

# ПРИМЕНЕНИЕ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО ПОДХОДА ПРИ ИМИТАЦИОННОМ МОДЕЛИРОВАНИИ МУЛЬТИСЕРВИСНЫХ СЕТЕЙ СВЯЗИ

В. П. Корчагин

Сибирский государственный университет телекоммуникаций и информатики,  
630102, Новосибирск, Россия

---

УДК 004.045

Статья посвящена проектированию объектной модели мультисервисных сетей связи. Описываются варианты декомпозиции предметной области на сущности, выделение классов и разработка на их основе общей структуры модели. С помощью программного средства, реализующего представленную объектную модель, проведен анализ влияния повторных вызовов на функционирование мультисервисной сети связи с адаптивной маршрутизацией.

**Ключевые слова:** мультисервисные сети связи, имитационное моделирование, объектный подход, адаптивная маршрутизация, повторные вызовы.

Article is devoted to the design of the object model of telecommunication networks. It described decomposition for telecommunication domain, emphasize decomposition into classes and develop process for general structure of the model. With a software that implements presented object model, we analyze the influence of repeated calls for the performance of telecommunication network with dynamic routing.

**Key words:** multiservice telecommunication networks, simulation, object-oriented design, dynamic routing, repeated calls.

**Введение.** Разработка и анализ алгоритмов маршрутизации для мультисервисных сетей связи (МСС) требуют принятия во внимание множества различных условий и характеристик сети, на базе которой предполагается функционирование конкретного алгоритма. Трудоемкость реализации аналитических и алгоритмических методов расчета МСС существенным образом зависит от учитываемых характеристик узлов и линий связи, сложности образованной структуры сети, характера поступающих информационных потоков. Чем подробнее рассматриваются подобные показатели, тем сложнее описывается соответствующий случайный процесс и, соответственно, труднее вести численный анализ модели. При имитационном моделировании (ИМ) эти факторы играют существенно меньшую роль, поскольку процесс моделирования структуры МСС и реализация соответствующих процедур ее ИМ в более слабой степени зависят от сложности отдельных компонент, чем при аналитической реализации. Особенно отчетливо это проявляется при дальнейшем усложнении модели.

**1. Постановка задачи.** Стоит отметить, что на сегодняшний день существует достаточно большое количество программных пакетов (как коммерческих, так и бесплатных с открытым кодом), которые позволяют в той или иной мере производить имитационное моделирование (ARENA, MATLAB, GPSS). Существуют даже программные системы, которые специально настроены на задачи моделирования компьютерных сетей (COMNET,

BONeS DESIGNER, OPNET Modeler, ns). Однако нужно сразу же сказать и про ограничения их применимости. Многие продукты в этой области (COMNET, BONeS DESIGNER, OPNET Modeler) ориентированы на типовые протоколы, применяемые в существующих сетях. Следовательно, возможности по анализу работы сети при других алгоритмах маршрутизации, которые требуется исследовать на определенном уровне абстракции, существенно затруднены. Необходимо также иметь в виду, что подавляющее большинство перечисленных систем моделирования уделяет главное внимание анализу работы сети только на уровне передачи пакетов, что существенно ограничивает область исследования современных алгоритмов маршрутизации.

Моделирование МСС при помощи универсальных систем широкого спектра применения (ARENA, MATLAB, GPSS) в конечном итоге по сложности работ часто оказывается не менее громоздким, чем при реализации полноценной модели на языке программирования высокого уровня. В итоге было принято решение построить собственную имитационную модель МСС. Как сущность МСС представляет собой композицию типовых объектов, поэтому при описании модели в полной мере эффективным будет использование объектно-ориентированного подхода.

Таким образом, если следовать выбранному подходу, то предстоит:

- 1) проанализировать основные сущности, относящиеся к области мультисервисных сетей связи, и выделить их классы;
- 2) на базе полученных классов спроектировать объектную модель для представления МСС;
- 3) на основе полученной объектной модели реализовать программный продукт для имитации работы исследуемой сети связи.

**2. Формирование объектной модели МСС.** В начале процесса выделения основных сущностей, необходимых для построения объектной модели МСС, требуется рассмотреть функции проектируемой модели.

Модель должна описывать топологию и трафик МСС. При этом в модели алгоритм маршрутизации и процесс симуляции должны являться отдельными компонентами относительно ранее указанной топологии и существующего в сети трафика. Таким образом, в первом приближении выделяются компоненты модели, показанные на рис. 1.

**2.1. Описание топологии МСС.** Термин „топология“ подразумевает, что сеть как совокупность узлов, связанных между собой линиями связи (дугами), представляется в виде графа. Таким образом, любой МСС будут присущи объекты классов „узел“ (Node) и „связь“ (Line). Для организации хранения информации о соединениях узлов (топологии) в класс „связь“ включены соответствующие атрибуты (для хранения идентификатора первого и второго соединяемых этой связью узлов). Подразумевается, что один экземпляр класса „связь“ предназначен для организации одного соединения для двух узлов.

Экземпляры классов „узел“ (Node) и „связь“ (Line), как правило, могут включать дополнительную информацию (например, уникальный номер и название объекта в системе, комментарий по эксплуатации, предназначение).

Реализация классов Node и Line может быть выполнена с учетом уже существующего сетевого оборудования (коммутатор, маршрутизатор конкретного типа, оптическое волокно некоторой марки и т. д.).

В зависимости от специфики и подробности описания модели МСС для узлов и линий связи характерен произвольный набор дополнительных свойств и функций. Решение подобной задачи в ООП реализации эффективно осуществляется с помощью задействова-

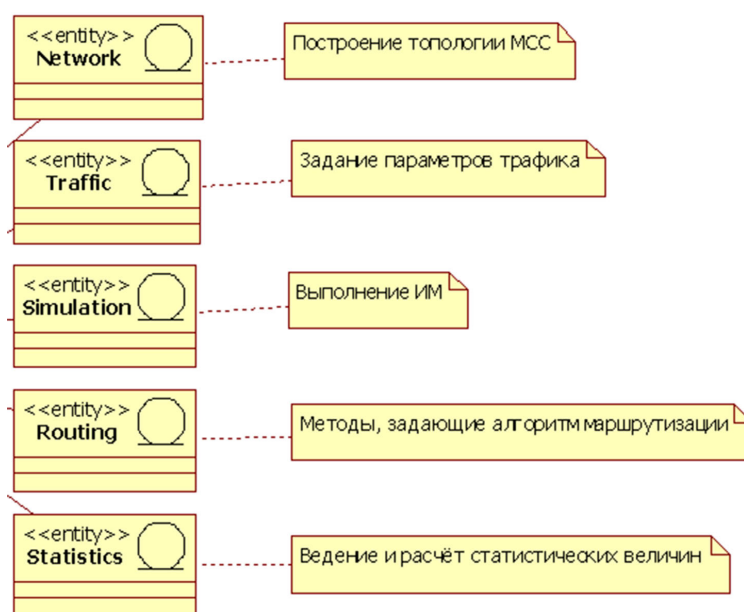


Рис. 1. Основные сущности модели МСС

ния инструмента наследования. Введены классы Router и Wire, дополненные требуемыми методами и атрибутами (в табл. 1 в качестве примера показаны такие самые общие атрибуты для различных моделей представления МСС, которые добавляются в класс Router). Здесь можно прийти к следующей идее: если в рамках каждого класса объектов МСС предусмотреть реализацию всего множества присущих ему параметров (т.е. индуцировать данный набор из большинства существующих моделей), то становится возможной конвергенция различных моделей построения. Таким образом, ИМ уже проводится в рамках одной универсальной модели.

Для реализации функций создания топологии и хранения сопутствующей этому информации вводится класс Network.

Данный класс является контейнером для объектов класса Router и Wire (`v_ListRouter` и `v_ListWire` соответственно; являются структурами для хранения множества объектов указанных типов). В его функции входит управление созданием и уничтожением объектов Router и Wire при процессе проектирования топологии сети.

В классе реализованы следующие методы (табл. 1).

**2.2. Описание сетевого трафика.** Для расчета сети при выбранном алгоритме маршрутизации, как правило, ограничиваются набором параметров, необходимых для исследования. Данный набор связан с тем, какие характеристики работы понадобятся при оценке сети, функционирование которой осуществляется в рамках вышеназванного алгоритма маршрутизации.

В общем случае, задача сети — осуществить передачу информации между отдельными ее точками [1]. Запрос на передачу информации осуществляется в результате заявки. Поступление заявки обычно носит вероятностный характер, т.е. поток заявок является стохастическим процессом. Когда МСС осуществляет обработку такого потока, ее состояние в любой момент времени задается распределениями вероятностей на множестве возможных состояний. Таким образом, МСС можно представить моделью стохастического

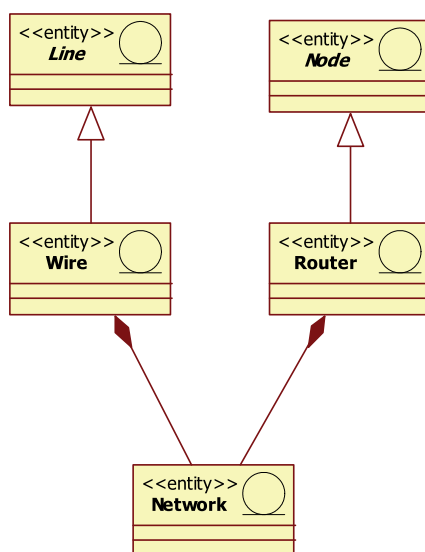


Рис. 2. Класс Network

объекта.

Для описания такого объекта предусмотрен набор классов, который позволяет при необходимости описать всю представленную сложность. Группа таких связанных классов выделена в общей структуре модели в сущность „Traffic“ (рис. 1).

В качестве примера в данной работе представлена одна из реализаций набора классов для описания сетевого трафика. На МСС в данной модели весь трафик рассматривается как совокупность информационных потоков с определенными характеристиками, передача каждого из которых организуется между определенной парой узлов.

Под информационным потоком (ИП) понимается совокупность поступающих заявок на запрос трафика, характеризующегося определенными свойствами [1]. По схожим свойствам весь трафик можно разбить на определенные группы (в данном случае — выделить определенные ИП), например, трафик передачи данных, видеопотоки, передача гипертекста. Каждый ИП здесь может характеризоваться своим значением интенсивности поступления, требованием к канальным ресурсам, приоритетом и т. д.

Таким образом, можно выделить класс информационного потока (ClaimType), каждый объект которого будет соответствовать некоторому ИП. На данный момент класс „ИП“ содержит следующие атрибуты: arrivalRate — интенсивность, sourceDemand — потребность в канальном ресурсе.

Для хранения объектов „ИП“ потребуется класс-менеджер, выполняющий сервис по созданию, редактированию и хранению объектов „ИП“ (класс Traffic). Диаграмма для указанных классов изображена на рис. 2.

Кроме того, в классе Traffic могут реализовываться различные инструменты для учета специфики исследования трафика на сети. В данной работе таковым являлось изучение влияния повторных вызовов на характеристики работы МСС. Для данных целей в класс были введены дополнительные атрибуты: arrivalRate — интенсивность поступления заявок повторного потока вызовов, H1 — вероятность повторного вызова от абонента при первичном отказе предоставления услуги, H2 — вероятность повторного вызова от аба-

Таблица 1

## Атрибуты класса CRouter

№	Наименование	Назначение
1	loss	вероятность потерь на сетевом узле (подобная характеристика может быть интересной при абстрактном представлении узла МСС как „черного ящика“)
2	serviceTime	интенсивность обслуживания узлом (например, скорость обработки пакетов)
3	portCount	максимальное количество подключений, которое возможно поддерживать узлом (на физическом уровне подобным может выступать количество портов коммутатора, на логическом — количество виртуальных соединений, количество маршрутов в таблице)
5	delay	время задержки сигнала (пакета) узлом
6	reliability	надежность узла (может выражаться, например, временем наработки на отказ)

нента при последующих отказах в предоставлении услуги (подробней данные параметры рассмотрены в разделе 4.2.).

Атрибут `arrivalIndex` является коэффициентом загруженности системы, предназначается для исследования поведения работы МСС в условиях повышенной загруженности линий связи.

Хранение объектов „ИП“ осуществляется в динамической структуре „вектор“ (vector), куда последовательно записываются добавляемые экземпляры. Так как формирование (добавление, удаление) объектов „ИП“ осуществляется строго на стадии описания, перед операциями ИМ, то необходимости в частом изменении порядка размещения таких объектов нет. Поэтому данный тип структуры для хранения объектов является адекватным.

2.3. *Описание алгоритма маршрутизации.* Для описания алгоритма маршрутизации предназначен класс `Algorithm` (рис. 3). Этот абстрактный класс состоит из виртуальных функций, общих для любого алгоритма маршрутизации (например, поиск наилучшего пути, создание и удаление маршрута). Абстрактный класс `Algorithm` предназначен для реализации такого инструмента, как „наследование интерфейса“ в объектно-ориентированном подходе к разработке, что необходимо для подключения к модели различных алгоритмов маршрутизации. В наследуемом от него классе описывается конкретный метод, по которому осуществляется маршрутизация. Для системы ИМ такой подход обеспечивает возможность существования в системе нескольких алгоритмов. При необходимости это позволяет сравнивать их эффективность в рамках одной построенной МСС.

Для рассматриваемого в работе алгоритма адаптивной маршрутизации по способу выбора пути с наименьшей загрузкой LLR разработан класс `Algorithm_LLRL`, опи-

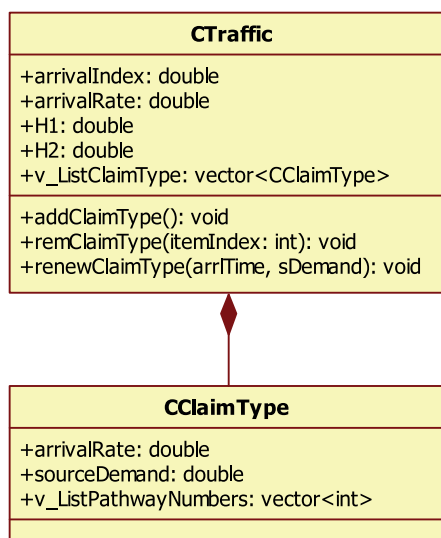


Рис. 3. Диаграмма классов для описания трафика МСС

сывающий данный метод (подробное описание метода можно посмотреть в [2]). В класс `Algorithm_LLР` также включены методы для занятия и освобождения ресурсов (`loadResource`, `freeResource`) для данного алгоритма. В общем случае данные процедуры могут включать все нюансы процессов занятия и освобождения (учитывать время между получением служебных пакетов, связанных с установлением соединения, осуществлять проверку пакетов на определенный набор требуемых флагов для успешного выполнения операции и т. д.), в данной же работе были реализованы лишь действия по изменению количества ресурсов соответствующих участков. Также в общем случае данный класс может содержать алгоритмы по проверкам достижимости узлов, алгоритмы резервирования ресурсов участков сети, иными словами, включать любые алгоритмы, связанные как с сигнализацией, так и с маршрутизацией на сети.

Большинство реальных алгоритмов маршрутизации работают на основе специальных таблиц, где хранятся списки маршрутов, записанных администратором или полученных в результате функционирования некоторого протокола маршрутизации. Для хранения информации о маршрутах в данной рассматриваемой работе для класса `Algorithm_LLР` создан подкласс `Pathway`. Класс `Pathway` содержит атрибуты для хранения идентификаторов узлов, между которыми проложен маршрут (`NodeA`, `NodeB`), а также последовательность транзитных узлов (`ListItemTrans`).

В качестве примера в работе рассматривается случай, когда список возможных маршрутов (из которых в каждом случае проводится выбор наиболее эффективного) заранее определен. Для управления созданием маршрутов и выполнения операций над ними создан класс `PathwayManager`. Класс `PathwayManager` содержит структуру для хранения объектов „маршрут“ (`ListPathwayNode`), а также методы для создания, редактирования, удаления маршрутов.

**2.4. Описание алгоритма имитации.** Для организации имитационного моделирования предназначен класс `Simulation`. Диаграмма класса показана на рис. 4. При имитации работы сети этот класс связывается со средствами маршрутизации (класс `Algorithm`), с классом `Network` (для получения сведений о пропускной способности линий связи), с классом `Traffic`

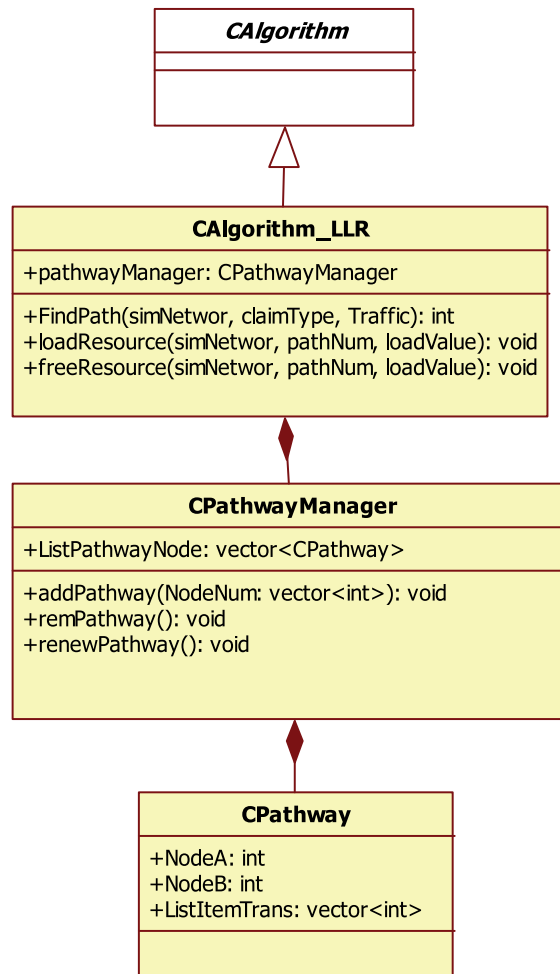


Рис. 4. Диаграмма класса Algorithm

(для получения данных о характеристиках входных информационных потоков), а также с классом `Statistics` (куда будет направляться информация для сбора статистических сведений). Класс `Simulation` содержит атрибуты проведения ИМ, такие как количество опытов (продолжительность) моделирование (`serieAmount`), количество серий экспериментов (`serieCount`), а также методы, реализующие алгоритмы имитационного моделирования сети (`DoSimulation`).

Кроме того, при проведении ИМ требуются структуры данных, позволяющие хранить сведения об сгенерированных заявках (`claimType` — тип ИП, `pathwayNumber` — номер маршрута, по которому требуется обслужить заявку, `timeProcessed` — время пребывания заявки в сети), а также методы по обслуживанию таких структур данных (`AddClaimProc` — разместить заявку, `RemClaimProc` — удалить заявку, `RenewClaimProc` — обновить параметры заявки, `ResetClaimProc` — очистить сведения о заявке). Подробнее об алгоритме имитации, использующем подобную систему с указанными заявками, рассказывается в разделе 3.

Для учета заявок потока повторных вызовов при проведении ИМ также потребуются соответствующие структуры данных. При имитации потребуется хранить информацию о типе заявок, поступивших повторно (массив данных `v_queries`), а также их количестве

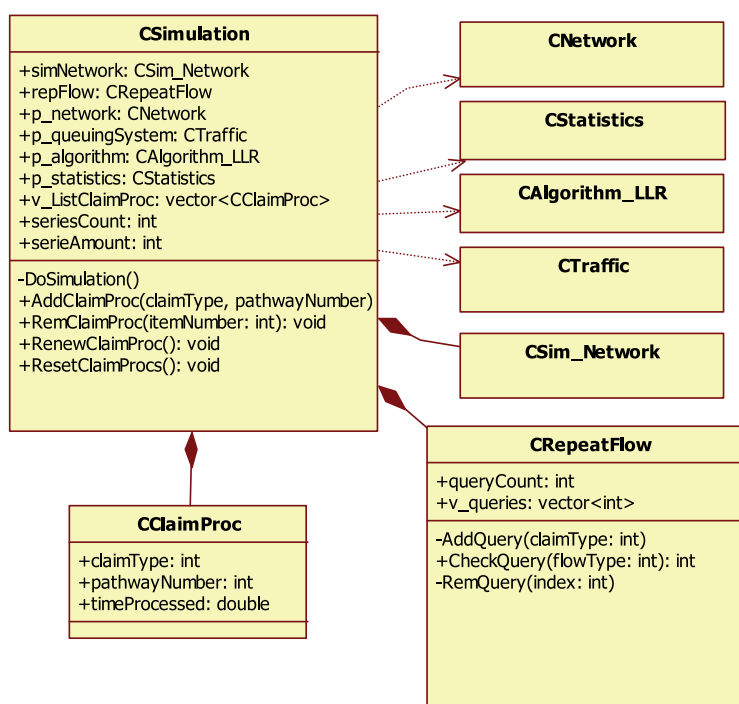


Рис. 5. Диаграмма класса Simulation

(queryCount). Для обслуживания массива используются методы `AddQuery` — добавить заявку повторного вызова, `RemQuery` — удалить заявку повторного вызова, `CheckQuery` — проверить поступление заявки повторного вызова.

Аналогичным образом обстоит дело с характеристиками узлов и линий связи сети, которые также постоянно изменяются в процессе симуляции. Чтобы отделить такие характеристики от фиксированных, задающихся на этапе до проведения ИМ, отдельно для процесса прогонки модели был введен класс `Sim_Network`. В данной работе в построенном конкретном примере системы ИМ при моделировании работы сети главным образом изменяются значения канального ресурса соединительных линий связи. Поэтому в классе `Sim_Network` реализуется только класс для учета изменений значений линий связи `Sim_Wire` (моделируется значение свободного канального ресурса, `rateFree`).

**2.5. Ведение и обработка статистики.** Накопление статистики по проводимому эксперименту, расчет итоговых показателей качества функционирования сети и оценка статистической погрешности полученных результатов обработки осуществляются с помощью класса `Statistics` (в процессе ИМ класс `Simulation` передает ему все необходимые первичные данные). Класс `Statistics` включает в себя классы `StatData` и `StatEstimation`. Первый класс связан с накоплением статистических данных, получаемых в процессе симуляции, второй класс связан с расчетом статистических величин.

**2.6. Хранение объектной модели.** С целью более простого хранения объектов модели и обращения к ним в процессе работы приложения следует добавить класс `Model`, который является контейнером для объектов классов проектируемой модели. В итоге главная структура модели выглядит так, как показано на рис. 6.

Данные компоненты являются независимыми. В результате этого каждый можно расширять независимо от другого, что дает большие возможности, связанные с детальностью

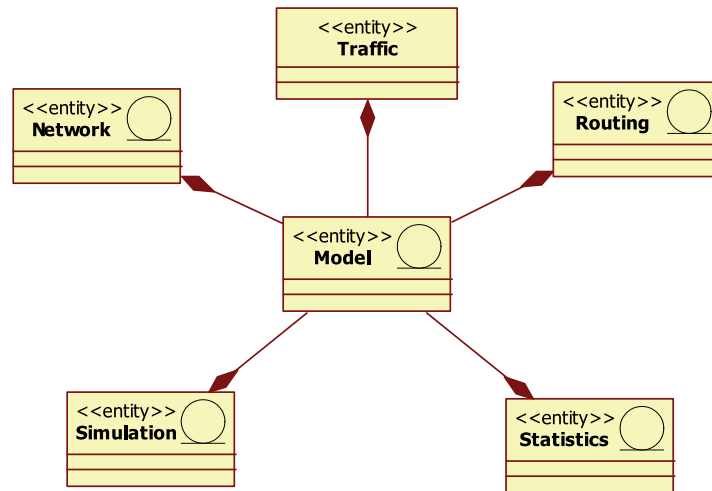


Рис. 6. Общая диаграмма класса модели

моделирования МСС.

**3. Проектирование алгоритма имитации работы МС.** В основе подхода лежит моделирование вложенной Марковской цепи. Это справедливо в случае, когда потоки вызовов и длительности обслуживания определяются Пуассоновскими распределениями. Тогда функционирование МСС описывается Марковским процессом  $x(t)$ , текущее состояние которого определяется количеством установленных соединений (для вызовов, принятых к обслуживанию), а также количеством источников повторных вызовов [3]. В момент выхода из этого состояния осуществляется переход в состояние  $n + 1$  с вероятностью (при этом событии происходит помещение еще одной (поступившей) заявки на обслуживание)

$$p = \frac{\sum_{n=1}^k \lambda_k}{\sum_{n=1}^k \lambda_k + n \cdot \mu} \quad (1)$$

и в состояние  $n - 1$  с вероятностью

$$q = \frac{n \cdot \mu}{\sum_{i=1}^k \lambda_k + n \cdot \mu}, \quad (2)$$

где  $n$  — количество вызовов на обслуживании;

$\lambda_k$  — интенсивность обслуживания.

При этом событии происходит завершение обслуживания одной из заявок, находящейся в системе. Определение номера заявки, которая выйдет из системы, будет осуществляться случайным образом.

При имитационном моделировании работы системы при подобном подходе происходят упрощение вычислений и, соответственно, экономия вычислительных средств.

Оценка точности результатов моделирования основана на центральной предельной теореме для Марковских процессов, которая говорит о сходимости выборочных статистик к нормальному закону распределения. Оценка точности чаще всего осуществляется по критерию Стьюдента [4]. При критерии Стьюдента сбор данных (для дальнейшего вычисления оценки) производится при помощи разделения основной реализации на  $N_{ser}$

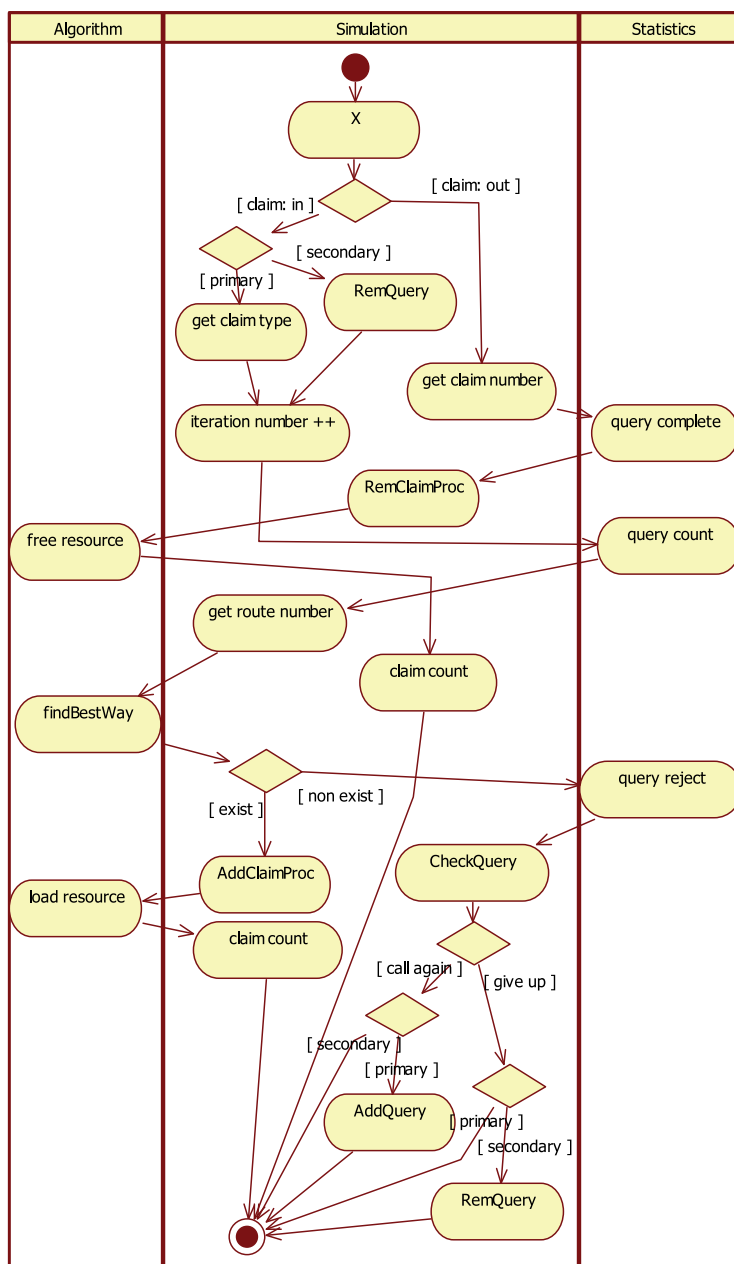


Рис. 7. Диаграмма алгоритма процедуры DoSimulation

независимых серий, каждая из которых характеризуется величиной периода проведения. Таким образом, эксперимент будет состоять из  $N_{ser}$  серий опытов, каждая из которых характеризуется количеством испытаний  $W_{end}$ . Опыт (испытанием) в данном случае является предположение о поступлении в систему некоторой заявки.

Алгоритм имитации работы МСС концептуально можно представить диаграммой (рис. 7). Подпрограммы алгоритма на диаграмме разбиты на дорожки, соответствующие объектам, которые занимаются их реализацией. На диаграмме показана одна итерация основного цикла опытов алгоритма.

После инициализации структур данных, необходимых для работы алгоритма, начи-

нается цикл генерации события. Сперва генерируется случайная величина  $X$ , затем в соответствии с подходом, который описан ранее и показан на рис. 7, определяется наступление одного из возможных событий: поступление заявки, завершение обслуживания заявки. При поступлении вызова осуществляется проверка, какому потоку (первичных вызовов, повторных вызовов) он принадлежит. Далее соответствующее событие записывается в объекте статистики. Определение вида ИП, которому принадлежит поступившая заявка, осуществляется по номеру интервала, в который попала величина  $X$ . При помощи алгоритма маршрутизации (на рис. 7 обозначено наименованием `findBestWay`) происходит поиск пути для возможности обслуживания данной заявки.

В случае существования пути осуществляется размещение на обслуживание заявки. Если заявка принадлежала потоку повторных вызовов, то она исключается из него. При помещении заявки на обслуживание происходит фиксирование требуемого канального ресурса.

Если же подходящего для обслуживания вызова пути не нашлось, то этот вызов получает отказ (происходит блокировка). Абонент, которому отказали в предоставлении услуги, может повторно осуществить вызов. Для проверки такого события вызывается метод `CheckQueue`, который его моделирует и определяет, поступит ли повторный вызов. Если поступил повторный вызов, и при этом он поступил от первичного потока, то осуществляется внесение его в поток повторных вызовов. Если же повторный вызов не поступил (т. е. абонент прекратил попытки получить услугу), то в случае принадлежности вызова потоку повторных вызовов осуществляется его исключение из данного потока.

При событии завершения обслуживания (выхода заявки из системы) определяется номер заявки. Событие записывается в объекте статистики. Далее путь разрушается и ресурсы, занимаемые данной заявкой, освобождаются.

После того, когда указанным образом реализовано очередное событие, происходит проверка на завершение моделирования.

В результате выполнения процедуры `DoSimulation` в объекте класса `Statistics` собираются все необходимые сведения для расчета интересующих характеристик и их статистической оценки.

В данной работе в объекте `Statistics` реализованы методы для расчета вероятности потерь, возникающих при недостаточности канального ресурса, при организации создания виртуальных путей для поступающих в систему заявок. Потери рассчитываются для каждого типа ИП. Также реализованы методы оценки СКО для таких потерь.

**4. Реализация контрольного примера для тестирования системы ИМ.** С целью контрольного примера промоделируем при помощи разработанной системы имитационного моделирования эксперимент работы сети, который рассмотрен в статье [2].

В данной статье рассматривается мультисервисная телекоммуникационная сеть с адаптивной маршрутизацией по методу LLR и анализируется качество функционирования сети на этапе установления соединения.

В качестве проверочных тестов в статье использовалось дискретно-событийное моделирование (`discrete event simulation, DES`). Были проведены серии анализов и получены результаты работы моделируемой системы. Возникает актуальный вопрос о сравнении результатов, получаемых в результате имитационного моделирования нашей системой и использованной в статье. Промоделируем пример, который в статье представлен под названием „Example One“.

Таблица 2

Описание методов и атрибутов		
№	Наименование метода	Назначение
1	iterationNumber	Номер итерации
2	claimCount	Количество обслуживаемых вызовов с сети
3	getClaimType	Определить тип заявки (номер ИП)
4	getClaimNumber	Определить номер заявки, завершающей обслуживание в сети
5	freeResource	Освободить каналные ресурсы сети, требуемые для обслуживания вызова
6	loadResource	Занять каналные ресурсы сети, требуемые для обслуживания вызова
7	findBestWay	Поиск лучшего пути для обслуживания вызова
8	getRouteNumber	Получить номер пути (−1 означает невозможность обслужить вызов)
9	AddClaimProc	Добавление (создание и размещение) заявки (объект Claim_Proc) на обслуживание в систему
10	RemClaimProc	Удаление заявки (объекта Claim_Proc) из системы
11	AddQuery	Добавление вызова в поток повторных вызовов
12	RemQuery	Исключение вызова из потока повторных вызовов
13	QueryCount	Количество поступивших вызовов
14	QueryComplete	Количество обслуженных вызовов
15	QueryReject	Количество вызовов, которым отказано в обслуживании

После проверки адекватности работы программы, восстановив в результате работы программы значения, сравнимые с полученными в статье, проанализируем влияние повторных вызовов на дальнейшую работу построенной сети. Изначально влияние повторных вызовов не учитывалось в данной статье, поэтому достаточно актуально узнать, как поведет себя система при их появлении.

Следует еще раз отметить, что благодаря разработанной объектной модели можно расширить и углубить исследование путем внесения в него дополнительных явлений и факторов, которые могут повлиять на качество работы сети (например, поток повторных вызовов).

4.1. *Исходные данные.* С помощью программного продукта, который реализован в соответствии с представленной концепцией объектной модели, проведены контрольные расчеты. Для расчетов была выбрана МСС, топология которой представляет собой полносвязную схему из 5 узлов (рис. 8). Как видно из рисунка, в сети имеется 10 цифровых линий связи, и все они эквивалентны по своей емкости. Следует отметить, что аналогичная МСС ранее исследовалась в [2], однако без учета эффекта повторных вызовов.

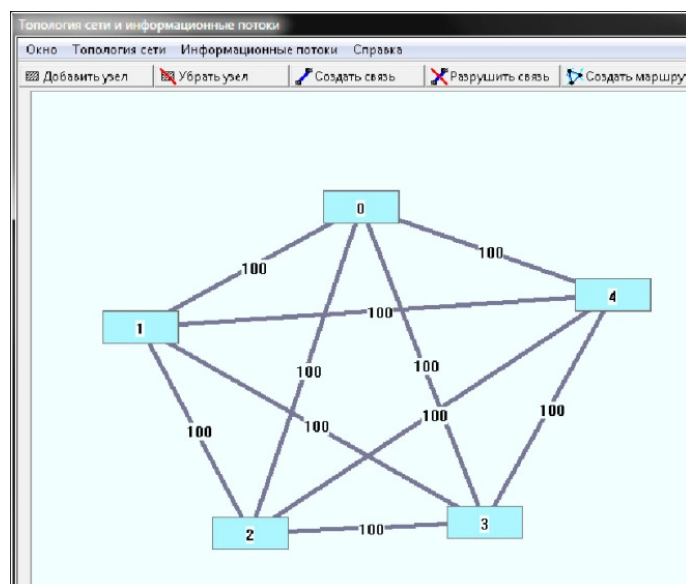


Рис. 8. Топология сети

Таблица 3

Характеристики повторных вызовов

Настойчивость абонентов	$H_1$	$H_2$
низкая	0,2	0,2
средняя	0,5	0,5
высокая	0,8	0,8

Каждый информационный поток (ИП<sub>*k*</sub>;  $k = \overline{1, K}$ ) относится к двум конечным пунктам сети и описывается следующими числовыми характеристиками: интенсивность поступления первичных вызовов ( $\lambda_k$ ); потребность в скорости передачи данных ( $b_k$ ), измеряемая количеством базовых передаточных единиц (Basic Bandwidth Unit — BBU). Для всех рассматриваемых ИП ( $K = 27$ ) значения  $\lambda_k$  принадлежат диапазону от 1,5 до 25,5.

4.2. *Сведения о параметрах характеристик при повторных вызовах.* В качестве набора параметров, характеризующих поведение абонента при неудачной попытке получить доступ к требуемой услуге, будем использовать интенсивность повторения ( $\mu$ ), а также вероятности повторения — после самой первой попытки ( $H_1$ ) и после любой по счету повторной попытки ( $H_2$ ). С точки зрения настойчивости абонентов рассматривались три режима работы сети (табл. 3), причем для всех этих режимов  $\mu = 1$ .

Зададим количество серий и опытов в каждой серии — 20 и 500 000 соответственно. В данном примере сеть нагружена не очень сильно, и для некоторых ИП вероятность того, что заявка получает отказ в обслуживании, составляет менее 0,1 %. Поэтому для получения достаточно стабильных результатов нужно заранее выбирать значительное число опытов, чтобы величина отброшенных заявок исчислялась хотя бы в десятках.

В результате проведения серий экспериментов для различных режимов работы МСС были получены результаты, которые для удобства анализа представлены в виде диаграммы (рис. 9).

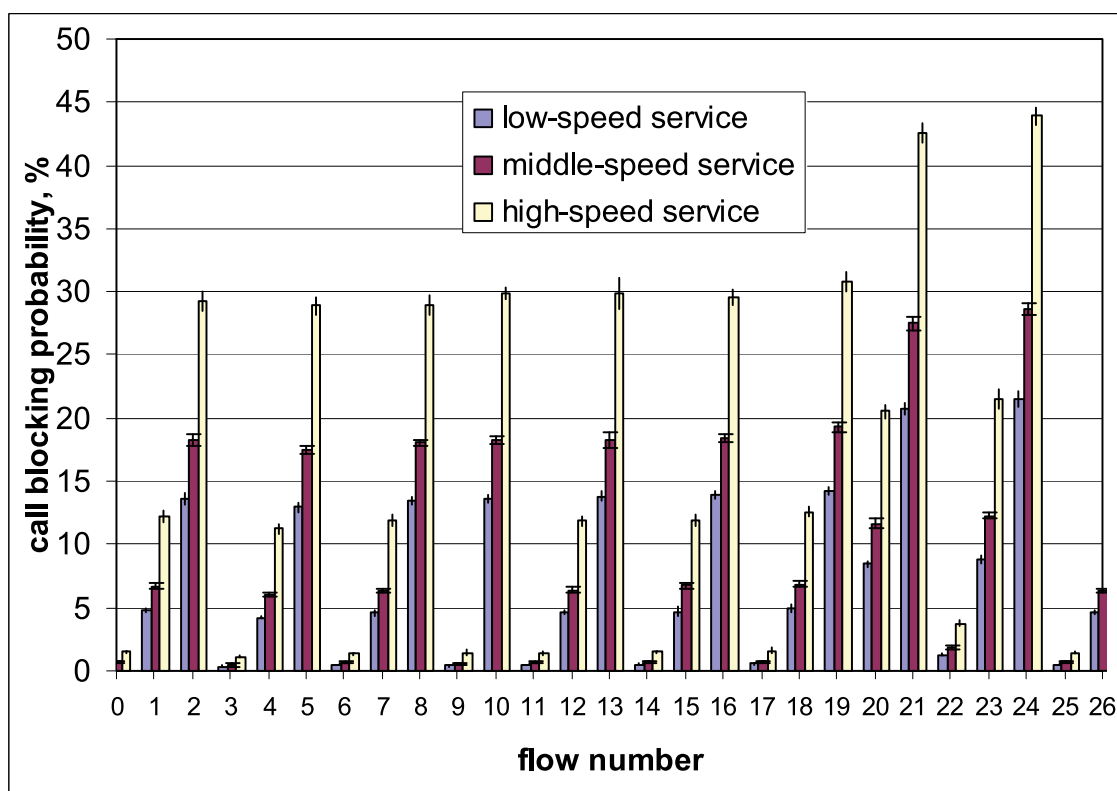


Рис. 9. Результаты эксперимента

Таблица 4

Итоговые значения по всем группам

Требование канального ресурса	Среднее по группе	
	Средняя/низкая	Высокая/средняя
1	1,45	2,15
2	1,40	1,82
3	1,34	1,60
Среднее	1,39	1,86

**5. Обсуждение результатов проведенного исследования по влиянию повторных вызовов.** Во-первых, стоит отметить, что характеристики работы сети при наличии повторных вызовов отличны от случая, в котором они отсутствовали. Это отчетливо прослеживается уже при сравнении результатов предыдущего эксперимента с результатами работы сети, полученными в случае, когда вероятность повторного вызова (параметр настойчивости абонента) еще относительно низка.

Для более детального изучения характера влияния повторных вызовов на качество функционирования сети были подготовлены вспомогательные промежуточные расчеты, чтобы сравнить вероятности потерь при разных уровнях настойчивости для всех информационных потоков, сгруппированных по величине потребляемого ими канального ресурса.

Анализируя значения, можно пронаблюдать, что первоначальное увеличение вероят-

ности повторения попытки вызова от 0,2 до 0,5 увеличивает вероятность блокировки поступившего вызова в среднем на 39 % для всех ИП. Дальнейший рост настойчивости абонентов (вероятности  $H_1$  и  $H_2$  достигают 0,8) увеличивает потери поступающих вызовов еще на 86 %. В среднем по всем ИП прослеживается нелинейное возрастание величины потерь при линейном возрастании показателя настойчивости абонента. Особенно выражен рост для низкоскоростных ИП: более 200 % при переходе от средней настойчивости (0,5) к максимальной (0,8).

Представленные результаты демонстрируют существенное изменение картины занятости ресурсов МСС под влиянием повторных вызовов. Здесь наглядно прослеживается лавинообразный эффект положительной обратной связи, когда перегрузка сети приводит к увеличению интенсивности входных потоков (за счет существенной добавки повторных вызовов), что, в свою очередь, еще больше усиливает степень ухудшения качества обслуживания вызовов.

Таким образом, указанные негативные обстоятельства требуют усложнения используемых математических моделей с целью более достоверного описания процесса взаимодействия между сетью связи и абонентами. Альтернативным вариантом можно считать использование такого эффективного инструмента, как ИМ, при котором достаточно легко преодолеваются многие трудности численного анализа современных систем связи с учетом эффекта повторных вызовов и других важных факторов, обусловленных развитием инфокоммуникационных технологий.

## Список литературы

1. СТЕПАНОВ С. Н. Основы телетрафика мультисервисных сетей. М.: Эко-Трендз, 2010.
2. LIU M., BARAS J. S. Fixed Point Approximation for Multirate Multihop Loss Networks with State-Dependant Routing // IEEE/ACM Trans. on Networking .2004. V. 12. N 2. P. 361–374.
3. ШНЕПС М. А. Системы распределения информации. Методы расчета: справ. пособие. М.: Связь, 1979.
4. Теория сетей связи: Учебник для вузов связи. / под ред. Рогинского. М.: Радио и связь, 1981.

*Корчагин Вячеслав Павлович — аспирант  
Сибирского государственного университета  
телекоммуникаций и информатики;  
e-mail: tbox54@gmail.com*

*Дата поступления — 21.01.2015*