

Министерство образования и науки Российской Федерации

---

САНКТ-ПЕТЕРБУРГСКИЙ  
ГОСУДАРСТВЕННЫЙ ПОЛИТЕХНИЧЕСКИЙ  
УНИВЕРСИТЕТ

---

В.С. Заборовский, А.С. Ильяшенко, В.А. Мулюха

**ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ  
ТЕЛЕМАТИЧЕСКИХ СИСТЕМ**

Учебное пособие

Санкт-Петербург  
Издательство Политехнического университета  
2014

УДК 004.414.23

ББК 32.973.202

З 12

Заборовский В.С., Ильяшенко А.С., Мулюха В.А. Имитационное моделирование телематических систем. Учеб. пособие. СПб: Изд-во СПбГПУ, 2013. 58 с.

Данное учебное пособие содержит описание методов имитационного моделирования телематических систем, сетей и устройств в среде GPSS, примеры описания типовых систем при помощи теории массового обслуживания, а также курс практических заданий по освоению рассмотренных инструментальных средств. Включает описание возможностей программы GPSS для моделирования систем массового обслуживания различной сложности и содержит методические рекомендации и материалы к циклу лабораторных работ студентов.

Практические задания в учебном пособии предназначены для освоения материала курса «Моделирование» студентами специальностей «Информатика и вычислительная техника», «Сети ЭВМ и телекоммуникации».

Табл. 1. Ил. 15. Библиогр.: назв. 9

© Санкт-Петербургский государственный  
политехнический университет, 2014

© Заборовский В.С., Ильяшенко А.С.  
Мулюха В.А., 2014

## ОГЛАВЛЕНИЕ

<b>СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ</b> .....	<b>5</b>
<b>ВВЕДЕНИЕ</b> .....	<b>6</b>
<b>1. ТЕЛЕМАТИЧЕСКИЕ СИСТЕМЫ</b> .....	<b>9</b>
1.1. ПОНЯТИЕ ТЕЛЕМАТИЧЕСКАЯ СИСТЕМА .....	9
1.2. ПОНЯТИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ.....	14
1.3. ИСПОЛЬЗОВАНИЕ СРЕДСТВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ ОПИСАНИЯ ТЕЛЕМАТИЧЕСКИХ СИСТЕМ .....	15
1.4. ИСПОЛЬЗОВАНИЕ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ ДЛЯ МОДЕЛИРОВАНИЯ ТЕЛЕМАТИЧЕСКИХ СИСТЕМ .....	18
<b>2. СРЕДА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD</b> .....	<b>23</b>
2.1. ОБЩЕЕ ОПИСАНИЕ СРЕДЫ МОДЕЛИРОВАНИЯ GPSS WORLD..	23
2.2. GPSS-МОДЕЛЬ .....	23
2.3. ТРАНЗАКЦИИ И МОДЕЛЬНОЕ ВРЕМЯ.....	25
2.4. НАЧАЛО РАБОТЫ СО СРЕДОЙ GPSS WORLD.....	28
2.5. СИСТЕМНЫЕ ЧИСЛОВЫЕ АТТРИБУТЫ .....	32
2.6. ВСТРОЕННЫЕ РАСПРЕДЕЛЕНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН .....	33
2.7. ОСНОВНЫЕ БЛОКИ GPSS .....	34
<b>3. МОДЕЛИРОВАНИЕ ТЕЛЕМАТИЧЕСКИХ СИСТЕМ С     ИСПОЛЬЗОВАНИЕМ GPSS WORLD</b> .....	<b>59</b>
3.1. МОДЕЛИРОВАНИЯ ЛОКАЛЬНОГО КОНЦЕНТРАТОРА .....	59
3.2. МОДЕЛИРОВАНИЯ ЛОКАЛЬНОГО КОММУТАТОРА.....	60
3.3. МОДЕЛИРОВАНИЯ МАГИСТРАЛЬНОГО МАРШРУТИЗАТОРА...	61
3.4. МОДЕЛИРОВАНИЕ ПРИЛОЖЕНИЙ С РАЗЛИЧНЫМИ ТИПАМИ ТРАФИКА.....	62

3.5. МОДЕЛИРОВАНИЕ ГРИД СИСТЕМ.....	64
3.6. МОДЕЛИРОВАНИЕ ПРИКЛАДНЫХ ТЕЛЕМАТИЧЕСКИХ СЕРВИСОВ.....	65
3.7. МОДЕЛИРОВАНИЕ ЛОКАЛЬНЫХ СЕТЕЙ С ОТКАЗАМИ .....	66
3.8. МОДЕЛИРОВАНИЕ ОБЛАЧНЫХ ТЕЛЕМАТИЧЕСКИХ СЕРВИСОВ.....	68
<b>СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....</b>	<b>71</b>

## **СПИСОК ИСПОЛЬЗУЕМЫХ СОКРАЩЕНИЙ**

ПО – Программное Обеспечение

ЭВМ – Электронная Вычислительная Машина

ARP – Address Resolution Protocol

BGP – Border Gateway Protocol

DARPA – Defense Advanced Research Projects Agency

ICMP – Internet Control Message Protocol

ID – Identification

IP – Internet Protocol

GPSS – General Purpose Simulation System

LAN – Local Area Network

NS2 – Network Simulator – 2

OSPF – Open Shortest Path First

RTCP – Realtime TCP

RTO – Retransmission TimeOut

RTT – Round Trip Time

SAMAN – Simulation Augmented by Measurement and Analysis for Networks

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

WDM – Wavelength Division Multiplexing

WiMAX – Worldwide Interoperability for Microwave Access

## **ВВЕДЕНИЕ**

Развитие сетевых технологий в направлении широкого использования ресурсоемких приложений и новых методов организации информационного взаимодействия предъявляет повышенные требования к характеристикам компьютерных сетей (КС) как защищенной среды доступа к информационным ресурсам. Таким образом, необходимо постоянное совершенствование моделей, средств и методов защиты информации в КС.

Анализ возможностей эффективного применения современных компьютерных сетей в условиях постоянного усложнения их топологии, а также появления новых телематических устройств, технологий передачи данных, протоколов и приложений является сложной научно-технической задачей. Так как современные компьютерные сети представляют собой объекты, характеризующиеся высокой степенью виртуализации, естественной распределенностью, реконфигурируемостью транспортной инфраструктуры и децентрализованностью управления проведение «натурных» экспериментов для изучения протоколов и особенностей функционирования телематических устройств вызывает организационно-технические сложности. В этих условиях совместное применение средств математического и имитационного моделирования позволят получить объективные данные о свойствах трафика и сетевых процессов в сложных телематических системах.

Особенности функционирования современных компьютерных сетей нельзя изучать без совместного рассмотрения свойств сетевой инфраструктуры, телематических устройств, а также используемых протоколов и прикладных сервисов. В основе телематики лежит совместный анализ возможностей передачи данных по телекоммуникационным каналам связи и осуществления транзакций в виртуальном пространстве для получения запрашиваемых информационных услуг.

Термин телематика появился благодаря переводу французского слова *télématique*, которое появилось из соединения слов *télécommunication* и *informatique*, и было предложено С. Норой и А. Минк в 1978 году в отчете под названием «Информатизация общества». Сфера телематики охватывает широкий и постоянно развивающийся спектр сетевых услуг и механизмов доступа к информационным ресурсам, включая службы электронной почты, цифровых данных, факсимильных, аудио- и видеосообщений.

Современные глобальные информационные сети реализованы на базе стека протоколов TCP/IP, именно поэтому Интернет можно считать типичным представителем телематических сетей или современной средой доступа к различным телематическим услугам.

В данном пособии изучаются методы описания и моделирования различных телематических устройств, используемых в современных магистральных сетях передачи данных. Учитывая широкое распространение телематических сервисов изучение механизмов сетевого информационного взаимодействия и средств математического и имитационного моделирования процессов в высокоскоростных компьютерных сетях является важной составляющей подготовки бакалавров и магистров по направлению «Информатика и вычислительная техника» и «Сети ЭВМ и телекоммуникации». Освоение материала учебного пособия позволит студентам глубже изучить возможности сетевых технологий, в том числе связанные с обработкой различных данных в телематических системах.

Исследование моделей, представленных в данном пособии, выполнено при финансовой поддержке РФФИ в рамках научного проекта № 13-07-12106 офи\_м.

Авторы выражают благодарность сотрудникам, студентам и аспирантам кафедры «Телематика (при ЦНИИ РТК)» Института прикладной математики и

механики без помощи которых написание данного учебного пособия было бы невозможно.



# 1. ТЕЛЕМАТИЧЕСКИЕ СИСТЕМЫ

## 1.1. ПОНЯТИЕ ТЕЛЕМАТИЧЕСКАЯ СИСТЕМА

Компьютеры и цифровые телекоммуникации преобразовали жизнь человека. Прошедшие изменения коснулись, прежде всего, «размерности» пространства существования человека, в котором отчетливо обозначилась информационная составляющая, образованная, в первую очередь, сетью Интернет и другими средствами массовой телекоммуникации. К настоящему времени Интернет превратилась в глобальную самоорганизующийся информационно-коммуникационную систему, которая вобрала в себя практически весь «информационный пепел», созданный за всю историю Человечества.

Одновременно с сетью Интернет появился термин телематика, который обычно встречается в литературе при описании услуг операторов связи, характеристик электронных устройств автомобилей, протоколов защиты информации или систем передачи мультимедиа данных. В результате, телематическими стали называть все телекоммуникационные услуги, которые не могут быть предоставлены на основе средств и технологий классической телефонии. Однако размытость представлений о спектре информационных приложений, которые соответствует этому термину, свидетельствует о том, что телематика как новая сфера информационных технологий еще находится на стадии становления, поэтому нуждается в теоретическом обосновании и дальнейшем исследовании. Именно этим вопросам посвящено настоящее учебное пособие.

Кратко сущность телематики можно охарактеризовать как «транзакции в информационном пространстве». Поясним, что это значит. С технической точки зрения, транзакции это целевые операции, выполнение которых не

нарушает целостность системы, в которой они производятся (как правило, это базы данных или файловые системы). Целостность достигается за счет того, что все производимые операции являются взаимозависимыми и либо все успешно завершаются, либо полностью и успешно отменяются. Телематические транзакции выполняются в Интернет, которая представляет собой сеть коммутации пакетов, поддерживающая информационные процессы с помощью стека протоколов TCP/IP. Используя каскадную организацию различных сегментов канальной инфраструктуры, связывающей сетевые узлы, транспортные протоколы формируют виртуальные соединения – логическую последовательность сетевых пакетов, каждый из которых имеет в своем заголовке специальную структуру данных «сокет», составленную из адреса сетевого узла и номера порта информационного приложения. Сокет и виртуальное соединение – это ключевые абстракции, которые положены в основу телематических сервисов, реализуемых на базе стека TCP/IP. Наряду с ними, существенную роль в реализации телематических сервисов играет концепция динамической маршрутизации пакетов через сеть, надежность функционирования узлов и каналов в которой не гарантируется. Эта концепция, дополненная принципом выполнения операций на основе транзакций, позволяет обеспечить непрерывную логическую связанность любых двух узлов, входящих в инфраструктуру сети и, следовательно, гарантировать завершенность проводимых телематических операций. Однако логическая связанность узлов сети не гарантирует определенных значений других важных характеристик информационных потоков, например пропускной способности или задержки в передаче пакетов. Дело в том, что на уровне протоколов сетевого уровня гарантия доставки данных достигается за счет адаптации пропускной способности виртуальных соединений к текущему состоянию сетевой среды, в том числе к отказу в передаче пакетов через тот или иной узел. Динамическая маршрутизация позволяет «внутри» уже существующего

транспортного соединения менять маршрут (трассу) прохождения пакетов, выбирая для коммутации только те узлы, расстояние между которыми в соответствии с используемой метрикой, минимально. Адаптация значения пропускной способности виртуальных соединений позволяет теоретически гарантировать доставку данных даже в том случае, когда некоторые узлы сети выходят из строя. Топологическая связанность на сетевом уровне являются функциональным инвариантом на базе которого создается телематический класс информационных сервисов, связывающих различные приложения, которые нуждаются в различных значениях пропускной способности, среднего времени или дисперсии (джиттере) задержки, уровни изохронности в передаче пакетов данных.

В телематических системах можно выделить несколько системных уровней:

- техническое обеспечение (маршрутизаторы, программное обеспечение, протоколы взаимодействия, и т. д.);
- коммуникационные возможности (структуры данных, типы информации, режимы взаимодействия и т. п.);
- системные службы (электронная почта, телеконференции, контроль доступа и т. п.);
- прикладные задачи (управление транспортными системами, предоставление сервиса в среде облачных вычислений, работа проектных групп и др.).

Телематические услуги (сервисы) для каждого конкретного потребителя определяются спецификой прикладной области и особыми требованиями к реализации коммуникационных задач. Среди телематических сервисов можно выделить: создание VPN сетей, организацию корпоративной системы голосовой почты, создание систем, обеспечивающих безопасную

многопользовательскую среду разработки или groupware, дистанционное обучение, телемедицину, управление транспортом, и пр.

Сфера телематики охватывает постоянно развивающийся спектр услуг доступа к информационным ресурсам, формируемым средствами контроля и телеуправления сложным оборудованием, хранение и обработка которых осуществляется при помощи среды облачных вычислений класса IaaS и PaaS.

Другое направление развития телематики – предоставление услуг по предоставлению вычислительных мощностей для физического размещения приложений на сервере (хостинг приложений) или услуги класса SaaS. В телематических системах этого класса одна часть приложения находится у корпоративного клиента, другая – у оператора, а третья – у фирм, которые предоставляют внешнее сопровождение (аутсорсинг) определенных услуг, например защиту и контроль доступа к базам данных CRM систем.

Реализация телематических услуг обычно осуществляется в режиме по требованию, поэтому может рассматриваться как технология комбинирования разных типовых сетевых сервисов, из которых оператор может выбрать только те, которые в данный момент требуются потребителю.

На сегодняшний день на телекоммуникационном рынке соседствуют два подхода к реализации телематических услуг. Одни операторы предоставляют телематические услуги, опираясь на традиционные инфраструктуры: выделенные каналы связи, построенные по классическим телекоммуникационным технологиям, сети, построенные на передаче радиосигнала и др.

Другие операторы предоставляют весь спектр телематических услуг поверх единой транспортной среды на основе IP-технологий.

Резкой границы и, тем более, противостояния между этими подходами нет, однако каждый из этих способов для потребителя имеет специфику. Подход традиционных операторов позволяет, во-первых, добавлять в пакет услуг новые

составляющие, реализуемые поверх различных транспортных сред, исходя из потребностей конкретного клиента, и, во-вторых, обеспечивать наивысший уровень качества. IP-провайдеры реализуют услуги в единой аппаратно-программной среде, что упрощает интеграцию составляющих комплексной телематической услуги. Кроме того, стоимость услуг, опирающихся на IP-технологии, в большинстве случаев остается существенно ниже услуг традиционных операторов.

Основное преимущество IP перед классическими технологиями основано на том, что передача любой информации осуществляется в едином транспортном потоке. В результате, услуги, которые получает абонент, легко комбинируются, масштабируются и изменяются. IP-технология экономит полосу пропускания за счет объединения всех видов трафика в один поток и использования разнообразных механизмов компрессии и технологий мониторинга, важных для реализации систем учета и контроля доступа.

Для того, чтобы количественно оценить возможности применения телематических сервисов в пособии предлагается использовать методы имитационного моделирования, основанные на теории массового обслуживания. Применение этих методов позволяет исследовать особенности организации виртуального канала и выбрать такие характеристики настройки параметров транспортных протоколов, чтобы задержки пакетов для всех приложений не превосходили максимально допустимого значения RTT (round trip time).

Другим приложением, методов моделирования, которые изучаются в данном учебном пособии, является отработка сетевых технологий, применяемых в различных сферах промышленности и научных исследований, в частности, при создании мобильных, беспроводных или сенсорных сетевых структур. Большое внимание уделяется вопросам имитационного моделирования, позволяющим анализировать процессы реализации транзакций

в сетевой среде, в которой для управления данными используются вероятностные механизмы приоритетного контроля длины очереди пакетов в буфере маршрутизатора, что важно для динамической настройки параметров алгоритмов сило-моментных воздействий при «очувствлении» задержек в канале передачи управляющих воздействий.

Все примеры и задачи заимствованы из исследований, проводимых специалистами кафедры «Телематика» СПбГПУ и ЦНИИРТК совместно с Институтом мехатроники и робототехники Германского аэрокосмического агентства (RM DLR) на Российском сегменте международной космической станции в рамках программы космического эксперимента «Контур-2».

## **1.2. ПОНЯТИЕ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ**

Моделирование – это один из самых распространенных методов решения задач, при использовании которого исследуемая система заменяется более простым объектом, называемым моделью и описывающим реальную систему с точки зрения исследуемых критериев и характеристик.

Моделирование применяется в случаях, когда проведение экспериментов с реальной системой невозможно или нецелесообразно: например, по причине хрупкости или дороговизны создания прототипа либо из-за длительности проведения эксперимента в реальном масштабе времени.

Различают физическое и математическое моделирование. Например, при использовании метода физического моделирования может быть создана уменьшенная или увеличенная копия объекта. Математическая модель – это некоторый эквивалент исследуемой системы или объекта, отражающий в математической форме важнейшие, с точки зрения исследователя, свойства, законы, которым он подчиняется, а также связи, присущие составляющим его частям. Особым видом математических моделей являются имитационные модели.

В настоящее время наиболее полным определением имитационной модели является следующее: имитационная модель – это компьютерная программа, которая описывает структуру и/или воспроизводит поведение реальной системы во времени. Имитационная модель позволяет получать подробную статистику о различных аспектах функционирования системы в зависимости от входных данных. Имитационное моделирование принято использовать в случаях, когда в силу сложности исследуемого объекта или большого числа взаимозависимых факторов, влияющих на состояние объекта, невозможно получить его аналитическую модель.

Имитационное моделирование на цифровых вычислительных машинах является одним из наиболее мощных средств исследования, в частности, сложных динамических систем. Как и любое моделирование, оно дает возможность проводить вычислительные эксперименты с проектируемыми системами и изучать системы, натурные эксперименты с которыми, из-за соображений безопасности или дороговизны, не целесообразны. В тоже время, благодаря своей близости по форме к физическому моделированию, это метод исследования доступен более широкому кругу пользователей.

В настоящее время, когда компьютерная промышленность, предлагает разнообразнейшие средства моделирования, любой квалифицированный инженер, технолог или менеджер должен уметь уже не просто моделировать сложные объекты, а моделировать их с помощью современных технологий, реализованных в форме специализированных сред разработки, графических сред или пакетов визуального моделирования.

### **1.3. ИСПОЛЬЗОВАНИЕ СРЕДСТВ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ ДЛЯ ОПИСАНИЯ ТЕЛЕМАТИЧЕСКИХ СИСТЕМ**

Среди средств моделирования компьютерных сетей в первую очередь необходимо выделить сетевой симулятор «Network Simulator-2» (NS-2),

который уже много лет успешно применяется на стадиях разработки новых сетевых технологий и отработки процессов информационного взаимодействия в сложных технических системах.

Сетевой симулятор NS2 (далее, симулятор) представляет собой программное средство для моделирования и анализа функционирования цифровых сетей с коммутацией пакетов. Первая версия симулятора появилась в 1989 году, когда быстрое распространение технологии межсетевого взаимодействия (internetworking) вызвало необходимость исследования и совершенствования протоколов составных сетей. Широкие возможности симулятора для исследования корректности и эффективности протоколов различных уровней, а также для моделирования разнородных приложений способствовали его быстрому распространению. С 1995 года совершенствование симулятора поддерживается агентством перспективных исследований министерства обороны (DARPA) США и национальным научным фондом США.

Симулятор предназначен для использования, как в исследовательских целях – для оценки влияния различных факторов на эффективность разрабатываемых протоколов и приложений, так и в учебных целях – для пояснения работы конкретных протоколов и алгоритмов управления процессами в сетях.

В зависимости от целей исследования, процессы в сети могут моделироваться на различных уровнях взаимодействия, с учетом особенностей традиционных и перспективных приложений, протоколов и технологий (WWW, Multicast, Mobile Networking, Satellite Networking, LAN и др.).

Основными особенностями симулятора NS2 являются:

- Модульный принцип построения и открытая архитектура симулятора, соответствующие многоуровневой архитектуре сетей и позволяющие легко



расширять его функциональные возможности путем добавления новых модулей и модификации имеющихся.

- Широкий диапазон методов и средств абстрагирования, предоставляющих возможности изменения уровня абстракции, как при анализе результатов, так и, непосредственно, при моделировании, что позволяет идентифицировать существенные эффекты на уровне макромоделей, а затем применять детальное моделирование для их более тщательного исследования.

- Наличие средств анимации, обеспечивающих возможность наблюдения конкретных реализаций и позволяющих выделить наиболее важные и интересные моменты поведения сетевой модели. (Отметим, что при рассмотрении только агрегированных статистических оценок много существенных явлений остается незамеченными).

- Наличие библиотеки сетевых топологий и генераторов трафика, облегчающих создание моделей сетей со сложной топологией и смешанной нагрузкой.

- Состав смешанного Интернет-трафика постоянно меняется (в настоящее время преобладают WWW-приложения, в ближайшем будущем, возможно, основную долю составят аудио- и видеоприложения). Топология сетей также постоянно изменяется. Важно облегчить исследователям возможность опережающего исследования влияния этих тенденций на эффективность работы сетей. Проблема задания топологии и динамики ее изменения особенно значима для мобильных сетей.

- Широкая известность NS2, как эталонного средства моделирования, свидетельствующая о высокой степени достоверности результатов.

Симулятор NS2 относится к свободно распространяемым программным средствам (freeware). На сайте разработчиков [3] доступно несколько версий симулятора, ориентированных на различные операционные системы и аппаратные платформы.

Как правило, совместно с симулятором NS2 используются дополнительные программные средства графического отображения результатов моделирования. В первую очередь, к ним относятся утилита сетевой анимации nam (network animator) и утилита построения графиков X-graph. Кроме того, вместе с симулятором распространяются средства отладки моделей, конвертации выходных данных, генераторы топологий сетей и сценариев моделирования, а также ряд других пакетов. Более полная информация об установке, составе, настройке и использовании отдельных пакетов, а также список основных объектов и функций симулятора содержится в [4].

Использование симулятора NS2 позволяет моделировать микро- и мезо-уровни информационного взаимодействия различных объектов компьютерных сетей, получать характеристики сетевого трафика с учетом специфики потока пакетов от каждого информационного сервиса и пользовательского приложения, а также моделировать пачечность, нестационарность и пиковость трафика на локальных участках сети. Для проведения макро исследований целесообразно использовать другие программные продукты, в основе которых лежит теория массового обслуживания – специализированного раздела теории вероятности, целью которого является анализ структуры системы в целом на основании информации о потоках пакетов.

#### **1.4. ИСПОЛЬЗОВАНИЕ ТЕОРИИ МАССОВОГО ОБСЛУЖИВАНИЯ ДЛЯ МОДЕЛИРОВАНИЯ ТЕЛЕМАТИЧЕСКИХ СИСТЕМ**

Применяемые в инженерной практике сетевые технологии чрезвычайно разнообразны. Они опережают в своем развитии и практическом внедрении разработку адекватных аналитических моделей и их теоретическое обоснование.

Основным фактором, осложняющим моделирование сетевого трафика, служит случайный характер потерь пакетов в виртуальных соединениях.

Стохастический характер процессов функционирования компьютерных сетей и структура формируемых при этом виртуальных соединений предопределяет возможность использования для их анализа методов и моделей теории массового обслуживания (ТМО).

Моделирование простейшей телематической системы можно осуществить при помощи однопоточковой системы массового обслуживания с простейшим входным потоком с интенсивностью  $\lambda$  и пуассоновским распределением интенсивности обслуживания  $\mu$ . Согласно предельной теореме, сумма многих случайных потоков может быть описана простейшим распределением. В процессе передачи данных телематические устройства, осуществляющие передачу данных, осуществляют одновременную обработку пакетов от десятков, сотен, а зачастую и тысяч различных соединений, таким образом использование пуассоновского закона при описании входного потока заявок и интенсивности обслуживания представляется адекватным. Очередь ограниченного размера  $k$  позволит моделировать память телематических устройств, осуществляющих передачу данных.

Однако анализ сетевых процессов на основе однопоточковых и однофазных моделей систем массового обслуживания (СМО), характерный для раннего этапа теоретического анализа сетевых взаимодействий, в настоящее время уже не отражает должным образом особенностей современных компьютерных сетей. Упомянутые выше модели позволяют получить лишь самые общие качественные характеристики процессов сетевого взаимодействия в целом, но не могут использоваться для оценки пропускной способности и задержек в каналах связи при передаче мультимедиа трафика, образованного данными от различных источников информации с определенными требованиями качества обслуживания. В этих условиях многопоточковые модели СМО позволяют более точно описывать сложные процессы передачи трафика в современных компьютерных сетях. При этом повышается не только детальность, точность и

надежность результатов, но и открываются возможности решения принципиально новых задач, важнейшими из которых являются задачи управления.

В качестве примера описания подобного телематического устройства можно рассматривать простейшую многопоточковую модель, учитывающую два потока данных. Приоритетные пакеты моделируют трафик с требованием высокого качества сервиса, неприоритетные – весь остальной трафик, который можно охарактеризовать как фоновый.

В рассматриваемой модели управление пакетными коммутациями осуществляется при помощи двух механизмов: во-первых, абсолютного приоритета в обслуживании пакетного трафика, сформированного разрешенными данными, и, во-вторых, вероятностного выталкивающего механизма, которым снабжена буферная память телематических устройств. В результате пакеты приоритетного типа с некоторой вероятностью  $0 \leq \alpha \leq 1$  способны выталкивать из буфера межсетевого экрана неприоритетные пакеты. С помощью выбора величины  $\alpha$  появляется возможность адаптивной настройки алгоритма управления трафиком. Это обстоятельство позволяет эффективно перераспределять доступную для различных виртуальных соединений пропускную способность сети, управляя, тем самым, взаимной пропускной способностью приоритетных и неприоритетных потоков данных.

Для примера рассмотрим одноканальную СМО ограниченной емкости  $k$  ( $1 < k < \infty$ ), на вход которой поступают два независимых простейших потока заявок (пакетов) интенсивностью  $\lambda_1$  и  $\lambda_2$ . Допустим, что длительность обслуживания (обработки любого пакета) распределена по одному и тому же показательному закону. Все эти длительности независимы в совокупности – обработка каждого из пакетов является транзакцией, что позволит нам не учитывать время дообслуживания вытесненного пакета. Интенсивность обслуживания пакетов обоих типов обозначается через  $\mu$ .

Пакетам первого типа предоставлен абсолютный приоритет в обслуживании. Пока все приоритетные запросы на обслуживание не будут обработаны, пакеты второго типа остаются в очереди. Вновь поступившие приоритетные запросы прерывают обработку неприоритетных и с вероятностью равной единице вытесняют их в накопитель (если есть свободные места ожидания), либо за пределы системы (если свободных мест нет). Вытесненные с обслуживания пакеты присоединяются к очереди неприоритетных требований и могут быть дообслужены после всех приоритетных. Накопитель является общим, свободные места ожидания полностью доступны для любого вновь поступившего запроса. Динамически формируются две отдельные очереди пакетов, их суммарная длина ограничена емкостью накопителя ( $k - 1$ ).

В отличие от типовых приоритетных СМО рассматриваемая система снабжена вероятностным выталкивающим механизмом. Приоритетный пакет, заставший все места ожидания занятыми в момент обработки другого приоритетного пакета, с заданной вероятностью  $\alpha$  вытесняет из накопителя один из менее приоритетных пакетов и занимает его место. Вытесненный пакет теряется.

Схема телематического устройства, моделируемого при помощи приоритетной системы массового обслуживания представлена на рисунке 1.

Для моделирования современных высокоскоростных распределенных телематических систем, в которых обработка данных осуществляется на нескольких независимых вычислительных устройствах можно использовать специализированные системы массового обслуживания с различными барьерными механизмами синхронизации данных. Параллельная обработка потока заявок на нескольких обслуживающих приборах с последующим объединением их в единый поток позволит описать подобную систему.

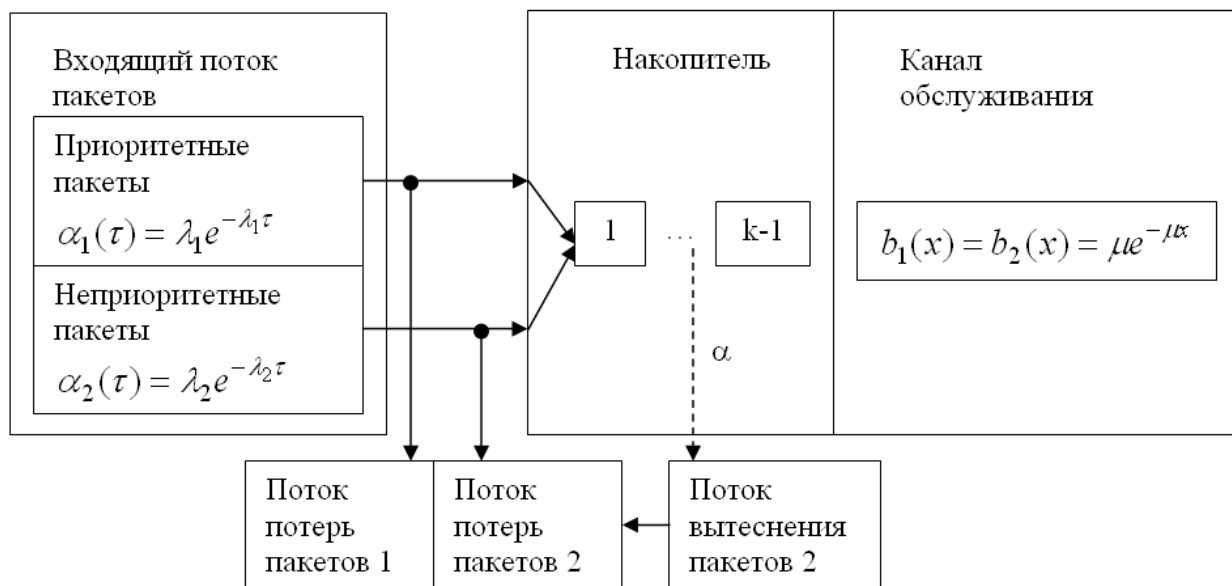


Рисунок 1.1 Схема приоритетной СМО класса  $\vec{M}_2 / M / 1 / k / f_2^1$

## **2. СРЕДА ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ GPSS WORLD**

### **2.1. ОБЩЕЕ ОПИСАНИЕ СРЕДЫ МОДЕЛИРОВАНИЯ GPSS WORLD**

GPSS (англ. General Purpose Simulation System) – система моделирования общего назначения, предназначенная для проведения исследования характеристик и поведения сложных систем с дискретным процессом их функционирования. Основные типы систем, которые могут быть замоделированы с помощью GPSS: системы массового обслуживания, сети массового обслуживания, автоматные модели, сети Петри и подобные системы.

Одной из самых распространенных реализаций систем GPSS является система GPSS World, разработанная для ОС Windows. Среда GPSS World обладает широким набором инструментов для анализа результатов моделирования и позволяет получать графики основных характеристик, а также динамически управлять процессом моделирования. Также, в библиотеках среды имеется ряд различных вероятностных распределений для генераторов случайных функций, что позволяет описывать широкий спектр моделей поведения потоков данных проходящих через моделируемые системы.

### **2.2. GPSS-МОДЕЛЬ**

В состав GPSS-модели входит программа, состоящая из последовательности операторов языка GPSS, описывающих поведение модели в процессе ее функционирования и последующего моделирования. Такие операторы делятся на команды и блоки. Команды предназначены для организации работы с элементами программы – переменными, функциями, а также для организации управления процессом моделирования.

Рассмотрим общий вид оператора GPSS:

<Метка> <Операция> <Операнды>;<Комментарий>

Рассмотрим каждый из элементов оператора:

- 1) <Метка>. В поле «Метка» разработчик может указать имя для приводимого далее оператора блока. Данное поле может быть использовано для организации условного перехода (возврата) к выполнению этого оператора заново;
- 2) <Операция>. Данное поле содержит в себе название одного из операторов языка GPSS. Многие из этих операторов будут далее разобраны. Пока что приведем список примеров: GENERATE, TERMINATE, QUEUE, DEPART, SEIZE, RELEASE, ENTER, LEAVE, ADVANCE, PRIORITY, GATE, ASSIGN, TEST и др.;
- 3) <Операнды>. Большинство операций в GPSS требуют передачи аргументов, требуемых для выполнения действий, связанных с этой операцией. Все операнды в GPSS разделяются или пробелом или запятой. Одновременно поставленные запятая и пробел, говорят интерпретатору, что параметр пропущен и должен быть выбран по умолчанию. Например, <A, B> и <A,B> - это разные наборы операндов. В первом случае задано три операнда, а во втором только два.
- 4) <Комментарий>. Комментирование кода позволяет упростить чтение текста модели и возможно объяснить некоторые хитрости и приемы, которыми пользовался разработчик при ее создании. В GPSS-модели могут быть добавлены комментарии следующим образом:
  - Если комментарий занимает всю строку, то тогда строка должна начинаться с символа «\*»



Пример:

*\*Начинается блок*

Оператор1

Оператор2

Оператор3

*\*Конец блока*

- Если комментарий следует сразу после описания оператора с набором операндов, то его отделяют от операндов символом «;».

Например, можно операторы прокомментировать следующим образом:

Оператор1; *Задержка требования в канале устройства на 10 минут.*

Оператор2; *Выдача товара.*

### **2.3. ТРАНЗАКЦИИ И МОДЕЛЬНОЕ ВРЕМЯ**

Суть процесса моделирования заключается в пропускании через описанную модель объектов, которые перемещаются от блока к блоку и выполняют все указания, описанные в них. Такими объектами являются транзакты. В процессе моделирования транзакты могут быть созданы оператором GENERATE и уничтожены оператором TERMINATE.

Моделирование всегда начинается с момента, когда в системе отсутствуют какие-либо транзакции. Далее в процессе работы они создаются при помощи оператора GENERATE и проходят через блоки модели именно в том порядке, в котором это было описано в модели, переходя из одного блока в другой в порядке описания блоков и осуществляя условные переходы, если это было предусмотрено в модели.

Среда моделирования GPSS в каждый момент времени следит за одним транзактом, который находится в блоке. Одновременно в разных блоках модели

могут находиться различные транзакты. Транзакт, попадая в блок, вызывает ряд действий, описанный этим блоком, после чего попадает в следующий блок, если это возможно. Все действия над транзакциями выполняются мгновенно с точки зрения модельного времени за исключением явных задержек в специальных блоках, а также ожидания определённых событий. Обработка конкретного транзакта на время прекращается, если транзакт входит в блок, который может задержать его внутри себя или в данный момент недоступен или содержит оператор `TERMINATE`, уничтожающий транзакт.

Каждый транзакт, который проходит через модель, имеет ряд атрибутов, которые описывают его текущее состояние в системе. Такими атрибутами могут быть:

- 1) параметры, которые за этим транзактом закрепляет пользователь для своих нужд;
- 2) приоритет транзакта, который используется при определении права на пользование ресурсом в модели;
- 3) время появления транзакта в системе;
- 4) текущий блок транзакта;
- 5) текущее состояние, в котором он сейчас находится (выполняется, ожидает, прерван более приоритетным, удален).

Процесс моделирования протекает во времени, отсчет которого ведется внутри среды моделирования GPSS World и который называется таймером модельного времени. Модельное время меняется только при наступлении какого-либо события, связанного с изменением состояния системы, которые очень удобно использовать для моделирования систем массового обслуживания и сетей массового обслуживания, например, в момент поступления новой заявки в систему, а также завершения обслуживания заявки в канале системы.

Одной из особенностей таймера среды GPSS World является его возможность принимать любые значения времени, которые удобны

пользователю. Единственное, что требуется от пользователя – это использовать для всех параметров системы уникальную единицу измерения времени, т.е. если требуется моделировать в секундах, то все параметры модели должны быть заданы в секундах.

Завершение моделирования происходит по следующим правилам:

- 1) если процесс моделирования идет слишком долго и Вам требуется его прекратить, то в меню Command в процессе моделирования можно вызвать команду HALT, которая незамедлительно прервет процесс;
- 2) предусмотреть в модели условие, при котором GPSS-модель остановит свою работы по команде STOP;
- 3) как только счетчик неуничтоженных транзактов станет меньше или равен нулю.

Счетчик неуничтоженных транзактов выставляется командой START, параметром которого является количество транзактов, которое должно пройти через систему. Команда TERMINATE уничтожает количество транзакций, указанное в качестве ее параметра, и вычитает это число из счетчика. Таким образом происходит процесс естественной остановки процесса моделирования. Также, можно не указывать параметр для оператора TERMINATE, тогда процесс будет длиться до момента, когда пользователь вызовет команду HALT.

Теперь, имея представление о процессе моделирования, можно привести простейший пример рабочей программы, написанной при помощи языка GPSS и провести первый имитационный эксперимент и исследовать процесс функционирования системы.

## 2.4. НАЧАЛО РАБОТЫ СО СРЕДОЙ GPSS WORLD

Для того, чтобы начать работы со средой GPSS World, требуется произвести установку этого пакета на свой персональный компьютер. Для этого требуется загрузить установочный пакет студенческой версии ПО с сайта компании Minuteman Software и пройти стандартную процедуру установки. В случае удачной установки, Вы сможете найти на своем компьютере установленную версию среды GPSS World Student Version и запустить ее.

Теперь все готово, чтобы провести первое моделирование с помощью среды GPSS World. Для этого потребуется уже установленный пакет GPSS World. Если Вы уже установили среду, то для запуска моделирования простейшей модели потребуется сделать следующее:

- 1) Запустить среду GPSS World. При удачном запуске Вы увидите следующее на экране (рис. 2.1).

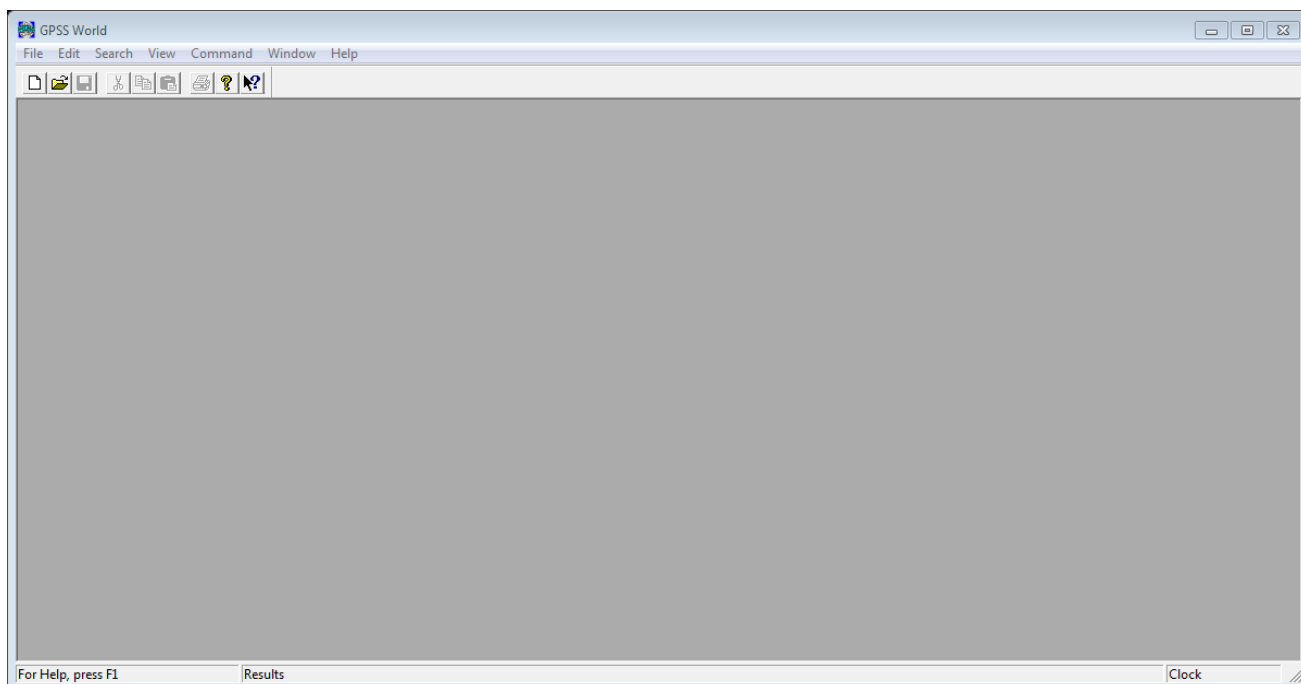


Рисунок 2.1 Окно GPSS World после запуска приложения

- 2) Выбрать в меню File -> New пункт Model, вид которого представлен на рис. 2.2.

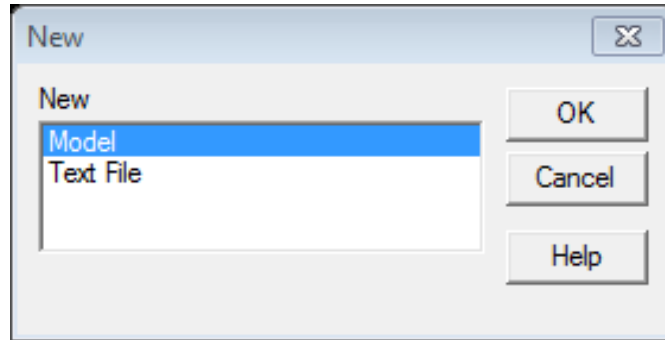


Рисунок 2.2 Вид меню File->New

- 3) Перед Вами появится текстовый редактор, который предназначен для создания и редактирования текста GPSS-модели (рис. 2.3).

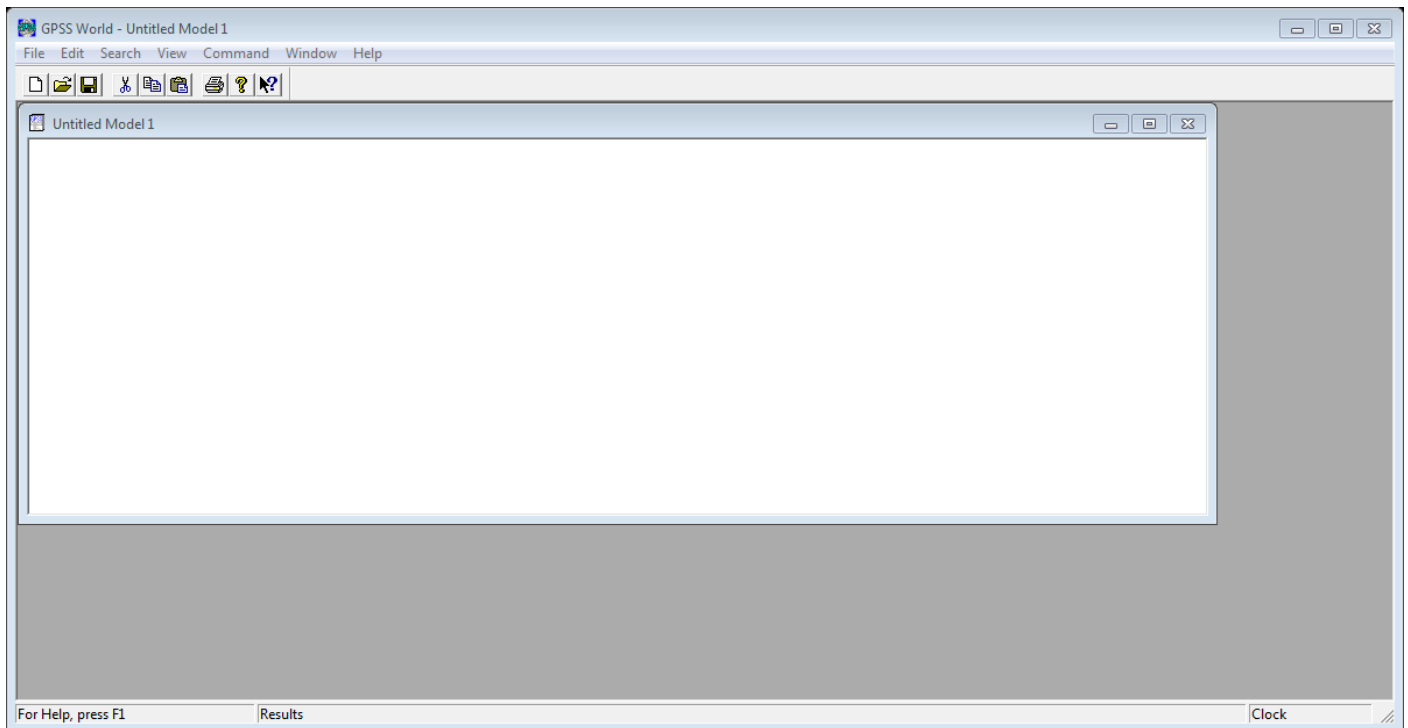


Рисунок 2.3 Окно редактирования модели

Сейчас мы напишем первую и простейшую модель GPSS, которую поместим в окно редактора моделей среды GPSS World и запустим процесс моделирования:

GENERATE 20

TERMINATE 1

START 1000

- 4) После создания модели требуется запустить сам процесс моделирования путем выбора команды меню Command -> Create Simulation.
- 5) По результатам моделирования на экране появляется отчет, который содержит в себе информацию о процессе моделирования и его результатах. Для заданной модели в п.3, отчет будет выглядеть следующим образом (см. рис. 2.4). Далее будет рассмотрено, как при помощи GPSS World получить в качестве результата моделирования не только текстовый отчет, а еще и результаты в графическом виде, с формированием гистограмм.

GPSS World Simulation Report - Untitled Model 1.2.1

Sunday, June 23, 2013 16:13:46

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	20000.000	2	0	0

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1000	0	0
	2	TERMINATE	1000	0	0

FEC XN	PRI	BDT	ASSEM	CURRENT	NEXT	PARAMETER	VALUE
1001	0	20020.000	1001	0	1		

Рисунок 2.4 Вид отчета о результате моделирования

б) Полученный в п.5 отчет позволяет получить сведения о состоянии модели в момент времени, когда процесс моделирования остановился, а также получить количественные сведения о том, как протекал процесс. Для данной модели можно провести анализ отчета по следующим характеристикам:

- Поля START TIME и END TIME – время начала и конца моделирования;
- Поле BLOCKS – количество блоков в модели;
- Колонка LABEL – метки блоков;
- Колонка LOC – номера блоков, задаваемые системой;
- Колонка BLOCK TYPE – имена блоков, если они заданы;
- Колонка ENTRY COUNT – количество транзакций, вошедших в блок за время моделирования;

- Колонка CURRENT COUNT – количество транзакций, содержащихся в блоке в момент окончания моделирования.

## 2.5. СИСТЕМНЫЕ ЧИСЛОВЫЕ АТТРИБУТЫ

Любые переменные и константы, которые разработчик использует в процессе моделирования, называются атрибутами модели. Разработчику в процессе создания модели могут потребоваться дополнительные переменные, которые он с легкостью может создать. Но любая GPSS-модель имеет ряд автоматически поддерживаемых атрибутов, которые требуются для организации управления транзактами и другими объектами, необходимыми для среды в процессе моделирования. Такие атрибуты называются системными числовыми атрибутами. Их значения изменяются в процессе моделирования, и пользователь может легко получать доступ к ним и пользоваться их значениями по своему усмотрению.

Общая структура имени атрибута следующая:

<Имя группы атрибутов><Число> или <Группа атрибутов>\$(Имя объекта)

Например: C1, QA\$Queue\_1, S13 и др.

Числовые атрибуты одноканальных устройств:

- F – проверка занятости устройства;
- FC – Сколько раз было занято устройство транзактами;
- FR – Относительная загрузка устройства;
- FT – Среднее время занятия устройства транзактом.

Числовые атрибуты многоканальных устройств:

- R – Количество свободных каналов;
- S – Количество занятых каналов;



- SA – Среднее количество занятых каналов;
- SR – Относительная загрузка устройства;
- ST – Среднее время пребывания транзакта в устройстве.

Числовые атрибуты очередей:

- Q – Текущая длина очереди;
- QA – Средняя длина очереди;
- QM – Максимальная длина очереди.

## **2.6. ВСТРОЕННЫЕ РАСПРЕДЕЛЕНИЯ СЛУЧАЙНЫХ ВЕЛИЧИН**

В среде GPSS World имеется ряд встроенных распределений случайных величин, которые могут быть использованы для введения случайного фактора в интервалы времени между появлениями транзактов и временами их задержек в каналах устройств. Добавление случайного фактора позволит сделать модели более правдоподобными и рассматривать модели, в которых имеет место факт случайности и распределения случайных величин отличаются от равномерного.

Основные распределения, которые представлены в GPSS World:

- Равномерное(Uniform);
- Экспоненциальное(Exponential);
- Геометрическое(Geometric);
- Пуассона(Poisson);
- Нормальное(Normal);
- Биномиальное(Binomial);
- и другие;

Вероятностное распределение может быть использовано для генерации случайной величины, распределенной по заданному закону, для задания требуемого распределения в модели. Обращение к функции распределения в общем виде выглядит следующим образом:

<Функция распределения>(G,A1,A2,...),

где G-номер генератора равномерно распределенных случайных чисел от 1 до 999, который служит в качестве сема для формирования случайных величин по заданному закону. А остальные аргументы – это аргументы, требуемые для вычисления значения функции распределения. Обычно этими аргументами являются моменты.

Равномерное распределение используется как:

UNIFORM(G, Min, Max),

где Min и Max – это границы для равномерной случайной величины.

Экспоненциальное распределение используется в виде:

EXPONENTIAL(G, Min, Mean),

где Min – это нижняя граница для значения случайной величины, а Mean – это математическое ожидание.

Данные функции активно используются в качестве первого операнда в операторах GENERATE и ADVANCE, однако, они должны быть заключены в скобки, чтобы считаться в качестве одного аргумента, а не набора.

## 2.7. ОСНОВНЫЕ БЛОКИ GPSS

### 2.7.1. Блоки создания и уничтожения транзактов

Данная группа блоков предназначена для описания источников и стоков транзактов модели, то есть они позволяют описывать появление новых транзактов и их уничтожение в модели. Пример использования этих блоков приведен в п. 2.4, когда мы осуществляли знакомство со средой GPSS World.

### 2.7.1.1. GENERATE

Блок GENERATE предназначен для создания транзакций, которые будут проходить по модели. С помощью этого блока предоставляется возможность создавать транзакты с заданными вероятностными распределениями и строить входящие потоки модели в более точном соответствии с реальными потоками данных.

GENERATE [<P1>],[<P2>],[<P3>],[<P4>],[<P5>],

где

P1 – среднее время между появлением транзактов. Можно использовать распределение из библиотеки GPSS World;

P2 – разброс равномерно распределенной случайной величины с мат. ожиданием, заданным в P1;

P3 – момент появления первого транзакта в модели;

P4 – максимальное количество транзактов, которые будут сгенерированы этим блоком;

P5 – номер приоритета транзактов (до 127).

### 2.7.1.2. TERMINATE

Блок TERMINATE предназначен для уничтожения транзактов, по результатам прохождения транзакта по всем блокам модели, которые были предусмотрены разработчиком.

TERMINATE [<P1>],

где P1 – количество, на которое уменьшается счетчик неуничтоженных транзактов, который инициализируется командой START. Если параметр не указан, то транзакт также удаляется, но при этом счетчик не уменьшается.

В качестве примера использования законов распределения можно привести следующую модель генератора заявок, который генерирует 100 транзактов с интервалом времени, распределенным по показательному закону.

```
GENERATE (EXPONENTIAL(1,0,6))
```

```
TERMINATE 1
```

```
START 100
```

Также, можно исследовать не только систем, в которых время моделирование определяется количеством прошедших через модель транзактов. В GPSS World имеется возможность запускать процесс моделирования по заданному времени. Для этого требуется создать два блока, которые ведут себя по-разному. Один будет в штатном режиме обслуживать транзакты, а второй будет создавать только один транзакт в течение заданного времени, которое и требуется для моделирования. Нам придется воспользоваться оператором TERMINATE без параметра, чтобы счетчик уничтоженных транзактов не уменьшался при работе первого блока. Рассмотрим пример такого моделирования.

```
GENERATE 8,2
```

```
SEIZE Device1
```

```
ADVANCE 7,5
```

```
RELEASE Device1
```

```
TERMINATE
```

```
GENERATE 100
```

```
TERMINATE 1
```

```
START 1
```

## 2.7.2. Моделирование каналов обслуживания

Для моделирования одноканальных систем массового обслуживания, требуется обладать инструментами, которые позволяют осуществлять обслуживание транзактов в течение времени, распределенного по произвольному заданному закону. Следующая группа операторов позволяет построить модель, в которой транзакты будут задерживаться в канале обслуживания на заданное время.

### 2.7.2.1. ADVANCE

Данный блок задерживает транзакт на указанное время. В качестве параметра времени также можно использовать случайные величины, распределения которых встроены в GPSS World.

ADVANCE [<P1>],[<P2>],

где P1 – среднее время задержки транзакта в этом блоке;

P2 – разброс равномерно распределенной случайной величины с мат. ожиданием, указанным в P1.

Для организации возможности наблюдения за характеристиками модели не достаточно только блока задерживания транзактов, а еще необходимо получить возможность следить за конкретным каналом обслуживания, с которым мы работаем в данный момент времени. Однако в модели их может быть целое множество. Для этого каждому прибору (в среде GPSS так принято называть одноканальное устройство, которое может быть в двух состояниях – занято/свободно) присваивается идентификатор в виде номера или имени.

### 2.7.2.2. SEIZE

Данный оператор используется для того, чтобы пометить указанное устройство как занятое. Транзакт при прохождении этого блока проверяет

занятость прибора и занимает его в случае, когда прибор свободен. В случае, когда прибор занят, транзакт встает в очередь этого прибора и ожидает вместе с другими транзактами, попавшими в такую же ситуацию, момента времени освобождения прибора.

SEIZE <P1>,

где P1 – имя или номер прибора, которые требуется занять.

### 2.7.2.3. RELEASE

Блок RELEASE противоположность блоку SEIZE. Если SEIZE занимает прибор, то RELEASE освобождает его, тем самым позволяя другим транзактам, стоящим в очереди, занять прибор.

RELEASE <P1>,

где P1 – имя или номер прибора, который требуется освободить;

При помощи приведенных операторов можно строить модели одноканальных систем массового обслуживания, в которых очередь имеет неограниченную длину. Однако, исследователя обычно не интересует только конечное состояние системы, а требуется получить информацию о процессе ее функционирования. В частности для одноканальных систем массового обслуживания с очередями интерес предоставляют характеристики очереди транзактов, которая заполняется в процессе моделирования. Чтобы исследовать очереди используются следующие операторы: QUEUE и DEPART.

### 2.7.2.4. QUEUE

Данный оператор предназначен для сбора статистики о транзактах, которые поступают на обслуживание прибору.

QUEUE <P1>,[<P2>],

где P1 – имя или номер очереди, которая формируется на входе прибора;  
P2 – число, на которое увеличивается количество транзактов в очереди при поступлении нового транзакта.

#### 2.7.2.5. DEPART

Данный оператор предназначен для сбора статистики о транзактах, которые проходят через прибор.

DEPART <P1>,[<P2>],

где P1 – имя или номер очереди, которая формируется на входе прибора;  
P2 – число, на которое уменьшается количество транзактов в очереди при уходе транзакта из нее.

Примером использования этих операторов может стать классическая система массового обслуживания с одним каналом обслуживания (прибором) М/М/1, которой на вход системы поступает простейший поток заявок, а время обслуживания заявки распределено по показательному закону.

GENERATE (Exponential(5,0,10))

QUEUE queue1

SEIZE device

DEPART queue1

ADVANCE (Exponential(25,4,12))

RELEASE device

TERMINATE 1

START 10000

Также приведем отчет на рис. 2.5, который получается при запуске процесса моделирования.

GPSS World Simulation Report - Untitled Model 1.1.1

Monday, July 01, 2013 04:17:15

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	160719.484	7	1	0

NAME	VALUE
DEVICE	10001.000
QUEUE1	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	16023	0	0
	2	QUEUE	16023	6022	0
	3	SEIZE	10001	1	0
	4	DEPART	10000	0	0
	5	ADVANCE	10000	0	0
	6	RELEASE	10000	0	0
	7	TERMINATE	10000	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
DEVICE	10001	1.000	16.066	1	10001	0	0	0	6022

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY	
QUEUE1	6027	6023	16023	1	3004.677	30138.561	30140.442	0

Рисунок 2.5 Отчет о результатах моделирования системы М/М/1

### 2.7.3. Моделирование многоканальных систем

#### 2.7.3.1. STORAGE

Оператор предназначен для описания емкости моделируемого устройства, чтобы указывать количество каналов обслуживания при построении модели многоканальной системы массового обслуживания



<Имя устройства> STORAGE P1,

где <Имя устройства> - имя многоканального устройства, в котором требуется описать несколько каналов;

P1 – число каналов обслуживания.

#### 2.7.3.2. ENTER

Данный блок предназначен для определения точки входа в многоканальное устройство, в момент прохождения которого происходит распределение по каналам обслуживания.

ENTER <Имя устройства>,[<P1>],

где <Имя устройства> - имя многоканального устройства, в котором требуется занять несколько каналов;

P1 – Количество каналов обслуживания, которое может занимать претендующий на обслуживание транзакт.

#### 2.7.3.3. LEAVE

Данный блок предназначен для определения точки выхода из многоканального устройства, в момент прохождения которого происходит освобождение каналов обслуживания.

LEAVE <Имя устройства>,[<P1>],

где <Имя устройства> - имя многоканального устройства, в котором требуется освободить несколько каналов;

P1 – Количество каналов обслуживания, которое освобождает покидающий систему транзакт.

#### 2.7.3.4. Пример моделирования многоканальной системы

Рассмотрим модель многоканальной системы массового обслуживания с количеством каналов равным двум и, в которой входящий поток требований является простейшим, а обслуживающие приборы затрачивают время на обслуживание, которое распределено по показательному закону. Код модели выглядит следующим образом:

```
Device1 STORAGE 2
```

```
GENERATE (EXPONENTIAL(1,0,5))
```

```
QUEUE Queue1
```

```
ENTER Device1
```

```
DEPART Queue1
```

```
ADVANCE (EXPONENTIAL(1,0,7))
```

```
LEAVE Device1
```

```
TERMINATE 1
```

```
START 1000
```

Рассмотрим отчет, полученный в результате моделирования этой системы и приведенный на рис. 2.6.

GPSS World Simulation Report - Untitled Model 1.9.1

Thursday, June 27, 2013 23:21:48

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	4802.719	7	0	1

NAME	VALUE
DEVICE1	10000.000
QUEUE1	10001.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	1002	0	0
	2	QUEUE	1002	0	0
	3	ENTER	1002	0	0
	4	DEPART	1002	0	0
	5	ADVANCE	1002	2	0
	6	LEAVE	1000	0	0
	7	TERMINATE	1000	0	0

QUEUE	MAX CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
QUEUE1	6	0	1002	778	0.190	0.911	4.074 0

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
DEVICE1	3	1	0	3	1002	1	1.539	0.513	0	0

Рисунок 2.6 Отчет о результатах моделирования многоканальной системы

## 2.7.4. Моделирование более сложных моделей массового обслуживания

### 2.7.4.1. TRANSFER

Для построения более сложных моделей, которые отличаются от простых систем массового обслуживания и состоят из нескольких частей, которые взаимодействуют друг с другом путем обмена транзактами, предназначены операторы, обеспечивающие возможность перенаправления транзакций из одного устройства в другое. Одним из таких операторов является оператор

TRANSFER. Этот оператор может быть использован большим количеством способов, которые будут приведены далее.

TRANSFER BOTH,[<Label1>],<Label2> ,

Данный оператор перенаправляет транзакт в блок, отмеченный меткой под номером 1, если в данный момент времени он свободен. Если Label1 не указан, и последующий блок свободен, то транзакт проходит через блок TRANSFER так, как будто его и вовсе нет. Но, если последующий или блок с меток Label1 занят, то транзакт переходит в блок с меткой номер 2, при условии, что тот свободен. Иначе транзакты начинают накапливаться в очереди перед этим блоком.

Рассмотрим пример описания модели одноканальной системы массового обслуживания с конечной очередью:

```
Queue1 STORAGE 3  
GENERATE 6,1  
TRANSFER BOTH,,Off
```

```
ENTER Queue1  
SEIZE Device1  
LEAVE Queue1  
ADVANCE 7,4  
RELEASE Device1
```

```
TERMINATE 1  
Off TERMINATE 1
```

```
START 150
```

## TRANSFER Probability,[<Label1>],<Label2>

Блок TRANSFER при таком использовании распределяет транзакты по блокам с вероятностью, а не по правилу, описанному ранее. С вероятностью 1-Probability транзакт определяется в блок с меткой Label1, а с вероятностью Probability в блок с меткой Label2.

## TRANSFER ALL,<Label1>,[<LabelN>],[NumOfBlocks]

Данный оператор работает по принципу оператора TRANSFER BOTH, только перенаправляет транзакты не на два блока, а на N блоков по тем же правилам. Параметр NumOfBlocks – это количество операторов, на которое отстоят блоки Label1, ..., LabelN друг от друга.

GENERATE 11,4

TRANSFER ALL,Label1,Label3,4

Label1SEIZE Device1

ADVANCE 8,6

RELEASE Device1

TRANSFER ,End

SEIZE Device2

ADVANCE 10,3

RELEASE Device2

TRANSFER ,End

Label3 SEIZE Device3

ADVANCE 14,10

RELEASE Device3

End TERMINATE 1

START 500

TRANSFER PICK,<Label1>,<LabelN>

Данный оператор перенаправляет транзакты на один из блоков с метками от Label1 до LabelN с одинаковой вероятностью, то есть  $1/N$ . Здесь же все блоки должны отстоять друг от друга на 1 блок. Для правильного использования этого оператора мы воспользуемся обычными переходами TRANSFER без каких либо условий и уже в местах, куда направлены переходы, описывать поведение каждого из блоков.

## 2.7.5. Моделирование приоритетных систем

### 2.7.5.1. PREEMPT

Далеко не всегда в моделируемой системе потоки данных имеют одинаковую значимость для исследователя. Иногда возникает задача исследования моделей, в которых есть данные, имеющие особую важность или обладающие особыми привилегиями и возможностями. Например, могут вытеснять менее важные данные из системы или просто обслуживаться раньше других. Для построения таких моделей используется оператор PREEMPT.

PREEMPT <P1>,[<P2>],[<P3>],[<P4>],[<P5>],

где P1 – имя или номер прибора, который будет выполнять приоритетную обработку транзактов;

P2 – условие, разрешающее занять прибор требованию (PR – разрешает занять если приоритет транзакта выше, чем у обслуживаемого, а пустое значение – режим прерывания);

P3 – имя блока, куда будет направлен вытесненный транзакт. По умолчанию транзакт встает обратно в очередь;

P4 – номер параметра, в который записывается оставшееся время обслуживания вытесненного транзакта, которое может быть использовано для систем с дообслуживанием;

P5 – если этот параметр принимает значение RE, то транзакт будет после выталкивания перенаправляться в блок, указанный в P3.

#### 2.7.5.2. RETURN

Когда приоритетный транзакт закончит использование канала обслуживания, то нам потребуется сообщить системе, что канал свободен. Для этого нужен оператор RETURN.

RETURN <Device> ,

где Device – имя устройства, которое освобождается после обработки приоритетного транзакта.

Пример модели с приоритетным выталкиванием:

GENERATE 5,2

SEIZE Facility1

ADVANCE 7,2

RELEASE Facility1

TERMINATE 1

GENERATE 25,2

PREEMPT Facility1,,Lost,,RE

ADVANCE 11,2

RETURN Facility1

Lost TERMINATE 1

START 100

### 2.7.5.3. PRIORITY

Для организации приоритетного обслуживания в системах с несколькими приоритетами у входных данных можно использовать оператор PRIORITY, позволяющий выставлять транзактами уровень приоритета, который в последствии будет использован при определении порядка обработки транзактов в момент выталкивания.

PRIORITY P1,[P2],

где P1 – уровень приоритета;

P2 – параметр определяет порядок обработки приоритетных транзактов в случае, когда на обслуживаемом приборе образовывается очередь из приоритетных транзактов. Если параметр указан как BU, то приоритетный транзакт ставится в конец очереди приоритетных транзактов, в другом случае ставится в начало.

### 2.7.5.4. Система с относительным приоритетом

GENERATE 5,3,,,2; Приоритет равный 2

QUEUE Queue1

SEIZE Device1

DEPART Queue1

ADVANCE 5,4

RELEASE Device1

TERMINATE 1



GENERATE 8,2; Приоритет равный 0(по умолчанию)  
QUEUE Queue2  
SEIZE Device1  
ADVANCE 12,7;  
DEPART Queue2  
RELEASE Device1  
TERMINATE 1

#### 2.7.5.5. Система с абсолютным приоритетом

GENERATE 15,6,,,2  
QUEUE Queue1  
PREEMPT Device1  
DEPART Queue1  
ADVANCE 20,3  
RETURN Device1  
TERMINATE 1

GENERATE 11,6  
QUEUE Queue2  
SEIZE Device1  
ADVANCE 10,9  
DEPART Queue2  
RELEASE Device1  
TERMINATE 1

#### 2.7.6. Моделирование систем с условными переходами

##### 2.7.6.1. TEST

Не всегда в модели можно обойтись безусловными переходами с помощью оператора TRANSFER. Есть возможность организовывать модели, у которых

поведение меняется в зависимости от изменяющихся в процессе моделирования условий. Для этого можно использовать оператор TEST.

TEST <Condition> <Val1>,<Val2>,[<Label>],

где Condition – условие сравнения (E,G,GE,L,LE,NE);

Val1,Val2 – значения, которые требуется сравнить;

Label – метка блока, в который перенаправлять транзакты при невыполнении условия. Если метка не указана, то обслуживание транзакта откладывается до момента, когда условие будет выполнено.

Пример системы с двумя входящими и двумя выходящими потоками, для транзактов которых предусмотрен общий накопитель бесконечной длины.

GENERATE 5,2

ASSIGN 8,1 ; Присвоить параметру №8 значение 1, чтобы отличать заявки друг от друга

TRANSFER ,Work

GENERATE 3,1

ASSIGN 8,2 ; Присвоить параметру №8 значение 2

Work SEIZE Device1

ADVANCE 4,3

RELEASE Device1

TEST E P8,1,Output2; Проверка типа транзакта и соответствующее его перенаправление

TERMINATE 1

Output2 TERMINATE 1

START 1000

#### 2.7.6.2. GATHER

Для некоторых систем требуется выполнение транзакций группами. Накопление транзакций одной группы (типа) можно осуществлять при помощи оператора GATHER

GATHER <Number>,

где Number – количество транзактов, которые требуется накопить перед отправкой в следующий блок.

#### 2.7.6.3. ADOPT

Для накопления транзактов требуется выставлять им номер группы, по которому они будут накапливаться при помощи оператора GATHER. Оператор ADOPT позволяет выставлять этот номер.

ADOPT <GroupNumber>,

где GroupNumber – номер группы.

#### 2.7.6.4. MATCH

Например, для организации систем с барьерными механизмами обработки возможно осуществление синхронизации путем использования блока MATCH с другими таким блоками.

MATCH <Label>,

где Label – метка блока MATCH, с которым требуется синхронизация.

#### 2.7.6.5. Пример системы с синхронизацией обслуживания

Рассмотрим пример модели с синхронизацией обработки транзакций. Модель обслуживает требования только тогда, когда сразу на двух устройствах имеются заявки одной группы. На входе в модель новый транзакт расщепляется оператором SPLIT, и далее транзакты задерживаются на разное время обслуживания. Для завершения обслуживания требуется, чтобы оба транзакта дошли до блока MATCH

```
GENERATE 10,3
```

```
SPLIT 1,Way2;    Дублировать 1 транзакт и направить его на блок Way2
```

```
SEIZE Device1
```

```
ADVANCE 5
```

```
RELEASE Device1
```

```
SyncPoint1 MATCH SyncPoint2; Ожидание синхронизации с блоком  
SyncPoint2
```

```
SEIZE Device2
```

```
ADVANCE 5
```

```
RELEASE Device2
```

```
TRANSFER ,Finish
```

```
Way2 SEIZE Device3
```

```
ADVANCE 9,4
```

```
RELEASE Device3
```

```
SyncPoint2 MATCH SyncPoint1; Ожидание синхронизации с блоком  
SyncPoint1
```

```
SEIZE Device4
```

ADVANCE 5

RELEASE Device4

Finish TERMINATE 1

START 200

Специально для работы с блоками модели существует оператор GATE, позволяющий перенаправлять транзакты в другие блоки аналогично оператору TEST, но в случаях, когда требуется изменить поведение модели при условиях, которые зависят от текущего состояния блока.

GATE <Condition> <Device>,[<Label>],

где Condition – условие проверки;

Device – имя устройства;

Label – блок для перехода в случае ложного результата проверки.

Проверка условия Condition может быть выполнено со следующими значениями:

- 1) FV, FNV – Устройство доступно или нет;
- 2) U,NU – Устройство используется или нет;
- 3) SE,SNE – Пусто или не пусто многоканальное устройство;
- 4) SF,SNF – Заполнено или нет многоканальное устройство;
- 5) SV,SNV – Доступно или нет многоканальное устройство.

#### 2.7.7. Построение графиков по результатам моделирования

Для получения результатов моделирования может служить не только отчет, который показывает статистические характеристики модели, которые

были получены в процессе работы модели. Одной из возможностей системы GPSS World является возможность построения графиков необходимых характеристик модели, которые позволяют получить представление о процессе функционирования системы в целом или отдельного ее компонента. Например, рассмотрим систему массового обслуживания с двумя каналами обслуживания и одним входящим потоком. Все времена обслуживания распределены равномерно, а входящий поток является простейшим. Тогда исходный код модели будет выглядеть следующим образом:

```
GENERATE (EXPONENTIAL(1,0,30))
QUEUE Queue1
TRANSFER BOTH,Chan_1,Chan_2
Chan_1 SEIZE Channel_1
DEPART Queue1
ADVANCE 45,10
RELEASE Channel_1
TRANSFER ,Finish
Chan_2 SEIZE Channel_2
DEPART Queue1
ADVANCE 60,15
RELEASE Channel_2
Finish TERMINATE

GENERATE 5000
TERMINATE 1
*START 1
```

Отчет о результатах моделирования, представленный на рис. 2.7.

GPSS World Simulation Report - Untitled Model 1.9.1

Friday, June 28, 2013 16:58:44

START TIME	END TIME	BLOCKS	FACILITIES	STORAGES
0.000	5000.000	15	2	0

NAME	VALUE
CHANNEL_1	10001.000
CHANNEL_2	10002.000
CHAN_1	4.000
CHAN_2	9.000
FINISH	13.000
QUEUE1	10000.000

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT COUNT	RETRY
	1	GENERATE	178	0	0
	2	QUEUE	178	0	0
	3	TRANSFER	178	8	0
CHAN_1	4	SEIZE	98	0	0
	5	DEPART	98	0	0
	6	ADVANCE	98	1	0
	7	RELEASE	97	0	0
	8	TRANSFER	97	0	0
CHAN_2	9	SEIZE	72	0	0
	10	DEPART	72	0	0
	11	ADVANCE	72	1	0
	12	RELEASE	71	0	0
FINISH	13	TERMINATE	168	0	0
	14	GENERATE	1	0	0
	15	TERMINATE	1	0	0

FACILITY	ENTRIES	UTIL.	AVE. TIME	AVAIL.	OWNER	PEND	INTER	RETRY	DELAY
CHANNEL_1	98	0.865	44.127	1	170	0	0	8	0
CHANNEL_2	72	0.846	58.759	1	171	0	0	8	0

QUEUE	MAX	CONT.	ENTRY	ENTRY(0)	AVE.CONT.	AVE.TIME	AVE.(-0)	RETRY
QUEUE1	9	8	178	42	1.839	51.662	67.616	0

Рисунок 2.7 Отчет о результатах моделирования системы с двумя каналами и общим накопителем

Для построения графиков придется закомментировать команду START. Мы будем запускать моделирование вручную при помощи пункта меню

Command->START, который станет активным после выбора команды Command->Create Simulation. Для запуска выберем команду Command->START и введем команду START 1. При таком способе запуска мы сможем построить график интересующей нас величины.

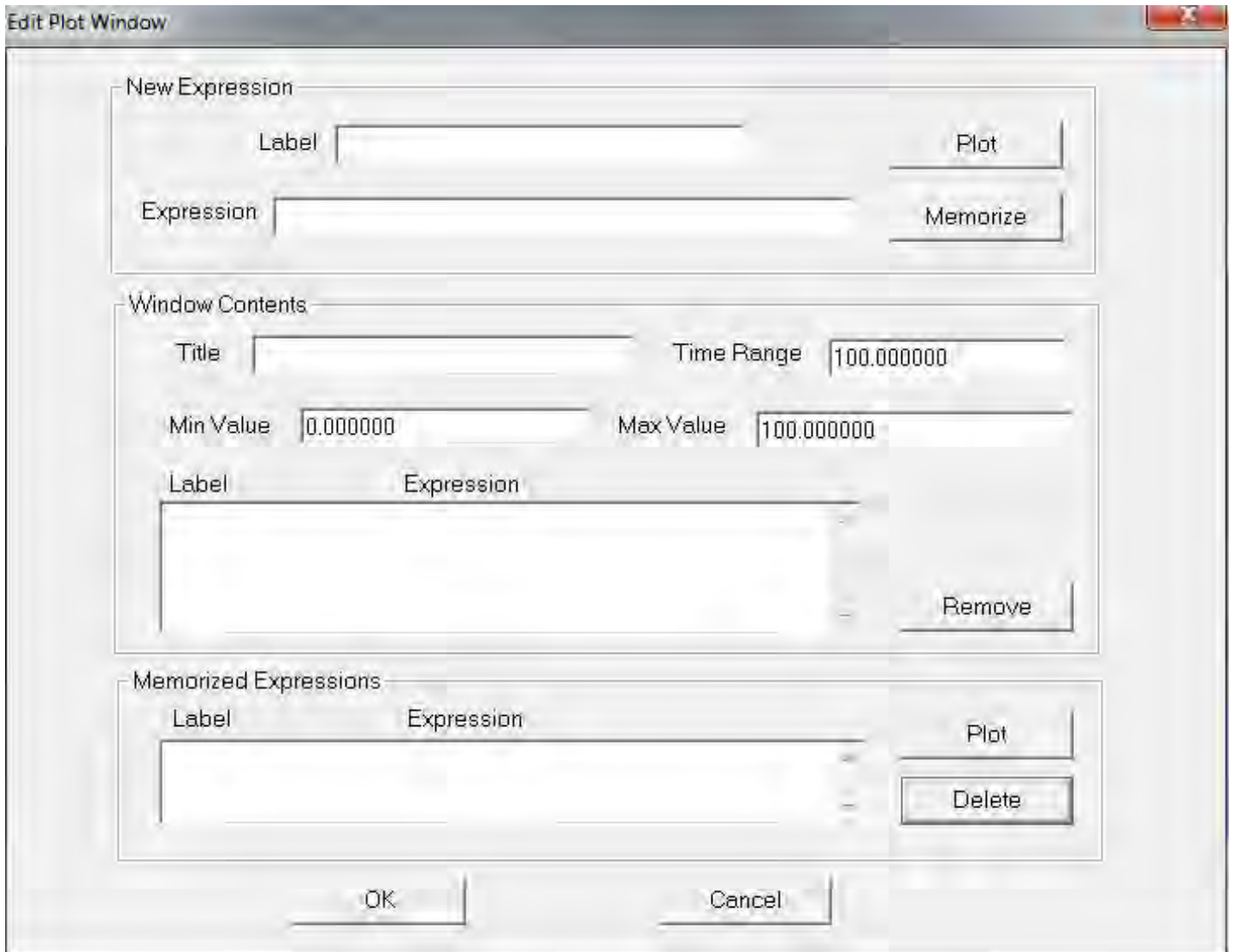


Рисунок 2.8 Окно создания графика величины, полученной в результате моделирования

В данном окне от нас потребуется ввести поле Label – поле с названием графика, который мы хотим построить. Также в поле Expression нужно будет ввести имя переменной, за которой мы будем наблюдать в процессе моделирования. В нашем случае я приведу графики для длины очереди системы



и средней длины очереди. Другие графики строятся аналогичным образом. После ввод всех требуемых данных требуется установить время моделирования в поле Time Range, в течение которого будет производиться построение графика, а также минимальное и максимальное значение измеряемой величины. После чего нужно последовательно нажать Plot и Memorize, чтобы применить все изменения к новому окну с областью рисования графиков.

Для рассматриваемой модели в поле Expression требуется ввести переменную, отвечающую за длину очереди и за максимальную длину очереди. Соответственно это Q\$Queue1 и QM\$Queue1, в соответствии с описанными выше атрибутами очередей из п.2.6. В итоге Вы должны получить следующее окно (рис. 2.9).

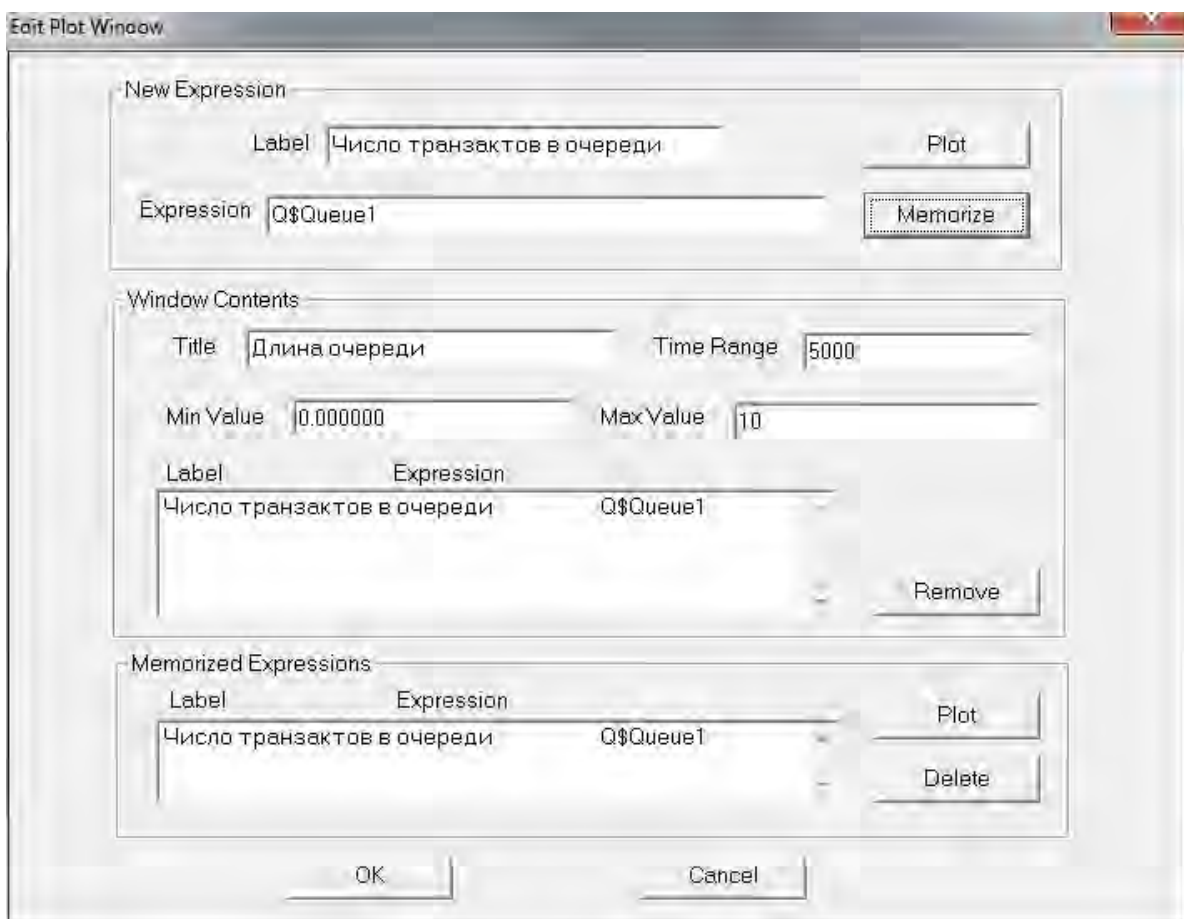


Рисунок 2.9 Заполненное окно создания графика числа транзактов в очереди, полученного в результате моделирования

В итоге график длины очереди будет иметь вид, представленный на рис. 2.10.

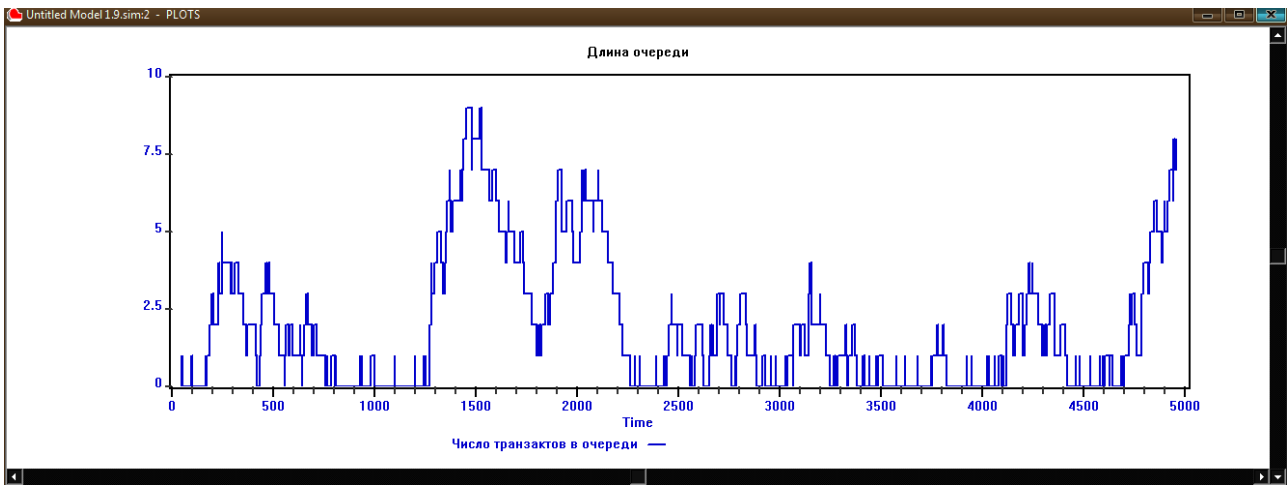


Рисунок 2.10 График количества транзактов в очереди модели в процессе моделирования

Таким же образом можно построить и график максимальной длины очереди, представленный на рис. 2.11:

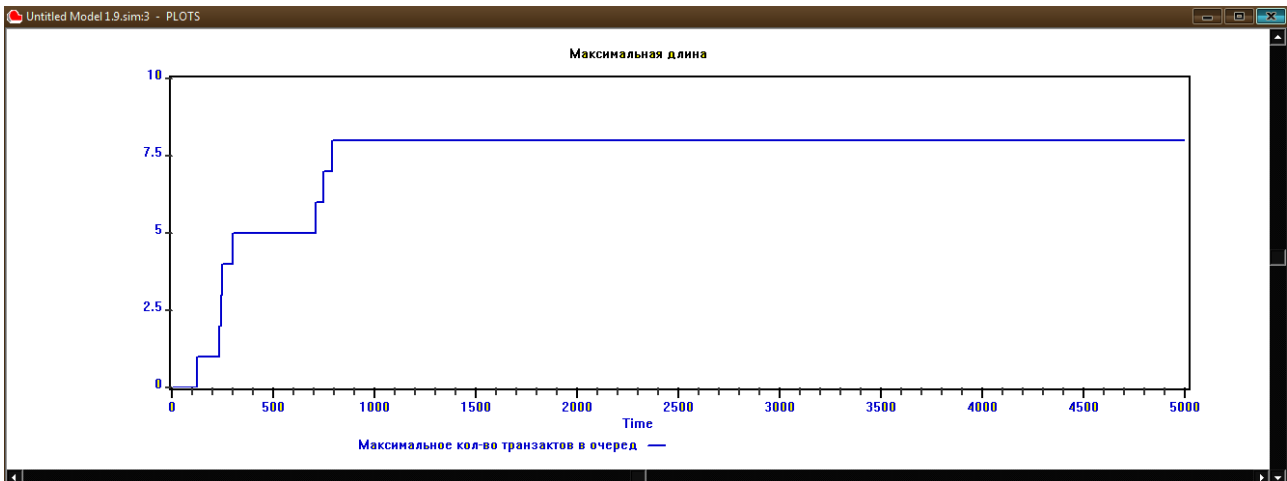


Рисунок 2.11 График максимального количества транзактов в очереди модели в процессе моделирования

### **3. МОДЕЛИРОВАНИЕ ТЕЛЕМАТИЧЕСКИХ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ GPSS WORLD**

Рассмотрим несколько классических телематических систем, которые могут быть представлены в виде моделей с использованием теории массового обслуживания.

#### **3.1. МОДЕЛИРОВАНИЯ ЛОКАЛЬНОГО КОНЦЕНТРАТОРА**

Локальный концентратор – простейшее сетевое устройство, обеспечивающее связанность одноранговых узлов в рамках сегмента сети. Характерными особенностями устройств такого типа можно считать использование разделяемых ресурсов, таких как вычислительный процессор, память и шины данных, а также низкая загруженность, относительно производительности самого концентратора. Таким образом, для моделирования данного телематического устройства целесообразно использовать модель массового обслуживания с одним входящим потоком, одним каналом обслуживания и бесконечной очередью, что в соответствии с классификацией Д.Кендалла обозначается G/G/1.

В рамках первой лабораторной работы необходимо:

Исследовать следующие системы при низкой и высокой загруженности, соответственно при интенсивности поступления транзактов равной 0,4 и 0,99 от интенсивности обслуживания:

- 1) M/M/1;
- 2) M/G/1;
- 3) G/G/1;
- 4) G/D/1.

Построить графики длины очереди от времени при обоих загрузках для всех вариантов системы, а также исследовать загруженность системы и сделать

выводы о влиянии типа обслуживания на длину очереди и загруженность системы.

### 3.2. МОДЕЛИРОВАНИЯ ЛОКАЛЬНОГО КОММУТАТОРА

Основным отличием коммутатора от концентратора является наличие возможности параллельно обрабатывать несколько сессий. С точки зрения задачи имитационного моделирования, это сводится к наличию нескольких независимых обслуживающих приборов. Также, в силу большей загруженности, на телематических системах такого класса возникают потери, вызванные переполнением буфера памяти сетевой карты, что может быть описано с достаточной степенью точности путем введения ограничений на размер очереди системы массового обслуживания.

В рамках второй лабораторной работы необходимо:

Исследовать следующие системы при низкой и высокой загруженности, соответственно при интенсивности поступления транзактов равной 0,9 и 2,4 от интенсивности обслуживания:

- 1) M/M/1;
- 2) M/M/2;
- 3) M/M/3;
- 4) M/G/2;
- 5) M/G/3;
- 6) M/D/2;
- 7) M/D/3.

Построить графики длины очереди от времени при обоих загрузках для всех вариантов системы, исследовать загруженность системы, среднее время пребывания заявки в системе, а также сделать выводы о влиянии типа обслуживания и количества обслуживающих каналов на длину очереди, загруженность системы и время пребывания пакета.

В рамках третьей лабораторной работы необходимо:

Исследовать следующие системы при низкой и высокой загруженности, соответственно при интенсивности поступления транзактов равной 0,2 и 0,4 от интенсивности обслуживания и при наличии конечной ( $k=30$ ) и бесконечной ( $k=\infty$ ) очередей:

- 1)  $M/M/1/k$  ;
- 2)  $\bar{M}_2/M/1/k$  ;
- 3)  $\bar{M}_3/M/1/k$  ;
- 4)  $\bar{G}_2/M/1/k$  ;
- 5)  $\bar{G}_3/M/1/k$  ;
- 6)  $\bar{D}_2/M/1/k$  ;
- 7)  $\bar{D}_3/M/1/k$  .

Построить графики длины очереди от времени при обеих загрузках для всех вариантов системы, исследовать загруженность системы, среднее время пребывания заявки в системе, а также сделать выводы о влиянии распределения интенсивности поступления транзактов на длину очереди, загруженность системы и время пребывания заявки. Сравнить полученные данные с данными второй лабораторной.

### 3.3. МОДЕЛИРОВАНИЯ МАГИСТРАЛЬНОГО МАРШРУТИЗАТОРА

Магистральный маршрутизатор – это оборудование операторского класса, хотя бы один интерфейс которого принадлежит магистральной зоне. Данный класс устройств отличается от рассмотренных ранее тем, что функционирует поверх нескольких различных проколов канального уровня, что оказывает существенное воздействие на интенсивность обслуживания.

Таким образом, отличительной особенностью устройств подобного класса, с точки зрения задачи моделирования, является наличие нескольких различных по своим характеристикам входных потоков транзактов, а также наличие нескольких обслуживающих приборов, отличающихся по своим характеристикам.

В рамках четвертой лабораторной работы необходимо:

Исследовать следующие системы при размере очереди 30 пакетов, а также низкой и высокой загруженности, соответственно при суммарной интенсивности поступления транзактов равной 0,4 и 1,2 от суммарной интенсивности обслуживания всех каналов:

- 1)  $\vec{M}_3/\vec{M}_3/3/k$ ;
- 2)  $\vec{M}_3/\vec{G}_3/3/k$ ;
- 3)  $\vec{M}_3/\vec{D}_3/3/k$ ;
- 4)  $\vec{G}_3/\vec{G}_3/3/k$ .

Построить графики длины очереди от времени при различных нагрузках для всех вариантов системы, исследовать загруженность системы, а также сделать выводы о влиянии распределения интенсивности поступления и обслуживания транзактов на длину очереди, загруженность системы и время пребывания заявки.

### **3.4. МОДЕЛИРОВАНИЕ ПРИЛОЖЕНИЙ С РАЗЛИЧНЫМИ ТИПАМИ ТРАФИКА**

В компьютерных сетях существует большое количество приложений, использующих несколько информационных потоков одновременно. Все виртуальные соединения, формируемые такими приложениями, можно классифицировать по требованиям к качеству канала связи, а также по критерию «важности» этих соединений для функционирования сервиса. Например, при формировании FTP-соединения, необходимо установить два

ТСР-соединения на 20 и 21 порт, при этом оба эти соединения участвуют в передаче данных между сервером и клиентом. В качестве примера информационных потоков с разными требованиями качества передачи можно привести Skype. При организации видеозвонка, в этом сервисе устанавливается несколько соединений таким образом, что аудио- и видео-данные передаются раздельно. При этом аудио данные считаются более «важными», т.е. с точки зрения математического моделирования, являются более приоритетными.

В рамках пятой лабораторной работы рассматриваются системы массового обслуживания с различными типами приоритета:

- без приоритета;
- с абсолютным приоритетом;
- с относительным приоритетом.

Использование различных типов приоритета позволяет лучше моделировать поведение телематических систем в результате перегрузки каналов связи.

Необходимо исследовать следующие системы при размере очереди 30 пакетов, с различным типом приоритета (без приоритета/с абсолютным и относительным приоритетами), а также низкой и высокой загруженности, соответственно при суммарной интенсивности поступления транзактов равной 0,4 и 1,2 от суммарной интенсивности обслуживания всех каналов:

- 1)  $\bar{M}_2/M/3/k$  ;
- 2)  $\bar{M}_2/G/3/k$  ;
- 3)  $\bar{M}_2/D/3/k$  ;
- 4)  $\bar{G}_2/G/3/k$  .

Построить графики длины очереди от времени при различных нагрузках и типах приоритета для всех вариантов системы, исследовать вероятности потери приоритетных и неприоритетных транзактов, а также времени пребывания каждого и типов заявок в системе.

### 3.5. МОДЕЛИРОВАНИЕ ГРИД СИСТЕМ

Грид система – это архитектура построения виртуальной вычислительной машины, состоящей из кластеров, соединенных при помощи компьютерных сетей. Распределенные вычисления в грид основываются на параллельной обработке данных на большом числе обычных компьютеров. При такой организации пользователь формирует запрос для грид системы, после чего запрос распараллеливается, отправляется на обработку на различные узлы грид системы, а после завершения всех вычислений снова собирается и отправляется пользователю.

Для моделирования грид систем необходимо использовать механизмы распараллеливания и синхронизации заявок.

В рамках шестой лабораторной работы необходимо:

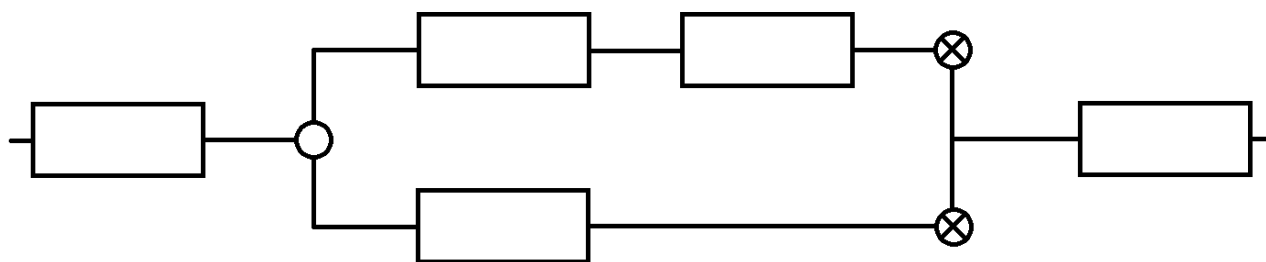


Рисунок 3.1 Схема сети систем массового обслуживания с синхронизацией

Создать имитационную модель в соответствии с рисунком 3.1. На вход системы поступает поток транзактов, моделирующих запросы пользователей, после обработки транзакты отправляются на параллельную обработку по двум направлениям. В первой цепи два обслуживающих прибора, а во второй – один. После чего заявки синхронизируются и обрабатываются с целью объединения результатов и отправки их пользователю.

Необходимо исследовать работу данной сети с учетом того, что размер очереди на каждом обслуживающем приборе равен 30 пакетам, а



распределение времени обслуживания описывается показательным (M), детерминированным (D) и произвольным (G) распределениями. Провести экспериментальные исследования при интенсивностях входящего потока равных 0.3, 0.5, 0.7, 0.9, 1 и 1.2 от интенсивности обслуживания. Выяснить при какой загрузке сети будут возникать потери пакетов, а также на каких узлах сети будет формироваться очередь транзактов.

Исследовать длинные очереди и загрузку систем в сети, а также сделать выводы о влиянии распределения интенсивности поступления и обслуживания заявок на длину очереди, загрузку системы и время пребывания транзакта.

### **3.6. МОДЕЛИРОВАНИЕ ПРИКЛАДНЫХ ТЕЛЕМАТИЧЕСКИХ СЕРВИСОВ**

Рассмотрим прикладные телематические сервисы на примере программы Skype. В рамках пятой лабораторной в качестве модели такой телематической системы как Skype мы рассматривали систему массового обслуживания с двумя входящими потоками, ограниченной очередью обслуживания и различными приоритетами. Данная модель достаточно хорошо описывает динамику движения пакетов в компьютерной сети, но не очень правдоподобно отражает состояние сетевых устройств при обработке двух разных типов трафика. Процесс передачи, обработки и хранения пакетов аудио- и видео-данных существенно различается, в первую очередь, из-за разной интенсивности поступления и обслуживания пакетов, но, наряду с этим, немаловажным фактом является размер самих пакетов. Во всех предыдущих лабораторных работах мы не учитывали размер передаваемых сообщений и считали, что все они одинаковые.

В рамках седьмой лабораторной работы необходимо исследовать систему массового обслуживания  $\vec{G}_2/\vec{G}_2/1/k$  при размере очереди 60 пакетов, с различным типом приоритета (без приоритета/с абсолютным и относительным

приоритетами). При этом необходимо учесть, что размер неприоритетных пакетов в 3 раза превышает размер приоритетных, а среднее время обслуживания неприоритетных пакетов в 2 раза превышает время обслуживания приоритетных. В работе требуется рассмотреть два варианта загрузки: в первом интенсивность поступления пакетов одинаковая и равняется 0,3 от времени обслуживания приоритетных пакетов, во втором – интенсивность поступления приоритетных пакетов должна составлять 0,5 от интенсивности их обслуживания, а для неприоритетных – 0,6 от интенсивности их обслуживания.

В отчете необходимо привести графики количества пакетов в очереди для каждого из приоритетов и общий размер очереди, а также долю потерь пакетов каждого типа и загрузку обслуживающего прибора пакетами каждого типа.

### **3.7. МОДЕЛИРОВАНИЕ ЛОКАЛЬНЫХ СЕТЕЙ С ОТКАЗАМИ**

В рамках восьмой лабораторной будет рассмотрен один из ключевых современных телематических сервисов, а именно – маршрутизация. Многие специалисты считают, что именно независимая маршрутизация пакетов, лежащая в основе IP протокола, явилась причиной такой популярности и распространенности сети Интернет. Каждый пакет, передаваемый по IP сети маршрутизируется независимо исходя из значений полей самого пакета, настроек маршрутизатора и текущего состояния среды передачи.

В рамках восьмой работы необходимо создать сеть систем массового обслуживания в соответствии с рисунком 3.2

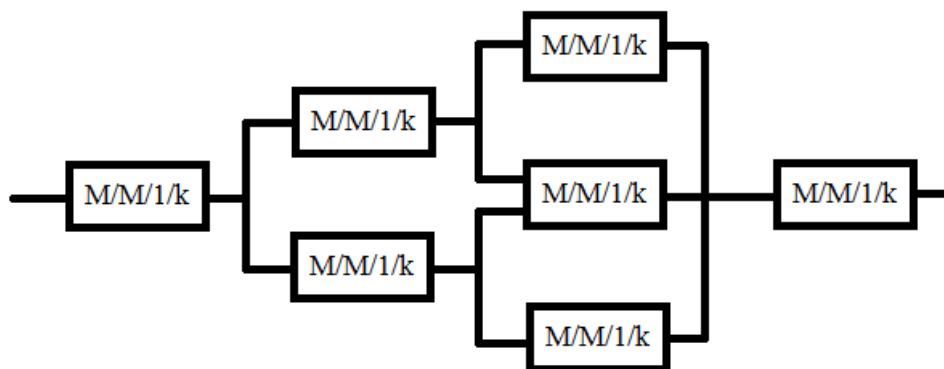


Рисунок 3.2 Схема сеть маршрутизаторов

Для упрощения будем считать, что все маршрутизаторы, входящие в сеть идентичные, размер очереди  $k$  у них равен 25, а интенсивность поступления заявок на вход первой системы равняется интенсивности обслуживания. Моделирование осуществлять по времени. При этом время моделирования должно в 1000 раз превосходить среднее время поступления заявок в систему.

Транзакты произвольно выбирают следующий маршрутизатор. На середине моделирования один из маршрутизаторов выходит из строя, при этом все последующие транзакты на него не направляются.

В отчете необходимо привести графики загрузки каждого из маршрутизаторов, общее количество транзактов на каждом из них, а также процент потерь.

В рамках девятой лабораторной работы необходимо собрать следующую сеть (см. рисунок 3.3):

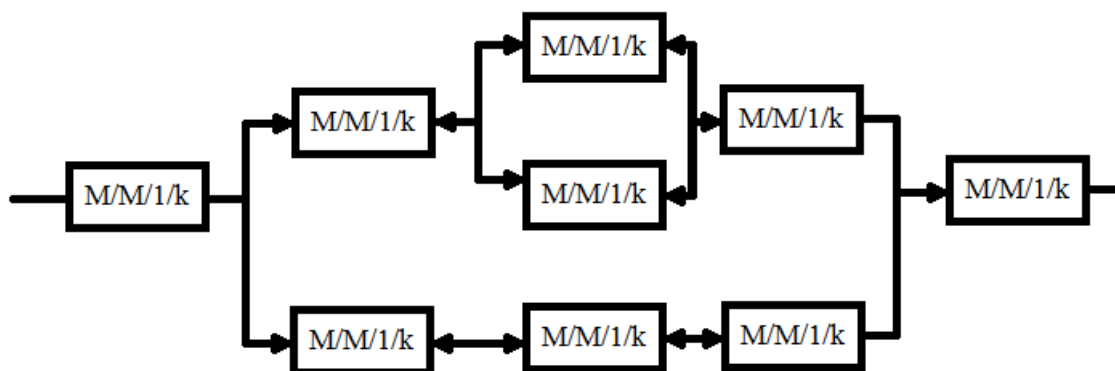


Рисунок 3.3 Схема сеть маршрутизаторов

Отличительной особенностью данной системы является тот факт, что все соединения маршрутизаторов внутри локального сегмента являются двусторонними, т.е. транзакт с выхода маршрутизатора может поступать, как на следующий (в штатном режиме работы), так и на предыдущий узел (в случае неисправности следующего). Все настройки соответствуют предыдущему варианту работы сети, но в данном случае в середине моделирования из строя должны выйти любые два маршрутизатора с условием, чтобы сохранилась возможность передачи транзактов через сеть.

### 3.8. МОДЕЛИРОВАНИЕ ОБЛАЧНЫХ ТЕЛЕМАТИЧЕСКИХ СЕРВИСОВ

Облачные технологии с каждым годом становятся все более популярными. Спецификой такого подхода к обработке и хранению информации является факт общей доступности ресурсов облака различным пользовательским задачам. В рамках заключительной десятой лабораторной работы рассматривается функционирование планировщика задач облачного вычислительного устройства. Современные высокопроизводительные вычислители являются гетерогенными многофункциональными устройствами, ориентированными на обработку различных типов данных. Использование

различных типов вычислителей позволяет обслужить большой диапазон различных запросов пользователей. Например, использование графических ускорителей (GPU) с огромным числом простых вычислительных ядер позволяет многократно сократить время выполнения матричных операций, в то время как для сложных последовательных процессов типа разбора XML или работы с файлами выигрыша не будет никакого. Сложные, но неизменные алгоритмы можно реализовывать используя специализированные чипы с программируемой пользователем вентильной матрицей (FPGA). Такие вычислители позволяют очень быстро выполнять достаточно сложные, но однотипные операции с большим объемом данных (например, шифрование сетевого трафика). Использование традиционных вычислителей позволяет решать большой круг задач, но в некоторых случаях производительность будет уступать специализированным решениям.

Таким образом, в рамках десятой лабораторной работы необходимо моделировать работу планировщика заданий в облачной системе, имеющей три типа вычислителей: традиционные CPU, многоядерные GPU и специализированные FPGA. Необходимо создать систему массового обслуживания класса  $\vec{G}_3/\vec{G}_3/3$ . Задача формулируется следующим образом:

Как и в прошлой лабораторной, время моделирования должно в 1000 раз превосходить среднее время поступления заявок первого типа в систему.

На вход планировщика с бесконечной очередью поступают 3 потока транзактов. Интенсивность первого потока в 1,5 раза больше, чем интенсивность второго и третьего. Время обслуживания транзакта зависит от типа потока и типа обслуживающего прибора в соответствии с таблицей, при условии, что интенсивность первого потока принята за единицу:

Таблица. Зависимость интенсивности обработки транзактов от типа потока и прибора обслуживания

Номер потока	Прибор обслуживания		
	CPU	GPU	FPGA
1	1,2	0,6	0,5
2	0,7	1,5	0,8
3	0,6	0,5	1,4

В первом варианте работы транзакты хранятся в общей очереди планировщика и попадают на первый освободившийся прибор обслуживания. Во втором – планировщик может отбрасывать транзакты и использовать различные типы приоритета для повышения производительности системы.

Необходимо оптимизировать работу планировщика таким образом, чтобы число обслуженных заявок было максимальным, при условии прохождения системы всеми типами заявок.

В отчете привести количество прошедших заявок в первом и во втором случаях, алгоритм работы планировщика и долю потерянных транзактов каждого типа.

## СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Олифер Н.А., Олифер В.Г., Компьютерные сети. Принципы, технологии, протоколы. СПб: Питер, 2006 – 672 с.
2. Таненбаум Э. Компьютерные сети. – Изд. 4-е, СПб: Питер, 2010 – 992 с.
3. Официальный сайт сетевого симулятора Network Simulator [Электронный ресурс]. – Режим доступа: <http://www.isi.edu/nsnam/ns>, свободный.
4. Заборовский В.С., Сети ЭВМ и телекоммуникации. Моделирование и анализ сетей связи с коммутацией пакетов. Network Simulator (Сетевой симулятор ns2): Учебное пособие. СПб: Изд-во СПбГТУ, 2001 – 108 с.
5. Заборовский В.С., Мулюха В.А., Новопашенный А.Г. Сети ЭВМ и телекоммуникации. Анализ трафика в сетях коммутации пакетов. Учебное пособие. – СПб: Изд-во СПбГПУ, 2010 – 90 с.
6. Кудрявцев Е.М., GPSS World. Основы имитационного моделирования различных систем. – М.: ДМК Пресс, 2004. – 320 с.
7. Безручко Б.П., Смирнов Д.А. Математическое моделирование и хаотические временные ряды. – Саратов: Издательство ГосУНЦ «Колледж», 2005. – 320 с.
8. Крылов В.В., Самохвалова С.С. Теория телетрафика и ее приложения. – СПб.: БХВ-Петербург, 2005. – 288 с
9. Алиев Т.И. Основы моделирования дискретных систем. – СПб: СПбГУ ИТМО, 2009. – 363 с.

В.С. Заборовский, А.С. Ильяшенко, В.А. Мулюха

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ  
ТЕЛЕМАТИЧЕСКИХ СИСТЕМ

Учебное пособие

Отпечатано с готового оригинал-макета, предоставленного авторами,  
типографии Издательства СПбГПУ.

195251, Санкт-Петербург, Политехническая ул., 29.