационного моделирования, так как аналитическим методам при изучении телекоммуникационных систем в направлении Инфокоммуникации посвящена отдельная дисциплина – Теория телетрафика.

Поскольку данное пособие является дополнительным материалом к лекционному курсу и к основной литературе по дисциплине «Мультисервисные сети связи», то в нём отсутствуют специальные разделы с детальным изучением принципов моделирования на языке *GPSS*. В то же время в 3 -м разделе приведены сведения, необходимые для понимания практических занятий, приведённых в 4-й части. Ряд справочных данных приведен в приложениях к пособию.

Целью изучения учебного пособия является:

- понимание основных принципов построения имитационных моделей, включая разработку небольших фрагментов;

- обеспечить освоение студентами навыков по составлению отдельных программ, моделирующих элементы телекоммуникационных сетей, прогон разработанных моделей и анализ результатов;
- изучение способов ввода исходных данных и измерения результирующих характеристик;
- приобретение навыков анализа функциональных зависимостей в телекоммуникационных системах и выработки решений по результатам их анализа.

В пособии рассматриваются вопросы изучения специализированной программы имитационного моделирования и развиваются навыки её использования для анализа функционирования сетей и сетевых элементов. В процессе обучения студенты проходят весь цикл исследования сетей, начиная от стратегического планирования (определение цели эксперимента), выбора факторов и откликов имитационной модели, разработки алгоритма модели, написание программы и заканчивая серией испытательных прогонов, анализа результатов и формулировки выводов.

В результате изучения настоящего учебного пособия в составе дисциплины «Мультисервисные сети связи» студенты получают представление:

- о современном состоянии прикладного направления – имитационное моделирование;
- об основных принципах построения специализированных программ имитационного моделирования.

Изучат:

- структуру имитационного программного комплекса GPSS World;
- основные операторы языка GPSS World;
- правила выбора операндов для построения конкретных моделей.

Овладеют:

- основными принципами имитационного моделирования;
- навыками системного мышления;

- способностью ставить задачи по моделированию телекоммуникационных сетей, реализовывать эту модель и проводить серию экспериментов;
- умением использовать методы анализа и оптимизации параметров системы.

В целом данное учебное пособие закладывает фундамент специальных знаний в области исследования различных телекоммуникационных сетей, в том числе мультимедийных сетей, мультисервисных и сетей следующего поколения (*NGN*).

Другой важной целью данного пособия и дисциплины «Моделирование систем и сетей связи» в целом является подготовка к проведению лабораторных работ в последующих семестрах при изучении специальных телекоммуникационных дисциплин.

В СКФ МТУСИ накоплена определённая база программных продуктов для проведения лабораторных работ и разработаны методические указания к проведению этих работ. В числе таких работ необходимо отметить следующие шесть лабораторных работ:

- лабораторная работа «Исследование сотовой сети подвижной связи стандарта GSM», проводится при изучении дисциплины «Мобильные системы связи»;
- лабораторная работа «Анализ качества обслуживания абонентов в сельской телефонной сети», проводится при изучении дисциплины «Сети связи»;
- лабораторная работа «Исследование технологии Frame Relay в сетях передачи данных», проводится при изучении дисциплины «Мультисервисные сети связи» (сети *NGN*);
- лабораторная работа «Исследование пропускной способности сети с коммутацией пакетов», проводится при изучении дисциплины «Мультисервисные сети связи» (сети *NGN*);



- лабораторная работа «Моделирование фрагмента мультисервисной сети коммутации пакетов», проводится при изучении дисциплины «Мультисервисные сети связи» (сети *NGN*);

- лабораторная работа «Моделирование мультисервисной сети связи», проводится при изучении дисциплины «Мультисервисные сети связи» (сети *NGN*).

Одна из этих работ (Исследование маршрутизатора сети коммутации пакетов методами имитационного и аналитического моделирования) включена в настоящее пособие (Раздел 5) как показательная работа.

# **1 Общие принципы имитационного моделирования систем**

## **1.1 Основные определения**

Имитационное моделирование (ситуационное моделирование) – метод, позволяющий строить модели, описывающие процессы так, как они проходили бы в действительности. Такую модель можно «проиграть» во времени как для одного испытания, так и заданного их множества. При этом результаты будут определяться случайным характером процессов. По этим данным можно получить достаточно устойчивую статистику. Моделирование – метод решения задач, при использовании которого, исследуемая система заменяется более простым объектом, описывающим реальную систему и называемым моделью.

Имитационное моделирование – это метод исследования, при котором изучаемая система заменяется моделью, с достаточной точностью описывающей реальную систему, с которой (с моделью) проводятся эксперименты с целью получения информации об этой системе. Экспериментирование с моделью называют имитацией (имитация – это постижение сути явления, не прибегая к экспериментам на реальном объекте).

Имитационное моделирование – это частный случай математического моделирования. Существует класс объектов, для которых по различным причинам не разработаны аналитические модели, либо не разработаны методы решения полученной модели. В этом случае аналитическая модель заменяется имитатором или имитационной моделью. Имитационным моделированием иногда называют получение частных численных решений сформулированной задачи на основе аналитических решений или с помощью численных методов.

Имитационная модель – это компьютерная программа, которая описывает структуру и воспроизводит поведение реальной системы во времени. Имитационная модель позволяет получать подробную статистику о различных аспектах функционирования системы в зависимости от входных данных.

Имитационная модель — логико-математическое описание объекта, которое может быть использовано для экспериментирования на компьютере в целях проектирования, анализа и оценки функционирования объекта.

## 1.2 Применение имитационного моделирования

– Имитационное моделирование применяется в случаях, когда проведение экспериментов над реальной системой невозможно или нецелесообразно, например:

- по причине хрупкости или дороговизны создания прототипа, либо из-за длительности проведения эксперимента в реальном масштабе времени;
- дорого или невозможно экспериментировать на реальном объекте;
- невозможно построить аналитическую модель: в системе есть время, причинные связи, последствие, нелинейности, стохастические (случайные) переменные и др.;
- необходимо имитировать поведение системы во времени.

Различают физическое и математическое моделирование. Примером физической модели является уменьшенная копия самолета, продуваемая в потоке воздуха. При использовании математического моделирования поведение системы описывается с помощью формул. Особым видом математических моделей являются имитационные модели.

Цель имитационного моделирования состоит в воспроизведении поведения исследуемой системы на основе результатов анализа наиболее существенных взаимосвязей между её элементами или другими словами – разработке симулятора (англ. *Simulation modeling*) исследуемой предметной области для проведения различных экспериментов.

Имитационное моделирование позволяет имитировать поведение системы во времени. Причём плюсом является то, что временем в модели можно управлять: замедлять в случае с быстропротекающими процессами и ускорять для моделирования систем с медленной изменчивостью. Можно имитировать поведение тех объектов, реальные эксперименты с которыми

дороги, невозможны или опасны. С наступлением эпохи персональных компьютеров производство сложных и уникальных изделий, как правило, сопровождается компьютерным трёхмерным имитационным моделированием. Эта точная и относительно быстрая технология позволяет накопить все необходимые знания, оборудование и полуфабрикаты для будущего изделия ещё до начала производства. Компьютерное 3D-моделирование теперь не редкость даже для небольших компаний.

Имитация как метод решения нетривиальных задач получила начальное развитие в связи с созданием ЭВМ в 1950-х – 1960-х годах. Можно выделить следующие разновидности имитации:

- метод Монте-Карло (метод статистических испытаний);
- метод имитационного моделирования (статистическое моделирование).

### 1.3 Виды имитационного моделирования

В настоящее время используются несколько видов имитационного моделирования, основные из которых следующие:

- агентное моделирование – относительно новое (1990 – 2000 гг.) направление в имитационном моделировании, которое используется для исследования децентрализованных систем, динамика функционирования которых определяется не глобальными правилами и законами (как в других парадигмах моделирования), а наоборот, когда эти глобальные правила и законы являются результатом индивидуальной активности членов группы. Цель агентных моделей – получить представление об этих глобальных правилах, общем поведении системы, исходя из предположений об индивидуальном, частном поведении её отдельных активных объектов и взаимодействии этих объектов в системе. Агент – некая сущность, обладающая активностью, автономным поведением, может принимать решения в соответствии с некоторым набором правил, взаимодействовать с окружением, а также самостоятельно изменяться;

– дискретно-событийное моделирование – подход к моделированию, предлагающий абстрагироваться от непрерывной природы событий и рассматривать только основные события моделируемой системы, такие, как: «ожидание», «обработка заказа», «движение с грузом», «разгрузка» и другие. Дискретно-событийное моделирование наиболее развито и имеет огромную сферу приложений – от логистики и систем массового обслуживания до транспортных и производственных систем. Этот вид моделирования наиболее подходит для моделирования телекоммуникационных процессов. Основан Джеффри Гордоном в 1960-х годах.

– системная динамика – парадигма моделирования, где для исследуемой системы строятся графические диаграммы причинных связей и глобальных влияний одних параметров на другие во времени, а затем созданная на основе этих диаграмм модель имитируется на компьютере. По сути, такой вид моделирования более всех других парадигм помогает понять суть происходящего и выявления причинно-следственных связей между объектами и явлениями. С помощью системной динамики строят модели бизнес-процессов, развития города, модели производства, динамики популяции, экологии и развития эпидемии. Метод основан Джейм Форрестером в 1950 годах.

## 1.4 Пример построения имитационной модели

Применение метода имитационного моделирования можно продемонстрировать на примере работы отделения банка по обслуживанию физических лиц. Допустим, что необходимо определить минимальное количество обслуживающего персонала, которое обеспечивает требуемое качество сервиса.

Критерий качества сервиса зададим правилом: средний размер очереди клиентов не должен превышать  $N$  человек. Очевидно, что для решения поставленной задачи необходимо иметь достаточные знания о системе: какие

клиенты посещают банк, какое количество клиентов приходит в течение рабочего дня, а также сколько времени занимает обслуживание одного клиента.

Хотя данная задача и может показаться специализированной, схожие проблемы возникают во многих областях, где задействованы людские и технические ресурсы. Оплата времени работы квалифицированного работника и времени использования сложной техники составляет немалую долю расходов компаний. Определение оптимального графика использования ресурсов, позволяющего системе эффективно выполнять поставленные задачи, позволяет снизить расходы, а значит увеличить прибыльность.

На первом этапе решения задачи создается модель, которая соответствует структуре и бизнес-процессам отделения банка. В ходе разработки модели учитываются только те детали, которые оказывают существенное влияние на изучаемые аспекты работы системы. Например, наличие отделения обслуживания юридических лиц или кредитного отдела не влияет на обслуживание физических лиц, поскольку они физически и функционально отделены от последнего. Схематично такую модель можно представить в виде последовательности следующих действий (рисунок 1.1).

На втором этапе на вход модели подаются исходные данные: интенсивность прихода клиентов, среднее время обслуживания клиентов, количество доступного персонала (таблица 1.1). На основании этих данных модель имитирует, или воспроизводит, работу банка в течение заданного промежутка времени, например, рабочего дня.

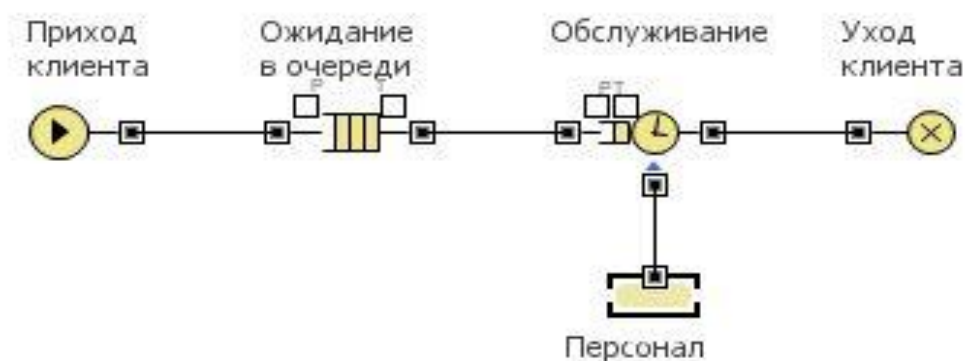


Рисунок 1.1 – Модель обслуживания клиентов банка

Таблица 1.1 – Динамика событий обслуживания клиентов банка

Время	Событие
19:54	клиент №167 пришёл и встал в очередь
19:56	клиент №168 пришёл и встал в очередь
19:57	клиент №164 закончил обслуживаться и ушёл
19:57	клиент №167 начал обслуживаться

Следующий этап заключается в анализе статистики, собранной и представленной моделью. Если средний размер очереди клиентов превышает выбранный предел в  $N$  человек, то количество доступного персонала следует увеличить и выполнить новый эксперимент (рисунок 1.2).



Рисунок 1.2 – Диаграмма продолжительности обслуживания клиентов банка в зависимости от доступного обслуживающего персонала

В результате проведения серии экспериментов над моделью пользователь может определить оптимальное количество персонала, при котором (по выбранному выше критерию) средний размер очереди клиентов не превышает  $N$  человек. Можно ориентироваться и на более тонкий критерий, например, чтобы вероятность задержки пользователя в очереди сверх допустимого времени не превышала некоторой обоснованной величины. Процесс подбора параметров может быть осуществлен также и с помощью встроенного оптимизатора, который в автоматическом режиме проверяет различные сочетания и находит лучшее решение.

## 1.5 Преимущества имитационного моделирования

Применение имитационных моделей дает множество преимуществ по сравнению с выполнением экспериментов над реальной системой и использованием других методов.

**Стоимость.** Допустим, компания уволила часть сотрудников, что в дальнейшем привело к снижению качества обслуживания и потери части клиентов. Принять обоснованное решение помогла бы имитационная модель, затраты на применение которой состоят лишь из цены программного обеспечения и стоимости консалтинговых услуг.

**Время.** В реальности оценить эффективность, например, новой сети распространения продукции или измененной структуры склада можно лишь через месяцы или даже годы. Имитационная модель позволяет определить оптимальность таких изменений за считанные минуты, необходимые для проведения эксперимента.

**Повторяемость.** Современная жизнь требует от организаций быстрой реакции на изменение ситуации на рынке. Например, прогноз объемов спроса продукции должен быть составлен в срок, и его изменения критичны. С помощью имитационной модели можно провести неограниченное количество экспериментов с разными параметрами, чтобы определить наилучший вариант.

**Точность.** Традиционные расчетные математические методы требуют применения высокой степени абстракции и не учитывают важные детали. Имитационное моделирование позволяет описать структуру системы и её процессы в естественном виде, не прибегая к использованию формул и строгих математических зависимостей.

**Наглядность.** Имитационная модель обладает возможностями визуализации процесса работы системы во времени, схематичного задания её структуры и выдачи результатов в графическом виде. Это позволяет нагляд-



но представить полученное решение и донести заложенные в него идеи до клиента и коллег.

**Универсальность.** Имитационное моделирование позволяет решать задачи из любых областей: телекоммуникации, производства, логистики, финансов, здравоохранения и многих других. В каждом случае модель имитирует и воспроизводит реальную жизнь и позволяет проводить широкий набор экспериментов без влияния на реальные объекты.

## 2 Место имитационного моделирования при анализе ТК-систем

### 2.1 Виды моделирования

Для исследования процессов в различных реальных системах, как правило, создаются модели, отражающие свойства системы, существенные для целей данного анализа. Например, для изучения зависимостей между напряжением, током и сопротивлением используется аналитическая модель – закон Ома ( $U = IR$ ).

Наиболее популярные модели, используемые для изучения телекоммуникационных систем и различающиеся формой отображения объекта, представлены на рисунке 2.1.



Рисунок 2.1– Основные виды моделирования

При натурном моделировании характеристики исследуются на реальном объекте. Исследователь, находясь на реальном объекте (АТС, узел связи, базовая станция сотовой сети) или наблюдая за процессами в ТК-сети с помощью средств дистанционного мониторинга, собирает статистику за некоторый интервал времени (день, неделя) и анализирует её результаты. Подсчитывая, например, число вызовов  $N$ , поступивших на АТС в час наибольшей нагрузки, и число  $n$  из них, которым было отказано в установлении соединения, исследователь может оценить вероятность потерь как:  $P = n / N$ .

Модель адекватна, но возможности ограничены. Нельзя, например, для целей эксперимента попросить абонентов звонить чаще или говорить меньше.

При математическом моделировании реальному объекту устанавливается соответствие некоторого математического объекта – модели. При этом модель должна отражать все элементы реальной системы (свойства входного потока, характеристики обслуживающих устройств, дисциплины обслуживания и др.). В зависимости от сложности модели создаётся та или иная степень приближения модели к реальному объекту. Различают аналитическое и имитационное математическое моделирование.

Аналитическое моделирование – процесс функционирования системы отображается функциональными соотношениями (уравнениями) или логическими условиями. Уравнения решаются в общем виде или численными методами. Примером аналитической модели может служить формула Эрланга (2.1), определяющая вероятность потери вызова в зависимости от интенсивности нагрузки и числа каналов:

$$P_v = \frac{\frac{\Lambda^v}{v!}}{\sum_{j=0}^v \frac{\Lambda^j}{j!}} \quad (2.1)$$

Имитационное моделирование (ИМ) – процесс функционирования системы воспроизводится с помощью моделирующего алгоритма – программы для ЭВМ. При многократном воспроизведении случайных событий, величин и функций и последующей статистической обработке результатов определяются статистические характеристики объекта исследования. Например, если по результатам прогона из  $M$  вызовов было потеряно  $m$  вызовов, то величина потерь определяется как и при натурном моделировании, а именно  $P = m / M$ . У имитационного моделирования практически неограниченный класс исследуемых систем.

По сравнению с аналитическим моделированием имитационное позволяет получить сколь угодно точное отображение реального объекта, правда

за счёт времени и денег, затраченных на разработку моделирующей программы, и машинного времени расходуемого на прогоны. Иногда на это уходят дни и месяцы.

Аналитическое моделирование при наличии предварительно разработанного математического аппарата позволяют легко получать нужные характеристики. Однако аналитическому описанию поддаётся только очень ограниченный круг систем. В основном, это системы без последствий. Поэтому при выборе подходящей аналитической модели из имеющегося ряда приходится делать допущения о характере реальных процессов, что, в конечном счёте, делает модель менее адекватной, а иногда и вовсе не пригодной.

## 2.2 Задачи анализа и синтеза, при проектировании и исследовании систем и сетей связи

Имитационное моделирование систем и сетей связи (СиСС) предназначено для изучения процессов обслуживания системами и сетями поступающих потоков сообщений и количественной оценки качества обслуживания.

Математическая модель процесса обслуживания включает четыре основных элемента:

- система распределения информации (система обслуживания – узлы, линии связи, сети);
- входной поток сообщений;
- дисциплина обслуживания;
- характеристики качества обслуживания.

Имитационная Модель (ИМ) решает следующие задачи:

- исследование входного потока вызовов и описание его характеристик;
- исследование и поиск оптимальной структуры системы обслуживания;
- исследование и определение оптимальных алгоритмов обслуживания вызовов;

- определение пропускных способностей структурных элементов системы (узлов и линий связи);
- определение качества обслуживания вызовов.

Доминирующее положение в имитационном моделировании занимают задачи анализа – определение характеристик качества обслуживания в зависимости от параметров и свойств потока вызовов, параметров и дисциплины системы обслуживания. Обратная задача – задача синтеза, т.е. нахождение параметров системы в зависимости от свойств входного потока и требуемого качества обслуживания методами имитационного моделирования не решается. Однако многочисленные анализы различных структур с оценкой качественных показателей и стоимости системы помогают проектировщику сделать правильный выбор наиболее приемлемого варианта.

Качество функционирования СиСС или её эффективность оценивается такими показателями как пропускная способность сети и отдельных её частей, вероятность потери вызовов, пакетов или сообщений, время доведения сообщений (пакетов) и др. Кроме того, в ряде случаев приходится учитывать и такие более сложные показатели как надёжность, помехозащищённость, устойчивость, стоимость и многие другие.

Понятие эффективность СиСС, как правило, относится к числу трудно-определимых, поскольку эффективность должна учитывать совокупность целого ряда характеристик, противодействующих друг другу. Например, как самое очевидное, – пропускная способность маршрутизатора (канала, адаптера) и его цена. Но есть и более скрытые противоречия. Например, характеристики потери пакета в сети и его задержки. Даже такой, на первый взгляд незначительный параметр, как емкость буфера для хранения пакетов в очереди, нуждается в тщательном выборе, так как малая емкость увеличивает потери пакетов, а большая – увеличивает задержки в маршрутизаторе и в сети в целом.

Аналитически рассчитать **оптимальный** объём буфера даже для одного направления пока невозможно. А в сети одновременно функционирует

множество маршрутизаторов и каналов, передаются потоки различных классов и, соответственно, различных требований к качеству обслуживания (*QoS*). И оптимизируемых характеристик в сети не две, а более десятка.

Имитационная модель должна выявлять зависимости этих характеристик от огромного числа сетевых параметров и в первую очередь такого трудно формализуемого, как конфигурация сети. Непосредственное написание моделирующей программы является менее значительной задачей по сравнению с идеологической разработкой модели, т.е. с определением необходимого перечня исследуемых характеристик и параметров, влияющих на эти характеристики.

Поэтому практически интересные модели редко создаются сразу, обычно они проходят некоторый цикл усовершенствования, как это показано на рисунке 2.2

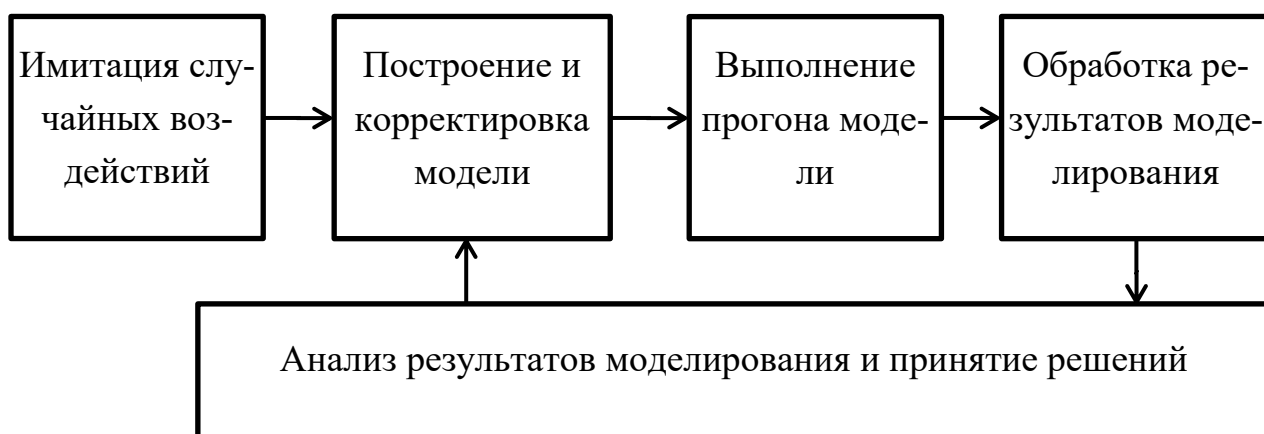


Рисунок 2.2 – Цикл модификаций модели

## 2.3 Реализация случайных воздействий в моделях СиСС

### 2.3.1 Случайные и стационарные процессы в телекоммуникациях

Большинство процессов в современных телекоммуникациях связаны со случайными событиями. В первую очередь это события, связанные с действиями абонента – человека:

- вызовы на установление телефонного соединения в фиксированных и мобильных сетях;
- файловый обмен с участием человека;
- продолжительность видео конференций;
- размеры передаваемых данных;
- размеры видео файлов и др.

Наряду с этим существуют и детерминированные процессы:

- телекоммуникационные средства мониторинга;
- размеры пакетов при фиксированной нарезке сообщений. Например, пакеты в 160 байт при IP-телефонии;
- покадровая передача телепрограмм;
- межмашинный обмен в системе M2M и др.

В связи с этим системы имитационного моделирования СиСС должны иметь в своём составе средства генерации случайных процессов и величин. В первую очередь это относится к определению случайных интервалов между вызовами в потоке требований на предоставление услуг связи и продолжительности реализации самой услуги.

### 2.3.2 Генераторы равномерно распределённых случайных чисел

Для работы стохастических имитационных моделей необходима генерация случайных чисел с равномерным законом распределения и возможностью использования этих чисел для генерации случайных величин с требуемыми характеристиками.

На ранних этапах развития имитационного моделирования для этой цели использовались физические датчики, построенные, как правило, с использованием генератора шума. Отличительной особенностью таких генераторов является неповторимость однажды сгенерированной последовательности. У этого свойства есть как достоинства (отсутствие корреляции между экспериментами), так и недостатки (невозможность в точности повторить экспери-

мент при изменении одного из параметров, чтобы проанализировать влияние на работу системы именно этого параметра).

Однако более существенным оказался недостаток физического датчика, заключающийся в его нестабильности и в первую очередь в температурной зависимости. Более удобными оказались датчики псевдо случайных чисел (ПСЧ), получаемые с помощью рекуррентных математических методов. ПСЧ нельзя назвать в полном смысле генераторами случайных чисел, поскольку последовательности этих чисел рассчитываются математически и могут быть в точности воспроизведены сколь угодно раз. Но с точки зрения решаемых с помощью этих ПСЧ задач они должны вести себя как идеально случайные числа. Важнейшим свойством последовательности ПСЧ должна быть её пригодность для решения определённого класса задач, в связи с чем любые алгоритмы и конкретные числовые значения должны проверяться на случайность генерируемых чисел, т.е. должны удовлетворять определённой группе тестов.

Ниже, в качестве примера рассмотрены два метода генерации ПСЧ.

а) Метод середины квадратов для генерации равномерно распределённых ПСЧ. Предложен Фон Нейманом в 1946 году. Алгоритм очевиден:

$X_0 = 4567$	$(X_0)^2 = 20857489$
$X_1 = 8574$	$(X_1)^2 = 73513476$
$X_2 = 5134$	$(X_2)^2 = 26357956$
$X_3 = 3579$	и т.д.

Полученную последовательность ПСЧ можно выстроить в ряд:

4567857451343579.....

Начальное значение в этом ряде ПСЧ (в данном случае  $X_0 = 4567$ ) выбирается произвольно, но при желании повторить прогон с тем же рядом случайных воздействий нужно начинать с одного и того же числа.



б) Смешанный конгруэнтный метод, являющийся в настоящее время наиболее популярным (2.2).

$$X_{i+1} = (aX_i + c) \bmod m. \quad (2.2)$$

Операция  $\bmod m$  означает получение остатка от деления некоторого числа (в данном случае содержимого скобок) на  $m$ .

Один из главных параметров генераторов ПСЧ – период повторения случайной последовательности – зависит от выбора параметров рекурренты. Для получения наибольшего периода рекомендуется выбирать  $c$  нечётное, а значение  $a$  подбирать из соотношения:

$a \bmod 4 = 1$ , т.е.  $a$  должно принимать одно из значений: 1, 5, 9, 13, 17 и т.д.

Для анализа качества генерируемой последовательности ПСЧ используется группа тестов. Наиболее важные из них:

а) Частотный тест. В десятичной системе каждая цифра должна появляться с частотой 0,1. В приведённой выше последовательности для метода середины квадратов из 16-и цифр цифра 5 встретила четыре раза, а цифры 0 и 2 – ни разу. Но делать заключение о непригодности данного метода получения ПСЧ по такой короткой выборке нельзя. Достаточно надёжную оценку можно получить при мощности выборки не менее  $N = 10000$ .

б) Сериальный тест. Частоты появления различных комбинаций. Так любые комбинации из трёх цифр (например, 222 или 357) должны появляться с частотой 0.001.

в) Интервальный тест. Подсчёт числа знаков между повторами каких-либо цифр и сравнение их с расчётными.

г) Автокорреляционные тесты. Корреляция между случайными числами  $X_i$  и  $X_{i+k}$ , где  $k$  – случайное смещение в последовательности чисел ( $k = 1, 2, 3, \dots$ ).

### 2.3.3 Получение случайных чисел с заданным законом распределения

Основной способ использует следующую зависимость:

Если случайная величина  $\eta$  имеет плотность распределения  $f(x)$ , то случайная величина  $\xi = \int_{-\infty}^{\eta} f(x)dx = F(\eta)$  имеет равномерное распределение, где  $F(\eta)$  – функция распределения. Таким образом, чтобы получить случайное число  $\eta$  с функцией распределения  $F(\eta)$ , следует взять равномерно распределённое случайное число  $\xi$  и решить уравнение  $F(\eta) = \xi$  относительно  $\xi$ .

Рассмотрим три наиболее распространённых случая.

а) Интервалы с экспоненциальным распределением.

Уравнение  $F(\eta) = \xi$  в данном случае принимает вид  $1 - e^{-\lambda t} = \xi^*$ . Отсюда  $e^{-\lambda t} = 1 - \xi^*$  и  $t = -\frac{1}{\lambda} \ln(1 - \xi^*)$ . Так как величина  $\xi^*$  распределена по равномерному закону, то можно произвести замену  $1 - \xi^* = \xi$ , которая упрощает основную формулу (2.3):

$$t = -\frac{1}{\lambda} \ln \xi. \quad (2.3)$$

Практически эта формула применяется следующим образом. С помощью датчика равномерно распределённых чисел выбирается случайное число  $0 < \xi < 1$  и для заданной интенсивности потока вызовов  $\lambda$  определяется интервал до очередного вызова. Например, при  $\lambda = 2$  выз/с получим: для  $\xi_1 = 0,7$  интервал  $t_1 = 0,18$  с, а для  $\xi_2 = 0,2$  получим  $t_2 = 0,8$  с. Формируя, таким образом длинную последовательность интервалов между вызовами, моделирующая программа имитирует пуассоновский поток вызовов.

Формирование интервалов, распределённых по экспоненте, можно проиллюстрировать и графически (Рисунок 2.3).

б) Равномерное распределение.

Для получения случайного числа  $x$ , равномерно распределённого между некоторыми числами  $a$  и  $b$ , нужно выбрать случайное число  $\xi$ , где  $0 < \xi < 1$ , и вычислить  $x = a + (b - a) \xi$ . Если  $a = 0$ , то  $x = b \xi$ .

Данный способ проиллюстрирован на рисунке 2.4.

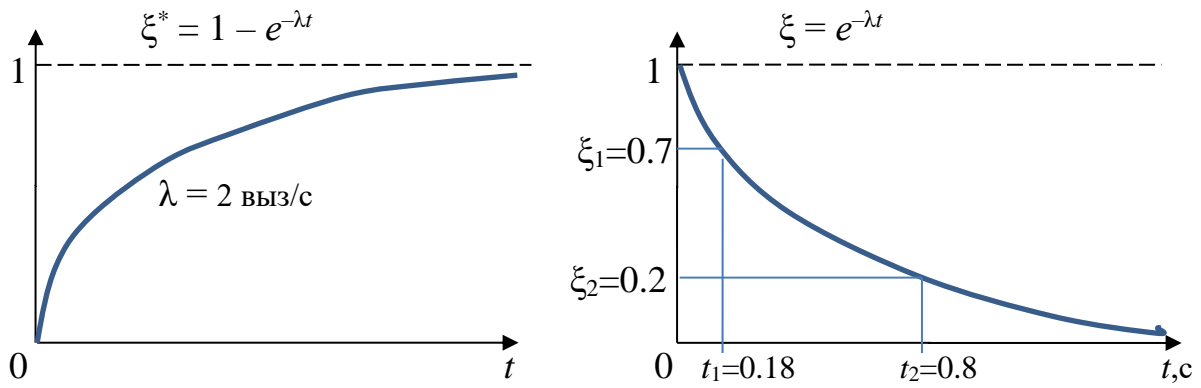


Рисунок 2.3 – Графическое определение интервалов  $t_1$  и  $t_2$  для  $\xi_1$  и  $\xi_2$

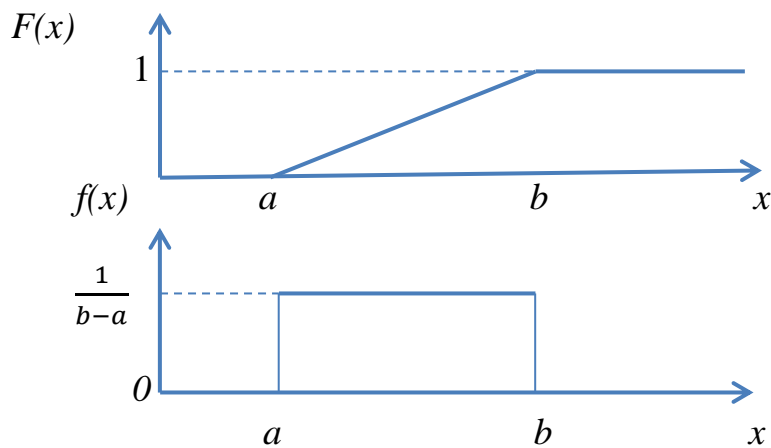


Рисунок 2.4 – Получение равномерно распределённого случайного числа

в) Выбор одного из равновероятных событий.

Такой выбор в практике имитационного моделирования встречается часто, например:

- выбор номера абонента, инициирующего в данный момент свой вызов;
- выбор номера порта коммутатора, на который поступает информационный кадр;
- выбор IP-адреса абонента, которому направляется информационный пакет, и т.д.

Во всех приведённых случаях принимается (в данной задаче), что рассматриваются равновероятные события.

Для получения случайного номера события нужно выбрать случайное число  $\xi$ , где  $0 < \xi < 1$ , и вычислить  $k = \xi N$ , где  $N$  – общее число событий. При дробном  $k$ , его значение округляется до целого в большую сторону.

Например, нужно выбрать один из 256-и входов коммутатора. Выбирается случайное число  $\xi$ , например,  $\xi = 0,3$  и вычисляется  $k = \xi N = 0,3 * 256 = 76,8$ . Иницируется 77-й порт коммутатора.

## 2.4 Механизм системного времени

### 2.4.1 Общие замечания

Функционирование имитационной модели должно протекать в искусственном времени, обеспечивая появление событий в нужном порядке и с необходимыми интервалами между ними. При этом удобно пользоваться не астрономическими единицами времени (минуты, секунды), а некоторыми условными единицами, так называемыми единицами модельного времени (е.м.в.). Соотношение между астрономическими единицами и е.м.в. остаются вне имитационной программы и устанавливаются программистом при планировании эксперимента. Например, если моделируется небольшая сельская или учрежденческая телефонная сеть с интенсивностью 4 вызова/мин (средний интервал между вызовами составляет 15 секунд), то целесообразно определить 1 е.м.в. = 1сек. При более интенсивных потоках, например, выдача IP-пакетов в канал связи с интенсивностью 1 млн. пакетов/с (средний интервал между пакетами составляет 1 мкс), можно выбрать 1 е.м.в. = 10 нс или 1 е.м.в. = 20 нс. Выбор значения е.м.в. произвольный, но направлен на удобство работы с операндами. Желательно избегать с одной стороны многозначных чисел, а с другой – дробных значений. Для операторов *generate* и *advance* удобнее подбирать двух – трёхзначные операнды.

Существует два основных механизма системного времени:

- шаг до следующего события;
- фиксированный шаг.

### 2.4.2 Механизм системного времени с шагом до события

Рассмотрим этот механизм на примере модели двухканальной системы обслуживания телефонных вызовов с потерями (Рисунок 2.5).

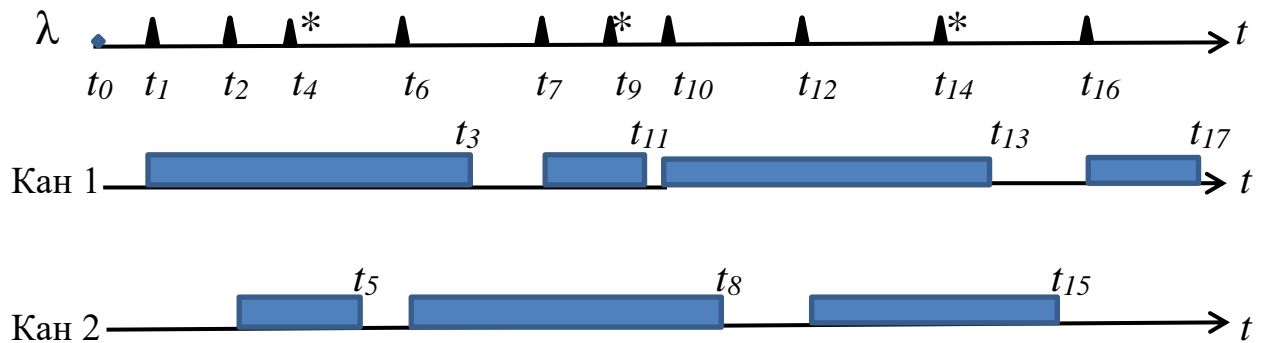


Рисунок 2.5 – Имитация двухканальной телефонной линии

В двухканальную систему поступает входящий поток  $\lambda$ . В начальный момент (системное время  $t_0 = 0$ ) оператор *generate* определяет интервал до 1-го вызова (момент  $t_1$ ) и программа сопровождения времени (ПСВ) помещает  $t_1$  в буфер будущих событий (ББС). Далее ПСВ выбирает в ББС самое меньшее время (в данный момент оно единственное и равно  $t_1$ ) и системное время (СВр) становится равным  $t_1$ . В этот момент программа должна выполнить следующие действия:

- оператор *generate* должен определить интервал до следующего вызова, т.е. найти момент  $t_2$ ;
- занять канал 1 и определить (с помощью оператора *advance*) момент его освобождения –  $t_3$ ;
- удалить из ББС время  $t_1$  и поместить туда времена  $t_2$  и  $t_3$ ;
- выбрать из ББС ближайшее время (в данном случае это  $t_2$ ) и сделать его новым системным временем, т.е. СВр =  $t_2$ .

В момент  $t_2$  необходимо:

- имитировать очередной вызов;
- определить время поступления следующего вызова ( $t_4$ );

- занять канал 2 и определить момент его освобождения –  $t_5$ ;
- удалить из ББС время  $t_2$  и поместить туда времена  $t_4$  и  $t_5$ ;
- выбрать из ББС ближайшее время (данном случае это  $t_4$ ) и сделать его новым системным временем, т.е.  $СВр = t_4$ .

В момент  $t_4$  необходимо:

- определить время поступления следующего вызова ( $t_6$ );
- зафиксировать потерю вызова, так как в момент  $t_4$  оба канала заняты;
- удалить из ББС время  $t_4$  и поместить туда время  $t_6$ ;
- выбрать из ББС ближайшее время (данном случае это  $t_5$ ) и сделать  $СВр = t_5$ .

В момент  $t_5$ .....и т.д.

В моменты времени  $t_3$  и  $t_5$  происходит освобождение соответствующего канала и учёт этого факта в собираемой статистике.

Рассмотрев весь прогон из 10 вызовов можно подсчитать, что 3 из них получили отказ в обслуживании (помечены \*), и сделать вывод, что вероятность потерь в этой системе составляет  $P = n/N = 3/10 = 0,3$ . Однако, делать выводы по таким коротким выборкам нельзя – слишком велика вероятность получить значительные отклонения полученного результата от истинного. Ниже (п. 2.6.2) будет показано, как зависит длительность прогона от желаемой достоверности результатов испытаний.

#### 2.4.3 Механизм системного времени с фиксированным шагом

При фиксированном шаге (ФШ) вся временная ось разделяется на длинную серию интервалов  $\Delta t$  и  $СВр$  при каждом цикле увеличивается на этот интервал. Проводится предварительный анализ событий, которые могут происходить в этих интервалах, и оценка вероятностей этих событий. В процессе моделирования для каждого интервала  $\Delta t$  происходит розыгрыш этих вероятностей и при наступлении какого-либо события проводится его реализация (поступление очередного вызова, занятие или освобождение канала и др.).

Рассмотрим этот процесс также на примере двухканальной системы с потерями.

Такая система может находиться в одном из трёх состояний (Рисунок 2.6). Перенумеруем эти состояния в соответствии с числом занятых каналов, т.е. состояния  $S0$ ,  $S1$  и  $S2$ .

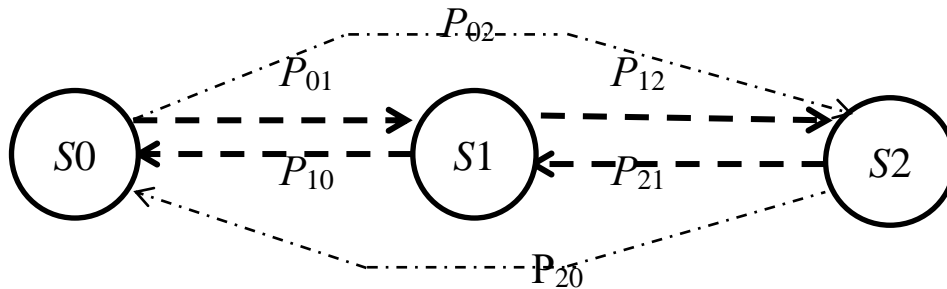


Рисунок 2.6 – Состояния системы и вероятности переходов

Модель двухканальной системы функционирует следующим образом:

- из состояния  $S0$  (оба канала свободны) система может выйти только в случае появления новых вызовов. Если за время  $\Delta t$  поступит 1 вызов, система перейдёт в состояние  $S1$ . Если – 2 вызова, то в состояние  $S2$ . Наконец, если за время  $\Delta t$  поступит более 2-х вызовов, то система перейдёт в состояние  $S2$ , а лишние вызовы будут потеряны;
- из состояния  $S1$  (занят один канал) система за время  $\Delta t$  может перейти в состояние  $S0$  (освободился один канал), в состояние  $S2$  (поступил один или более вызовов), или остаться в том же состоянии, если освободился один канал и поступил один вызов;
- из состояния  $S2$  (оба канала заняты) система может перейти в состояния  $S0$  или  $S1$ , если освободятся два или один канал, соответственно. При поступлении новых вызовов в состоянии  $S2$  они теряются;
- система может остаться в том же состоянии, если за время  $\Delta t$  не было ни новых вызовов, ни освобождений каналов.

При построении модели с фиксированным шагом (ФС), принципиальным вопросом является выбор интервала  $\Delta t$ . Число событий, которые могут наступить в каждом интервале может исчисляться десятками, и при малых  $\Delta t$

на розыгрыш вероятностей этих событий может уходить значительное время, что увеличит общее время прогона.

Выбирать большие интервалы  $\Delta t$  нельзя по следующим причинам:

- при ФШ за момент наступления события принимается конец интервала  $\Delta t$ , т.е. события в модели могут наступать только в определённые дискретные моменты времени, а не в произвольные моменты непрерывного времени как у реального объекта. Нарушается адекватность модели объекту и, при больших  $\Delta t$  это приведёт к несостоятельным оценкам;

- в большом интервале  $\Delta t$  может одновременно произойти больше одного события, а принцип ФШ определяет только факт возникновения события в интервале  $\Delta t$ , а не конкретный момент. И, например, в ситуации, когда в некотором интервале поступил новый вызов, и произошло освобождение канала, имеет значение очерёдность этих событий. Если вызов поступит раньше освобождения, то он будет потерян (при отсутствии других свободных каналов), а если позже, то он займёт только что освободившийся канал.

Единственный способ борьбы с этими неприятностями – уменьшение длительности интервала  $\Delta t$  настолько, чтобы вероятность наступления двух или более событий была пренебрежимо мала. Этого можно добиться, доведя,

например, вероятность появления интервала с одним событием до  $P_1 = 0,1$ . В таком случае вероятность нулевого интервала составит  $P_0 = 0,9$  и значит длительность прогона возрастёт на порядок. Пример расчёта системного времени с ФШ приводится ниже.

В настоящее время чаще применяется механизм системного времени с шагом до события.



## 2.5 Моделирование методом вложенной марковской цепи

Рассмотрим систему, которая может принимать одно из  $k$  состояний полной группы. Цепь называется марковской, если условная вероятность распределения  $P_{ij}(m)$  (перехода из состояния  $i$  в состояние  $j$ ) не зависит от номера испытания  $m$ , т.е.  $P_{ij}(m) = P_{ij}$  и результаты  $m$ -го испытания не зависят от предистории событий.

Рассмотрим, как и прежде, двухканальную систему с потерями (Рисунок 2.6) и механизм системного времени с фиксированным шагом. В каждый момент времени система может находиться в одном из трёх состояний в зависимости от числа занятых каналов  $S$  (т.е.  $S1, S2, S3$ ). Функционирование такой системы отображается марковской цепью с тремя состояниями (2.2):

$$P = \begin{vmatrix} P_{00} & P_{01} & P_{02} \\ P_{10} & P_{11} & P_{12} \\ P_{20} & P_{21} & P_{22} \end{vmatrix} \quad (2.2)$$

Чтобы определить значения представленных в матрице вероятностей перехода  $P_{ij}$ , нужно конкретизировать законы распределения случайных величин, с помощью которых описываются случайные процессы в имитационной модели. Пусть в систему поступает пуассоновский поток вызовов с параметром  $\lambda$ , а каналы обслуживают вызовы с экспоненциальным распределением длительности обслуживания  $t_{\text{обсл}}$  и с интенсивностью  $\mu$ , где  $\mu = 1/t_{\text{обсл}}$ .

Тогда вероятность поступления вызова за время  $\Delta t$  определится как  $P_\lambda = 1 - e^{-\lambda\Delta t}$ , а вероятность освобождения за время  $\Delta t$  занятого канала составит  $P_\mu = 1 - e^{-\mu\Delta t}$ . Подставляя полученные значения в матрицу (2.2), после несложных преобразований получим (2.3):

$$P = \begin{vmatrix} e^{-\lambda\Delta t} & 1 - e^{-\lambda\Delta t} & 0 \\ 1 - e^{-\mu\Delta t} & e^{-(\lambda+\mu)\Delta t} & 1 - e^{-\lambda\Delta t} \\ 0 & 1 - e^{-2\mu\Delta t} & e^{-2\mu\Delta t} \end{vmatrix} \quad (2.3)$$

Отметим, что данная матрица носит приближённый характер, поскольку предполагает, что  $\Delta t \rightarrow 0$ . Именно поэтому вероятности наступления двойных событий  $P_{02}$  (приход сразу двух вызовов) и  $P_{20}$  (освобождение сразу двух каналов) приняты равными нулю.

От данной матрицы легко перейти к процедуре розыгрыша событий в имитационной модели (Рисунок 2.7).

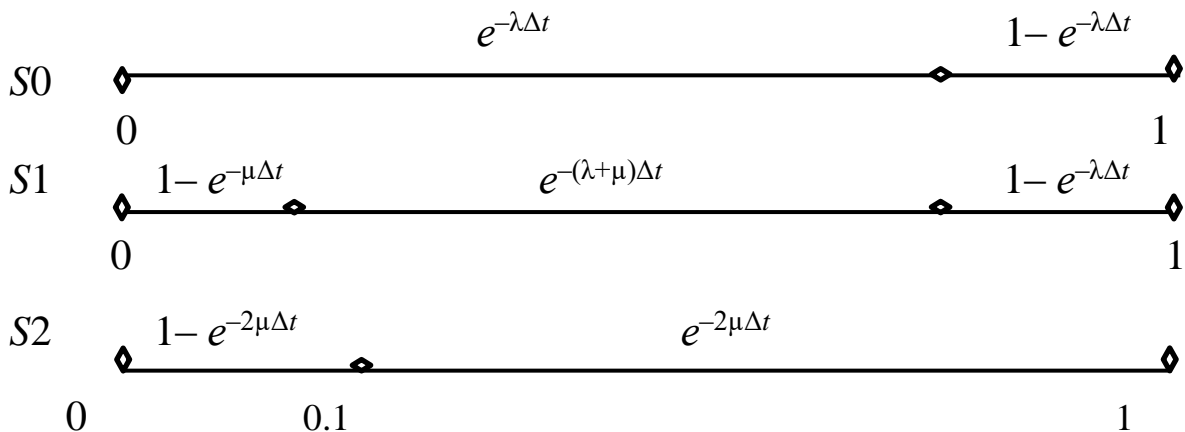


Рисунок 2.7 – Розыгрыш переходов в марковской цепи.

## 2.6 Методы планирования эксперимента

### 2.6.1 Стратегическое планирование

Важнейшим этапом имитационного моделирования является планирование эксперимента, которое разделяется на стратегическое и тактическое.

Стратегическое планирование – это определение цели эксперимента, т.е. того, ради чего предполагаются затраты материальных и людских ресурсов. Примерами таких целей могут быть следующие:

- приобретение нового оборудования – не слишком дорогого, но в то же время обеспечивающего необходимую пропускную способность;
- определение необходимого числа базовых станций для покрытия некоторого городского района с заданным качеством обслуживания (с допустимой вероятностью потерь);
- определение потока мультимедийного трафика между ЛВС предприятия и *Internet* для выбора пропускной способности канала доступа;

- определение интенсивности трафика между узлами магистральной сети для выбора пропускных способностей магистральных каналов;
- выбор дисциплины обслуживания разно приоритетных потоков пакетов в маршрутизаторе сети для получения наилучшего качества обслуживания (*QoS*).

При стратегическом планировании в первую очередь определяются выходные переменные, называемые реакцией (откликом) системы, которые интересуют исследователя. Как правило, это то, ради чего руководство компании решается на проведение имитационного моделирования.

Не менее ответственной задачей является выбор входных переменных (факторов). Естественно, что среди множества имитируемых факторов необходимо выбрать только те, которые влияют на выбранные отклики (Рисунок 2.8). Факторы бывают количественные (число абонентов, величина входного потока, пропускная способность канала и др.) и качественные (дисциплина обслуживания, распределение приоритетов и др.).

Не всегда удаётся сразу определить нужный перечень факторов, а тем более – необходимые пределы варьирования численных значений этих факторов. Поэтому, как показано на рисунке 2.2, необходим этап постепенного уточнения структуры модели. Проводя последовательно ряд экспериментов, можно уточнять перечень факторов и придаваемых им значений. При этом можно отказываться от варьирования факторов, не влияющих на выбранные отклики. Необходимо уточнять значения, присваиваемые переменным, сильно влияющим на отклики.



Рисунок 2.8 – Факторы и отклики при стратегическом планировании

### 2.6.2 Тактическое планирование

Под тактическим планированием понимают планирование одного эксперимента. При этом чаще всего решаются следующие вопросы:

- определение размера выборки;
- учёт начальных условий;
- сходимости результата;
- методы снижения дисперсии результата.

Рассмотрим два наиболее важных вопроса.

#### а) Определение размера выборки.

Размер выборки – это число единичных испытаний. Например, число пакетов, поступающих в систему в одном прогоне, или число телефонных вызовов в сотовой системе мобильной связи за прогон. Наиболее просто размер выборки определяется по оценке доверительного интервала для биномиальной схемы (2.4):

$$\varepsilon = \pm t_{\beta} \sqrt{\frac{D}{N}} = \pm t_{\beta} \sqrt{\frac{\tilde{p}(1-\tilde{p})}{N}}, \quad (2.4)$$

где  $\tilde{p}$  – оценка вероятности потери пакета или вызова,

$D$  – дисперсия оценки  $\tilde{p}$ ,

$N$  – длина прогона (размер выборки),

$\beta$  – доверительная вероятность,

$t_{\beta}$  – функция доверительной вероятности,

$\varepsilon$  – доверительное значение (допустимое отклонение от  $\tilde{p}$ ).

Истинное значение вероятности потерь  $p$  может отличаться от его измеренного на модели значения (оценки)  $\tilde{p}$ . Причём это отличие (разность  $|p - \tilde{p}|$ ) с доверительной вероятностью  $\beta$  будет не больше  $\varepsilon$  (2.5), т.е.

$$P(|p - \tilde{p}| < \varepsilon) = \beta \quad (2.5)$$

Сказанное проиллюстрировано на рисунке 2.9. Для рассматриваемой биномиальной схемы, отклонение различных значений оценок  $p$  от их среднего значения подчиняется нормальному закону распределения. При

этом, в выражении (2.5)  $\beta$  определяет вероятность того, что истинное значение измеряемой величины потерь  $p$  находится в доверительном интервале от  $(\bar{p} - \varepsilon)$  до  $(\bar{p} + \varepsilon)$ , как показано на заштрихованной части нормальной плотности распределения. Значение  $t_\beta$  в выражении (2.4) определяется по специально составленной таблице (2.1) для функции нормального распределения, фрагмент которой представлен ниже.

Таблица 2.1

$\beta$	0.8	0.9	0.95	0.99
$t_\beta$	1.28	1.64	1.96	2.58

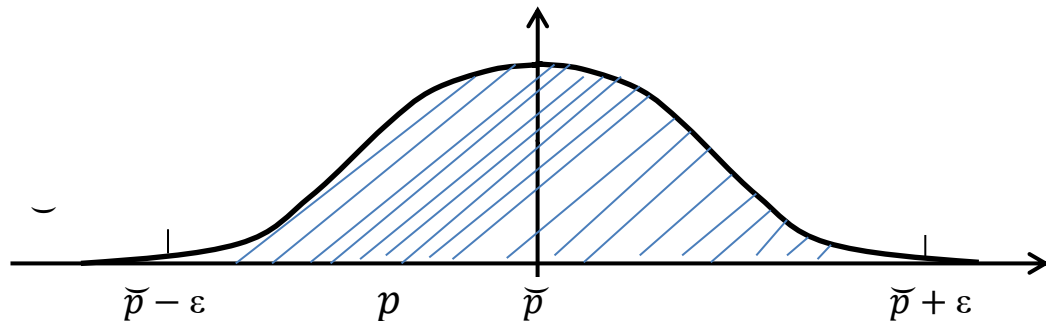


Рисунок 2.9 – определение доверительного интервала

Рассмотрим на частном примере – как влияет длительность прогона на величину доверительного интервала. Пусть в формуле (\*)  $\bar{p} = 0.1$  и  $t_\beta = 1.96$ , что соответствует доверительной вероятности  $\beta = 0.95$  (заштрихованная часть на рисунке 2.9). Тогда при длительности прогона  $N = 1000$  доверительный интервал окажется равным  $0.081 \div 0.119$ , т.е. с вероятностью 95% истинное значение вероятности потерь будет  $0.081 < p < 0.119$ . Если увеличить продолжительность прогона в 100 раз, т.е. сделать  $N = 100000$ , то доверительный интервал сократится в 10 раз и тогда, получим  $0.0081 < p < 0.0119$ . Напомним, что за измеренное значение вероятности потерь было выбрано  $\bar{p} = 0.1$ , т.е. в данном случае относительная ошибка не превышает 2%.

б) Учёт начальных условий – это учёт влияния на результаты эксперимента начального периода. Дело в том, что сразу после запуска модели все каналы и все места в очередях ещё пусты. Поэтому в этот период потери вы-

зовов в принципе невозможны и результаты прогона могут отметить слишком хорошее качество системы.

Предположим, что имитационная модель сети с коммутацией пакетов содержит 16 магистральных каналов с очередями. Длина каждой очереди равна 50, т.е. общее количество мест ожидания в системе равно 800. А теперь представим, что малоопытный экспериментатор установил длину прогона  $N = 600$ . Очевидно, что в этом случае даже при отказе всех каналов не будет зафиксировано ни одной потери пакета (все 600 введённых в систему пакетов займут места в очередях), и экспериментатор доложит руководству, что пропускные способности каналов, которые фирма собирается приобрести удовлетворяют заданным требованиям.

Для снижения влияния начального периода на результаты эксперимента рекомендуется:

- использовать максимально возможные по длине прогоны;
- исключать программными средствами начальный период из учитываемой статистики;
- выбирать начальные условия близкие к типичным (предварительная загрузка каналов и очередей).

### 3 Краткие сведения по структуре языка *GPSS*

#### 3.1 Модель как система массового обслуживания

В математических моделях (ММ) сложных объектов, представленных в виде систем массового обслуживания (СМО), фигурируют средства обслуживания, называемые обслуживающими аппаратами (ОА), и обслуживаемые заявки, называемые транзактами. Так, в модели многоканальной телефонной соединительной линии ОА отображают телефонные каналы, а транзакты – поступающие на обработку вызовы (заявки на телефонные разговоры).

Состояние СМО характеризуется состояниями ОА, транзактов и очередей к ОА. Параметры входных потоков заявок – внешние параметры СМО. Выходными параметрами являются величины, характеризующие свойства системы – качество ее функционирования. Примеры выходных параметров:

- производительность СМО – среднее число заявок, обслуживаемых в единицу времени;
- коэффициенты загрузки оборудования – отношение времен обслуживания (занятости ОА) к общему времени в каждом ОА;
- среднее время обслуживания одной заявки.

Основное свойство ОА, учитываемое в модели СМО, – это затраты времени на обслуживание. Например, длительность разговора, время передачи по пакета или кадра. Поэтому внутренними параметрами в модели СМО являются величины, характеризующие это свойство ОА. Обычно время обслуживания рассматривается как случайная величина и в качестве внутренних параметров фигурируют параметры законов распределения этой величины.

Имитационное моделирование позволяет исследовать СМО при различных типах входных потоков и интенсивностях поступления заявок на входы, при вариациях параметров ОА, при различных дисциплинах обслуживания заявок.

Дисциплина обслуживания – правило, по которому заявки поступают из очередей на обслуживание. Величина, характеризующая право на перво-

очередное обслуживание, называется приоритетом. В моделях СМО с очередями заявки, приходящие на вход занятого ОА, образуют очереди, отдельные для заявок каждого приоритета. При освобождении ОА на обслуживание принимается заявка из непустой очереди с наиболее высоким приоритетом.

Основной тип ОА – устройства. Именно в них происходит обработка транзактов с затратами времени. К ОА можно отнести также накопители (память), отображающие средства хранения сообщений, файлов, пакетов и других видов информации в узлах коммутации теле и инфокоммуникационных систем. Накопители характеризуются не временами обслуживания заявок, а емкостью – максимально возможным количеством одновременно находящихся в накопителе заявок.

К элементам имитационных моделей СМО, кроме ОА, относят также узлы и источники заявок. Связи ОА между собой реализуют узлы, т.е. характеризуют правила, по которым заявки направляются к тому или иному ОА.

Для описания моделей СМО при их исследовании на ЭВМ разработаны специальные языки имитационного моделирования. Существуют общецелевые языки, ориентированные на описание широкого класса СМО в различных предметных областях, и специализированные языки, предназначенные для анализа систем определенного типа. Примером общецелевых языков служит широко распространенный язык *GPSS*.

*GPSS (General Purpose Simulation System)* – общецелевая система моделирования сложных систем, разработанная Джеффри Гордоном. Первоначально разрабатывалась и поддерживалась компанией *IBM*. В настоящее время имеются версии различных разработчиков. *GPSS World* – самая современная версия *GPSS* для персональных ЭВМ и ОС *Windows*. Разработана компанией *Minuteman Software*. В России и странах СНГ распространяется ООО «Элина-Компьютер» (Казань).

Система *GPSS World* – это мощная среда компьютерного моделирования общего назначения, разработанная для профессионалов в области моделирования. Это комплексный моделирующий инструмент, охватывающий



области как дискретного, так и непрерывного компьютерного моделирования, обладающий высочайшим уровнем интерактивности и визуального представления информации.

Использование *GPSS World* дает возможность оценить эффект конструкторских решений в чрезвычайно сложных системах реального мира.

Система *GPSS* предназначена для написания имитационных моделей систем с дискретными событиями. Наиболее удобно в системе *GPSS* описываются модели систем массового обслуживания, для которых характерны относительно простые правила функционирования составляющих их элементов.

В системе *GPSS* моделируемая система представляется с помощью набора абстрактных элементов, называемых объектами. Каждый объект принадлежит к одному из типов объектов. Объект каждого типа характеризуется определенным способом поведения и набором атрибутов, определяемых типом объекта. Например, если рассмотреть работу телеграфиста почтового отделения связи, принимающего, сортирующего и отправляющего телеграммы, и работу маршрутизатора сети ЛВС или сети *Internet*, принимающего, обрабатывающего и выдающего пакеты в направления связи, то можно заметить большое сходство в их функционировании. В обоих случаях имеются объекты, постоянно присутствующие в системе (почта или маршрутизатор), которые обрабатывают поступающие в систему объекты (телеграммы или пакеты). В теории массового обслуживания эти объекты называются приборами и заявками. Когда обработка поступившего объекта заканчивается, он покидает систему. Если в момент поступления заявки прибор обслуживания занят, то заявка становится в очередь, где и ждет до тех пор, пока прибор освободится.

Очереди также можно представлять себе как некоторые объекты, функционирование которых состоит в хранении других объектов. Каждый объект может характеризоваться рядом атрибутов, отражающих его свойства. Например, прибор обслуживания имеет некоторую производительность, выражаемую числом заявок, обрабатываемых им в единицу времени. Сама за-

явка может иметь атрибуты, учитывающие время её пребывания в системе, время ожидания в очереди и т.д. Более сложные заявки, такие как информационные пакеты, могут иметь гораздо большее число атрибутов (параметров). Например, для *IP*-пакета в сети *Internet* атрибутами являются адреса отправителя и получателя, приоритет, *CoS* (класс сервиса) и др.

Характерным атрибутом очереди является ее текущая длина, наблюдая за которой в ходе работы системы (или ее имитационной модели), можно определить ее среднюю длину за время работы (или моделирования) и среднюю продолжительность ожидания.

В языке *GPSS* определены классы объектов, с помощью которых можно задавать приборы обслуживания, потоки заявок, очереди и т.д., а также задавать для них конкретные значения атрибутов.

Несколько часов, недель или лет работы исследуемой системы могут быть промоделированы на ЭВМ за несколько минут. Соотношение времени реального процесса и времени прогона модели зависит от характера реальных процессов (мощные потоки пакетов, телефонные вызовы, редкие отказы оборудования) и производительности моделирующей ЭВМ.

## 3.2 Программирование в системе *GPSS World*

### 3.2.1 Общецелевая система моделирования *GPSS World*

*GPSS World* – общецелевая система моделирования, основана на оригинальном языке компьютерного моделирования *GPSS*, имеет расширенные возможности, включая пользовательскую среду с интегрированными функциями работы с Интернет.

*GPSS World* разработана для оперативного получения достоверных результатов с наименьшими усилиями. В соответствии с этими целями в *GPSS World* хорошо проработана визуализация процесса моделирования, а также встроены элементы статистической обработки данных. Сильная сторона *GPSS World* – это его прозрачность для пользователя.

Прозрачность для пользователя ценна по трем причинам. Во-первых, опасно полагаться на непрозрачное моделирование типа “черный ящик”, внутренние механизмы функционирования которого скрыты от пользователя. Мало того, что в этом случае нельзя быть уверенным, подходит ли оно для какого-либо конкретного случая, но и невозможно гарантировать, что оно работает, как задумано. Во-вторых, удачные имитационные модели являются очень ценными и пригодны в течение длительного периода времени. Возможно, потребуется, чтобы новые сотрудники ознакомились с внутренними процессами модели, а это почти невозможная задача, если модель не имеет высокого уровня прозрачности. В-третьих, одним из наиболее эффективных, но наименее известных преимуществ компьютерного имитационного моделирования является возможность проникновения в самую суть поведения системы, когда опытный профессионал в области моделирования может видеть внутреннюю динамику в наиболее важные моменты времени процесса моделирования.

*GPSS World* был разработан с целью решить все эти проблемы. *GPSS World* является объектно-ориентированным языком. Его возможности визуального представления информации позволяют наблюдать и фиксировать внутренние механизмы функционирования моделей. Его интерактивность позволяет одновременно исследовать и управлять процессами моделирования. С помощью встроенных средств анализа данных можно легко вычислить доверительные интервалы и провести дисперсионный анализ. Кроме того, теперь есть возможность автоматически создавать и выполнять сложные отсеивающие и оптимизирующие эксперименты.

*GPSS World* был разработан, чтобы полностью использовать возможности имеющейся вычислительной системы. Использование механизма виртуальной памяти позволяет моделям реально достигать размера миллиарда байт. Вытесняющая многозадачность и многопоточность обеспечивают высокую скорость реакции на управляющие воздействия и дают возможность *GPSS World* одновременно выполнять множество задач. Это также означает,

что система моделирования *GPSS World* может использовать вычислительные возможности, предоставляемые симметричными многопроцессорными архитектурами (*SMP*).

*GPSS World* сочетает в себе функции дискретного и непрерывного моделирования. Возможность перехода из дискретной фазы моделирования в непрерывную фазу и обратно обеспечивает тесную связь с непрерывным моделированием. В непрерывной фазе могут быть установлены пороговые значения, управляющие созданием транзактов в дискретной фазе.

### 3.2.2 Основные элементы и свойства

**Транслятор.** Высокоэффективный транслятор является частью программы *GPSS World*, который создает машинную программу (программу для процессора) из исходной программы разработчика. Перед включением в объект “Процесс моделирования” все операторы модели проходят трансляцию. Точно так же интерактивные операторы транслируются в глобальной области видимости прежде, чем они будут переданы существующему объекту “Процесс моделирования”.

**Архитектура.** На уровне интерфейса *GPSS World* представляет собой реализацию архитектуры “документ-вид”, общей для всех приложений операционной системы *Windows*. Объекты могут быть открыты в нескольких окнах, изменены и сохранены на постоянных носителях информации. Привычное меню главного окна и блокировка недоступных команд меню, не отвлекая внимания, направляет пользователя к конечной цели. *GPSS World* был разработан с целью достичь тесной интерактивности даже в многозадачной среде с использованием виртуальной памяти.

**Многозадачность.** Многопоточная архитектура *GPSS World* позволяет совместно запускать несколько процессов моделирования и экспериментов. Одновременно выполняются не только обновление окон, пользовательский ввод, дисковый ввод-вывод, печать и процесс моделирования, но также в од-

но и то же время может быть запущено любое количество процессов моделирования.

**Виртуальная память.** Процессы моделирования непосредственно не ограничиваются размером физической памяти с произвольным доступом (ОЗУ), в которой выполняется объект “Процесс моделирования”. Используя механизм виртуальной памяти, модели могут достигать размера до гигабайта. Количество объектов также ограничивается только обеспечиваемым размером файла подкачки. Для достижения оптимальной производительности необходимо использовать значительный объем реальной памяти. Выделение и управление памятью для объектов происходит невидимо для пользователя. Объекты автоматически создаются до тех пор, пока не потребуются дополнительная информация.

**Интерактивность.** *GPSS World* поддерживает высокий уровень интерактивности даже во время процесса моделирования. Используя команду главного меню окна модели ***Command* (Команда)**, ускоряющие клавиши или настройки модели, закрепляя за функциональными клавишами собственные команды, можно передавать существующему объекту “Процесс моделирования” любой оператор. Можно использовать диалоговое окно “*Simulation Command*” (“Команда”) для ввода операторов, отсутствующих в выпадающем меню, а с помощью команды *include* можно посылать процессу моделирования интерактивные операторы любой сложности.

**Визуализация.** *GPSS World* отличается высоким уровнем визуализации выполняющегося процесса моделирования. Для наблюдения и взаимодействия с процессом моделирования используются двадцать различных окон, соответствующих большей части объектов *GPSS*. Для получения, сохранения и печати визуального представления состояния процесса моделирования не требуется дополнительных усилий, кроме операций с окнами.

**Анимация.** В *GPSS World* существует ряд анимационных возможностей. Уровень их реализма изменяется от абстрактной визуализации, не тре-

бующей никаких усилий, до высоко реалистических динамических изображений, включающих в себя сложные элементы, созданные пользователем.

**Абстрактная анимация.** В *GPSS World* встроена функция комплексной стилизованной анимации. Эта функция обеспечивается большим количеством окон, которые динамически отображают объекты *GPSS* в процессе моделирования по мере изменения их состояния. Для создания такой анимации не требуется дополнительных усилий. Графические изображения могут быть сохранены для включения в отчеты и/или распечатаны.

**Пост-процессорная анимация.** *GPSS World* имеет внешний интерфейс, который может поддерживать анимационные пакеты, управляемые трассировочными данными. Таким способом могут быть разработаны фотореалистичные анимации. Для использования этого интерфейса необходимо создать выходной поток, содержание и форматирование данных которого удовлетворяет правилам анимационного пакета, поставляемого сторонним разработчиком.

**Оперативная анимация.** *GPSS World* имеет набор *PLUS*-процедур динамического вызова, позволяющих вызывать функции во внешних исполняемых файлах. Это обеспечивает оперативную связь с анимационными пакетами других разработчиков.

**Язык моделирования.** *GPSS World* является реализацией *GPSS*, общецелевой системы моделирования, улучшенной встроенным языком программирования *PLUS* – языком программирования низкого уровня моделирования.

Эта версия *GPSS* включает в себя 53 типа блоков и 25 команд, а также более чем 35 системных числовых атрибутов, которые обеспечивают текущие переменные состояния, доступные в любом месте модели.

*PLUS* – это небольшой, но эффективный процедурный язык программирования, созданный из 12 типов операторов. Его эффективность во многом обеспечивается большой библиотекой процедур, содержащей математиче-

ские функции и функции манипуляции со строками, и большого набора вероятностных распределений.

В *GPSS World* модель определяется как последовательность операторов. Это операторы *GPSS*, операторы *PLUS*-процедур или операторы *PLUS*-экспериментов. За исключением списков данных функции, все операторы *GPSS* должны состоять из одной текстовой строки длиной до 250 символов. Любой оператор *GPSS* может входить в модель и сохраняться в файле модели или может быть передан процессу моделирования в интерактивном режиме.

**Полиморфные типы данных.** Переменные могут принимать значения одного из четырех типов. Переменные, управляемые пользователем, такие, как ячейки, элементы матриц, параметры транзактов и переменные пользователя, могут принимать целочисленное, вещественное, строковое и неопределенное значение. Значения времени могут быть целыми или вещественными двойной точности.

Преобразования типов происходит автоматически. Процедуры, которые требуют в качестве аргумента значение определенного типа данных, принудительно преобразуют аргумент к соответствующему типу. Например, если вы передаёте строку процедуре, которая требует числовое значение, будет использован числовой эквивалент этой строки. Точно так же, если вы пытаетесь записать (*write*) числовое значение, оно будет автоматически преобразовано в текстовую строку.

Строковые величины имеют много вариантов использования. Они могут использоваться в потоках данных для создания отчетов и файлов с результатами, а также для прямого доступа к внутренним данным. Строковые константы обозначаются заключением строки между парой двойных кавычек. Для представления двойных кавычек внутри строки используется пара двойных кавычек. Например, для представления одной строки внутри другой строки необходимо использовать 6 символов кавычек. Внутренняя строка (подстрока) заключается между двумя парами двойных кавычек, а итоговая

строка будет заключаться между еще одной парой двойных кавычек. Библиотека процедур содержит много строковых процедур, которые могут использоваться для создания и манипулирования строковыми данными.

Теперь ячейкам и матрицам может быть присвоено значение *unspecified* (неопределенный тип данных). Команда *initial* позволяет присваивать это значение ячейкам, элементам матрицы и даже целым матрицам. Если элемент данных *unspecified* был использован в операции, которая требует некоторого значения, произойдет останов по ошибке. Библиотечная процедура дисперсионного анализа (*ANOVA*) теперь распознает элементы *unspecified* в матрице результатов как недоступные результаты прогонов эксперимента.

**Многомерные матрицы.** Матрицы могут иметь до 6 измерений. Динамические окна “*Matrix*” (“Матрица”) позволяют просматривать любые поперечные сечения матрицы, одновременно можно открыть любое количество таких окон. Для использования во время исполнения процедуры могут быть созданы временные матрицы. Для инициализации матрицы с тремя и более измерениями используются присваивающие *PLUS*-операторы.

**Выражения.** *GPSS World* поддерживает широкое использование выражений. Они могут использоваться в *PLUS*-процедурах или в операторах *GPSS* (если заключены в скобки). Это означает, что в операндах блоков и команд могут производиться эффективные вычисления. Выражения могут производить простые вычисления, вызывать процедуры, выполняющие математические или строковые операции, производить выбор вероятностного распределения или выполнять заданные пользователем алгоритмы, включая файловый ввод-вывод.

### 3.2.3 Классы языка *GPSS*

Язык *GPSS* состоит из наборов элементов, разделяемых на 5 классов:

- динамические;
- аппаратно-ориентированные;



- статистические;
- операционные;
- вычислительные.

**Динамические** элементы, соответствующие заявкам в системе массового обслуживания (СМО), называются *GPSS* транзактами. Они создаются и уничтожаются так, как это нужно в процессе моделирования. С каждым транзактом может быть связано некоторое число параметров, несущих в себе необходимую информацию о каждом транзакте. Например, при моделировании сети коммутации пакетов такими параметрами могут быть: время возникновения пакета, адрес получателя, адрес отправителя, категория срочности, принадлежность данного пакета к определённому сообщению и др.

**Аппаратно-ориентированные** элементы, соответствующие обслуживающим приборам СМО, включают в себя одноканальные и многоканальные устройства. Одноканальное устройство или просто устройство, может обслуживать одновременно только один транзакт (передавать один пакет, поддерживать один телефонный разговор и др.). Многоканальное устройство (МКУ) – несколько транзактов одновременно. Количество каналов в МКУ указывается в начале программы специальным оператором *storage*.

**Статистические** элементы *GPSS* служат для сбора и обработки статистических данных о функционировании моделируемой системы. Это очереди и таблицы. Каждая очередь обеспечивает сбор и обработку данных о транзактах, задержанных в какой-либо точке модели, например, перед устройством обслуживания (коммутатор, маршрутизатор). При этом, собирается подробная статистика о таких параметрах очереди как: средняя длина очереди, средняя продолжительность ожидания, максимальная наблюдавшаяся в данном прогоне длина очереди и других. Информация по аппаратно-ориентированным элементам содержит среди прочих данных и такие важные показатели как коэффициент использования канала, а в МКУ – среднее число использованных за время прогона каналов.

Таблицы используются для получения распределений некоторых случайных величин. Например, время пребывания транзакта в модели, время задержки пакета в сети, время нахождения пакета в очереди, количество повторных вызовов, сделанных абонентом до установления соединения и др.

**Операционные** элементы *GPSS*, называемые блоками, формируют логику работы модели, давая транзактам указания, куда идти и что делать дальше. Модель системы на *GPSS* – совокупность блоков, объединённых в соответствии с логикой работы реальной системы в блок-схему. Блок-схема – это фигуры, соединённые между собой линиями, показывающими взаимодействие блоков в процессе моделирования. Блоки могут включать части программы, отображающие, например, работу маршрутизатора, процессы, происходящие в мобильном центре коммутации, работу одной базовой станции и др.

К **вычислительным** элементам относятся арифметические переменные и функции. Они используются для вычисления некоторых величин, заданных арифметическими выражениями или табличными зависимостями. Арифметические переменные и функции описываются в самом начале программы. Например, описание функции назначения приоритетов поступающим в систему пакетам:

*prior function rn1,d3* ;аргумент – случайное число (*rn1*)  
*.5,1/.8,2/1,7* ;функция – значения приоритетов 1, 2 или 7.

По этой функции 1-й приоритет будет присваиваться пакетам в 50% случаев, 2-й – в 30% и 7-й – в 20%. Обращение к этой функции из программы происходит по её символическому имени (в данном случае это метка – *prior*).

Выполнение программы *GPSS*-модели происходит с помощью управляющей программы – симулятора. Симулятор перемещает транзакты от блоков к блокам подобно тому, как в моделируемой СМО движутся заявки (пакеты), изображаемые транзактами.

### 3.2.4 Блоки *GPSS*

Каждый блок содержит некоторую детализирующую информацию, которая может относиться к трём различным категориям:

а) Местоположение. Каждый блок занимает определённое место в блок-схеме. Это место может быть помечено и символическим именем (меткой).

б) Операция – функциональное назначение блока.

в) Операнды – задают информацию, специфичную для данного типа блока. Операнды обозначаются буквами *A, B, C, D, E, F, G*. Есть обязательные и необязательные операнды. Для записи операндов используется позиционный принцип, т.е. запятые присутствуют всегда, кроме конечной. Некоторые операнды задаются по умолчанию.

Рассмотрим один из наиболее сложных блоков, каким является блок формирования транзактов *generate A,B,C,D,E* (Таблица 3.1)

Таблица 3.1 – Параметры блоков транзактов

Операнд	Значение	Значение по умолчанию
<i>A</i>	Средняя величина интервалов между поступлениями транзактов	Обязательный операнд
<i>B</i>	Модификатор разброса интервалов между транзактами	Ноль
<i>C</i>	Смещение начала выдачи транзактов	Ноль
<i>D</i>	Ограничение числа выдаваемых транзактов	Ноль
<i>E</i>	Уровень приоритета выдаваемого транзакта	Ноль

### 3.2.5 Движение транзактов во времени

В начальный момент в модели нет ни одного транзакта. В процессе моделирования транзакты возникают (блок *generate*) и уничтожаются (блок *terminate*) так же как вызовы (требования, пакеты) входят и выходят из моделируемой системы.

В модели одновременно может существовать множество транзактов, но симулятор в каждый момент продвигает только один транзакт, при этом транзакт продвигается от блока к блоку до одного из следующих событий:

- транзакт входит в блок *advance* и задерживается в нём на время, указанное в его операндах;
- транзакт входит в блок *terminate* и покидает модель;
- транзакт пытается войти в следующий блок, но тот отказывается его принять. Например, это может произойти, если следующим устройством является одноканальное устройство, которое в данный момент занято. Транзакт задерживается в предыдущем блоке, пока появится возможность продвижения (например, пока заняты все каналы).

В этих случаях симулятор прекращает обработку данного транзакта и начинает перемещение очередного транзакта.

Программисту доступны через стандартный числовой атрибут (СЧА)  $W$  – текущее содержимое блока (например, число каналов в многоканальном устройстве или число транзактов в очереди), а через СЧА  $N$  – общее количество входов в блок с начала прогона.

### 3.2.6 Таймер модельного времени

Каждое продвижение транзакта это событие, происходящее в определённый момент времени. Для соблюдения правильной временной последовательности симулятор имеет таймер модельного времени, который корректируется в соответствии с логикой программы.

Особенности работы таймера:

- единица модельного времени определяется разработчиком, который задаёт все временные интервалы в данной программе в одних и тех же единицах;

– симулятор продвигает таймер к моменту очередного ближайшего события, т.е. в *GPSS* реализована процедура модельного времени «до ближайшего события», как это было описано в п. 2.4.2.

Значения таймера доступны программисту через системные СЧА: *c1* (относительное время) и *ac1* (абсолютное время).

Для определения очередного инициализируемого транзакта симулятор ведёт два списка транзактов:

– список текущих событий (СТС) содержит транзакты, планируемое время продвижения которых равно или меньше текущего модельного времени (оно меньше для тех, которые были заблокированы, как например, вызовы или пакеты, стоящие в очереди и ждущие освобождения канала);

– список будущих событий (СБС) содержит транзакты, планируемое время продвижения которых больше текущего. Это, например, вызовы или пакеты, обрабатываемые в каналах связи, когда время окончания обработки уже определено. СБС организуется в порядке возрастания планируемого времени продвижения транзактов и используется каждый раз, когда нужно увеличить системное время до очередного ближайшего события.

СТС и СБС идентифицируют транзакты по их порядковому номеру в текущем прогоне.

### 3.2.7 Управляющие операторы *GPSS*

Оператор *simulate* – начало.

Оператор *Start* – длина прогона. Например, 10000.

Оператор *clear* – обнуляет все значения, набранные в модели, кроме значений генераторов *c1* и *ac1*.

Оператор *reset* – сброс накопленной статистической информации. При этом состояния аппаратных (числа каналов), динамических и запоминающих (длины очередей) объектов и генераторов случайных чисел сохраняются. Таймер *c1* – обнуляется, *ac1* – сохраняется.

### 3.3 Краткое описание языка GPSS

#### 3.3.1 Структура оператора

Модель (программа) на языке *GPSS* представляет собой последовательность операторов (их также называют блоками), отображающих события, происходящие в СМО при перемещениях транзактов. Поскольку в интерпретаторах *GPSS* реализуется событийный метод, и в СМО может быть одновременно много транзактов, то интерпретатор будет попеременно исполнять разные фрагменты программы, имитируя продвижения транзактов в текущий момент времени до их задержки в некоторых устройствах или очередях.

В записи оператора выделяют четыре части: метку, название, поле переменных и комментарий.

Пример оператора. Цифрами указаны позиции в строке, на которых могут располагаться четыре части оператора.

```
met   generate   30,5   ;генератор телефонных вызовов
2....6    8.....18 .....19.....70
<метка> <название>  <поле переменных> <комментарий>
```

Метка может занимать позиции, начиная со второй, имя оператора – с восьмой, поле операндов – с девятнадцатой, комментарий обязательно отделяется от поля операндов пробелом и знаком «;».

Поле операндов может быть пусто, иметь один или более операндов, обозначаемых ниже при описании блоков символами *A, B, C, ...* Операндами могут быть идентификаторы устройств, накопителей, служебные слова и стандартные числовые атрибуты (СЧА).

#### 3.3.2 Стандартные числовые атрибуты

К СЧА относятся величины, часто встречающиеся в разных задачах. К числу наиболее употребимых СЧА относятся следующие операнды:

- $s$  – объем занятой памяти в накопителе или число занятых каналов в многоканальном устройстве;
- $r$  – объем свободной памяти в накопителе (свободных каналов);
- $f$  – состояние устройства (заполнено, свободно, частично заполнено);
- $q$  – текущая длина очереди;
- $p$  – параметр транзакта (каждый транзакт может иметь до 12 параметров);
- $v$  – целочисленная переменная (вещественная и булева переменные обозначаются  $fv$  и  $bv$  соответственно);
- $x$  – хранимая переменная (переменная, для которой автоматически подсчитывается статистика);
- $ac1$  – текущее время;
- $fn$  – функция;
- $rn$  – случайная величина;
- $rn1$  – случайная величина, равномерно распределенная в диапазоне  $[0, 1]$  и другие.

При этом ссылки на СЧА записываются в виде:  $\langle \text{СЧА} \rangle \$ \langle \text{идентификатор} \rangle$ .

Например,  $q\$ord$  означает очередь с именем  $ord$ , а  $fn\$cos$  – ссылка на функцию с именем  $cos$ .

Стандартные числовые атрибуты служат для сокращенного указания различных величин, фигурирующих в модели.

Примеры СЧА:  $v2$  – переменная номер 2;  $q4$  – длина очереди номер 4;  $x5$  – хранимая величина номер 5;  $fn7$  – функция номер 7;  $p4$  – значение параметра номер 4 транзакта;  $*6$  – содержимое параметра номер 6 транзакта;  $s*3$  (или  $fn*3$  – память (или функция), определенная в параметре номер 3 транзакта).

Каждому объекту в *GPSS* (транзакт, блок, память, очередь и др.) соответствуют атрибуты, описывающие его состояние в данный момент времени

(арифметические или логические). СЧА бывают адресуемые (относящиеся к конкретному объекту) и безадресные.

Примеры адресуемых СЧА:

- $n\$met1$  – число транзактов, входивших в блок с меткой  $met1$ ;
- $s\$kan3$  – текущее содержимое памяти (например, число занятых каналов в многоканальном устройстве) с именем  $kan3$ ;
- $r\$kan3$  – число свободных ячеек памяти (например, число свободных каналов в многоканальном устройстве) с именем  $kan3$ ;
- $q\$vetv5$  – текущая длина очереди к каналам ветви с именем  $vetv5$ ;
- $qa\$vetv5$  – средняя длина очереди к каналам ветви с именем  $vetv5$ .

Безадресные СЧА:

- $c1$  – текущее значение относительного времени с начала процесса моделирования;
  - $ac1$  – текущее значение абсолютного времени с начала процесса моделирования;
- $rn1 \div rnN$  – случайные числа, выдаваемые одним из датчиков случайных чисел.

Полный перечень СЧА языка *GPSS* с пределами их изменений приведён в приложении Г.

### 3.3.3 Примеры наиболее часто встречающихся операторов

Рассмотрим наиболее часто встречающиеся операторы, сопровождая знакомство с ними простыми примерами моделей.

а) Источники заявок обычно описываются блоком:

*generate A,B,C,D,E*

Здесь  $A$  и  $B$  служат для задания интервалов в единицах модельного времени (е.м.в.) между появлениями заявок, при этом можно использовать один из следующих вариантов:



- интервал – равномерно распределённая в диапазоне  $[A - B, A + B]$  случайная величина;
- интервал – значение функции, указанной в  $B$ , умноженной на  $A$ ;
- $C$  – задержка в выработке первого транзакта в е.м.в.;
- $D$  – число вырабатываемых источником транзактов. Если  $D$  пусто, то число вырабатываемых транзактов не ограничено;
- $E$  – приоритет заявок. Если  $E$  пусто, то приоритет заявки равен 0 (самый низший). Чем выше значение  $E$ , тем выше приоритет.

Например:

*generate 6,fn\$exp,15*

Этот оператор описывает источник, который вырабатывает 15 транзактов с интервалами, равными произведению числа 6 и значения функции EXP;

*generate 36,12*

Здесь число транзактов не ограничено, интервалы между транзактами – случайные числа в диапазоне  $[24, 48]$ , т.е. диапазон определяется как  $36 \pm 12$ .

б) Функции, на которые имеются ссылки в операторах, должны быть описаны с помощью блока следующего типа:

*met1 function A, B*

За ним следует строка, начинающаяся с первой позиции:

$x_1, y_1/x_2, y_2/x_3, \dots/x_m, y_n$

Здесь метка *met1* – идентификатор функции,  $A$  – аргумент функции,  $B$  – тип функции,  $x_i$  и  $y_i$  – координаты узловых точек функции, заданной таблично. Например:

*exp function rn1, c12*

0,0/.2,.22/.4,.51/.5,.6/.6,.92/.7,1.2/.8,1.61/.9,2.3/.95,3/.99,4.6/.999,6.9/1,1000

Это описание непрерывной ( $c$ ) функции *exp*, заданной таблично 12-ю узловыми точками, аргументом является случайная величина (*rn1*), равномерно распределённая в диапазоне  $[0, 1]$ . Или:

*bbb function \*4, d6*

1,2/2,5/3,11/4,20/5,18/6,12/7,9

Дискретная (*d*) функция *bbb* задана 6-ю узловыми точками, аргумент – четвертый параметр транзакта, возбудившего обращение к функции с меткой *bbb*.

Здесь аргумент задан с использованием косвенной адресации, признаком которой является символ \*, т.е. запись \*4 означает, что аргументом является величина, указанная в 4-м параметре транзакта, вызвавшего функцию (в данном примере можно использовать равноценную запись \**p*4). В общем случае косвенная адресация выполняется путем записи операнда в виде СЧА\*СЧА, например: *s*\**x*2, что означает емкость памяти, занятая в накопителе, именем которого является значение переменной *x*2, или *q*\**p*5 – длина очереди с именем, записанным в параметре 5 транзакта.

в) Транзакты могут порождаться и оператором размножения:

*split*    *A, B, C*

Новые транзакты порождаются, когда в данный блок входит некоторый транзакт. При этом создается семейство транзактов, включающее основной (вошедший в блок) транзакт и *A* его копий. Основной транзакт переходит в следующий по порядку блок, а его копии переходят в блок с меткой *B*. Для различения транзактов параметр *C* основного транзакта увеличивается на 1, а транзактов-копий – на 2, 3, 4,... и т.д.

Обратное действие – сборка транзактов выполняется операторами:

*assemble*    *A*

*gather*    *A*

Согласно оператору *assemble* первый из вошедших в блок транзактов выйдет из него только после того, как в этот блок придут еще *A* – 1 транзактов того же семейства. Этот оператор удобен, например, для сборки сообщения, состоящего из нескольких пакетов. Второй оператор отличается от предыдущего тем, что из блока выходят все *A* транзактов.

г) Операторы занятия транзактом и освобождения от обслуживания одноканального устройства *A*:

*seize*    *A*

*release A*

Задержка в движении транзакта по СМО описывается оператором:

*advance A,B*

Здесь операнды *A* и *B* имеют тот же смысл, что и в операторе *generate*.

Пример:

Обслуживание транзакта в устройстве *kan* продолжительностью  $\mu$  единиц модельного времени, где  $\mu$  – равномерно распределенная в диапазоне [7,11] случайная величина, описывается следующим фрагментом программы:

*seize kan*

*advance 9,2*

*release kan*

Аналогично описывается занятие транзактом памяти в накопителе, например, в многоканальном устройстве:

*enter vetv,b*

Здесь помимо имени накопителя (*vetv*) указывается объем требуемой данным транзактом памяти (*b*).

Освобождение *b* ячеек памяти в накопителе с именем *vetv* выполняется оператором:

*leave vetv,b*

Отсутствие операнда *b* означает, что занимается (освобождается) только один канал.

Для накопителей в модели нужно задавать общий объем памяти, что делается в следующем описании накопителя:

*vetv storage n*

Здесь: *vetv* – имя накопителя,

*n* – объем его памяти (число каналов в многоканальном устройстве с именем *vetv*).

В целом данный фрагмент программы примет следующий вид:

*vetv storage n* ;определить устройство с *n* каналами

*enter vetv,b* ;занять *b* каналов

*advance* 9,2 ;задержать транзакт на  $9 \pm 2$  е.м.в.  
*leave vetv,b* ;освободить  $b$  каналов

д) Если транзакт приходит на вход занятого устройства или на вход накопителя с недостаточным объемом свободной памяти, то транзакт задерживается в очереди к этому устройству или накопителю. Слежение за состоянием устройств и очередей выполняет интерпретатор. Но если в модели требуется ссылаться на длину очереди или собирать статистику по ее длине, то требуется явное указание этой очереди в модели. Делается это с помощью операторов входа в очередь и выхода из очереди:

*queue vetv,b*  
*depart vetv,b*

Согласно этим операторам очередь  $A$  увеличивается и уменьшается на  $b$  единиц соответственно, если  $b=1$ , то поле  $B$  можно оставить пустым, т.е.:

*queue vetv*  
*depart vetv*

### 3.3.4 Изменение маршрутов транзактов.

Изменение линейной последовательности выполнения блоков в системе GPSS осуществляется с помощью следующих блоков:

а) Условные переходы.

Блок *transfer A,B,C* например:

– *transfer .75,met1,met3*

с вероятностью 0.75 активный транзакт перейдет к блоку с именем *met3*, а с вероятностью 0.25 – к блоку с именем *met1*;

– *transfer .75,met3*

с вероятностью 0.75 активный транзакт перейдет к блоку с именем *met3*, а с вероятностью 0.25 – к следующему блоку;

– *,fin*

Безусловный переход к блоку с именем *fin*.

б) Блок *test x A,B,C*

В данном случае переход к оператору, помеченному меткой  $C$ , происходит, если не выполняется условие  $A \times B$ . Операнд  $x$  может принимать следующие значения:  $e$  – равно;  $ne$  – неравно;  $l$  – меньше;  $le$  – меньше или равно;  $g$  – больше;  $ge$  – больше или равно. Например:

*test l q\$vetv5,10,met7*

Если текущая длина очереди в ветви 5 ( $q$vetv5$ ) меньше 10-и, выполняется следующая команда. В противном случае осуществляется переход к блоку с меткой  $met7$ . С помощью этой конструкции можно проверять возможность установки пакета (вызова) в очередь ограниченной длины. В данном случае число мест ожидания в очереди  $vetv5$  установлено равным 10-и.

в) Блок *gate x kan3,met4*

Условный переход к метке  $met4$  в зависимости от состояния операнда  $x$ , характеризующего многоканальное устройство с именем  $kan3$ . Операнд  $x$  может принимать следующие значения:  $u$  (устройство занято);  $ni$  (устройство свободно);  $i$  (устройство захвачено);  $ni$  (устройство не захвачено);  $se$  (МКУ пусто);  $sne$  (МКУ не пусто);  $sf$  (МКУ заполнено);  $snf$  (МКУ не заполнено).

г) Оператор перехода в циклических процедурах выглядит так:

*loop a,b*

Здесь  $a$  – номер параметра транзакта, в котором содержится число повторений (витков) цикла,  $b$  – метка оператора, с которого начинается повторяющаяся часть.

### 3.3.5 Способы задания экспоненциального закона распределения

#### а) Аналитический способ

Экспоненциальный закон распределения описывается плотностью

$$P(t) = \lambda \exp(-\lambda t),$$

где  $\lambda$  – интенсивность потока событий с функцией распределения экспоненциального закона (3.1):

$$F(T) = \int_0^T p(t) dt = 1 - \exp(-\lambda T). \quad (3.1)$$

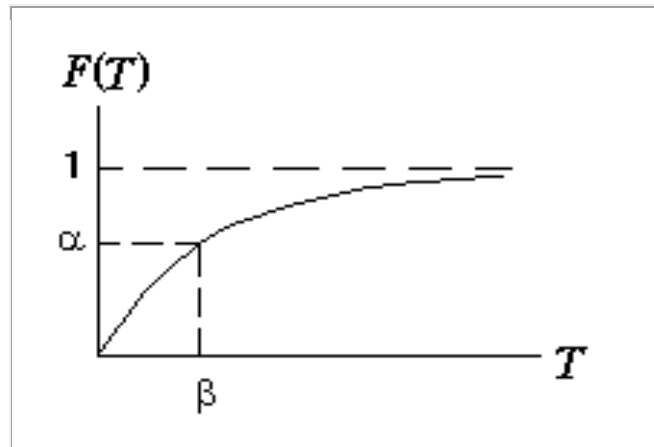


Рисунок 3.1 Функция экспоненциального закона распределения

Из рисунка 3.1 видно, что поскольку искомыми являются значения  $\beta$  случайной величины  $T$ , то, задавая значение  $\alpha$ , как равномерно распределенное в диапазоне  $[0,1]$  случайной величины, по формуле (3.2)

$$\beta = \frac{1}{\lambda} \ln \frac{1}{1-\alpha}, \quad (3.2)$$

находим искомое значение. Например, пара операторов:

*eks fvariable* –  $\log((1+rn8)/1000)$  ;экспоненциальное распределение

*generate (2#v\$eks)* ;генерация пуассоновского потока

обеспечит генерацию потока событий с экспоненциальным распределением интервалов между ними, при средней длительности интервалов равной 2-м.

б) Табличный способ

Этот способ описан в пункте 3.3.3. б).

в) С помощью встроенной библиотеки процедур GPSS

3.3.6 Оператор изменения параметров транзактов:

*assign A,B*

где  $A$  – номер параметра транзакта,  $B$  – присваиваемое ему значение.

В следующих операторах параметр  $A$  увеличивается (уменьшается) на значение  $B$ :

*assign A+,B*

*assign A-,B*

### 3.3.7 Вычислительный оператор *variable*

Вычислительный оператор *variable* присваивает переменной с номером *M* значение арифметического выражения *A*:

*M variable A*

Например, следующая пара операторов присвоит 3-му параметру транзакта разность чисел 216 и объема занятой памяти в накопителе *mem2*:

*met5 variable k216-s\$mem2*

*assign 3,v\$met5*

В большинстве операндов можно использовать знаки арифметических операций сложения, вычитания, умножения и деления (+, −, # и /) соответственно.

### 3.3.8 Оператор присвоения

Следующий оператор присваивает хранимой переменной, указанной в *A*, значение, записанное в *B*:

*savevalue A,B*

### 3.3.9 Представление гистограмм

Часто сведения о некоторых величинах, характеризующих моделируемый процесс, удобно представлять в виде гистограмм. Задание гистограммы выполняют в разделе описаний с помощью оператора:

*T1 table A,B,C,D*

Здесь *T1* – имя гистограммы; *A* – табулируемая величина; *B* – граница левого интервала гистограммы; *C* – ширина интервалов; *D* – число интервалов. Формирование гистограммы происходит с помощью оператора:

*tabulate A*

Выполнение этого оператора увеличивает на единицу число попаданий в *i*-й интервал гистограммы величины, имя которой указано в *A*. При этом *i*-й

интервал соответствует текущему значению переменной, являющейся аргументом для гистограммы.

### 3.3.10 Удаление транзактов

С помощью следующего оператора транзакты выводятся из системы и в дальнейшей статистике не учитываются.

*terminate A*

Согласно этому оператору из итогового счетчика вычитается число *A*. В начале исполнения программы в счетчик заносится число, указанное в команде *start*, т.е. задаётся длительность прогона.

Моделирование прекращается, когда содержимое счетчика будет равно или меньше нуля. Операнд *A* в операторе *terminate* в зависимости от места его установки может быть равен нулю, единице и даже большей величине. При этом хотя бы в одном из операторов *terminate* значение *A* должно быть не нулевым, иначе прогон никогда не закончится.

Список основных операторов языка *GPSS* с кратким пояснением их функций даны в Приложениях Б и В.



## 4 Практические занятия по изучению способов моделирования систем и сетей связи

### 4.1 Замечания по применению системы моделирования *GPSS World*

Дисциплина «Моделирования систем и сетей связи» предполагает одновременное изучение как взаимосвязей основных характеристик и параметров связанных систем (влияние факторов на отклики), так и принципов использования конкретной моделирующей системы (в данном случае *GPSS*) для анализа этих взаимосвязей. Однако, объём аудиторных занятий, выделенных для изучения этой дисциплины (18 часов для дневной формы обучения и 12 часов – для заочной), не предполагает детального изучения всех огромных возможностей системы *GPSS*.

В то же время дружелюбный пользовательский интерфейс системы *GPSS World* позволяет работать с ней, владея лишь общей компьютерной грамотностью, и во многих случаях рассматривать приведённый ниже набор программ как некоторый чёрный ящик. В то же время необходимо дать хотя бы минимальные сведения о порядке запуска программ, способах изменения параметров и доступа к результатам моделирования.

#### 4.1.1 Запуск моделирующей программы.

Щёлкнуть на ярлыке *GPSSW* на рабочем столе и далее по кнопке "Открыть" (2-я кнопка слева на шкале инструментов). В появившемся перечне выбрать один из модулей по указанию преподавателя и вывести его на рабочий стол.

Для запуска программы, нажмите последовательно: *Command* → *Create Simulation* → *Command* → *Start*. В появившемся окне *Start Command* установите продолжительность эксперимента в виде числа вызовов (транзактов) вводимых в систему. Учтите, что чем продолжительнее эксперимент, тем точнее результат моделирования (меньше дисперсия). Для целей лабораторной работы можно ограничиться числом транзактов, равным, например,

10000 или 100000. Нажав “ОК”, Вы запускаете моделирующую программу, а после её завершения и доступ к стандартному отчёту.

#### 4.1.2 Доступ к результатам моделирования

Основные результаты моделирования представляются в стандартном отчёте, который автоматически выводится на рабочий стол по окончании прогона. Конкретные значения отчёта рассматриваются последовательно по мере выполнения занятий.

Для доступа к матрицам и гистограммам, при их наличии в данном модуле, необходимо нажать следующие кнопки: *Window*→*Simulation Window* и далее *Matrix Window* или *Table Window*.

#### 4.1.3 Внесение изменений в параметры моделируемой системы

Параметры, подлежащие изменению (интенсивность входного потока, число каналов, канальная скорость и др.), указываются в каждом конкретном задании.

Для внесения изменений в исходный модуль программы нужно вызвать этот модуль, нажав кнопку *Window* и в открывшемся меню вызвать имя программы под номером 1. Необходимо помнить, что ЭВМ выполняет вычисления не с исходным, а с машинным модулем. Поэтому после любых изменений в исходном модуле необходимо снова запускать программу транслятор запуском: *Command* → *Create Simulation* (или *Retranslate*).

Не забывайте после каждого прогона обнулять предыдущую статистику нажатием кнопок *Clear* или *Reset*. При выходе из программы по окончании занятия помните, что все изменения и получаемые результаты сохранению не подлежат.

Для более детального изучения основ компьютерного моделирования можно обратиться к монографии [1], а для овладения навыками моделирования на *GPSS World* следует обратиться к учебному пособию [2].

В отчёте по практическим занятиям необходимо по каждому пункту представлять программы с комментариями и основные результаты прогонов.

Изучить функции использованных операторов и операндов, а также содержание стандартных отчётов.

## 4.2 Практические занятия

### Практическое занятие №1

Объект исследования. Одноканальное устройство с неограниченной очередью.

Одноканальные устройства широко применяются в современных телекоммуникационных системах. Вся сеть *Internet* на 3-м уровне семиуровневой модели представляет собой множество маршрутизаторов, связанных между собой одноканальными линиями связи. Основная работа маршрутизатора состоит в следующем:

- получив *IP*-пакет, маршрутизатор анализирует его адресную часть и определяет направление его дальнейшей передачи;
- если канал в выбранном направлении свободен, то он занимается для передачи этого пакета;
- если канал занят передачей ранее пришедшего пакета, то данный пакет устанавливается в очередь (записывается в буфер ожидания) и будет обслужен по мере освобождения канала.

При анализе работы *IP*-сетей наибольший интерес представляет время задержки пакета в маршрутизаторе (ожидание в очереди + передача по каналу).

В реальных маршрутизаторах объём буфера (длина очереди), конечно, ограничен. Но в данном занятии мы предварительно рассмотрим теоретическую модель с неограниченной очередью, как более простую.

Строго говоря, поступивший в маршрутизатор пакет не сразу поступает на обработку в одно из исходящих из маршрутизатора направлений. Предварительно пакет должен обработаться в устройстве анализа адреса (обычно

специальный микропроцессор) для выбора направления дальнейшей передачи пакета. С точки зрения теории массового обслуживания это типичная двухфазная схема. Однако, пренебрегая временем работы микропроцессора по сравнению со временем передачи пакета по каналу (что, как правило соответствует действительности), рассмотрим процесс передачи пакета через маршрутизатор как работу одноканального устройства.

Листинг 1 – Имитационная модель – модуль 1 – Модель одноканального устройства с неограниченной очередью.

```
; Модель одноканального устройства с неограниченной очередью.
generate      20      ;генерация транзактов с интервалом 20 е.м.в.
queue         fff      ;установить транзакт в очередь с именем fff
seize         kkk      ;занять канал с именем kkk
depart        fff      ;уйти из очереди fff
advance       19      ;задержать транзакт в канале на 19 е.м.в.
release       kkk      ;освободить канал kkk
terminate     1        ;вычесть 1 из длины прогона
```

е.м.в. – единица модельного времени.

**Задание.** Тщательно изучите каждый оператор и приведенные операнды по комментариям и по справочным данным в Приложениях Б и В.

Произвести запуск программы (*command* → *start*) на длину прогона в 10000 транзактов.

Изучить содержимое стандартного отчета (*REPORT*), особенно в части характеристик обслуживающего устройства (канала) и очереди.

Обратите внимание на то, что данная модель статистически устойчива только, если длительность задержки транзакта (операнд *A* оператора *advance*) меньше интервала между моментами появления транзактов (операнд *A* оператора *generate*).

Убедитесь в этом, изменив соотношение операндов на противоположное и проведя два эксперимента с различной длиной прогона. Объясните причину зависимости длины очереди (*AVE.CONT*) от длины прогона.

В исходном варианте отсутствие в операторах *generate* и *advance* операндов *B* исключает случайность в работе устройства, что, как правило, не соответствует реальной ситуации. Исключение составляют, например, систе-

мы наблюдения, в которых различные датчики через равные промежутки времени выдают в центральный пункт сообщения одинаковой длины.

Внесите случайность в работу модели, добавив операнды *B*:

*generate* 20,20 – длительность интервала между моментами появления транзактов будет меняться от 0 до 40 .

*advance* 19,10 – длительность задержки транзакта в канале будет меняться от 9 до 29.

Сравните данные в новом отчёте с предыдущими и объясните произошедшие изменения.

## Практическое занятие №2

Объект исследования. Одноканальное устройство с ограниченной очередью.

Наиболее типичная ситуация для маршрутизаторов в сетях с коммутацией пакетов. Важно отметить, что ограничение очереди в маршрутизаторах выполняется не столько с целью экономии буферной памяти (память, как известно, стремительно дешевеет), сколько из необходимости ограничить время пребывания пакетов в узлах сети. Застаревший сверх допустимого времени пакет, во-первых, утрачивает свою актуальность (возможно, что он уже не нужен получателю), а во-вторых, находясь в очереди, этот пакет задерживает тех, которые приходят в узел после него. Выбор наиболее целесообразного ограничения для длины очереди – сложнейшая оптимизационная задача. Слишком длинные очереди в сети приводят, как уже отмечалось, к большим задержкам пакетов в сети, а слишком короткие – увеличивают потери пакетов, т.к. пакет, поступивший в узел при занятости всех мест ожидания в нужном направлении, покидает систему и считается потерянным.

Ограничить очередь в имитирующей программе можно различными способами. Например, можно добавить в предыдущий модуль оператор *test*:

*test L q\$fff,10,met* ;очередь меньше десяти?

Данный оператор проверяет на «меньшее» текущую длину очереди с именем *fff* (стандартный числовой атрибут *q\$fff*, см. Приложение Г) с числом 10 и, если оно не меньше (вспомогательный операнд *L*), то направляет транзакт к метке *met*.

Таким образом, обслуживание транзакта (пакета) продолжается только в том случае, если в очереди есть свободные места. В противном случае с помощью оператора *savevalue* к переменной *otk* прибавляется единица.

Листинг 2 – Имитационная модель – модуль 2 – Модель одноканальной системы с ограниченной очередью.

```
; Модель одноканальной системы с ограниченной очередью.
generate 20,20 ;генерация транзактов с интервалом 20 е.м.в.
test L Q$fff,10,met ;есть свободные места в очереди?
queue fff ;Да. Установить транзакт в очередь с именем fff
seize kkk ;занять канал с именем kkk
depart fff ;уйти из очереди fff
advance 19,10 ;задержать транзакт на 19 е.м.в. (в среднем)
release kkk ;освободить канал kkk
terminate 1 ;вычесть 1 из длины прогона
met savevalue otk+,1 ;зафиксировать отказ в переменной otk
terminate 1 ;вычесть 1 из длины прогона
```

**Задание.** Запустите программу и проанализируйте отчет.

Снимите зависимости:

- $T_{ож}(\rho)$  для 4-х значений  $\rho$  в пределах от 0,5 до 1,5;
- $P(\rho)$  – вероятность потери транзакта.

Здесь:  $T_{ож}$  – среднее время ожидания в очереди (*AVE.TIME* в *REPORT*), а  $\rho = t_{зад}/t_{инт}$  – загрузка канала, определяемая как отношение времени задержки транзакта в операторе *advance* к интервалу между моментами появления транзактов в операторе *generate*. Вероятность потерь определяется как отношение числа потерянных пакетов (переменная *otk*) к общему числу вошедших в систему пакетов (устанавливается командой *Start*).

В теории телетрафика величина  $\rho$  является главным параметром, определяющим степень загруженности обслуживающей системы. При этом,  $\rho$  определяется как  $\rho = \lambda t$ , где  $\lambda$  – интенсивность входного потока требований, а

$t$  – длительность обслуживания. Значение  $\rho$  может меняться от 0 до 1 и при приближении к 1 такие параметры как длительность ожидания или длина очереди резко возрастают (теоретически до бесконечности).

### Практическое занятие №3

Построение гистограммы распределения времени ожидания в очереди.

Дополнить программу Модуль 2 оператором:

`wtime qtable fff,0,50,40`

поставив его перед оператором *generate*. Операнд  $A$  (*fff*) – имя очереди, операнд  $B$  (0) – начало гистограммы по оси абсцисс, операнд  $C$  (50) – длина интервала в единицах модельного времени, операнд  $D$  (40) – число интервалов. Подобрать операнды  $B$ ,  $C$ , и  $D$  для наглядного представления гистограммы на экране монитора.

Листинг 3 – Имитационная модель – модуль 3 – Построение гистограммы  
; Пример построение гистограммы  
`wtime qtable fff,0,50,40` ;описание формата гистограммы  
`generate 20,20` ;генерация транзактов с интервалом 20 е.м.в.  
`test L Q$fff,10,met` ;есть свободные места в очереди?  
`queue fff` ;Да. Установить транзакт в очередь *fff*  
`seize kkk` ;занять канал с именем *kkk*  
`depart fff` ;уйти из очереди *fff*  
`advance 19,10` ;задержать транзакт в канале на 19 е.м.в.  
`release kkk` ;освободить канал *kkk*  
`terminate 1` ;вычесть 1 из длины прогона  
`met savevalue otk+,1` ;зафиксировать отказ в переменной *otk*  
`terminate 1` ;вычесть 1 из длины прогона

**Задание.** Зарисовать гистограмму в отчёте. Сопоставить ориентировочные данные, представляемые гистограммой, с точными значениями, приведёнными в отчёте в соответствующей таблице.

### Практическое занятие №4

Объект исследования. Многоканальное устройство с ограниченной очередью.

Многоканальные устройства (МКУ) являются наиболее массовой структурой в телефонных сетях с коммутацией каналов. Узлы коммутации и многочисленные АТС связываются между собой стандартными телефонными каналами, число которых может достигать нескольких тысяч.

Принцип работы многоканального устройства с явными потерями состоит в следующем: поступивший вызов пытается занять свободный канал в некоторой соединительной линии для установления телефонного соединения и, если такой свободный канал находится, то он занимает. Если в многоканальном устройстве (в пучке каналов) все каналы заняты, то вызов получает отказ и считается потерянным. По соотношению числа потерянных вызовов ( $n$ ) к общему числу вызовов, поступивших за время наблюдения ( $N$ ), определяется главная характеристика системы с потерями – вероятность потери вызова  $P = n/N$ .

В отличие от предыдущей модели в данную модель внесено изменение, связанное с реализацией многоканальной схемы.

В системе *GPSS* используются разные операторы для занятия и освобождения каналов в одноканальных и многоканальных устройствах, а именно *seize* или *release* для одноканальных систем и *enter* или *leave* – для многоканальных. При этом число каналов в многоканальной системе задаётся отдельным оператором *storage* (см. листинг) с именем МКУ, в котором указывается число каналов (в данном случае определено, что в МКУ с именем *ggg* имеется 10 каналов). Оператор *test l* передаст транзакт на дальнейшее обслуживание только при наличии свободных мест в очереди, т.е. в случае, когда 1-й операнд (число занятых мест в очереди) меньше 2-го (общее число мест). В противном случае, транзакт будет учтён как потерянный в переменной с именем *otk*.

Листинг 4 – Имитационная модель – модуль 4 – Анализ потерь в многоканальной системе

```
;Вероятность потерь в системе с ограниченной очередью
kan storage 10 ;установка числа каналов
generate 8,8 ;генерация интервалов между вызовами
```



test 1	Q\$och, 7, met	; есть свободные места в очереди?
queue	och	; Да. Установить транзакт в очередь
enter	kan	; занять канал
depart	och	; освободить место в очереди
advance	100, 100	; задержать тр. на время обслуживания
leave	kan	; освободить канал
terminate	1	; покинуть систему
met savevalue	otk+, 1	; увеличить число потерь на единицу
terminate	1	; покинуть систему

Задание. Запустите программу и проанализируйте отчет.

Снимите зависимости:

- *Тожд ( $v$ ) для 4-х значений  $\rho$  в пределах от 0,5 до 1,5;*
- *$P(v)$  – вероятность потери транзакта.*

В данном случае  $\rho = t_{\text{зад}}/vt_{\text{инт}}$ , где  $v$  – число каналов, задаваемое оператором *storage*.

## Практическое занятие №5

Объект исследования. Модель разделения потоков транзактов на 3 раз-  
но приоритетных потока.

Телекоммуникационные системы, как правило, рассматривают вызовы, обладающие различными приоритетами. Особенно необходимо дифференцировать качество обслуживания вызовов (*QoS*) в мультисервисных сетях. Известно, что потоковые виды трафика (речь, видео) требуют первоочередного обслуживания, так как допустимое время задержки в сети для них находится в пределах 150 – 250 мс, а файловый обмен данными не критичен к задержкам, и их значения могут достигать до нескольких секунд.

Реальные маршрутизаторы анализируют требования по приоритетности передачи (классы обслуживания – *COS*), находящиеся в заголовке пакета, и в соответствии с этими требованиями организуют несколько очередей для выдачи пакетов в каждый канал. Наиболее часто реализуется, так называемая процедура относительных приоритетов, в которой при каждом освобождении канала выбирается очередной пакет из наиболее приоритетной очереди.

Имитационная модель. При генерации транзакта оператором *generate* ему присваивается самый низший приоритет «0», если иное не указано в операнде 5 оператора *generate*. Этот приоритет может быть, при необходимости изменён в любом месте программы оператором *priority* с указанием нового приоритета (например, *priority 17*) или в более сложных случаях с помощью функции, как в приведённом ниже примере.

#### Листинг 5 – Имитационная модель – модуль 5 – Распределение приоритетов

```
;Распределение транзактов по приоритетам
sss function rn1,d3 ;аргумент – случайное число (rn1)
.5,1/.8,15/1,7 ;функция – числа 1, 15 или 7
generate 30,20 ;генерация интервалов между вызовами
priority fn$sss ;присвоение приоритетов по функции sss
advance 1000
terminate 1
```

Операторы *priority* и *function* с именем *sss* разделят общий поток транзактов случайным образом (датчик случайных чисел *rn1*) на 3 потока. При этом 50% всех транзактов получит приоритет №1, 30% – №15 и 20% – №7. Аргументом функции в данном случае является случайное число, выбираемое в интервале 0 – 1, а функцией – номер нового приоритета для транзакта. Оператор *advance* с чрезмерно большой задержкой (операнд 1000) позволяет сохранить в отчёте все сгенерированные транзакты и наблюдать присвоенные им приоритеты.

Задание. Объясните структуру оператора *priority*. Измените операнды оператора *function* для деления общего потока транзактов на 2, а потом на 4 приоритета.

#### Практическое занятие №6

Объект исследования. Модель системы с 3-я разно приоритетными очередями транзактов.

Исследуется маршрутизатор IP-пакетов в сети с коммутацией пакетов. Анализируется процедура выдачи пакетов в канал связи с заданной пропуск-

ной способностью. Разно приоритетные потоки для выдачи в этот канал поступают из нескольких входящих каналов с суммарной интенсивностью  $\Lambda = \sum \lambda_i$ , где  $\lambda_i$  – интенсивность потока  $i$ -го приоритета ( $i = 0 \div 127$ ).

Соотношение между потоками определяется исследователем в зависимости от конкретной ситуации. Основные исследуемые характеристики – длительности задержек пакетов в узле в зависимости от канальной скорости, интенсивности суммарного потока и соотношением между интенсивностями потоков различных приоритетов.

#### Листинг 6 – Имитационная модель – модуль 6 – Исследование очереди с приоритетами

```
; Модель очередей с приоритетами

sss    function rn1,d3 ;функция случ. распределения приоритетов
.5,1/.8,2/1,5          с вер. 0.5, 0.3 и 0.2
och    function pr,d3  ;функция распределения длин очередей по
1,30/2,15/5,7          приоритетам
        generate 20,20 ;генерация транз. с интервалом 20 е.м.в.
        priority fn$sss ;выбор номера приоритета
        assign   ocher,pr ;присвоение очереди номера приоритета
        test L   Q*ocher,fn$och,met ;есть свобод. места в очереди?
        queue    pr      ;Да. Установить транзакт в очередь
                        с именем 1,2 или 5
        seize    kkk      ;занять канал с именем kkk
        depart   pr       ;уйти из очереди
        advance  19,10    ;задержать транзакт в канале на 19 е.м.в.
        release  kkk      ;освободить канал kkk
        terminate 1       ;вычесть 1 из длины прогона
met     savevalue otk+,1 ;зафиксировать отказ в переменной otk
        terminate 1
```

Особенностью данной программы является применение в операторах *queue* и *depart* в качестве имени очереди номера приоритета (*pr*). Это дает возможность применять одни и те же операторы для транзактов любого приоритета. В данном случае именами очереди будут цифры 1, 2 или 5.

Для этого применяется оператор *assign*, который может вошедшему в него транзакту определить имя параметра и присвоить этому параметру значение. В данном случае, параметру с именем *ocher* – присваивается числовое

значение по номеру приоритета (1, 2 или 5). В дальнейшем этот параметр будет определять имя очереди в операторе *test L*.

Оператор *test L* сравнивает на “меньше” текущую длину очереди под номером приоритета ( $Q*ocher$ ), хранящегося в параметре транзакта *ocher*, с числом мест в приоритетных очередях, определяемых функцией с именем *och*. В исходном варианте данного модуля длины допустимых очередей по приоритетам равны соответственно 30-й, 15-й и 7-й.

Задание. Выполните прогон Модуля 6 и проанализируйте отчёт в части очередей с именами 1,2 и 5. Измените длины очередей в функции с именем *och*, произведите прогон и объясните изменения в отчёте.

Измените операнды оператора *generate* так, чтобы суммарный поток был равен 400 пакетов/с. Меняя с помощью оператора *advance* пропускную способность канала от 200 до 600 пакетов/с, снимите зависимости длительностей задержек пакетов  $T_{ожі}$  и вероятностей потерь пакетов  $P$  по каждому приоритету в зависимости от пропускной способности канала.

Измените программу Модуль 6 так, чтобы она имитировала многоканальную систему. Снимите зависимости средней длительности ожидания  $T_{ожі}$  (значение *AVE.TIME* в *REPORT*) и средней длины очереди  $D_{очі}$  (значение *AVE.CONT* в *REPORT*) по каждой категории от числа каналов. Если значения откликов  $T_{ожі}$  и  $D_{очі}$  исчезающе малы, измените параметры системы так, чтобы исследуемые зависимости оказались наиболее наглядными.

## Практическое занятие №7

Объект исследования. Пуассоновское распределение потоков различных событий.

Один из наиболее часто используемых в теории телетрафика простейший поток вызовов описывается пуассоновским распределением (4.1):

$$P_i(t) = \frac{(\lambda t)^i}{i!} e^{-\lambda t}. \quad (4.1)$$

где:  $\lambda$  – интенсивность входящего потока вызовов [выз/ед.вр.],

$t$  – интервал времени [ед. вр],

$P_i(t)$  – вероятность появления ровно  $i$  вызовов в интервале  $t$ .

Пуассоновский поток вызовов непосредственно связан с экспоненциальным распределением тем, что интервалы ( $z$ ) между вызовами пуассоновского потока распределены по экспоненте:  $P(z < t) = 1 - e^{-\lambda t}$ . А так как в операторе генерации транзактов *generate* устанавливаются именно интервалы между вызовами, то в модуле 7 реализован экспоненциальный генератор со средним интервалом равным 2 (это означает, что интенсивность вызовов составляет 0,5 выз/ед. вр.).

Программа работает следующим образом. Оператор «*generate 100*» нарезает временной процесс на равные интервалы времени в 100 е.м.в. (единицы модельного времени). В это время оператор «*generate (2#v\$eks)*» генерирует случайное число транзактов, которое фиксируется в счётчике *sch*. Содержимое этого счётчика в начале каждого интервала в 100 е.м.в. табулируется и счётчик обнуляется.

Листинг 7 – Имитационная модель – модуль 7 – Построение пуассоновского распределения

```
;Построение пуассоновского распределения
tab table      x$sch,10,2,40          ;построение гистограммы
eks fvariable  - log((1+rn8)/1000)    ;экспоненциальное распред.
generate      (2#v$eks)               ;генерация пуасс. потока
savevalue     sch+,1                  ;счётчик событий в интервале t
terminate
generate      100                     ;генерация интервалов t
tabulate      tab                     ;фиксация числа событий в интервале t
savevalue     sch,0                   ;обнуление счётчика событий в интервале t
terminate     1                       ;общее число интервалов t
```

**Задание.** Снять гистограмму пуассоновского распределения. Рассчитать теоретическое распределение по формуле Пуассона. Проверить гипотезу о законе распределения исследуемой случайной величины по критерию согласия  $\chi^2$  (см. приложение А).

## Практическое занятие №8

**Объект исследования.** Пуассоновское распределение входящего потока и экспоненциальное распределения длительности обслуживания.

Рассмотренные в предыдущих занятиях объекты функционировали при интервалах между вызовами и длительностях обслуживания распределёнными по закону равномерной плотности. Для этого в операторах *generate* и *advance* в качестве операнда *B* использовалось постоянное число. Однако в реальных телекоммуникационных системах гораздо чаще встречаются интервалы, распределённые по экспоненциальному закону. Это относится как к пуассоновскому входящему потоку, в котором интервалы между вызовами распределены по экспоненте, так и к длительности обслуживания.

Как правило, вызова подчиняются пуассоновскому закону, если они исходят от большого количества источников (теоретически бесконечного), ни один из которых не является превалирующим. Экспоненциальное распределение длительности занятия каналов характерно для телефонных разговоров. Применительно к передаче данных время передачи пакета по каналу прямо пропорционально его длине, т.е. количеству байт (бит) в нём. И если длину пакета аппроксимировать экспоненциальным распределением (что нередко соответствует действительности), то можно и длительность занятия канала передачей одного пакета считать распределённой по экспоненте.

Реализация экспоненциального распределения для длительностей обслуживания и интервалов между вызовами осуществляется, как показано в разделе 2.3.3, с помощью логарифмической функции  $t = -\frac{1}{\lambda} \ln \xi$ . В листинге программы эту функцию выполняет оператор *fvariable* (4.2):

$$fvariable - \log ((1+rn8)/1000). \quad (4.2)$$

При этом среднее значение интервалов между транзактами устанавливается непосредственно в операторе *generate* (средний интервал между вызовами в данном случае установлен равным 2 е.м.в.), а средняя длительность

задержки в операторе *advance* (транзакты задерживаются в среднем на 3 е.м.в.).

### Листинг 8 – Имитационная модель – модуль 8 – Реализация пуассоновского и экспоненциального распределений

```
;реализация пуассоновского входного потока и экспоненциального
;распределения длительности обслуживания
tab    table        m1,0,0.2,100          ;гистограмма задержек
eks    fvariable    -log((1+rn8)/1000)    ;экспоненц. распределение
        generate    (2#v$eks)             ;генерация потока
        advance     (3#v$eks)             ;задержка транзакта
        tabulate    tab                   ;табуляция задержек
        terminate   1
```

В операторе *fvariable* датчик случайных чисел *rn8* при каждом обращении к нему выбирает случайное целое число в диапазоне от 0 до 999. Делением на 1000 эти числа переносятся в диапазон от 0 до 0.999. Чтобы избежать ситуации «log 0» к значению *rn8* прибавляется 1 и окончательно диапазон чисел для логарифмирования устанавливается от 0.001 до 1.

Для построения гистограммы задержек в операторе *table* присутствует операнд *m1*, являющийся СЧА, определяющим время жизни транзакта. Операнды *B*, *C* и *D* определяют формат гистограммы по оси абсцисс.

**Задание.** Зафиксировать 3 – 4 гистограммы задержек, при различных значениях входящего потока.

### Практическое занятие №9

**Объект исследования.** Многоканальное устройство с явными потерями.

В данном занятии проводится анализ многоканального устройства средствами двух методов математического моделирования: аналитического и имитационного. Методы аналитического моделирования многоканальных устройств описаны в многочисленных работах по теории телетрафика. В большинстве случаев эти работы рассматривают системы с пуассоновским входящим потоком вызовов и с экспоненциальным распределением длитель-

ности обслуживания. В связи с этим для сопоставительного анализа многоканальной системы двумя методами необходимо использовать модель системы с экспоненциальным распределением как для интервалов между вызовами, так и для длительностей задержек.

#### Аналитическая модель

Наиболее реальную ситуацию при ограниченном числе каналов ( $v < \infty$ ) отражает распределение Эрланга или усечённое распределение Пуассона (4.3):

$$P_i = \frac{\frac{Y^i}{i!}}{\sum_{j=0}^v \frac{Y^j}{j!}} \quad i = 0, v. \quad (4.3)$$

где  $P_i$  – вероятность того, что в случайный момент времени в  $v$ -канальном устройстве занято  $i$  каналов,

$Y = \lambda t$  – нагрузка на  $v$ -канальную систему, зависящая от интенсивности потока вызовов  $\lambda$  и средней длительности занятия каналов  $t$ .

Наиболее важным в распределении Эрланга является состояние  $i = v$ , т.е. состояние, когда заняты все каналы. Разумеется, что это то состояние, когда наступают потери (отказы в обслуживании). Вероятность этого состояния определяется формулой Эрланга (4.4):

$$P_i = P_v = P_t = P_e = P_n = E_v(Y). \quad (4.4)$$

Здесь значения  $P_b$ ,  $P_e$  и  $P_n$  являются вероятностями потерь по времени, по вызовам и по нагрузке соответственно.

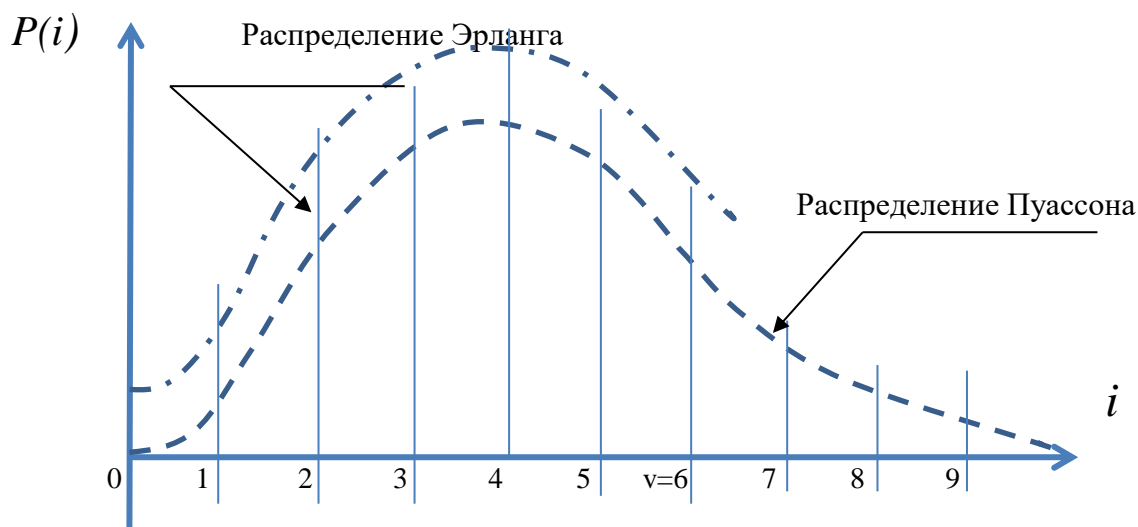




Рисунок 4.1. Огибающие распределений вероятностей чисел занятых каналов для простейшего входящего потока.

Последнее обозначение  $E_v(Y)$  является общепринятым обозначением формулы Эрланга. Наиболее частая практическая задача, определения необходимого число линий в пучке для обслуживания поступающей нагрузки, при заданной вероятности потерь данной формулой в явном виде не решается. Потому что невозможно выразить число каналов  $v$ , как функцию от  $Y$  и  $P$ . В связи с этими трудностями, формула Эрланга табулирована в различных справочниках и задачниках. Кроме того в последнее время появилось несколько удобных программ, позволяющих очень легко из трёх величин – нагрузка ( $Y$ ), число каналов ( $v$ ) и вероятность потерь  $P$  – найти любую при двух других известных.

**Имитационная модель.** В программном модуле, реализованы экспоненциальные распределения для интервалов между вызовами и длительностей задержек. Кроме того, реализован модуль, обеспечивающий построение гистограммы для сопоставления результатов имитационного моделирования с аналитическими результатами по формуле Эрланга.

Листинг 9 – Имитационная модель – модуль 9 – многоканальная система с потерями

```
;анализ многоканальной системы с потерями
raspr table      s$ggg,0,1,12 ;гистограмма Pi
ggg storage      10           ;установка числа каналов
eks fvariable    -log((1+rn8)/1000) ;экспоненц. распределение
generate (2#v$eks) ;генерация вызовов
gate snf ggg,met ;многокан. устройство не заполнено?
enter ggg ;Да. Занять канал
advance (18#v$eks) ;задержать транзакт
leave ggg ;освободить канал
terminate ;вывести транзакт из системы
met savevalue ot+,1 ;зафиксировать очередной отказ
terminate ;вывести транзакт из системы

generate 50
tabulate raspr
terminate 1
```

Задание. Открыть Модуль 9 и запустить модель.

1 Построить распределение Эрланга по формуле (4.1), при нагрузке  $Y = 9$  Эрл и числе каналов  $\nu = 10$ . Нагрузка находится как  $Y = \lambda t$ , где  $\lambda$  определяется по операнду  $A$  оператора *generate* ( $\lambda = 1/t_{\text{интерв}}$ ), а  $t$  – по операнду  $A$  оператора *advance* ( $t$  – длительность задержки).

2 Снять гистограмму для распределения числа занятых каналов при той же нагрузке  $Y = 9$  Эрл. для пуассоновского входного потока и экспоненциального обслуживания. В исходном варианте (модуль 9) установлено  $Y = 0.5 \cdot 18 = 9$  Эрл.

3 Снять гистограмму для распределения числа занятых каналов при той же нагрузке  $Y = 9$  Эрл. при равномерно распределённом входном потоке и равномерно распределённой длительности обслуживания. Установить для этого *generate* 2,2 и *advance* 18,18.

4 Сравнить полученные экспериментальные зависимости (п.2 и п.3) с теоретическим распределением Эрланга по критерию согласия  $\chi^2$ . (См. приложение А).

## Практическое занятие №10

Объект исследования. Одноканальная система с бесконечной очередью, с пуассоновским входным потоком и экспоненциальным обслуживанием.

Такая, чисто теоретическая система является наиболее исследованной в теории массового обслуживания и поэтому в наибольшей степени подходит для сопоставительного изучения методов аналитического и имитационного моделирования.

### Аналитическая модель

Основной характеристикой, наиболее часто используемой для практических расчётов, является вероятность ожидания для поступившего вызова  $P(\gamma > 0)$ , где  $\gamma$  – время ожидания. По существу, это вероятность того, что вызов

до начала обслуживания будет ожидать освобождения канала. Для простейшего потока вызовов эта вероятность совпадает с вероятностью занятости всех каналов или с вероятностью потерь по времени (4.5):

$$P(\gamma > 0) = P_{\geq v} = P_t = \sum_{i=v}^{\infty} P_i = \frac{E_v(Y)}{1 - \left(\frac{Y}{v}\right)[1 - E_v(Y)]} = D_v(Y). \quad (4.5)$$

В данном случае суммирование проводится по всем состояниям ( $i$ ), при которых заняты все каналы и имеется очередь любой длины: от нуля до бесконечности. Расчётное соотношение для вероятности ожидания полученное Эрлангом, называется второй формулой Эрланга, обозначается как  $D_v(Y)$  и табулировано для практического применения. Таблицы, как и в случае первой формулы –  $E_v(Y)$ , позволяют по любым двум из трех параметров –  $Y$ ,  $v$ ,  $P$  – определить третий. Здесь  $D_v(Y)$  является общепринятым обозначением 2-й формулы Эрланга, которая табулирована во многих источниках.

Очень важно отметить, что одноканальная система с неограниченной очередью может быть статистически устойчива только при  $Y < 1$ . В противном случае очередь к каналу будет расти до бесконечности, в чём можно убедиться, сопоставив результаты прогонов разной длительности (например, в 10000 и 50000 транзактов).

Выше приведённые соотношения сделаны для произвольного числа каналов  $v$ . В данном занятии  $v = 1$ , а в следующих занятиях будут рассмотрены системы с  $v > 1$ .

Листинг 10 – Имитационная модель – модуль 10 – одноканальная система с бесконечной очередью

```
*анализ одноканальной системы с бесконечной очередью
eks  fvariable  -log((1+rn8)/1000) ;экспоненц. распределение
      generate  (20#v$eks)         ;генерация транз.интерв. 20 е.м.в.
      queue    fff                 ;установить транз. в очередь fff
      seize    kkk                 ;занять канал с именем kkk
      depart   fff                 ;уйти из очереди fff
      advance  (19#v$eks)         ;задержать транзакт на 19 е.м.в.
      release  kkk                 ;освободить канал kkk
      terminate 1                  ;вычесть 1 из длины прогона
```

Данная имитационная модель почти полностью аналогична модели занятия №1. Исключение составляют операнды в операторах *generate* и *advance*, в которых равномерные распределения интервалов заменены на экспоненциальные.

Задание. Открыть Модуль 10 и запустить модель.

1 Построить по таблицам  $D_v(Y)$  график зависимости вероятности ожидания  $P(\gamma > 0)$  от нагрузки  $Y$  при изменении аргумента в пределах  $0,2 < Y < 0,9$ .

2 Построить на том же графике эмпирическую зависимость вероятности ожидания (параметр канала *UTIL*) от тех же аргументов. Напомним, что нагрузка подсчитывается как  $Y = t_3 / t_{\text{и}}$ , где  $t_3$  – время задержки транзакта в операторе *advance*, а  $t_{\text{и}}$  – длительность интервалов между вызовами в операторе *generate*.

3 Построить 3 графика по п. 1 и 2, меняя длительность прогона (10000, 100000 и 1000000). Объяснить имеющиеся различия.

## Практическое занятие №11

Объект исследования. Многоканальная система с бесконечной очередью с пуассоновским входным потоком и экспоненциальным обслуживанием. В данном занятии исследуются различные характеристики многоканальных устройств с бесконечной очередью.

### 11.1 Средняя длина очереди

Аналитически средняя длина очереди определяется по формуле (4.6):

$$\bar{S} = \sum_{i=v}^{\infty} (i - v) P_i = Y D_v(Y) / (v - Y) = \lambda P(\gamma > 0) / (v - Y). \quad (4.6)$$

Здесь  $v$  – число каналов в системе, а остальные параметры имеют тот же смысл, что и в занятии №10. Отметим, что в многоканальных системах условием статистической устойчивости является неравенство  $Y < v$ , т.е. только в этом случае очередь не будет расти до бесконечности.

Экспериментально средняя длина очереди ( $\bar{S}$ ) по результатам прогона фиксируется непосредственно в стандартном отчёте в параметре *AVE.CONT*.

Листинг 11 – Имитационная модель – модуль 11 – анализ средней длины очереди

```
;Определение средней длины очереди
ggg storage 5 ;число каналов в пучке
eks fvariable - log((1+rn8)/1000) ;экспон. распределение
generate (2#v$eks) ;генерация пуассоновского потока
queue fff ;установить транзакт в очередь fff
enter ggg ;занять канал
depart fff ;уйти из очереди fff
advance (3#v$eks) ;обслуживание с экспон. распределением
leave ggg ;освободить канал ggg
terminate 1
```

**Задание.** Исследовать зависимость расхождений результатов аналитической и имитационной моделей от величины  $P(\gamma > 0)$ . Параметры моделей ( $\lambda$ ,  $v$ ,  $n$ ) установить идентичными. Результаты представить графически.

### 11.2 Вероятность превышения длиной очереди заданной величины $n$

Аналитически эта вероятность определяется с использованием второй формулы Эрланга (4.7):

$$P(S > n) = \sum_{i=v+n+1}^{\infty} P_i = (Y/v)^{n+1} D_v(Y) = (Y/v)^{n+1} P(\gamma > 0). \quad (4.7)$$

Для получения аналогичной вероятности на имитационной модели необходима достаточно сложная программа. Поэтому ниже приводится программа, позволяющая получить несколько приближённую оценку. Причём разница в результатах аналитической и имитационной моделей будет тем меньше, чем меньше эти вероятности.

## Листинг 11a – Имитационная модель – модуль 11a

```

;Вер. превышения установленной длины очереди (приблизительно)
ggg storage 5 ;число каналов в пучке
eks fvariable -log((1+rn8)/1000) ;экспон. распределение
generate (2#v$eks);генерация пуассоновского потока
test L Q$fff,10,met ;есть свободные места в очереди?
queue fff ;Да.Установить транзакт в очередь fff
enter ggg ;Занять канал
depart fff ;уйти из очереди fff
advance (8#v$eks);обслуживание с экспон. распределением
leave ggg ;освободить канал
terminate 1
met savevalue otk+,1 ;зафиксировать отказ в переменной otk
terminate 1

```

Приближённый характер данной модели заключается в том, что реализуется очередь ограниченной длины, равной параметру  $n$ , и вместо вероятности превышения длиной очереди заданной величины  $n$ , т.е.  $P(\gamma > 0)$ , имитационная модель определяет вероятность потери транзакта из-за ограниченной очереди.

Задание. Исследовать зависимость расхождений результатов аналитической и имитационной моделей от величины  $P(\gamma > 0)$ . Параметры моделей  $(\lambda, \nu, n)$  установить идентичными. Результаты представить графически.

## 11.3 Вероятность ожидания свыше допустимого времени.

Это так называемые условные потери.

Аналитически эта величина получается достаточно просто (4.8):

$$P(\gamma > t_d) = \sum_{i=\nu}^{\infty} P_i(\gamma > t_d) P_i = D_{\nu}(Y) e^{-(\nu-Y)t_d} = P(\gamma > 0) e^{-(\nu-Y)t_d} \quad (4.8)$$

Здесь величина  $t_d$  выражена в условных единицах времени (у.е.в.). Условные единицы времени широко применяются в теории телетрафика, т.к. позволяют абстрагироваться от конкретных астрономических единиц (час, мин, мкс) и получать математические выражения в унифицированном виде. Определяется как  $1 \text{ у.е.в.} = t_{\text{ср.обсл.}}$  т.е. через среднюю длительность обслуживания.

На имитационной модели вероятность  $P(\gamma > t_d)$  определяется по гистограмме в стандартном отчёте. Точнее по таблице в отчёте, в которой по вер-

тикали откладывается время, а в последнем столбце – накопленные проценты.

### Листинг 116 – Имитационная модель – модуль 116

```
;Вероятность ожидания свыше допустимого времени t
ggg      storage      5                ;число каналов в пучке
wtime    qtable       fff,0,0.5,40    ;распределение длит. ожидания
eks      fvariable    - log((1+rn8)/1000) ;экспон. распределение
generate (2#v$eks)      ;генерация пуассоновского потока
queue     fff          ;установить транз. в очередь fff
enter     ggg          ;Занять канал
depart    fff          ;уйти из очереди fff
advance   (3#v$eks)     ;обслуживание с экспон. распредел.
leave     ggg          ;освободить канал
terminate 1
```

**Задание.** Снять зависимость вероятности  $P(\gamma > t_d)$  от допустимого времени  $t_d$ . Пределы изменения  $t_d$  подобрать так, чтобы  $P(\gamma > t_d)$  изменялось примерно от 0,001 до 0,1. При этом соответствующим образом нужно подбирать операнды в операторе *qtable*.

11.4 Средняя длительность ожидания рассматривается в двух вариантах:

- $\bar{\gamma}_3$  – средняя длительность ожидания для задержанных вызовов, т.е. подсчитанная только среди тех, которые поступили на обслуживание после некоторого ожидания;
- $\bar{\gamma}$  – средняя длительность ожидания для всех вызовов, т.е. для любого поступившего независимо от процедуры ожидания.

Аналитически оба параметра подсчитываются по формулам (4.9):

$$\bar{\gamma}_3 = 1/(\nu - Y), \quad \bar{\gamma} = \bar{\gamma}_3 D_\nu(Y) = D_\nu(Y) / (\nu - Y) = P(\gamma > 0) / (\nu - Y). \quad (4.9)$$

Здесь величины  $\bar{\gamma}_3$  и  $\bar{\gamma}$  выражены в условных единицах времени (у.е.в.).

Эмпирически эти параметры выводятся непосредственно в стандартном отчёте в строке *queue*:

$$\bar{\gamma}_3 - AVE.(-0), \quad \bar{\gamma} - AVE.TIME$$

Задание. Построить на одном графике зависимости средних длительностей ожидания от степени загрузки канала:  $\bar{\gamma}_3(\rho)$  и  $\bar{\gamma}(\rho)$  для аналитических и эмпирических наблюдений. Объяснить результаты наблюдений.

## Практическое занятие №12

### Модель двухфазной системы с потерями.

Реальные телекоммуникационные системы никогда не состоят из одного обслуживающего устройства. Например, в ЛВС созданный в компьютере пользователя пакет, прежде чем попасть в канал доступа к *Internet*, в зависимости от конфигурации ЛВС пройдет через несколько обслуживающих устройств (фаз обслуживания): каналы, коммутаторы, маршрутизатор. Каждое из этих устройств, строго говоря, тоже нельзя рассматривать как отдельное обслуживающее устройство. Например, в коммутаторе мы можем отдельно рассматривать работу входного порта (контроллера), центрального процессора, различных модулей ПО, общей шины, выходного порта. Причём такую детализацию можно продолжить. Например, в работе центрального процессора можно отдельно рассматривать работу АЛУ, управляющего устройства, памяти, интерфейсов.

В данном занятии исследуется система с двухфазным обслуживанием. Каждый транзакт должен обслуживаться вначале каналом с именем *rrr*, а затем каналом с именем *kkk*. Если к моменту поступления транзакта к любому из каналов данный канал занят (проверка оператором *gate nu*), то транзакт фиксируется как потерянный в переменных *ot1* или *ot2* в зависимости от номера фазы.

### Листинг 12 – Имитационная модель – модуль 12

```
;модель двухфазной системы обслуживания с потерями
;с одним входным потоком
generate 20,20 ;генерация транз. с интерв. 20 е.м.в.
```



```

gate nu      rrr,otk1 ;устройство rrr свободно?
seize        rrr      ;Да. Занять канал с именем rrr
advance      29,20     ;задержать транзакт в а 29 е.м.в.
release      rrr      ;освободить канал с именем rrr
vtor gate nu  kkk,otk2 ;2-я фаза. Устройство kkk свободно?
seize        kkk      ;Да. Занять канал с именем kkk
advance      19,19     ;задержать транзакт на 19 е.м.в.
release      kkk      ;освободить канал kkk
terminate    1         ;вычесть 1 из длины прогона
otk1 savevalue ot1+,1   ;Потеря транзакта в 1-й фазе
terminate    1         ;вычесть 1 из длины прогона
otk2 savevalue ot2+,1   ;Потеря транзакта во 2-й фазе
terminate    1         ;вычесть 1 из длины прогона

```

Задание. Откройте программу Модуль 12 и запустите её. Определите степень загрузки каждого канала и долю потерянных транзактов в каждой фазе. Определите суммарную вероятность потери транзактов в двух фазах.

### Практическое занятие №13

Добавление 2-го потока, входящего сразу во 2-ю фазу (во второй канал).

Случаи, когда транзакты (пакеты) могут поступать не только в начальную фазу, но и в промежуточные очень многочисленны. Например, на рисунке 4.2 представлена схема доступа в *Internet* с двумя входными потоками  $\lambda_1$  и  $\lambda_2$ . УК1 и УК2 – узлы доступа абонентов, а узел УК3 – узел выхода в *Internet*. Канал 1 связывает узлы УК1 и УК2 (1-я фаза обслуживания), а канал 2 – УК2 и УК3 (2-я фаза обслуживания).

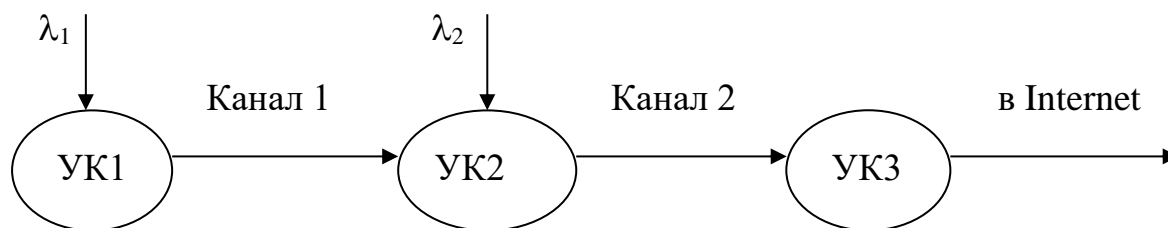


Рисунок 4.2 – Вариант доступа в сеть *Internet*

### Листинг 13 – Имитационная модель – модуль 13

```

;модель двухфазной системы обслуживания с потерями
;с двумя входными потоками
generate     25,20      ;генерация второго потока

```

	transfer	, vtor	; безусловный переход
	generate	20, 20	; генерация транзактов (пакетов)
	gate nu	rrr, otk1	; канал rrr свободен?
	seize	rrr	; занять канал с именем rrr
	advance	29, 20	; задержать транзакт в канале
	release	rrr	; освободить канал rrr
vtor	gate nu	kkk, otk2	; канал kkk свободен?
	seize	kkk	; занять канал с именем kkk
	advance	19, 15	; задержать транзакт в канале
	release	kkk	; освободить канал kkk
	terminate	1	; вычесть 1 из длины прогона
otk1	savevalue	ot1+, 1	; потеря транзакта в 1-й фазе
	terminate	1	; вычесть 1 из длины прогона
otk2	savevalue	ot2+, 1	; потеря транзакта во 2-й фазе
	terminate	1	; вычесть 1 из длины прогона

Задание. Выполните прогон модуля 13 и в отчёте (*Report*) в самом начале (массив с названием *LABEL*) по данным колонки *ENTRY COUNT* проследите за движениями транзактов по блокам программы. Установите и объясните местонахождение всех транзактов, выданных двумя операторами *generate*. Определите в какой фазе обслуживания вероятность потери транзактов выше и измените входные параметры модели (факторы) так, чтобы соотношение откликов изменилось на обратное.

## Практическое занятие №14

Изучение основных элементов стандартного отчёта и стандартных числовых атрибутов.

Ознакомиться на конкретных примерах с основными элементами стандартного отчета и способами использования стандартных числовых атрибутов (СЧА).

Рассмотреть модель многоканальной системы с простейшим потоком вызовов, экспоненциальным обслуживанием и ограниченной очередью.

Листинг 14 – Имитационная модель – модуль 14. Изучение стандартных числовых атрибутов.

```
;модель для изучения стандартного отчёта и СЧА
vspot    fvariable    N$met/N$fin    ;подсчёт вероятности потерь
kan      storage      10              ;установка числа каналов
eks      fvariable    -log{(1+rn1)/1000} ;вычисление экспоненты
          generate     (10#v$eks)      ;генерация простейшего потока
          test 1       Q*och,7,met     ;очередь меньше 7-и?
          queue        och             ;Да.Поставить транз. в очередь
          enter        kan             ;занять канал
          depart       och             ;покинуть очередь
          advance       (11#v$eks)      ;задержать транзакт
          leave        kan             ;освободить канал
          transfer      ,fin           ;безусловный переход
met       avevalue     otk+1           ;увеличить на 1 число потерь
fin       test e       tgl,1,final     ;данный транзакт последний?
          savevalue     verp,v$vpot    ;да. Вычислить вер. потерь
final     terminate    1              ;вывести транзакт из системы
```

Выполнить запуск модели и выписать в тетрадь содержимое стандартного отчёта аналогично приведённым ниже фрагментам.

Фрагмент отчёта №1

Объясните зафиксированное в отчёте число транзактов, поступивших в каждый блок (*ENTRY COUNT*), и число транзактов, присутствующих в некоторых блоках на момент окончания прогона (*CURRENT*).

label	loc	block type	entry count	current
	1	generate	100010	0
	2	test	100010	0
	3	queue	85887	5
	4	enter	85882	1
	5	depart	85881	0
	6	advance	85881	4
	7	leave	85877	0
	8	transfer	85877	0
met	9	savevalue	14123	0
fin	10	test	100000	0
	11	savevalu	1	0
final	12	terminate	100000	0

Фрагмент отчёта №2

Содержания отчёта в части очереди с именем *och*:

```
queue  max  cont.  entry  entry(0)  ave.cont.  ave.time
```

<i>och</i>	7	6	85887	16103	3.357	7.821
------------	---	---	-------	-------	-------	-------

- *max* – 7 – максимальное значение длины очереди за время прогона;
- *cont* – 6 – текущее значение длины очереди в момент окончания прогона;
- *entry* – 85887 – число транзактов, вошедших в очередь;
- *entry(0)* – 16103 – число транзактов, вошедших в нулевую очередь, т.е. тех транзактов из общего числа, которые вошли в очередь и тут же перешли в канал без ожидания;
- *ave.cont* – 3.357 – средняя длина очереди;
- *ave.time* – 7.821 – среднее время ожидания транзакта в очереди.

### Фрагмент отчёта №3

Содержания отчёта в части многоканального устройства с именем *kan*:

<i>storage</i>	<i>cap.</i>	<i>min.</i>	<i>max.</i>	<i>entries</i>	<i>avl.</i>	<i>ave.cont.</i>	<i>util.</i>
<i>kan</i>	5	0	5	85882	1	4.696	0.939

- *cap.* – 5 – число каналов в многоканальном устройстве;
- *min.* – 0 – минимальное число занятых каналов;
- *max.* – 5 – максимальное число занятых каналов;
- *entries* – 85882 – число транзактов, вошедших в многоканальное устройство;
- *avl.* – состояние памяти к моменту завершения процесса моделирования 1 – доступно, 0 – не доступно.
- *ave.cont* – 4,696 – среднее число занятых каналов;
- *util* – 0,939 – степень загрузки канала или доля времени, в течение которого канал занят.

В указанном программном модуле использованы следующие стандартные числовые атрибуты:

- СЧА –  $q*och$  – означает число транзактов, стоящих в данный момент в очереди с именем *och*.
- СЧА –  $N\$met$  – счетчик входов в оператор с именем *met*. Переменная *vpot* подсчитывает вероятность потерь, как отношение числа транзактов, прошедших через оператор с меткой *met* (СЧА –  $N\$met$ ) к общему числу транзактов (СЧА –  $N\$fin$ ).
- СЧА – *tg1* – содержимое счётчика завершения. В данной программе подсчёт переменной *vpot* производится не при всех транзактах, а в целях экономии времени, только при проходе последнего транзакта, т.е. только тогда, когда *tg1*=1.
- СЧА –  $rn1...rn8$  Генераторы случайных чисел вырабатывают случайные числа в интервале 0.....0,999999, используемые как аргументы функций, например, для определения номера узла отправителя. Во всех остальных случаях  $rn1...rn8$  – случайные числа в интервале 0...999. В данной программе *rn1* используется для определения переменной с плавающей запятой, распределенной по экспоненциальному закону. Такое распределение дает функция –  $\log(rn1/1000)$ . В программе используется функция –  $\log\{(1+rn1)/1000\}$  во избежание возникновения случая с вычислением несуществующего логарифма нуля.
- СЧА – *m1* – время жизни транзакта. Удобно использовать для формирования таблицы распределения времени обслуживания.

Полный список СЧА приведён в Приложении Г.

## Практическое занятие №15

### Ознакомление с процедурами генерации вероятностных распределений

Данное занятие предназначено для ознакомления студентов с 3-мя способами получения вероятностных распределений: табличный, аналитический и с помощью встроенной в GPSS библиотеки процедур теоретических вероятностных распределений. Библиотека GPSS содержит 24 распределения.

а) Листинг 15а – Программа формирования временных интервалов с экспоненциальным распределением с помощью аналитического выражения.

```
*экспоненциальное распределение длительности обслуживания
tab  table      m1,0,0.2,100          ;гистограмма задержек
eks  fvariable  -log((1+rn8)/1000)    ;экспоненциальное распр.
      generate   2                    ;генерация потока
      advance    (3#v$eks)            ;задержка транзакта
      tabulate   tab                  ;табуляция задержек
      terminate  1
```

б) Листинг 15б – Программа формирования временных интервалов с экспоненциальным распределением при табличном задании функции.

```
*экспоненциальное распределение длительности обслуживания
tab  table      m1,0,0.2,100          ;гистограмма задержек
eks  function    rn1,c24              ;табличное экспоненциальное распр.
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.
38/.8,1.6/.84,1.85/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.9
6,3.2/.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/1,8
      generate   2                    ;генерация потока
      advance    3,fn$eks             ;задержка транзакта
      tabulate   tab                  ;табуляция задержек
      terminate  1
```

в) Листинг 15в – Программа формирования временных интервалов различных распределений с помощью встроенных процедур.

```
*экспоненциальное распределение длительности обслуживания
tab  table      m1,0,0.2,100          ;гистограмма задержек
      generate   2                    ;генерация потока
      advance    (exponential(1,0,3));библиотека процедур
*аргументы: Stream (rn),Locate (сдвиг),Scale (сжатие)
      tabulate   tab                  ;табуляция задержек
      terminate  1
```

```
*равномерное распределение длительности обслуживания
tab  table      m1,0,0.2,100          ;гистограмма задержек
      * generate  2                    ;генерация потока
      advance    (uniform(1,2,10))    ;библиотека процедур
*аргументы: Stream (rn), Min, Max
      tabulate   tab                  ;табуляция задержек
      terminate  1
```

```
*треугольное распределение длительности обслуживания
tab  table      m1,0,0.2,100          ;гистограмма задержек
      * generate  2                    ;генерация потока
      advance    (triangular(1,2,12,5));библиотека процедур
*аргументы: Stream (rn), Min, Max, Mode
      tabulate   tab                  ;табуляция задержек
      terminate  1
```

Задание.

1. Произвести запуск программ с 3-мя вариантами получения экспоненциального распределения. Рассмотреть гистограммы.

Внимание. При каждом новом запуске программы необходимо удалять с экрана предыдущие гистограммы. В противном случае программа по каждому транзакту будет обращаться к экрану, что многократно увеличит длительность прогона.

2. Провести сопоставительную оценку производительности рассматриваемых вариантов.

С этой целью произвести запуск каждой программы с длиной прогона, равной  $N = 5000000$ . С помощью секундомера измерить продолжительности прогонов и зафиксировать результаты в отчёте.

Производительность программы становится важным параметром, если в результате прогона необходимо собрать достаточную статистику по редко возникающим событиям. Например, когда нужно оценить площадь «хвоста» распределения.

3. Для экспоненциального распределения 15в произвести серию прогонов с длительностями  $N_1 = 1000$ ,  $N_2 = 10000$ ,  $N_3 = 100000$  и  $N_4 = 1000000$ . Объяснить различную степень сглаженности полученных гистограмм.

4. Закрыть с помощью \* программу экспоненциального распределения и открыть программу равномерного распределения. Зафиксировать гистограмму в отчёте.

5. Закрыть программу равномерного распределения и открыть программу треугольного распределения. Зафиксировать гистограмму в отчёте.

6. Запустить одновременно все три программы из листинга 15в, убрав звёздочки перед всеми операторами `generate`. Объяснить странную форму полученной гистограммы.

Варьируя (осмысленно) параметрами распределений (операнды в операторе `advance`), получить несколько «экзотических» по форме гистограмм.

## 5 Лабораторные занятия

### Лабораторная работа № 1

Исследование маршрутизатора сети коммутации пакетов методами имитационного и аналитического моделирования

#### 5.1 Цель работы:

5.1.1 Освоить приёмы конфигурации фрагментов сети коммутации пакетов на примере исследования маршрутизатора методами имитационного и аналитического моделирования.

5.1.2 Освоить способы изменения параметров маршрутизатора (входные потоки, каналные скорости, допустимые длины очередей и др.) для удовлетворения требований к системе по качеству обслуживания (*QoS*), при заданных параметрах входящих потоков пакетов.

5.1.3 Освоить способы построения основных функциональных зависимостей фрагментов сети.

5.2 Объект исследования. Схема исследуемого маршрутизатора представлена на рисунке 5.1.

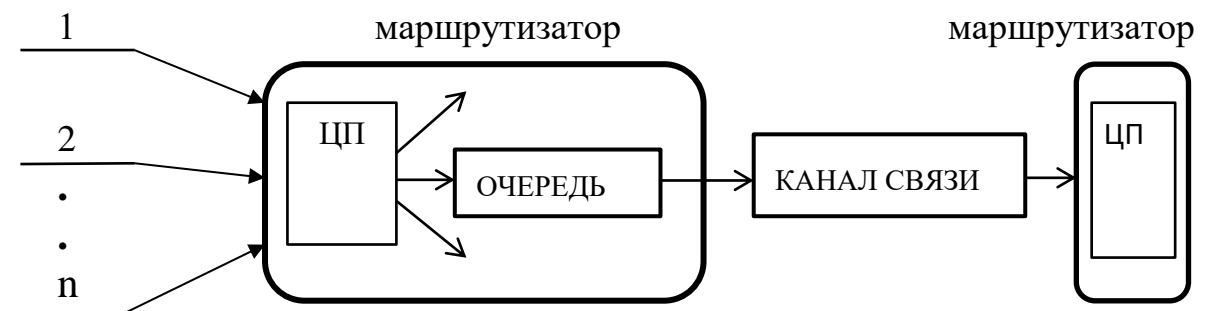


Рисунок 5.1 Схема исследования маршрутизатора

В интересующей нас части, маршрутизатор работает следующим образом:



- все пакеты входящие из разных направлений поступают в центральный процессор (или специализированный процессор выбора пути), который выбирает канал (направление) дальнейшей передачи пакета;
- при наличии свободных каналов, процессор немедленно приступает к передаче пакета;
- если канал занят, то пакет ставится в соответствующую очередь и будет передан по мере освобождения канала.

В теории телетрафика такой вид обслуживания классифицируется как обслуживание с неявными потерями, а конкретнее – обслуживание с очередями.

Основными характеристиками, определяющими качество функционирования сетей передачи данных, как и любых сетей с коммутацией пакетов, являются длительности задержек пакетов, включая джиттер, и вероятности потерь пакетов. Источниками задержек и потерь являются узлы коммутации (маршрутизаторы и коммутаторы) и каналы связи, через которые эти пакеты передаются. Каждый узел или канал, встречающиеся на пути продвижения пакета, вносят свою долю в эти характеристики.

Ниже приводятся методики определения основных вероятностно-временных характеристик маршрутизатора с помощью аналитической и имитационной моделей.

### 5.3 Аналитическая модель

Возможности аналитического моделирования, как правило, более ограничены по сравнению с имитационным, поэтому в данной лабораторной работе мы ограничимся расчётом только двух характеристик: средняя длина очереди пакетов и среднее время обслуживания пакета маршрутизатором, которое состоит из времени ожидания обслуживания и времени передачи пакета по каналу. При этом в обеих моделях (аналитической и имитационной) рассматривается только система  $M/M/1/\infty$  (одноканальная система с пуассо-

новским потоком пакетов на входе, экспоненциальным распределением времени обслуживания при бесконечной очереди).

Эти характеристики легко рассчитываются по формулам Хинчина-Полячека и Литтла, которые для системы  $M/M/1/\infty$  принимают следующий вид (5.1):

$$q = \rho / (1 - \rho) \quad \text{и} \quad t_q = q / \lambda, \quad (5.1)$$

где  $q$  – средняя длина очереди в рассматриваемой системе,

$\lambda$  – интенсивность входного потока пакетов к рассматриваемому каналу,

$t$  – средняя длительность обслуживания (передачи пакета по каналу),

$(\rho = \lambda t)$  – нагрузка одноканальной системы массового обслуживания,

$t_q$  – среднее время обслуживания пакета.

## 5.4 Имитационная модель

Моделирующая программа представлена ниже.

```
;ПРОГРАММА МОДЕЛИРОВАНИЯ РАБОТЫ МАРШРУТИЗАТОРА (РОУТЕРА).
;1 е.м.в. = 1 мкс
interval equ 50 ;интервал между пакетами
skor equ 100 ;установка канальной скорости, Мбит/с
ochmax equ 100 ;максимальная длина очереди
tab table m1,0,200,30 ;Занести значение СЧА m1 (время
; жизни транзакта)
dlina equ 500 ;длина передаваемого gfrtnf в
байтах
tprd fvariable 8#dlina/skor ;вычисление времени передачи
; пакета по каналу
eks fvariable -log((1+rn8)/1000) ;экспоненц. распределение
;-----
generate (interval#v$eks) ;генерация потока пакетов.
test L Q$och,ochmax,otk ;есть места в очереди?
queue och ;Да. Установка пакета в очередь.
seize kan ;Занять канал.
depart och ;Покинуть очередь
advance (v$tprd#v$eks) ;Задержать пакет в канале на
; время его передачи.
release kan ;Освободить канал.
tabulate tab ;Зафиксировать событие в таблице tab
terminate 1 ;покинуть систему
otk savevalue ot+,1 ;Зафиксировать потерю пакета
terminate 1 ;покинуть систему
```

В исходной версии модель характеризуется следующими параметрами:

- Входной поток пакетов  $\lambda = 20$  кПакетов/с.

В моделях *GPSS* генерация потоков пакетов (транзактов) определяется в операторе *generate* операндом *A* как интервал между моментами поступления пакетов  $t_{\text{инт}}$ , который подсчитывается как  $t_{\text{инт}} = 1/\lambda = 1/2 \cdot 10^4 = 50$  мкс.

В *GPSS* все временные интервалы задаются в единицах модельного времени (е.м.в.), а в данной программе принято 1 е.м.в. = 1 мкс. Поэтому в первом операторе (эквивалент) метке *interval* присвоено значение 50 е.м.в. Это значение используется в операторе *generate* совместно с функцией экспоненциального распределения этих интервалов.

- Канальная скорость задаётся вторым оператором. В исходном варианте установлена скорость 100 Мбит/с.

- В третьем операторе установлен максимально допустимый размер очереди пакетов (объём буферного накопителя). В исходном варианте установлена максимальная длина очереди – 100 пакетов.

- В четвёртом операторе устанавливается средняя длина передаваемых пакетов в байтах, а в следующем операторе в зависимости от этой длины и канальной скорости *skor* вычисляется среднее время передачи пакета по каналу в е.м.в. Это среднее время (метка *trpd*) совместно с экспоненциальной функцией распределения используется в операторе *advance* для задержки пакета на время его передачи по каналу.

Здесь необходимо отметить, что экспоненциальное распределение допускает появление в серии опытов с бесконечно малой вероятностью бесконечно большой величины (в нашем случае длины пакета). Реальные же технологии всегда ограничивают длину пакетов (кадров) какой-либо аргументированной величиной. Например, в *ATM* это 53 байта, в *Ethernet* – 1500 байт, в *Frame Relay* – 4500 байт, а в *IP*-пакете версии *IPv4* длина пакета ограничена только размером поля в заголовке пакета, указывающим на его длину. Это

поле состоит из 2-х байтов и, следовательно, длина пакета не может превышать 65535 байт.

В данной модели реализовано экспоненциальное распределение длин пакетов по двум причинам:

- для упрощения модели, стремление приблизить процесс формирования пакетов к реальному, потребовало бы учёта очень многих факторов, в том числе и таких, как пачечный характер потока пакетов;
- невозможность сопоставления результатов аналитического и имитационного моделирования, так как система  $M/M/1/\infty$  предполагает экспоненциальное распределение длительности обслуживания, а, следовательно, и длины пакетов.

В то же время такое допущение несколько не мешает достижению основной цели лабораторной работы – овладение приёмами изучения телекоммуникационных систем с помощью аналитических и имитационных моделей.

## 5.5 Лабораторное задание

5.5.1 Открыть программу моделирования *router*.

5.5.2 Зафиксировать в рабочей тетради исходные параметры модели:

- входной поток –  $\lambda = 1/interval = 1/50 \text{ емв} = 1/50 * 10^{-6} \text{с} = 20 \text{ кПак/с}$ ;
- канальная скорость –  $scor = 100 \text{ бит/емв} = 100 \text{ Мбит/с}$ ;
- максимальная длина очереди –  $ochmax = 100$ ;
- средняя длина пакета –  $dlina = 500 \text{ байт}$ ;
- время передачи по каналу пакета средней величины –  $tprd = 8 * dlina / scor = 8 * 500 / 100 = 40 \text{ е.м.в.} = 40 \text{ мкс}$ .

5.5.3 Определить, в соответствии с исходными данными п.5.5.2 расчётные значения для средней длины очереди и среднего времени обслуживания по п. 5.3:  $q = \rho / (1 - \rho)$  и  $t_q = q / \lambda$ ,

где  $\rho$  – определяется как  $\rho = \lambda t_{prd} = 20 * 10^3 * 40 * 10^{-6} = 0.8$

#### 5.5.4 Запустить модель и зафиксировать основные результаты прогона:

- числа потерянных пакетов (величина  $ot$ );
- средняя длина очереди ( $AVE.CONT$ );
- среднее время ожидания ( $AVE.TAIM$ );
- коэффициенты использования канала ( $UTIL$ ).

Длительность прогона на данном этапе принять равной 1000000 транзактов.

#### 5.5.5 Сопоставить результаты, полученные по пп. 5.5.3 и 5.5.4, т.е.

$q$  с  $q_{\text{имит}}$  и  $t_q$  с  $t_{q\text{имит}}$

При этом длина очереди  $q_{\text{имит}}$  определяется непосредственно по результату прогона как  $AVE. CONT$ , а длительность обслуживания вычисляется суммированием длительности ожидания и времени передачи по каналу (5.2), т.е.

$$t_{q\text{имит}} = AVE.TAIM + t_{prd} \quad (5.2)$$

Напомним, что вышеприведённая аналитическая модель предполагает наличие бесконечной очереди, т.е. применима только при  $\rho < 1$ , иначе очередь будет расти до бесконечности и её состояние будет неопределённым (конкретнее – оно будет зависеть от продолжительности её работы). В связи с этим следующие пункты задания, проводимые с увеличением нагрузки (а, следовательно, и значения  $\rho$ ), будут выполняться только с имитационной моделью.

#### 5.5.6 При дальнейших погонах фиксировать в таблице 5.1 следующие результаты:

- вероятность потери пакетов ( $P_{\text{пот}}$ ). Определяется по отношению числа потерянных пакетов ( $savevalue\ ot$  в отчёте) к общему числу поступающих в направление пакетов (длина прогона, задаваемая в команде  $start$ );
- средняя длина очереди ( $AVE\ CONT$ );
- среднее время ожидания ( $AVE\ TIME$ );
- коэффициенты использования канала ( $UTIL$ ).

– вероятность превышения общего времени задержки пакетов (ожидание + передача) допустимой величины ( $t_{\text{доп}}$ ), т.е.  $P_{\text{прев}} = P(t_{\text{зад}} > t_{\text{доп}})$ . Эта вероятность определяется ориентировочно по гистограмме и точно – по таблице *tab* в отчёте. Например, если  $t_{\text{доп}} = 1\text{мс} = 1000$  е.м.в, то в последнем столбце таблицы (%) в строке интервала в 1000 е.м.в. будет указана доля пакетов, обработка которых уложилась в  $t_{\text{доп}}$ .

Таблица 5.1 – Зависимость качества обслуживания от входной нагрузки и параметров системы

Номер опыта	Входные данные			Выходные данные				
	$\lambda$	<i>scor</i>	<i>ochmax</i>	$P_{\text{пот}}$	<i>CONT</i>	<i>TIME</i>	<i>UTIL</i>	$P_{\text{прев}}$
1								
2								
.								
<i>N</i>								

5.5.7 Провести 3 эксперимента для значений входного потока  $\lambda$ ,  $1.25\lambda$  и  $1.5\lambda$ . Допустимое время задержки принять равным  $t_{\text{доп}} = 4\text{мс} = 4000$  е.м.в. Результаты занести в таблицу 5.1.

5.5.8 Изменяя размер максимальной очереди (*ochmax*) при входном потоке равном  $1.5\lambda$ , определить влияние этой величины на вероятности  $P_{\text{пот}}$  и  $P_{\text{прев}}$ . Пределы изменения *ochmax* выбрать самостоятельно для получения наиболее наглядных зависимостей. Результаты занести в таблицу 5.1.

5.5.9 При входном потоке равном  $1.5\lambda$ , изменением пропускной способности каналов (оператор *skor*) и максимальной очереди (*ochmax*) добиться выполнения основных качественных показателей маршрутизатора  $P_{\text{пот}}$  и  $P_{\text{прев}}$  (параметры *QoS*) при минимальной канальной скорости. Величину допустимого времени задержки принять равной  $t_{\text{доп}} = 4\text{мс} = 4000$  е.м.в. Значения вероятностей  $P_{\text{пот}}$  и  $P_{\text{прев}}$  взять из таблицы 5.2 индивидуально по бригадно. Результаты занести в таблицу 5.1.

Обратите внимание: при увеличении канальной скорости, улучшаются оба показателя – вероятности  $P_{\text{пот}}$  и  $P_{\text{прев}}$ , а при изменении максимальной очереди, например, при её уменьшении вероятность  $P_{\text{пот}}$  будет увеличиваться, а вероятность  $P_{\text{прев}}$  будет уменьшаться. Поэтому при оптимизации системы, если не удовлетворяются оба показателя, то увеличивается канальная скорость, а если только один из них, то выполняется соответствующая корректировка максимальной очереди.

Таблица 5.2 Предельные показатели качества для оценки бригадами

Бригада	1	2	3	4	5	6	7	8
$P_{\text{пот}}$	0.01	0.01	0.02	0.02	0.03	0.03	0.04	0.04
$P_{\text{прев}}$	0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17

5.5.10 Провести оценку доверительного интервала для вероятности потерь пакетов (5.3):

$$E = \pm t_{\beta} \sqrt{\frac{P(1-P)}{N}}. \quad (5.3)$$

где  $P$  – наблюдаемая оценка вероятности,

$N$  – длительность прогона (число пропущенных системой пакетов-транзактов),

$t_{\beta}$  – коэффициент, зависящий от доверительной вероятности  $\beta$ . Для  $\beta=0,95$  величина  $t_{\beta} = 1,96$ .

5.5.11 Повторить п.5.5.5 (сопоставление результатов аналитического и имитационного моделирования) при различных длительностях прогонов. Представить графически, как изменяется относительная погрешность моделирования при изменении длительности прогона, т.е. представить функцию  $\delta(N)$ . Относительную погрешность определять по формуле (5.4):

$$\delta = (S_{\text{расч}} - S_{\text{опыт}}) / S_{\text{расч}}. \quad (5.4)$$

где  $S$  – произвольный рассчитываемый или измеряемый параметр.

## 5.6 Содержание отчета

5.6.1 По каждому пункту лабораторного задания зафиксировать входные и выходные данные и занести их в таблицу.

5.6.2 По пп. 5.5.7, 5.5.8 и 5.5.11 результаты экспериментов представить графически.

5.6.3 По п. 5.5.9 зарисовать гистограмму и указать на ней значение допустимого времени задержки  $t_{\text{доп}}$ .

## 5.7 Контрольные вопросы

5.7.1 Какие факторы моделируемой системы влияют на функционирование маршрутизатора и каким образом?

5.7.2 Какие отклики моделирующей системы характеризуют качество работы системы ( $QoS$ )?

5.7.3 Как проводится оптимизация системы по необходимой скорости канала и максимальной длине очереди для выполнения параметров  $QoS$ ?

## Лабораторная работа № 2

### Оптимизация структурных параметров в сетях

#### с коммутацией пакетов

#### 1 Цель работы.

Изучить принципы организации информационного обмена в сети с коммутацией пакетов в дейтаграммном режиме. Исследовать зависимость качественных показателей сети от параметров её структурных элементов.

#### 2 Краткие сведения об объекте моделирования

##### 2.1 Общие принципы построения сетей с КП.

Коммутация пакетов (КП) является в настоящее время единственным способом коммутации, предназначенным для обмена всеми видами информации в сетях следующего поколения. КП всецело используется в сети Internet и для передачи данных в сотовых сетях.

КП предусматривает разбивку длинных сообщений на отдельные порции - пакеты, снабжение каждого пакета адресом получателя и другими необходимыми для передачи по сети атрибутами и последовательную передачу этих пакетов через сеть от одного маршрутизатора к другому до маршрутизатора получателя. При этом сеть в соответствии с протоколами сетевого уровня (3-ий уровень ЭМВОС) выполняет процедуры маршрутизации, защиты от ошибок, организации очередей (буферизация в маршрутизаторах), управления потоками и др.



Для каждого пакета в соответствии с его адресными признаками мы будем различать узел–отправитель и узел–получатель. Как правило, это маршрутизаторы. Первый обеспечивает абоненту–отправителю точку входа в сеть КП, а второй – точку выхода из сети к абоненту–получателю.

□ Длина поля данных в различных протоколах строго не оговаривается, но в заголовке IP-пакета версии IPv4 для указания общей длины (заголовков + данные) отведено 2 байта, т.е. не более 65535 байт. Практически чаще всего применяются либо очень короткие пакеты (100÷200 байт для пакетов управления и для пакетов IP-телефонии), либо пакеты длиной 1500 байт по формату информационной части кадра Ethernet.

Дейтаграммный режим передачи пакетов, рассматриваемый в данной работе, в отличие от режима виртуальных каналов предполагает выдачу пакетов в сеть без предварительного установления соединения. При этом пакеты передаются независимо друг от друга, могут доводиться до узла получателя по различным маршрутам и, в общем случае, могут поступать к абоненту (на 4-ый уровень) неупорядоченно, а также с потерями или с дублями пакетов.

В любом случае, каждый маршрутизатор содержит таблицу маршрутизации (статическую или динамически обновляемую), в которой каждому адресу получателя соответствует одно из исходящих из данного узла направлений передачи данных (например, портов маршрутизатора).

## 2.2 Проблемы параметрической оптимизации в сетях с КП.

Чрезвычайная сложность параметрической оптимизации в сетях с КП вызвана, во-первых, большим количеством показателей качества функционирования этих сетей, а во-вторых, большим количеством структурных параметров сети, влияющих на эти показатели. Причём, как правило, эти влияния разнонаправлены.

Среди показателей качества отметим следующие:

- вероятность потери пакета;
- время задержки пакета в сети;
- вероятность доступа абонентов к услугам сети;
- спектр услуг доступных абоненту в произвольный момент времени;
- показатели надёжности и живучести сети;
- вопросы информационной безопасности (доступность, конфиденциальность, целостность информации) и др.

Структурные параметры, влияющие на эти показатели:

- конфигурация сети;
- распределение канальных скоростей по ветвям сети;
- интенсивность входного потока пакетов с детализацией по видам трафика, размерам и приоритетностью пакетов;
- матрицы тяготений информационных потоков по видам трафика;
- размеры буферов в маршрутизаторах для ожидающих передачи пакетов;

- производительность узлов и длительность обработки пакетов в узлах.
- алгоритмы маршрутизации пакетов в сети и др.

Очевидно, что невозможно проводить всеобъемлющее исследование в рамках одной лабораторной работы. Поэтому в лабораторном задании (Раздел 5) проводится принцип варьирования только одного или двух параметров при фиксированном значении остальных.

В условиях разнонаправленного влияния какого-либо параметра сети на два или более показателя качества её функционирования, особую важность приобретает вопрос правильного выбора критерия оптимальности. Это, например, то, что часто называют соотношением цена/качество устройства. При этом, не детализируется, что есть качество (число портов, пропускная способность, надёжность, удобство обслуживания и т.д.).

Одной из наиболее часто используемых форм обобщённого критерия является сумма показателей с их весовыми коэффициентами. Например:

$$K = \sum_{i=1}^m a_i c_i ,$$

где:  $c_i$  — значение  $i$ -го показателя сети (системы),

$a_i$  — весовой коэффициент с которым следует учитывать  $i$ -й показатель в обобщённом критерии. Выбором этих критериев должны заниматься высококвалифицированные эксперты.

Подобный критерий используется, например, в протоколе маршрутизации внутренних роутеров (Interior Gateway Routing Protocol-IGRP), а также в его поздней версии EIGRP (усовершенствованный внутренний протокол маршрутизации шлюзов).

Наилучший путь в этом протоколе в упрощённом варианте выбирается с использованием обобщённой метрики, вычисляемой по формуле:

$$R = [(K_1 / B_e) + (K_2 * D_c)] \cdot r ,$$

где:  $B_e$  = пропускная способность ( $\Delta F$ ) самого медленного канала в пути (в отсутствии загрузки) \*  $(1 - Z)$ . Здесь загрузка  $Z$  измеряется как доля от 1. Фактически  $B_e = \Delta F(1-Z)$  — это ширина *свободной* полосы канала в кбит/с;

$D_c$  = задержка пакета в незагруженной сети в с;

$r$  = относительная надёжность (функция доли пакетов, успешно передаваемых по данному пути);

$K_1, K_2$  — константы, с помощью которых можно учитывать пользовательские предпочтения при выборе альтернативных маршрутов.

В протоколе IGRP по умолчанию установлено:  $K_1=10^7$ ;  $K_2=10^5$ ;  $D_c=21\text{мс}$  (для канала T1; 1,544 Мбит/с);  $r = 1$ .

Путь, имеющий наименьшую обобщённую метрику, считается лучшим. В такой схеме появляется возможность, используя весовые коэффициенты, адаптировать выбор маршрутов к задачам конечного пользователя. Например, увеличивая  $K_1$  или уменьшая  $K_2$ , можно показать, что малое время доведения пакета важнее высокой пропускной способности пути.

В данной работе (п. 5.5) в качестве обобщённого критерия используется отношение  $K = C / (-\lg P_{\text{сум}})$ . Вопрос об объективности такого критерия может

быть предметом отдельного маркетингового анализа, так как требуется сопоставление ущерба (возможно финансового) от потери пакетов с затратами на аренду достаточно широкополосных каналов.

### 2.3 Исследуемые зависимости.

К числу главных показателей качества функционирования сети, которые можно наблюдать на данной имитационной модели, относятся:

- вероятность потери пакетов в процессе передачи по узлам сети;
- время доведения пакетов от узла-отправителя до узла-получателя (часто это время называется временем задержки пакета в сети).

Существует несколько причин потери пакетов в сети КП. Например:

- неисправимые ошибки при передаче по каналу;
- сбои при обработке в узлах сети;
- истечение срока жизни пакета (TTL);
- отсутствие свободных мест в очереди пакетов на выдачу в канал связи.

Последняя причина является наиболее существенной, особенно в загруженных сетях и подробно анализируется в настоящей работе. Вопрос о правильном выборе объёма буфера для очереди пакетов осложняется тем, что этот объём влияет одновременно на оба главных показателя качества сети. Чем больше мест в очереди, тем меньше потери пакетов, но больше длительность задержки, и наоборот. Поэтому вопрос об объёме буферов должен решаться компромиссом между допустимыми потерями и допустимой задержкой. В идеале необходим некоторый обобщённый критерий, учитывающий в абстрактных единицах как ущерб от потери пакетов, так и от их чрезмерной задержки.

Допустимое время доведения пакета в сети определяется характером обработки того или иного вида информации на пользовательском уровне. Например, упорядочение речевых пакетов в IP-телефонии, видео кадров в телевидение и вообще любой обмен в режиме реального времени, когда пришедший с опозданием пакет уже не нужен и приравнивается к потерявшему.

Характерной особенностью требований к времени доведения является то, что средние значения этого времени, которые, как правило, легко определяются аналитическими методами, не интересуют заказчика сети. Более важным является показатель вероятности превышения времени доведения определённого порога, задаваемого для каждой пары узлов сети, либо для сети в целом. Таким образом, перед аналитиками стоит задача в определении распределения  $P(t)$  - вероятностей доведения пакета за время  $t$ .

Практически требование заказчика может задаваться в виде допустимой доли пакетов, доводимых за сверхдопустимое время  $P_{\text{дов}}(t > t_{\text{доп}}) \leq Q$ . Например:  $P_{\text{дов}}(t > 1\text{с}) \leq 0.01$ , устанавливает требование к сети, чтобы не менее 99% всех переданных пакетов было доведено за время, не превышающее 1с. Так требования МСЭ-Т к качеству передачи пакетов в Internet для класса реального времени (РВ) составляет до 100 мс задержки при джиттере до 50 мс, а для эластичного трафика (ЭТ) – до 1 с задержки.

Время доведения пакетов зависит практически от всех параметров сети, приведённых выше (п. 2.2).

### 3 Вопросы для самостоятельной подготовки

- 3.1 Опишите процедуру передачи сообщений в сети с коммутации пакетов с привязкой к уровням модели ВОС?
- 3.2 Какие существуют режимы коммутации пакетов и в чем их различие?
- 3.3 Что может быть источником пакетов для узла – отправителя?
- 3.4 Как выбирается длина пакета при разбивке сообщения?
- 3.5 Как организуется обработка пакетов в узлах?
- 3.6 Назовите основные качественные показатели функционирования сети.
- 3.7 От каких параметров и как (качественно) зависит время доведения пакетов?
- 3.8 Как задаются требования ко времени доведения пакетов?
- 3.9 Чем определяются явные и неявные потери пакетов в сети?
- 3.10 От каких параметров зависит вероятность явной потери пакетов?
- 3.11 От каких параметров зависит вероятность неявной потери пакетов?
- 3.12 В каких случаях возникает необходимость использования обобщённых критериев оптимизации сети (системы)?
- 3.13 Как выбираются обобщённые критерии?
- 3.14 Как определяются весовые коэффициенты в обобщённых критериях?
- 3.15 Как действуют весовые коэффициенты при выборе оптимального маршрута в протоколе IGRP?

### 4 Описание имитационной модели

#### 4.1 Реализация основных элементов модели.

Моделируемая сеть представляет собой 5-и узловую структуру кольцевой конфигурации (Рисунок 1).

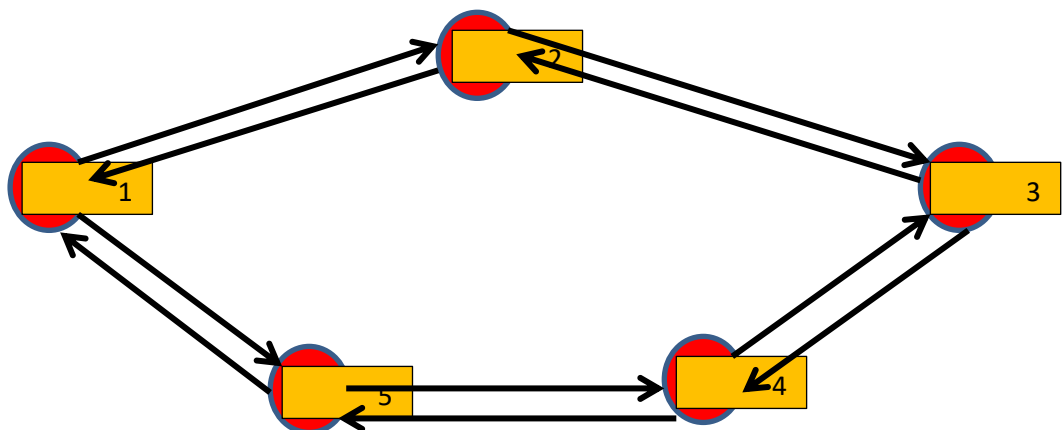


Рисунок 1. Схема моделируемой сети.

В сеть поступают 2 вида трафика: аудио трафик (пакеты IP-телефонии) и трафик данных (пакеты файловой информации). Суммарный входящий в сеть поток пакетов задаётся оператором `vhdrpot` как среднее число входящих пакетов в секунду.

Этот поток оператором `priority` разбивается на поток данных с приоритетом 1 и аудио поток с более высоким приоритетом 2. Оператор `prior` устанавливает соотношение между потоками пакетов D и A. В исходном варианте принято соотношение 1:1. Оператор `kanskor` устанавливает одну и ту же битовую скорость для всех 10-и каналов моделируемой сети.

В моделируемой сети имитируются два типа трафика: прямой (передаётся только между смежными узлами, т.е. по одной ветви) и транзитный (передаётся с одним транзитом, т.е. последовательно по двум ветвям). В исходном варианте оператор `transfer` с вероятностью 0,5 либо заканчивает обслуживание пакета первой ветвью (прямой поток), либо передаёт его на дообслуживание в следующую ветвь (транзитный поток). Вероятность пакета быть транзитным задаётся оператором `Ptransz`.

В каждом узле в каждый исходящий канал организуются две очереди для ожидающих D и A пакетов. Количество мест ожидания (объём буфера) задаётся операторами `bufD` (для пакетов данных) и `bufA` (для аудио пакетов).

Длины пакетов устанавливаются оператором `dlpakD` (для пакетов данных) и оператором `dlpakA` (для аудио пакетов). Четыре оператора (`tdop1D`, `tdop2D`, `tdop1A`, `tdop2A`) устанавливают допустимые времена задержек в зависимости от вида трафика и наличия транзита.

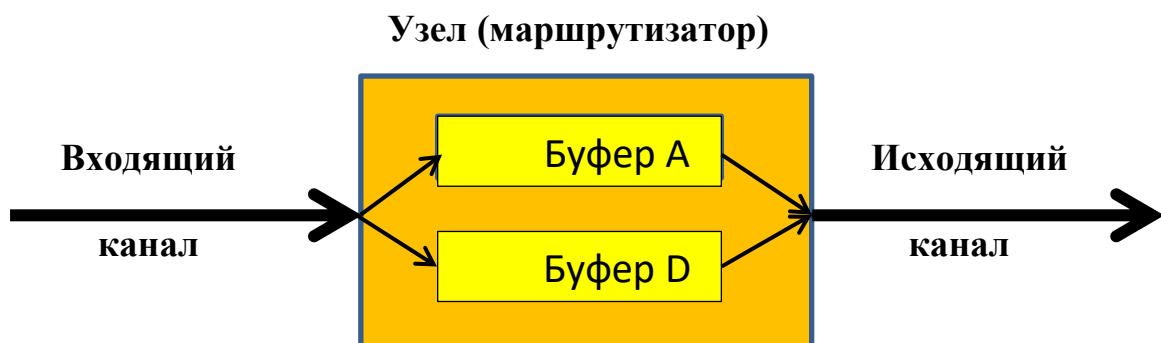


Рисунок 2. Схема прохождения пакетов в узле.

В целом работа модели происходит следующим образом:

- для каждого пакета с равной вероятностью выбирается одна из 10 ветвей сети;
- в выбранной ветви пакет или сразу поступает в канал на выдачу, или поступает в буфер ожидания, или, при отсутствии свободных мест в буфере, покидает систему и фиксируется как потерянный;
- по окончании обслуживания в первой ветви пакет или фиксируется как доведённый до узла получателя (прямой пакет) или передаётся на дообслуживание в следующую ветвь по кольцу (транзитный пакет).

## 4.2 Параметры сети, устанавливаемые в модели.

Первые 12 операторов, приведённые в листинге программы (п. 4.4), позволяют создавать широкий диапазон условий проведения экспериментов. Однако в данном лабораторном задании (п. 5) вариациям подвергаются только первые 4 параметра (входной поток, канальная скорость и объёмы буферов D и A). Остальные 8 параметров, определяемые характеристиками пользовательских потоков и требованиями к ним, не меняются.

## 4.3 Средства наблюдения за результатами прогонов.

Результаты прогона отображаются гистограммами времён доведения и счётчиками событий.

Распределение времён доведения:

- tdov1A – гистограмма времени доведения прямых A-пакетов;
- tdov1D – гистограмма времени доведения прямых D-пакетов;
- tdov2A – гистограмма времени доведения транзитных A-пакетов;
- tdov2D – гистограмма времени доведения транзитных D-пакетов.

Счётчики событий:

- число пакетов, сброшенных из-за отсутствия мест в буфере для аудио пакетов otkbufA и для пакетов данных otkbufD. Число явных потерь;
- число пакетов превысивших допустимое время доведения otkdop1D и otkdop1A (для прямых маршрутов) и otkdop2D, otkdop2A (для транзитных маршрутов). Число не явных потерь;
- sum – суммарное число потерянных пакетов (явные и не явные потери).

□

## □ 4.4 Листинг моделирующей программы

```
* модель сети коммутации пакетов
* 1 единица модельного времени равна 10 мкс
* Установка параметров модели-----
vhdprot      equ      100000 ;интенс. вх. потока(число пак/сек)
prior        function  rn1,d2 ;функция распр.пакетов по приоритетам.
.5,1/1,2      ;аргумент-случайное число rn1,
kanskor      equ      200    ;канальная скорость в ветвях, Mbit/c
bufA         equ      5      ;число мест ожидания для A
bufD         equ      10     ;число мест ожидания для D
dlpakD       equ      1500   ;длина пакетов D
dlpakA       equ      160    ;длина пакетов A
Ptranz       equ      .5     ;вероятность транзитного пакета
tdop1D       equ      50     ;допустимое время доведения прямых пакетов D
tdop2D       equ      150    ;допустимое время доведения транзитных пакетов D
tdop1A       equ      10     ;допустимое время доведения прямых пакетов A
tdop2A       equ      30     ;допустимое время доведения транзитных пакетов A
*-----
tdov1A       table     m1,0,2,40 ;гистогр.доведения для прямых A
tdov1D       table     m1,0,10,40 ;гистогр.доведения для прямых D
tdov2A       table     m1,0,2,40 ;гистогр.доведения транзитных A
tdov2D       table     m1,0,10,40 ;гистогр.доведения транзитных D
kan1         function  rn1,d10 ;выбор 1-го канала
.1,12/.2,23/.3,34/.4,45/.5,51/.6,15/.7,54/.8,43/.9,32/1,21
```

```

kan2      function  P$kan,d10      ;выбор 2-го канала
12,23/15,54/21,15/23,34/32,21/34,45/43,32/45,51/51,12/54,43
raspr     function  rn1,c24        ;экспоненциальное распределение
0,0/.1,.104/.2,.222/.3,.355/.4,.509/.5,.69/.6,.915/.7,1.2/.75,1.38/
.8,1.6/.84,1.85/.88,2.12/.9,2.3/.92,2.52/.94,2.81/.95,2.99/.96,3.2/
.97,3.5/.98,3.9/.99,4.6/.995,5.3/.998,6.2/.999,7/1,8
vint      variable  100000/vhdpot  ;интервал между пакетами
tkanD     variable  8#dlpakD/kanskor/10 ;время передачи пакета D
tkanA     variable  8#dlpakA/kanskor/10 ;время передачи пакета A
sumD      variable  p$kan          ;номер очереди в 1-й ветви
sumA      variable  p$kan+100      ;номер очереди во 2-й ветви
DoA1      function  pr,d2          ;выбор гистограммы для прямых
1,tdov1D/2,tdov1A
DoA2      function  pr,d2          ;выбор гистограммы для транзитных
1,tdov2D/2,tdov2A
Sum variable x$otkbufD+x$otkbufA+x$otkdop1D+x$otkdop2D+x$otkdop1A+x$otkdop2A
*расчёт суммарных потерь (явных и не явных)
*-----
      generate      v$vint,fn$raspr ;генерация входного потока
      priority      fn$prior        ;разделение потока на приоритеты.
      assign        tranz,0         ;признак "без транзита (прямая)"
      assign        kan,fn$kan1     ;выбор 1-го канала
      test e        pr,1,mett       ;приоритет 1 ?
      assign        tkan,v$tkanD    ;Да.Время передачи для D
      assign        ocher,v$sumD    ;номер очереди в 1-м канале
      test l        Q*ocher,bufD,otbd ;есть место в очереди D?
      transfer      ,rab1           ;Да.К обслуживанию
mett      assign    tkan,v$tkanA     ;Время передачи для A
      assign        ocher,v$sumA    ;номер очереди в 1-м канале
      test l        Q*ocher,bufA,otba ;есть место в очереди A?
rab1      queue      p$ocher        ;Установка в очередь
      seize        p$kan           ;Занять канал
      depart       p$ocher         ;освободить очередь
      advance      p$tkan,fn$raspr ;передача пакета по каналу
      release      p$kan           ;освободить канал
      tabulate     fn$DoA1         ;допустимое t для прямых
      transfer     Ptranz,met       ;есть транзит?
*-----
      assign        tranz,1         ;Да.Признак "с транзитом"
      assign        kan,fn$kan2     ;выбор 2-го канала
      test e        pr,1,mtt       ;приоритет 1 ?
      assign        tkan,v$tkanD    ;Да.Время передачи для D
      assign        ocher,v$sumD    ;номер очереди во 2-м канале
      test l        Q*ocher,bufD,otbd ;есть место в очереди D?
      transfer      ,rab2           ;Да.К обслуживанию
mtt      assign    tkan,v$tkanA     ;Время передачи для A
      assign        ocher,v$sumA    ;номер очереди во 2-м канале
      test l        Q*ocher,bufA,otba ;есть место в очереди A?
rab2      queue      p$ocher        ;Установка в очередь
      seize        p$kan           ;Занять канал
      depart       p$ocher         ;освободить очередь
      advance      p$tkan,fn$raspr ;передача пакета по каналу
      release      p$kan           ;освободить канал
      tabulate     fn$DoA2         ;допустимое t для транзитных
*-----
met       test e    p$tranz,0,tr     ;пакет прямой?
      test e      pr,1,mlt          ;Да.Пакет D?
      test l      m1,tdop1D,ot1D    ;Да.Время для D превышено?
      transfer    ,fin              ;Нет
mlt       test l   m1,tdop1A,ot1A    ;время для A превышено?
      transfer    ,fin              ;Нет
tr        test e   pr,1,m2t         ;пакет транзитный
      test l      m1,tdop2D,ot2D    ;Да.Время для D превышено?

```

	transfer	,fin	;Нет
m2t	test 1	m1,tdop2A,ot2A	;время для A превышено?
	transfer	,fin	;Нет
*-----			
otbd	savevalue	otkbufD+,1	;число отказов в очереди D
	transfer	,fin	
otba	savevalue	otkbufA+,1	;число отказов в очереди A
	transfer	,fin	
ot1D	savevalue	otkdop1D+,1	;число отказов прямых D
	transfer	,fin	
ot1A	savevalue	otkdop1A+,1	;число отказов прямых A
	transfer	,fin	
ot2D	savevalue	otkdop2D+,1	;число отказов транзитных D
	transfer	,fin	
ot2A	savevalue	otkdop2A+,1	;число отказов транзитных A
	transfer	,fin	
*-----			
fin	savevalue	summa,v\$sum	;подсчёт суммарных потерь
final	terminate	1	;покинуть систему

## 5 Лабораторное задание

5.1 Изучить модель сети, способы ввода исходных данных и считывания результатов эксперимента.

5.2 Измерить выходные параметры для исходного состояния системы (т.е. для установленных в программе исходных данных). Зафиксировать 4 гистограммы (схематично) и значения 7-и счётчиков событий, приведённых в п. 4.3. Параметры гистограмм (величину интервала и число интервалов) подобрать для наиболее наглядного представления функциональных зависимостей.

5.3 Изменяя нагрузку на сеть с помощью параметров входной поток или канальная скорость, построить график зависимости суммарных потерь от среднего коэффициента использования каналов ( $P_{\Sigma}$  (UTIL)).

5.4 Изменяя канальную скорость добиться снижения вероятности суммарных потерь пакетов (явных и не явных) до  $P_{\Sigma} < 0,01$ .

5.5 Варьируя 4-мя параметрами (входной поток, канальная скорость, размеры буферов) подобрать 4-5 наиболее интересных состояний системы, ориентируясь на коэффициент использования каналов (например, перегрузки или недогрузки). Объяснить оригинальность этих состояний, т.е. цель исследования системы в этих состояниях.

При анализе гистограмм произвести с помощью команды Window>Tile совмещение всех 4 гистограммы в одном окне, зафиксировать эти гистограммы в отчёте (схематично) и объяснить их содержание и причины различия. Для каждой гистограммы указать:

- допустимое время доведения пакетов на временной оси  $tdop_{xx}$ ;
- площади «хвостов» (вероятность превышения допустимого времени  $P(t_{\text{дов}} > t_{\text{доп}})$ );
- средние значения представленных случайных величин (Mean:);
- вероятности суммарных потерь пакетов  $P_{\Sigma}$  (явных и неявных);
- наблюдавшиеся в каждом опыте коэффициенты использования каналов



(UTIL).

5.6 При одном из значений коэффициента использования каналов (наиболее характерном с точки зрения функциональных зависимостей) снять следующие три зависимости вероятности потерь пакетов от размеров буферов ожидания ( $b$ ) и построить их на одном графике:

- вероятность явных потерь пакетов  $P_{\text{явн}}(b)$ ;
- вероятность не явных потерь пакетов  $P_{\text{неявн}}(b)$ ;
- суммарная вероятность потерь пакетов  $P_{\Sigma}(b)$ .

В суммарные потери включаются как явные потери (из-за занятости буферов ожидания), так и неявные (из-за превышения допустимого времени задержки). При выборе размеров буферов для сокращения общего поля значений аргументов придерживаться правила: размер буфера пакетов данных  $b_D$  должен вдвое превышать размер буфера аудио пакетов  $b_A$ . Зафиксировать в отчёте размеры буферов минимизирующих потери. Если функция  $P_{\Sigma}(b)$  не имеет явного минимума, изменить (осмысленно) исходные данные так, чтобы такой минимум появился.

5.7 Для полученного в п. 5.6 оптимального значения размеров буферов  $b_A$  и  $b_D$ , варьируя канальную скорость, провести оптимизацию сети по обобщённому критерию  $K = C / (-\lg P_{\Sigma})$ . Зафиксировать в отчёте значение канальной скорости, оптимизирующей обобщённый критерий  $K$ .

## 6 Отчёт по работе

По результатам выполнения п.п. 5.2÷5.7 лабораторного задания представить графики, гистограммы и показания счётчиков потерь пакетов. Сделать выводы по каждому пункту задания.

## 7 Контрольные вопросы

7.1 Как вводятся в модель основные параметры сети?

7.2 Объяснить на гистограммах вероятностно-временные характеристики процесса передачи пакета в сети. В чём отличие гистограмм для слабо и сильно загруженных сетей?

7.3 На какие показатели системы и как влияют размеры буферов ожидания? Показать по результатам экспериментов.

7.4 Представить и объяснить результаты оптимизации сети по размерам буферов.

7.5 Представить и объяснить результаты оптимизации сети по обобщённому критерию.

## Критерии согласия для проверки гипотез

Критерии согласия применяются для проверки гипотезы о законе распределения исследуемой случайной величины. Во многих практических задачах точный закон распределения неизвестен. Поэтому выдвигается гипотеза о соответствии имеющегося эмпирического закона, построенного по наблюдениям, некоторому теоретическому. Данная гипотеза требует статистической проверки, по результатам которой будет либо подтверждена, либо опровергнута.

Критерий согласия Пирсона  $\chi^2$  – один из основных, который можно представить как сумму отношений квадратов расхождений между теоретическими и эмпирическими частотами к теоретическим частотам:

$$\chi^2 = \sum_{i=0}^s \frac{(m_i - n_i)^2}{n_i} \quad (A1)$$

где  $s$  – число групп, на которые разбито эмпирическое распределение,

$m_i$  – наблюдаемая частота признака в  $i$ -й группе,

$n_i$  – теоретическая частота.

Для распределения  $\chi^2$  составлены таблицы, где указаны критические значения критерия согласия  $\chi^2$  для выбранного уровня значимости  $\alpha$  и степеней свободы  $k$ .

Число степеней свободы  $k$  определяется как число групп в ряду распределения ( $s$ ) минус число связей:  $k = s - r$ . Под числом связей понимается число показателей эмпирического ряда, использованных при вычислении теоретических частот, т.е. показателей, связывающих эмпирические и теоретические частоты. Например, при выравнивании по кривой распределения Эрланга или нормального распределения имеется три связи. Поэтому число степеней свободы определяется как  $k = s - 3$ . Для распределения Пуассона число  $k = s - 2$ .

При полном совпадении теоретического и эмпирического распределений  $\chi^2 = 0$ , в противном случае  $\chi^2 > 0$ . Если  $\chi^2_{\text{расч}} > \chi^2_{\text{табл}}$ , то при заданном уровне значимости и числе степеней свободы гипотезу о несущественности (случайности) расхождений отклоняем. В случае, если  $\chi^2_{\text{расч}} < \chi^2_{\text{табл}}$ , гипотезу принимаем и с вероятностью  $P$  можно утверждать, что расхождение между теоретическими и эмпирическими частотами случайно.

Пример: Проверим гистограмму, полученную в занятии 9 для 9-и канального устройства, на соответствие распределению Эрланга. В данном случае число групп в ряду распределения (число интервалов гистограммы) равно 10. Поэтому эмпирические и расчётные данные сводим в таблицу А.1.

Таблица А.1 – Расчёт критерия согласия Пирсона  $\chi^2$

$i$	$P_i$	$n_i$	$m_i$	$n_i - m_i$	$(n_i - m_i)^2$	$(n_i - m_i)^2/n_i$
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
$\chi^2 = \Sigma$						

Эмпирические данные для данной таблицы  $m_i$  берутся непосредственно из отчёта модуля 9. Расчётные значения вычисляются как  $n_i = P_i * N$ , где  $P_i$  определяется по формуле Эрланга (можно воспользоваться программой *elangcalc*), а  $N$  соответствует длине прогона (число, устанавливаемое при команде *Start*).

После этого проверяется гипотеза по таблице А.2, о соответствии гистограммы распределению Эрланга. В нашем случае число степеней свободы  $k = s - 3 = 10 - 3 = 7$ . Пусть в результате расчета по таблице 1 оказалось, что  $\chi^2 = 2,5$ . Тогда по таблице 2 для  $k = 7$  находим, что  $2.17 < \chi^2 < 2.83$ , и делаем

вывод о том, что с вероятностью  $P$ , лежащей между 0.9 и 0.95, гипотеза о соответствии эмпирического распределения распределению Эрланга является правдоподобной.

Таким образом, любой критерий согласия не подтверждает однозначно правильность гипотезы о законе распределения исследуемой случайной величины. Критерий лишь определяет **вероятность правильности гипотезы**. А при проверке нескольких гипотез (например, соответствие эмпирического распределения нормальному или эрланговскому) можно принять более вероятную.

Таблица А2 Значения квантили  $\chi_{\alpha}^2$ , в зависимости от уровня вероятности и числа степеней свободы  $k$

k	Вероятность $P(\chi^2 > \chi_{\alpha}^2)$								
	0.99	0.95	0.90	0.70	0.50	0.30	0.10	0.05	0.01
1	0.00016	0.0039	0.016	0.148	0.455	1.074	2.71	3.84	6.63
2	0.020	0.103	0.211	0.713	1.39	2.41	4.61	5.99	9.21
3	0.115	0.352	0.584	1.424	2.37	3.66	6.25	7.81	11.3
4	0.297	0.711	1.06	2.20	3.36	4.88	7.78	9.49	13.3
5	0.554	1.15	1.61	3.00	4.35	6.06	9.24	11.1	15.1
6	0.872	1.64	2.20	3.83	5.35	7.23	10.6	12.6	16.8
7	1.24	2.17	2.83	4.67	6.35	8.38	12.0	14.1	18.5
8	1.65	2.73	3.49	5.53	7.34	9.52	13.4	15.5	20.1
9	2.09	3.33	4.17	6.39	8.34	10.66	14.7	16.9	21.7
10	2.56	3.94	4.87	7.27	9.34	11.78	16.0	18.3	23.2
11	3.05	4.57	5.58	8.15	10.3	12.90	17.3	19.7	24.7
12	3.57	5.23	6.30	9.03	11.3	14.01	18.5	21.0	26.2
13	4.11	5.89	7.04	9.93	12.3	15.12	19.8	22.4	27.7
14	4.66	6.57	7.79	10.82	13.3	16.22	21.1	23.7	29.1
15	5.23	7.26	8.55	11.72	14.3	17.32	22.3	25.0	30.6
16	5.81	7.96	9.31	12.62	15.3	18.42	23.5	26.3	32.0
17	6.41	8.67	10.1	13.53	16.3	19.51	24.8	27.6	33.4
18	7.01	9.39	10.9	14.44	17.3	20.6	26.0	28.9	34.8
19	7.63	10.1	11.7	15.35	18.3	21.7	27.2	30.1	36.2
20	8.26	10.9	12.4	16.27	19.3	22.8	28.4	31.4	37.6

Таблица Б1 – Список блоков *GPSS*

	Операнды				
	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>ADVANCE</i> (задержать)	Среднее время { @ }	Модификатор разброса [ <i>K</i> , <i>cra</i> <i>j</i> , <i>cra</i> <i>\$</i> имя <i>cra</i> <i>*</i> <i>j</i> ] кроме <i>fn</i> <i>j</i> , <i>fn</i> <i>\$</i> имя <i>fn</i> <i>*</i> <i>j</i> ] или модификатор функций [ <i>fn</i> <i>fn</i> <i>\$</i> имя, <i>fn</i> <i>*</i> <i>j</i> ]			
<i>ASSEMBLE</i> (соединить)	Счетчик соединения { @ }				
<i>ASSIGN</i> (назначить)	Номер параметра { @ }	Назначаемая величина { @ }	Номер модифицирующей функции [ @ ]		
<i>BUFFER</i> (возобновить просмотр)					
<i>DEPART</i> (покинуть очередь)	Имя очереди {имя}	Число единиц [ @ ]			
<i>ENTER</i> (войти)	Имя многоканального устройства {имя}	Число единиц [ @ ]			
<i>GATE</i> (впустить)	Имя логического переключателя {имя}	Следующий блок при невыполнении логического условия [имя]			
<i>GATE</i> (впустить)	Имя блока <i>ASSEMBLE</i> <i>GATHER</i> или <i>MATCH</i> {имя}	Следующий блок при невыполнении логического условия [имя]			
<i>GATE</i> (впустить)	Имя прибора {имя}	Следующий блок при невыполнении логического условия [имя]			
<i>GATE</i> (впустить)	Имя многоканального устройства {имя}	Следующий блок при невыполнении логического условия [имя]			
<i>GATHER</i> (собрать)	Счетчик сборки { @ }				
<i>GENERATE</i> (генерировать)	Среднее время интервала {имя}	Модификатор разброса [ <i>k</i> , <i>cra</i> <i>j</i> , <i>cra</i> <i>\$</i> имя, кроме <i>FN</i> <i>J</i> , <i>FN</i> <i>\$</i> имя] или модификатор функции [ <i>FN</i> <i>J</i> , <i>FN</i> <i>\$</i> имя]	Интервал смещения [ <i>k</i> , <i>cra</i> <i>j</i> , <i>cra</i> <i>\$</i> имя]	ограничитель [ <i>k</i> , <i>cra</i> <i>j</i> , <i>cra</i> <i>\$</i> имя]	Уровень приоритета [ <i>k</i> , <i>cra</i> <i>j</i> , <i>cra</i> <i>\$</i> имя]

## Продолжение таблицы Б

<i>LEAVE</i> (выйти)	Имя многоканального устройства { @ }	Число единиц [ @ ]			
<i>LINK</i> (ввести в цепь)	Имя цепи пользователя { имя }	Критерий присоединения { <i>LIFO, FIFO, PJ</i> }	Альтернативный блок { имя }		
<i>LOGIC {I,R,S}</i> (воздействовать на логический переключатель)	Имя логического переключателя { имя }				
<i>LOOP</i> (организовать цикл)	Номер параметра { @ }	Следующий блок { имя }			
<i>MARK</i> (отметить)	Номер параметра { @ }				
<i>MATCH</i> (синхронизировать)	Адрес сопряженного блока { имя }				
<i>MSAVEVACVE</i> (сохранить значение)	Имя матрицы { имя } [ ±1 ]	Номер строки { @ }	Номер столбца { @ }	Величина, которую следует сохранить { @ }	Тип матрицы [ <i>X, H, V</i> или <i>L</i> ]
<i>PREEMPT</i> (захватить)	Имя прибора { имя }	Возможность захвата по приоритету [ <i>PR</i> ]	Блок, в который переходит захваченный транзакт [ имя ]	Параметры захваченного транзакта [ @ ]	
<i>PRINT</i> (напечатать)	Нижний предел [ имя ]	Верхний предел [ имя ]	Мнемоническое обозначение элемента(1)	Указатель разбивки по страницам [ любая буква ]	
<i>PRIORITY</i> (назначить приоритет)	Новое значение приоритета { @ }	Возможность возобновления просмотра- [ <i>BUFFER</i> ]			
<i>QUEUE</i> (встать в очереди)	Имя очереди { имя }	Число единиц [ @ ]			
<i>RELEASE</i> (освободить)	Имя прибора { имя }				
<i>RETURN</i> (вернуть)	Имя прибора { имя }				
<i>SAVEUALVE</i> (сохранить значение)	Имя сохраняемой величины { имя } [ ±1 ]	Величина, которую следует сохранить { @ }			
<i>SEIZE</i> (занять)	Имя прибора { имя }				

## Продолжение таблицы Б

<i>SELECT</i> (в логическом режиме) { <i>V</i> , <i>NV</i> , <i>I</i> , <i>NI</i> , <i>SE</i> , <i>SNE</i> , <i>SF</i> , <i>SNF</i> , <i>LR</i> , <i>LS</i> } (выбрать)	Параметр, в который заносят номер элемента { @ }	Нижний предел { @ }	Верхний предел { @ }		
<i>SELECT</i> (в режиме max и min) (выбрать)	Параметр, в который заносят номер элемента { @ }	Нижний предел { @ }	Верхний предел { @ }		Исследуемый атрибут (любой СЧА, кроме МХ, МН)
<i>SELECT</i> (в режиме отношения) (выбрать)	Параметр, в который заносят номер элемента { @ }	Нижний предел { @ }	Верхний предел { @ }	Величина для сравнения { @ }	Исследуемый атрибут
<i>SPLIT</i> (расщепить)	Число потомков { @ }	Следующий блок для потомков { имя }	Параметр, в который заносят порядковый номер { @ }	Число параметров у каждого потомка { @ }	
<i>TABULATE</i> (табулировать)	Имя таблицы { имя }	Весовой фактор { @ }			
<i>TERMINATE</i> (завершить)	Счетчик числа завершений [ @ ]				
<i>TEST</i> { <i>G</i> , <i>GE</i> , <i>E</i> , <i>NE</i> , <i>LE</i> , <i>L</i> } (проверить)	Первая величина { @ }	Вторая величина { @ }	Следующий блок при невыполнении условия { имя }		
<i>TRANSFER</i> (в условном режиме) (передать)	Режим выбора BOTH	Первый исследуемый блок { имя }	Второй исследуемый блок { имя }		
<i>TRANSFER</i> (в статическом режиме) (передать)	Режим выбора { @ }	Первый блок { имя }	Второй блок { имя }		
<i>TRANSFER</i> (в безусловном режиме)	Режим выбора не используется	Блок в который переходит транзакт { имя }			
<i>UNLINK</i> (вывести из цепи)	Имя цепи пользователя { имя }	Блок, в который входят выведенные транзакты { имя }	Счетчик выводимых транзактов { @, all }	Номер параметра [ @ ] или [BACK]	Аргумент парности { @ }

## Примеры применения операторов GPSS

В данном приложении показаны примеры использования операторов языка GPSS с конкретными значениями подполей в поле переменных и с пояснениями особенностей выбора этих операндов. В приложение включены только те операторы, которые применяются в практических занятиях настоящего пособия.

***generate 12,4,50, 5,1*** – генерация транзактов, интервалы времени между появлениями транзактов распределены равномерно в диапазоне  $[12 - 4, 12 + 4]$ , первый транзакт появится с задержкой в 50 единиц модельного времени, всего будет создано 5 транзактов, приоритет транзактов равен единице.

***generate 12,4,50,1*** – то же, но количество генерируемых транзактов не ограничено.

***generate 6, fn\$fff,50,5,1*** – то же, но интервал времени между появлениями транзактов есть целая часть произведения числа 6 на значение функции FFF.

***fff function rn1,c4***

**0,0/0.1,0.8/0.5,1.6/1.0,1.9** – описание функции *fff*, ее аргументом является случайная величина (на это указывает значение *rn1*), равномерно распределенная в диапазоне  $[0,1]$ , функция является непрерывной числовой (указатель *c*), заданной таблично четырьмя точками: (0;0), (0.1; 0.8), (0.5,1.6), (1.0; 1.9).

***fff function \*2,d4***

**0,12/1,9/2,8/3,6** – то же, но аргументом является значение второго параметра транзакта, для которого вычисляется значение дискретной величины (*d*) числовой функции *fff*, заданной таблично четырьмя узловыми точками. Это текущее значение округляется до ближайшего большего значения аргумента в узловой точке.

***enter mem,12*** – занятие транзактом 12 единиц емкости в накопителе *mem*.



***leave mem\*2*** – освобождение  $k$  единиц памяти в накопителе *mem*, где  $k$  – значение 2-го параметра транзакта.

***str storage 4096*** – описание накопителя *str* емкостью 4096 единиц.

***terminate 3*** – удаление транзакта из системы, при этом содержимое итогового счетчика уменьшается на 3 единицы, моделирование заканчивается, если содержимое счетчика станет равным или меньше нуля.

***advance a,6*** – задержка транзакта на время, определяемое содержимым полей  $A$  и  $B$ , смысл величин, записываемых в этих подполях, такой же, как и в блоке *generate*.

***split 3,III,6*** – копирование транзактов. В данном случае создаются три копии исходного транзакта. Исходный транзакт направляется в следующий по порядку блок, а созданные копии – в блок с меткой *III*. При этом, параметр 6 основного транзакта увеличивается на единицу, а транзактов-копий – на 2, 3, 4 соответственно.

***assemble 5*** – объединение транзактов. Первый, из вошедших в блок транзактов продолжит движение в системе после того, как в блок придут еще четыре транзакта.

***assign 2, nap*** – изменение параметров транзактов, в данном случае второй параметр транзакта получит значение *NAP*.

***assign 3+,v4*** – изменится значение третьего параметра транзакта – к нему прибавится значение *V4*.

***transfer , mmm*** – безусловная передача управления оператору с меткой (номером) *mmm*.

***transfer ,both,lab1,unn*** – переход к оператору с меткой *lab1*. Если он невозможен, то к оператору с меткой *unn*, если и он невозможен, то транзакт задерживается до следующего момента дискретного модельного времени, в который повторяются указанные попытки перехода.

***transfer .4,aaa,lab*** – транзакт с вероятностью 0.4 переходит к оператору с меткой *lab* и с вероятностью 0.6 к оператору с меткой *aaa*.

***transfer pick,smk7,smk21*** – равновероятный переход к операторам с номерами *smk7*, *smk7+1*, *smk7+2*,..., *smk21*.

***transfer fn,aaa,5*** – переход к оператору, метка которого равна сумме значения функции *aaa* и числа 5.

***loop 6,met*** – организация цикла – переход 5 раз к оператору с меткой *met* и на шестой раз – к следующему по порядку оператору.

***test e v7,k256,lav*** – переход по условию (условная передача управления): в позициях 13 – 18 записывается знак отношения, в первых двух подполях поля переменных записываются сравниваемые величины, если условие выполняется, то перехода нет, иначе переход есть к оператору с меткой *lav*. Символы отношений: *g* – больше, *l* – меньше, *e* – равно, *ne* – не равно, *le* – меньше или равно, *ge* – больше или равно. В данном примере перехода нет, если *v7* = 256, иначе переход к оператору с номером *lav*.

***queue sqv*** – оператор организации очереди, длина очереди *sqv* увеличивается на единицу.

***depart sqv*** – то же, но длина очереди уменьшается на единицу.

***priority 2*** – транзакту присваивается приоритет 2.

***tabulate mat*** – в соответствующий интервал гистограммы с именем *mat* добавляется единица.

***5 variable x2 – k25*** – вычислительный оператор, в данном случае из хранимой величины номер 2 вычитается число 25 и результат присваивается переменной номер 5.

***savevalue 5,\*3*** – хранимая величина номер 5 (*x5*) получает значение третьего параметра транзакта.

## Приложение Г

Таблица Г – Список стандартных числовых атрибутов

Типы объектов	Символическое обозначение	Диапазон изменения	Описание
Транзакты	<i>PF</i>	$\pm(2^{31}-1)$	Текущее значение параметра (формата <i>F</i> «слово», <i>H</i> – «полуслово», <i>B</i> – «байт», <i>L</i> – «плавающая точка») транзакта, обрабатываемого в данный момент.
	<i>PH</i>	$\pm(2^{15}-1)$	
	<i>PB</i>	$\pm(2^7-1)$	
	<i>PL</i>	$\pm(2^{24}-1)$	
	<i>PR</i>	0... 127	Приоритет обрабатываемого транзакта.
	<i>MI</i>	$\pm(2^{31}-1)$	Транзактное время
	<i>MP</i>		Параметрическое транзактное время
Блоки	<i>N</i>	$2^{24}-1$	Счетчик входов в блок
	<i>W</i>	$2^{15}-1$	Счетчик текущего содержимого блока
Память	<i>F</i>	Логические «1» или «0»	Состояние устройства
	<i>FR</i>	0... 999	Коэффициент использования (в долях от 1000)
	<i>FC</i>	$2^{31}-1$	Счетчик входов
	<i>FT</i>	$2^{31}-1$	Среднее время/ транзакта
	<i>S</i>	$2^{31}-1$	Текущее содержимое памяти
	<i>R</i>	$2^{31}-1$	Число свободных единиц памяти
	<i>SR</i>	0... 999	Коэффициент использования (в долях от 1000)
	<i>SA</i>	$2^{31}-1$	Среднее содержимое памяти
	<i>SC</i>	$2^{31}-1$	Счетчик входов
	<i>ST</i>	$2^{31}-1$	Среднее время транзакта
	<i>SM</i>	$2^{31}-1$	Максимальное содержимое памяти

Продолжение таблицы Г

Переменные	$V$	$\pm(2^{31}-1)$	Значение арифметической переменной
	$FV$	$10^{-78} \dots 10^{75}$	Переменные с плавающей запятой
	$BV$	1 или 0	Значение булевой переменной
Функции	$FN$	$\pm(2^{31}-1)$	Значение функции
Очереди	$Q$	$2^{31}-1$	Текущая длина очереди
	$QA$	$2^{31}-1$	Средняя длина очереди
	$QM$	$2^{31}-1$	Максимальная длина очереди
	$QC$	$2^{31}-1$	Общее число элементов в очереди
	$QZ$	$2^{31}-1$	Число элементов, не задержанных в очереди
	$QT$	$2^{31}-1$	Среднее время пребывания транзакта в очереди
	$QX$	$2^{31}-1$	Среднее время транзакта в очереди без учета нулевых входов
Таблицы	$TB$	$\pm(2^{31}-1)$	Среднее значение аргумента таблицы
	$TC$	$2^{31}-1$	Счетчик входов в таблицу
	$TD$	$2^{31}-1$	Среднее квадратичное отклонение аргумента
Матрицы	$MX(a, b)$	$\pm(2^{31}-1)$	Текущее содержание матрицы (формата $X$ – «слово», $H$ «полуслово», $B$ – «байт», $L$ – «плавающая точка») строка– $a$ , столбец– $b$
	$MH(a, b)$	$\pm(2^{15}-1)$	
	$MB(a, b)$	$\pm(2^7-1)$	
	$ML(a, b)$	$\pm(2^{24}-1)$	
Ячейки	$XF$	$\pm(2^{31}-1)$	Текущее содержание матрицы (формата $F$ – «слово», $H$ «полуслово», $B$ – «байт», $L$ – «плавающая точка»)
	$XH$	$\pm(2^{15}-1)$	
	$XB$	$\pm(2^7-1)$	
	$XL$	$\pm(2^{24}-1)$	

Продолжение таблицы Г

Группы	$G$	$2^{15} - 1$	Число членов группы в текущий момент времени
Список пользователя	$CA$	$2^{15} - 1$	Среднее число элементов в списке
	$XF$	$2^{15} - 1$	Текущее число элементов в списке
	$XH$	$2^{15} - 1$	Максимальное число элементов в списке
	$XB$	$2^{31} - 1$	Общее число элементов в списке
	$XL$	$2^{31} - 1$	Среднее время пребывания элементов в списке
Системные атрибуты	$RN1...RN8$	$0...0,999999$	Случайное число, используемое как аргумент функции
		$0...999$	Случайное число в других случаях
	$C1$	$2^{31} - 1$	Текущее значение относительного времени
	$AC1$	$2^{31} - 1$	Текущее значение абсолютного времени
	$TG$	$2^{31} - 1$	Содержание счетчика завершения

## Список использованных источников

1. Шрайбер Т.Дж. Моделирование на GPSS. М. Машиностроение. 1980.
2. Руководство пользователя по GPSS World. /перевод с английского/. – Казань: Изд-во «Мастер Лайн», 2002. – 384 стр.
3. Учебное пособие по GPSS World. /перевод с английского/. – Казань: Изд-во «Мастер Лайн», 2002. – 272 стр.
4. Олифер В.Г., Олифер Н.А. Компьютерные сети. Принципы, технологии, протоколы. Учебник для ВУЗов. 5-е издание. Издательский дом – Питер. 2016. 992 стр.
5. Нерсисянц А.А. Изучение способов анализа структурных элементов инфокоммуникационных систем методами имитационного моделирования в системе GPSS World. Методические указания к выполнению лабораторного практикума. Ростов-на-Дону: ЮФУ, 2013. – 23 с.
6. Нерсисянц А.А. Изучение маршрутизатора сети коммутации пакетов методами имитационного моделирования. Методические указания к выполнению лабораторного практикума. Ростов-на-Дону: ЮФУ, 2013. – 9 с.
7. Оптимизация структурных параметров в сетях с коммутацией пакетов. Методические указания по выполнению лабораторной работы. Ростов-на-Дону: МТУСИ СКФ, 2017. – 12 с.