

**Міністерство освіти і науки України**  
**Запорізький державний університет**

До захисту допущений  
Зав. кафедрою

ММ та ІТ

\_\_\_\_\_  
(підпис)

Борю Сергій Юрійович

(прізвище, ім'я, по батькові)

\_\_\_\_\_  
(дата)

**КВАЛІФІКАЦІЙНА РОБОТА**  
**МАГІСТРА**

**МОДЕЛЮВАННЯ РОЗПОДІЛУ**  
**РЕСУРСІВ ЗА ДОПОМОГОЮ АУКЦІОНІВ**

Виконав	<u>Холод Сергій Олександрович</u>		
ст. групи	М-1 (шифр)	(підпис і дата)	С. О. Холод (ім'я, по батькові, прізвище)
Керівник	доцент (посада)	(підпис і дата)	В. А. Єрмолаєв (ім'я, по батькові, прізвище)
Нормоконтролер		(підпис)	(ім'я, по батькові, прізвище)



## **ЗАВДАННЯ**



## РЕФЕРАТ

Квалификационная работа бакалавра: 42 с., 9 рис., 1 приложение, 22 источника.

Объект исследования - модели и алгоритмы распределенного планирования в динамических системах с элементами автономного поведения.

Цель работы - исследование и анализ методов распределенного планирования в динамических системах с элементами автономного поведения; разработка модели, динамических алгоритмов планирования, не основывающихся на знании внутренних предпочтений субъектов системы.

Методы исследования - теоретические методы построения алгоритмов и мультиагентских систем; методы дискретной математики; методы исследования операций; методы математического программирования.

Данная квалификационная работа магистра посвящена исследованию и разработке моделей и алгоритмов распределенного планирования в сообществах независимых субъектов, действующих автономно, используемых для моделирования процессов распределения ресурсов. Главной задачей данной работы является разработка алгоритма распределенного планирования, не основывающегося на знании предпочтений субъектов, участвующих в процессе планирования, и автономное поведение последних. Алгоритм планирования, предложенный в этой работе, основан на алгоритме «сетевого планирования».

АЛГОРИТМ, АГЕНТ, МУЛЬТИ-АГЕНТНАЯ СИСТЕМА,  
ПЛАНИРОВАНИЕ, АУКЦИОН, АВТОНОМНОЕ ПОВЕДЕНИЕ,  
МАКРОМОДЕЛЬ.

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>7</b>
<b>1. ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ .....</b>	<b>9</b>
1.1. Задачи моделирования компьютерных сетей.....	9
1.2. Цели моделирования вычислительных сетей .....	10
1.3. Особенности моделирования вычислительных сетей .....	12
1.3.1. Методы расчета характеристик компьютерных сетей .....	12
1.3.2. Модель и уровни OSI.....	16
1.3.3. Программные системы моделирования компьютерных сетей .....	19
1.4. Диакоптический подход моделирования сложных систем .....	22
1.5. Постановка задачи.....	24
<b>2. РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ КОМПЬЮТЕРНОЙ СЕТИ .....</b>	<b>25</b>
2.1. Структура модели сети как сложной системы .....	25
2.2. Основные предположения и допущения .....	28
2.3. Математическое описание элементов сложной системы .....	29
2.4. Разработка модели сопряжения компонент системы .....	35
2.5. Разработка модели взаимодействия компонент системы .....	38
<b>3. РАЗРАБОТКА АЛГОРИТМИЧЕСКОЙ МОДЕЛИ КОМПЬЮТЕРНОЙ СЕТИ .....</b>	<b>41</b>
3.1. Разработка концептуальной модели мониторинга системы .....	41
3.2. Разработка алгоритмической модели монитора системы .....	42
3.3. Разработка алгоритмических моделей компонент .....	47
3.4. Получение и анализ результатов моделирования .....	49
<b>ВЫВОДЫ .....</b>	<b>52</b>
<b>СПИСОК ССЫЛОК.....</b>	<b>54</b>

## ВВЕДЕНИЕ

Успешная работа многих организаций и компаний сегодня напрямую зависит от средств коммуникаций. Большую роль в деловой жизни стали играть Internet и мультимедиа. Поэтому очень важно сделать правильный стратегический выбор на пути развития сети своего предприятия. Чтобы наилучшим образом подобрать сетевое оборудование и программное обеспечение необходимо иметь разноплановую информацию о сетевых технологиях, знать их возможности и уметь оценивать стоимость. Однако во многих случаях этих знаний оказывается недостаточно для создания сети с желаемыми параметрами. Дело в том, что современные компьютерные сети представляют собой сложные гетерогенные системы с множеством взаимодействующих компонент. Следовательно, для получения приемлемых оценок параметров будущей сети недостаточно знать параметры аппаратного и программного обеспечения, нужно также учитывать их взаимодействие в процессе работы сети.

Рассмотрим такой пример. Одно из отделений организации намеревается оборудовать все свои рабочие места и персональными компьютерами и настроить их для работы в локальной сети с сервером. Конечно, можно, основываясь на опыте какой-либо родственной организации, которая уже проделала подобную работу, оценить примерную загрузку, генерируемую каждым ПК, и на этом основании оценить требуемую производительность всей локальной сети и сервера в частности. Однако такой подход позволяет получить только качественные оценки будущей сети, основанные на интуитивных предположениях.

Таким образом, при проектировании новой сети или модернизации уже существующей возникает необходимость в применении научно обоснованных методов получения характеристик системы, позволяющих разработчику сделать те или иные выводы о целесообразности будущих действий.

Одним из методов, позволяющих проводить априорную оценку сети до ее развертывания, является математическое моделирование.

Для оценки поведения системы практически на всем диапазоне загрузки может быть использован аналитический метод. При его практическом применении приходится решать набор уравнений, после чего удается получить параметры, необходимые для оценки системы (время ответа, пропускную способность и т. д.).

Однако, когда нужно построить модель проанализировать сеть со сложной топологией, аналитическая модель становится слишком грубым приближением к действительности. Тогда на первый план выступает имитационное моделирование. В имитационной модели поведение компонент системы описывается набором алгоритмов (программ), которые реализуют ситуации, возникающие в реальной системе.

Существуют специальные, ориентированные на моделирование вычислительных сетей программные системы, в которых процесс создания модели упрощен. Это ПО само генерируют модель сети на основе исходных данных о ее топологии и других характеристиках. Такие системы имеют ряд неоспоримых достоинств, например, поддержка большого числа устройств разных производителей, графические средства для построения моделей и визуализации процесса моделирования, широкий спектр отслеживаемых параметров и т.д. Однако им же присущи и несколько характерных недостатков, а именно: 1) невысокая устойчивость моделирующих алгоритмов; 2) замкнутость системы моделирования и, как следствие, ограниченные возможности по включению в состав модели компонент, разработанных сторонними производителями; 3) относительно высокая стоимость; 4) фиксированная степень детализации модели сети.

В настоящее время, по-прежнему есть необходимость в создании методов и алгоритмов моделирования компьютерных сетей, позволяющих разработать ПО полностью или частично указанные недостатки систем моделирования сетей.



# 1. ОСОБЕННОСТИ МОДЕЛИРОВАНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ

## 1.1. Задачи моделирования компьютерных сетей

В условиях стремительного роста информационного обмена в современных сетях часто возникает необходимость предсказания характеристик проектируемой вычислительной сети, последствий изменений в системе и т.д. Последствия могут оцениваться с точки зрения влияния на производительность, время ответа сети, доступность тех или иных сервисов и пр. Желательно также проводить априорную оценку производительности сети до ее развертывания. Представим себе следующую ситуацию. В организации установлено определенное количество рабочих станций, подключенных к одной локальной сети. Руководитель недавно сформированного отделения организации собирается подключить новые рабочие станции своих сотрудников к действующей сети. Перед всей организацией сразу встает вопрос – сможет ли существующая локальная сеть справиться с возросшей нагрузкой или для этого отделения необходимо будет создавать вторую локальную сеть и объединить обе сети мостом? При использовании интерактивных приложений реального времени в качестве основной оценочной характеристики обычно используется время ответа сети (иногда оно называется временем реакции сети). В других случаях ориентируются на пропускную способность сети [1].

Существуют и другие случаи, когда достаточно сложно быстро получить ответ на вопрос о том, насколько возрастет нагрузка на сеть при тех или иных изменениях, и справится ли с ней сеть. С точки зрения проектирования сети это означает, что не существует четкого однозначного метода, позволяющих на основе существующих требований к сети вычислить параметры и конфигурацию будущей системы.

Рассмотрим другой пример. Одно из отделений организации намеревается оборудовать все свои рабочие места и персональными компьютерами и настроить их для работы в локальной сети с сервером. Конечно, можно, основываясь на опыте какой-либо родственной организации, которая уже проделала подобную работу, оценить примерную загрузку,

генерируемую каждым ПК, и на этом основании оценить требуемую производительность всей локальной сети и сервера в частности. Естественно, основным критерием при оценке совокупности параметров сети в данном случае является её производительность в целом. Однако такой подход позволяет получить только качественные оценки будущей сети, основанные на интуитивных предположениях.

Таким образом, при проектировании новой сети или модернизации уже существующей возникает необходимость в применении научно обоснованных методов получения характеристик системы, позволяющих разработчику сделать те или иные выводы о целесообразности будущих действий.

## **1.2. Цели моделирования вычислительных сетей**

Рассмотрим, для чего могут использоваться системы моделирования. Даже грубый анализ позволяет выделить шесть случаев, к которым (хотя и с натяжками) могут быть сведены практически все разумные потребности в системах моделирования вычислительных сетей.

Чаще всего пользователь системы моделирования имеет следующие цели.

1. Получение числовых характеристик динамики работы модели сети за некоторое время.
2. Параметризация модели по определенному показателю и моделирование работы сети при некоторых значениях параметра (нагрузка на процессор, использование канала, задержка сообщения и прочее).
3. Составление и анализ общих агрегированных отчетов по числовым характеристикам работы модели сети.
4. Нахождение "узких мест" в проекте сети на основе анализа динамики работы ее модели.
5. Выработка рекомендаций к модернизации проекта сети и определение возможных этапов будущего развития (определение оптимальной топологии, адекватный выбор сетевого оборудования, замена ПО и т. д.).
6. Аргументация правильности некоторого "интуитивного" представления о работе проектируемой или модернизируемой сети и использование отчета о

моделировании как иллюстрации при подготовке технической документации.

В соответствии с поставленными целями разработчику проекта могут потребоваться самые различные характеристики как компьютерной сети в целом, так и ее отдельных компонент. В процессе моделирования можно выяснить следующие параметры:

- Предельные пропускные способности различных фрагментов сети и зависимости потерь пакетов от загрузки отдельных станций и внешних каналов.
- Время отклика основных серверов в самых разных режимах, в том числе таких, которые в реальной сети крайне нежелательны.
- Влияние установки новых серверов на перераспределение информационных потоков (Proxy, Firewall и т.д.).
- Решение оптимизации топологии при возникновении узких мест в сети (размещение серверов, DNS, внешних шлюзов, организация опорных каналов и пр.).
- Выбор того или иного типа сетевого оборудования (например, 10BaseTX или 100BaseFX) или режима его работы (например, cut-through, store-and-forward для мостов и переключателей и т.д.).
- Выбор внутреннего протокола маршрутизации и его параметров (например, метрики).
- Определение предельно допустимого числа пользователей того или иного сервера.
- Оценка необходимой полосы пропускания внешнего канала для обеспечения требуемого уровня QoS.
- Оценка влияния мультимедийного трафика на работу локальной сети, например, при подготовке видеоконференций.

Следует иметь в виду, что для моделирования поведения реальной сети, надо знать все ее рабочие параметры: длины кабеля от концентратора до

конкретной ЭВМ, задержки используемых кабелей, задержки концентраторов и т.д. Параметры могут быть определены и прямым измерением.

На модели можно опробовать широковещательный шторм или реализовать режим коллапса (для Ethernet), что вряд ли кто-то может себе позволить в работающей сети.

### **1.3. Особенности моделирования вычислительных сетей**

При моделировании компьютерных сетей мы имеем дело с такими объектами, как реальная работающая сеть, отдельные устройства сети, каналы связи, компьютерные модели отдельных устройств, протоколов, каналов и сетей, модели нагрузок (расписания, объемы информации и сценарии ввода информации в модель сети), числовые параметры сети и отчеты, снятые, например, системой управления сетью, отчеты о работе модели сети при тех или иных нагрузках. В связи с вышесказанным проблемы, связанные с понятием моделирования сети, можно разбить на две группы: построение моделей сетевых устройств, каналов и сети; работа с моделью, анализ и интерпретация результатов моделирования.

#### *1.3.1. Методы расчета характеристик компьютерных сетей*

В этом разделе рассматриваются математические инструменты построения моделей сетевых устройств, каналов передачи данных и сети в целом, излагаются наиболее важные особенности и сферы применения указанных средств математического моделирования. На сегодняшний день можно выделить четыре основных способа получения данных о действующей сети или расчета характеристик проектируемой сети:

1. Проведение анализа производительности сети после её внедрения, основываясь на значениях показателей, которые актуальны в данном конкретном случае;
2. Выполнение простой оценки работоспособности будущей системы, основанной на существующем опыте разработки и построении подобных сетей;

3. Разработка и применение аналитической модели;
4. Создание имитационной модели и основанной на ней программы, воспроизводящей поведение сети.

Первый вариант предполагает пассивную позицию разработчика сети. «Разработчик» просто ожидает результатов своей деятельности. Полученным результатом, как правило, оказываются недовольны и пользователи, и руководители организации. Их можно понять – они понесли неоправданные затраты, но так и не получили сети с желаемыми параметрами.

Второй вариант может дать, как правило, лучшие результаты. При анализе будущей сети на основании имеющегося опыта можно увидеть, что при наличных возможностях (в том числе финансовых) и ограничениях бессмысленно ожидать, что сеть будет удовлетворять тем или иным требованиям. То есть этот метод позволяет с достаточной уверенностью сказать, что *не сможет* сделать проектируемая сеть. С точки зрения выполнения предъявляемых требований, метод, основывающийся на опыте, может дать только достаточно расплывчатые предположения, носящие качественный характер. Абсолютно бессмысленно пытаться выполнять на основе этого метода более или менее точную количественную оценку необходимых параметров. Другая проблема, связанная с этим подходом, заключается в том, что поведение большинства систем при изменении загрузки будет не таким как ожидалось. Если существует среда в которой есть разделяемые каналы связи, то производительность такой системы, как правило, экспоненциально уменьшается при увеличении нагрузки. В результате наблюдается расхождение ожидаемых значений и наблюдаемых.

Для проведения оценки поведения системы практически на всем диапазоне загрузки может быть использован аналитический метод. При его практическом применении приходится решать набор уравнений, после чего удастся получить параметры, необходимые для оценки системы (время ответа, пропускную способность и т. д.).

Наиболее полное исследование можно провести в том случае, когда будут получены явные зависимости, связывающие искомые величины с параметрами системы и начальными условиями её изучения. Однако это удастся выполнить только для сравнительно простых систем. Для сетей со сложной топологической структурой (наличие коммутаторов, повторителей, нескольких серверов разной производительности, мостов и т.д.) при выводе формул, на которых основывается аналитическая модель и которые используются для расчета интересующих нас параметров приходится идти на упрощения представления реальных явлений. Упрощения дают возможность описать поведение сети и представить взаимодействие между её компонентами. Это позволяет изучить хотя бы некоторые общие свойства сложной системы, например оценить ее устойчивость. Для построения аналитических моделей имеется мощный математический аппарат (алгебра, функциональный анализ, теория вероятностей, математическая статистика, теория массового обслуживания и т. д.).

Одним из таких математических инструментов, предназначенных для разработки сетевых и коммуникационных структур, может быть аналитическая модель основанная на *теории очередей* (теории *массового обслуживания (МО)*). Существует расширение этой теории, называемое сетями МО. Сеть МО представляет собой совокупность конечного числа обслуживающих центров, в которой циркулируют сообщения, переходящие в соответствии с маршрутной матрицей из одного центра в другой. Под центром обслуживания понимают систему МО. Применение теории сетей МО для анализа компьютерных сетей подробно изложено в [2]. Отметим, что использование теории очередей дает достаточно точную оценку, которая, в большинстве случаев, хорошо соответствует действительности. Недостатком теории очередей, является то, что как отмечено выше, для применения этой теории при построении модели компьютерной сети необходимо принять определенные допущения. Для сравнительно простых сетей оказывается, что эти допущения вполне оправданы, а получающие результаты близки к тем, которые

получаются при программном моделировании сети с такими же параметрами. Преимуществом теории очередей по сравнению с имитационным моделированием является то, что анализ системы может быть выполнен за сравнительно короткий срок (для большинства реальных ситуаций) по сравнению со сроками построения программной модели.

Однако, когда явления в системе настолько сложны и многообразны, что аналитическая модель становится слишком грубым приближением к действительности, то исследователь вынужден использовать имитационное моделирование. В имитационной модели поведение компонент системы описывается набором алгоритмов (программ), которые реализуют ситуации, возникающие в реальной системе. Моделирующие алгоритмы позволяют по исходным данным, содержащим сведения о начальном состоянии системы, и фактическим значениям параметров сети отобразить реальные явления в системе и получить сведения о возможном поведении сети для данной конкретной ситуации. На основании этой информации исследователь может принять соответствующие решения. Имитируя различные реальные ситуации с помощью программы, моделирующей поведение сети, исследователь получает возможность решения таких задач, как оценка эффективности различных принципов управления системой, сравнения вариантов топологии (структуры) сети, определения степени влияния изменений параметров системы и начальных условий имитации на показатель эффективности функционирования сетевых приложений. Отметим при этом, что предсказательные возможности имитационного моделирования значительно меньше, чем у аналитических моделей.

Однако имитационные модели (ИМ) наряду с характерными для них достоинствами имеют ряд существенных недостатков. Разработка хорошей ИМ часто обходится дороже создания аналитической модели и требует больших временных затрат. Иногда может показаться, что ИМ точно отражает реальное положение дел в системе, а в действительности это не так. Р. Шеннон показал в [3], что ИМ в принципе не точна и мы не в состоянии измерить степень этой

неточности. Тем не менее имитационное моделирование является одним из наиболее широко используемых методов при решении задач синтеза и анализа сложных систем. Из достоинств имитационного моделирования выделим: возможность описания поведения компонент системы на высоком уровне детализации, отсутствие ограничений на вид зависимостей между параметрами ИМ и состоянием внешней среды, возможность исследования динамики взаимодействия компонент во времени и пространстве параметров системы. Указанные достоинства обеспечивают имитационному методу широкое распространение.

### *1.3.2. Модель и уровни OSI*

Все устройства, работающие в одной сети, должны общаться на одном языке – передавать данные в соответствии с общеизвестным алгоритмом в формате, который будет понят другим устройством. Различия во взглядах разработчиков на фундаментальные основы сети могут привести к несовместимости устройств.

Взаимодействие устройств подразумевает, что все устройства следуют общепризнанным правилам. В 1984 году с целью упорядочения описания принципов взаимодействия устройств в сетях Международная организация по стандартизации (International Organisation of Standardization – ISO) предложила семиуровневую эталонную коммуникационную модель «Взаимодействие Открытых Систем» (Open System Interconnection, OSI). Модель OSI стала основой для разработки стандартов на взаимодействие систем. Она определяет только схему выполнения необходимых задач, но не дает конкретного описания их выполнения. Это описывается конкретными протоколами или правилами, разработанными для определенной технологии с учетом модели OSI. Уровни OSI могут реализовываться как аппаратно, так и программно.

Эталонная модель OSI сводит передачу информации в сети к семи подзадачам. Каждая из них соответствует своему строго определенному уровню модели OSI. Тем не менее, в реальности некоторые аппаратные и программные средства отвечают сразу за несколько уровней.



Существует семь основных уровней модели OSI (см. табл. 1.1). Первый уровень представляет собой аппаратную среду передачи сигнала. Каждый уровень предоставляет услуги для более высокого уровня. Седьмой уровень обслуживает непосредственно пользователей.

Таблица 1.1 – Уровни эталонной модели OSI

Уровень	Ключевое слово	Данные	Ответственность
Прикладной (Application)	Разделение	Сообщение	Предоставляет сетевой адрес
Представления (Presentation)	Форматирование	Пакет	Трансляция данных и файлов; шифрование данных; сжатие данных
Сеансовый (Session)	Диалог	Пакет	Управление сессией; диалог; контроль за ошибками; обработка транзакций; поддержка вызова удаленных процедур (RPC)
Транспортный (Transport)	Надежность	Сегмент, дейтаграмма, пакет	Надежность передачи; гарантированная доставка; мультиплексирование сессий верхнего уровня
Сетевой (Network)	Дейтаграмма	Дейтаграмма	Маршрутизация логических адресов; создание и ведение таблиц маршрутизации; фрагментация и сборка данных; неориентированная на соединение и ненадежная доставка
Канальный (Data Link)	Кадр	Кадр, пакет	Окончательная доставка по физическому адресу устройства; синхронизация кадров; доступ к среде передачи
Физический (Physical)	Биты	Биты	Синхронизация битов; сигнализация аналоговая или цифровая

Проходящие через уровни данные имеют определенный формат. Сообщение, как правило, делится на заголовок и информационную часть. Конкретный формат зависит от функционального назначения уровня на котором информация находится в данное время (табл. 1.1). Например, на сетевом уровне информационный блок состоит из сетевого адреса и

следующими за ними данными. Данные сетевого уровня, в свою очередь, могут содержать заголовки более высоких уровней. И, наконец, физический уровень не нуждается в присоединении заголовков.

Основная идея модели OSI заключается в том, что одни и те же уровни на разных системах, не имея возможности связываться непосредственно, должны работать по абсолютно одинаковым алгоритмам. Одинаковым должен быть и сервис между соответствующими уровнями различных систем. Нарушение этого принципа может привести к тому, что информация, посланная от одной системы к другой, после всех преобразований не будет идентична исходной.

Модель OSI описывает путь информации через сетевую среду от одной прикладной программы на одном компьютере до другой программы на другом компьютере. При этом пересылка информации происходит вниз через все уровни системы. Уровни на разных системах не могут общаться между собой напрямую. Это умеет только физический уровень. По мере прохождения информации вниз она преобразуется к виду, удобному для передачи по физическим каналам связи. После получения адресатом информации она проходит через все уровни вверх. По мере прохождения информация преобразуется в первоначальный вид.

В конкретных случаях может возникать потребность в реализации лишь части функций, тогда соответственно в сети имеется лишь часть уровней. Так, в простых (неразветвленных) ЛВС отпадает необходимость в средствах сетевого и транспортного уровней. В то же время сложность функций канального уровня делает целесообразным его разделение в ЛВС на два подуровня: *управление доступом к каналу* (MAC - Medium Access Control) и *управление логическим каналом* (LLC - Logical Link Control). К подуровню LLC в отличие от подуровня MAC относится часть функций канального уровня, не связанных с особенностями передающей среды.

Эталонная модель OSI не определяет реализацию сети. Она только описывает функции каждого уровня и общую схему передачи данных в сети. Она служит основой сетевой стратегии в целом.

### *1.3.3. Программные системы моделирования компьютерных сетей*

В последнее время в практику проектирования и планирования предоставления сетевых услуг стали внедряться коммерческие программные продукты, предназначенные для моделирования процессов передачи информации в различного типа локальных и глобальных сетях. В настоящее время имеется уже более десятка таких систем моделирования сетевых процессов. (Обзор некоторых из них приведен, например, в [4], [5]). Хотя эти системы различаются как по объектам моделирования (корпоративные, локальные, глобальные сети, ориентация на определенных производителей сетевого оборудования и т. д.), так и по функциональным возможностям, идеология их использования и принципы построения могут быть рассмотрены с некоторых общих позиций.

Сразу скажем, что речь идет не о моделировании сетей, а о моделировании процессов. Программные комплексы первого типа носят, как правило, справочный характер и дают возможность проектировщику проверить совместимость фрагментов сети (сетевых устройств, каналов, протоколов) и просчитать некоторые характеристики сети. Во втором же случае речь идет о том, чтобы смоделировать динамику работы сети при разных режимах ее работы. То есть с понятием моделирования ассоциируется здесь целый комплекс задач, которые будут рассмотрены в контексте имитационного моделирования с применением диакоптического подхода следующих разделах данной работы.

Кроме того, в основу разных систем моделирования могут быть заложены разные подходы. К примеру, в одном случае программа базируется на аналитическом моделировании с применением теории очередей и/или дифференциальных уравнений, а в другом – на имитационной модели, детально воспроизводящей поведение сети и отдельных её составляющих. В этом разделе мы коснемся программных систем использующих принципы имитационного моделирования с дискретным характером моделируемого процесса. Типичными примерами такого рода систем являются довольно

известная в нашей стране система COMNET III компании CACI, а также система QASE компании AST. Использование компонент COMNET III для моделирования и анализа сетей описано в [6].

Программы имитационного моделирования сети используют в своей работе информацию о пространственном расположении сети, числе узлов, конфигурации связей, скоростях передачи данных, используемых протоколах и типе оборудования, а также о выполняемых в сети приложениях.

Обычно имитационная модель строится не с нуля. Существуют готовые имитационные модели основных элементов сетей: наиболее распространенных типов маршрутизаторов, каналов связи, методов доступа, протоколов и т.п. Эти модели отдельных элементов сети создаются на основании различных данных: результатов тестовых испытаний реальных устройств, анализа принципов их работы, аналитических соотношений. В результате создается библиотека типовых элементов сети, которые можно настраивать с помощью заранее предусмотренных в моделях параметров.

Системы управления сетью обычно рекламируются как всеохватывающие и всемогущие. Средства моделирования работы сети назвать таковыми никак нельзя. Разброс цен на эти средства составляет от 129 дол. (за работающую под Windows программу LANModel от Network Performance Institute) до 40 тыс. дол. (за COMNET III от CACI, которая может работать под Windows 95, Windows NT и Unix).

Однако ни одно из средств не способно охватить все задачи, поэтому если необходимо смоделировать сеть и проанализировать ее работу, придется покупать несколько продуктов. К примеру, на сегодняшний день, ни один из известных программных продуктов не способен моделировать поведение сети с учетом нескольких верхних уровней модели OSI. Это связано не только с большими трудозатратами на создание такой модели, но и с трудностями создания программы, обладающей широкими возможностями подстройки под конкретную сеть, использующую различные протоколы, сетевое оборудование

разных производителей и т.д. Имеются также заметные различия между продуктами, которые, как утверждается, решают одни и те же задачи.

В [4] рекомендуется использовать коммерческие пакеты моделирования сетей для проактивного планирования нагрузки, для нахождения узких мест в сети, моделирования работы клиент-серверных приложений. Среди выводов любопытны следующие. Автор утверждает, что "после выявления проблем с нагрузкой и производительностью все готово для создания модели вашей среды". Конечно. Под моделью можно понимать очень разные вещи. Кроме того, если проблемы с нагрузкой и производительностью решены без всякой модели, то цели моделирования требуют уточнения. Из статьи ясно, что автор предполагает верифицировать модель на реальной сети, а затем использовать эту модель для прогнозирования поведения сети при изменении параметров модели. Это понятная и естественная задача, но как быть тем, кто в силу разных причин не может провести проверку адекватности модели на реальной сети?

В обзоре систем моделирования [7] автор очень сдержан в оценках. Здесь описываются свойства, сравниваются качества и даются некоторые рекомендации по выбору и применению инструментариев моделирования. Однако автор не забыл отметить и, например, два следующих фактора. "Ни один из продуктов нельзя рассматривать как полностью готовое к употреблению средство, способное в точности смоделировать работу существующей или даже вновь спроектированной сети. Необходимо потратить значительные средства на обучение, прежде чем станут возможными построение корректных моделей и интерпретация полученных результатов. Затем понадобится еще в течение шести-девяти месяцев непрерывно подстраивать модель, и только после этого она будет хотя бы приблизительно приведена в соответствие с действительностью". Но далее следует убийственное замечание о цене этой самой приблизительности". К сожалению, одна-две ошибки в начальной информации могут испортить все".

Эта фраза подтверждает скептицизм по отношению к заявлениям поставщиков новых услуг, обещающих за короткий срок и небольшие деньги построить модель вашей сети и решить проблемы с ее узкими местами.

Перечисленные в п. 1.2. цели моделирования компьютерных сетей предъявляют различные требования к программам. В одних случаях достаточно провести моделирование на физическом (MAC) уровне, в других нужен уже уровень транспортных протоколов (например, UDP и TCP), а для наиболее сложных задач нужно воспроизвести поведение прикладных программ. Все это должно учитываться при выборе или разработке моделирующей программы. Также нужно принять во внимание, что ЭВМ, на которой выполняется программа моделирования, должна в той или иной мере воспроизвести действия всех компонент моделируемой сети. Таким образом, машина эта должна быть достаточно быстродействующей и, несмотря на это, моделирование одной секунды работы сети может занять при определенных условиях не один час.

Из всего вышесказанного можно сделать вывод, что сегодняшние программные продукты моделирования компьютерных сетей далеки от совершенства и по-прежнему есть необходимость в создании продукта, обладающего возможностями производить моделирование на одном или нескольких различных уровнях модели OSI, поддерживать библиотеки сетевого оборудования разных производителей и варьировемыми затратами машинного времени в зависимости от точности моделирования и устойчивости результатов.

#### **1.4. Диакоптический подход моделирования сложных систем**

Построение математической модели непосредственно для сколько-нибудь сложного объекта требует немалого искусства. Зачастую «барьер сложности», когда человек не в состоянии мысленно охватить всевозможные переплетения событий и явлений процесса функционирования моделируемой системы, или рентабельности приводят к тому, что создание модели оказывается практически нецелесообразным. Одним из способов преодоления указанного барьера

является представление объекта как сложной системы, которую затем расчлениают на достаточно простые части и занимаются изучением частей с учетом их взаимодействия. Моделирование систем по частям – в этом и заключается принцип диакоптики.

Понятие сложной системы интуитивно ассоциируется с объектом, состоящим из большого числа элементов и подсистем с разветвленными многоуровневыми связями и сложным взаимодействием между ними. Если обобщить подобные представления, то мы приходим к концепции *сложной системы (СС)*, состоящей в следующем:

1. Рассматриваемая система (объект материального мира) может быть разделена (не обязательно единственным образом) на конечное число частей, называемых *подсистемами СС*; каждая подсистема в свою очередь может быть расчленена на конечное число более мелких подсистем и т. д. – до получения, в результате конечного числа шагов, таких частей, называемых элементами СС, относительно которых имеется договоренность, что в условиях данной задачи они не подлежат дальнейшему расчленению на части.
2. Элементы СС функционируют не изолированно друг от друга, а во взаимодействии, при котором свойства одного в общем случае зависят от условий определяемых поведением других элементов.
3. Свойства СС в целом определяются не только свойствами элементов, но и характером взаимодействия между элементами (две СС, состоящие из попарно одинаковых элементов, взаимодействие между которыми имеет различную структуру, рассматривается как различные СС).

Представляется достаточно очевидным, что гетерогенная информационно-вычислительная сеть как нельзя лучше соответствует концепции СС, поэтому дальнейшее рассмотрение компьютерных сетей будет проводиться именно с таких позиций.

Один из вариантов дискретного моделирования СС с использованием принципа диакоптики изложен в [8].

Для того, чтобы задать СС, необходимо (и достаточно) представить описание всех ее элементов описания взаимодействия между элементами. Ниже мы рассмотрим обе эти проблемы применительно к вычислительным сетям передачи данных.

### **1.5. Постановка задачи**

Цель данной работы – разработка модели и алгоритмов функционирования компьютерной сети как среды обмена данными между компонентами информационной системы. Основные функции указанной модели заключаются в имитации процессов поглощения, передачи и генерации информации в вычислительных сетях с возможностью получения сведений о системе в некоторые дискретные моменты времени. Назначение модели – предоставить пользователю возможность получить необходимые данные о работе уже существующей или проектируемой сети. В работе будут определены временные статистические параметры передачи пакетов между любыми компонентами исследуемой сети. Модель компьютерной сети позволит выяснить зависимость указанных временных характеристик передачи информации от настроек и свойств сетевого оборудования. В работе предполагается определить качественные зависимости времени передачи пакетов от пропускных способностей сетевых устройств.



## 2. РАЗРАБОТКА МАТЕМАТИЧЕСКОЙ МОДЕЛИ КОМПЬЮТЕРНОЙ СЕТИ

### 2.1. Структура модели сети как сложной системы

Компьютерную сеть в целом можно представить как динамическую систему. Под *динамической системой* понимается объект, находящийся в каждый момент времени  $t$  (из множества  $T$ ) в одном из возможных состояний  $z_t$  (из множества  $Z$ ) и способный переходить во времени из одного состояния в другое под действием внешних и внутренних причин.

Таким же способом может быть описан и каждый компонент сети. Что конкретно понимается под компонентом зависит от уровня декомпозиции модели.

Функционирование элемента, как динамической системы, представляется последовательностью перемещений состояния (изображающей точки) в пространстве состояний.

Выходные данные, выдаваемые одними элементами сети, поступая к другим элементам как входные данные, приводят к изменению поведения компонент. Получающееся при этом переплетение перемещений и случайных скачков состояний различных элементов сети составляет процесс функционирования СС во всем многообразии событий и явлений.

Несмотря на сложность процесса функционирования СС и необозримость всевозможных сочетаний общесистемных ситуаций, путь для количественного и качественного математический исследований не является полностью закрытым. Правда если в рамках традиционного подхода иметь в виду уравнения для достаточно богатого набора общесистемных характеристик, описывающих свойства изучаемого объекта, то для сложных моделей вычислительных сетей мы неизбежно столкнемся с трудностями составления и решении уравнений. Основным препятствием здесь оказывается сложность непосредственной формализации и математического описания общесистемных ситуаций.

Если воспользоваться принципом диакоптики для построения модели, то нам не требуется составлять уравнения относительно характеристик системы и на человека не возлагается непосильное бремя разложения сложного процесса функционирования системы в целом в последовательность простых явлений и событий доступных непосредственному математическому описанию. Умозрительный анализ остается инструментом, помогающим описать связи элементов в сети, а также формализовать процессы функционирования лишь элементов системы.

При моделировании информационной сети с применением принципа диакоптики представляется целесообразным использовать такой подход к расчленению системы на части:

- каждое устройство реальной сети представляется в виде элемента или объединения элементов модели системы;
- каждому элементу ставится в соответствие моделирующая программа – часть модели сети.

Отметим, что указанное соответствие является биекцией, т. е. каждый элемент моделируется одной подпрограммой, но, в общем случае, одна компонента может моделировать работу нескольких элементов.

Т.о. модель представляет собой совокупность взаимодействующих компонент, часть из которых моделирует работу фрагментов сети (сети образующих устройств), то есть, получив исходные данные, такая компонента производит требуемые вычисления, а результаты решения передает вовне (другим компонентам или на выход всей системы моделирования).

В качестве метода исследования компьютерной сети будет использовано имитационное моделирование. Это позволит при помощи составленных алгоритмов функционирования вычислительной сети производить расчеты нужных характеристик на ЭВМ.

Разработка имитационной модели компьютерной сети с использованием диакоптического подхода требует решения некоторого круга основных задач, а именно:

1. Задача составления моделей и базовых алгоритмов работы компонент системы, соответствующих концепции генерации, передачи и поглощения информации в вычислительной сети.
2. Задача построения модели представления топологии сети, т.е. информации о связях компонент системы.
3. Задача формирования модели взаимодействия составляющих вычислительной сети, методов и способов обмена информацией между компонентами в модели системы.
4. Задача разработки способов представления данных необходимых для моделирования. В эту задачу также можно включить разработку методов хранения и доступа к информации о системе и ее составляющих.
5. Задача создания модели программы управления системой, выполняющей мониторинг общесистемных процессов таких как активизация компонент, обеспечение доступа составляющих системы к необходимым данным, контроль использования ресурсов среды и прочее. Предназначение монитора – создать прозрачную среду взаимодействия составляющих модели и обеспечить совместную работу относительно независимых компонент как целостной системы.
6. Задача разработки модели взаимодействия монитора и компонент системы. Способы и методы этого взаимодействия могут отличаться от способов и методов взаимодействия компонент в системе.
7. Задача разработки алгоритмов и моделей элементов, выполняющих анализ, форматированный ввод и вывод информации, необходимой пользователю модели.

Более подробно каждая из вышперечисленных задач будет раскрыта в следующих разделах данной работы.

Отметим также, что поскольку построение модели компьютерной сети будет вестись с использованием диакоптического подхода, то решать указанные задачи следует в рамках требований указанных в п 1.4. Также решение этих задач должно проводится с учетом метода моделирования.

## 2.2. Основные предположения и допущения

Так как речь идет не о реальном процессе, а о его дискретном моделировании, то мы вынуждены сделать ряд предположений, которые будут использованы при построении модели системы.

*Предположение 1.* *Время* в модели дискретно, т. е. существует фиксированная величина  $\Delta t$ , зависящая от степени *детализации* модели сети, а любой временной промежуток кратен этой величине.

*Предположение 2.* Любое действие в моделируемой сети осуществляется за время, кратное  $\Delta t$ .

Что касается информации передаваемой между элементами модели системы, то представляется целесообразным принять следующие допущения.

*Предположение 3.* Данные между компонентами передаются не непрерывным потоком, а в виде *информационных сообщений* (или просто *сообщений*). Между передачей сообщений допускается наличие некоторого временного интервала (так называемого *перерыва*) в течении которого сообщения не передаются.

Согласно предположению 2, передача информационного сообщения может происходить только целое число промежутков  $\Delta t$ . То же справедливо и для перерывов.

*Предположение 4.* С достаточной для практических целей общностью любое сообщение циркулирующее между элементами СС, описывается при помощи конечного набора характеристик, т.е. представимо в виде

$$\mathbf{x} = (x_1, x_2, \dots, x_n), \quad (2.1)$$

где  $x_k \in X_k$ ,  $X_k$  – заданное множество,  $k = 1, 2, \dots, n$ .

Следовательно, все сообщения передающиеся в системе имеют один и тот же набор характеристик.

Каждому информационному сообщению  $\mathbf{x}$  ставится в однозначное соответствие натуральное число – *мера* сообщения. Мера является размерной величиной и измеряется в единицах количества информации (бит и его производные единицы). Мера сообщения  $\mathbf{x}$  обозначается как  $\mu(\mathbf{x})$ .

*Предположение 5.* Любое информационное сообщение в системе представляется в виде  $m$  частей, где  $m = \mu(x)$ .

Эти части мы в дальнейшем будем называть *следами* сообщений.

Каждый след информационного сообщения  $x$  характеризуется своим порядковым номером  $k$  и обозначается как пара  $(x, k)$ .

*Предположение 6.* Каждый след информационного сообщения имеет единичную меру.

Т.о. суммарная мера всех следов любого информационного сообщения равняется его мере.

*Предположение 7.* В каждый момент времени  $t$  в системе циркулирует некоторое неотрицательное целое число  $N$  информационных сообщений  $X_t$  и соответствующее множество следов  $T_t$ .

Указанные множества имеют структуру:

$$X_t = \begin{cases} \{x_i\}, & \text{если } N \in \mathbb{N} \\ \emptyset, & \text{если } N = 0 \end{cases} \quad (2.2)$$

и

$$T_t = \begin{cases} \{(x_i, k_i)\}, & \text{если } N \in \mathbb{N} \\ \emptyset, & \text{если } N = 0 \end{cases} \quad (2.3)$$

где  $i=1, \dots, N$ ;  $k_i=1, \dots, \mu(x_i)$ .

Список предположений, приведенных в этом пункте не является окончательным, однако последующие допущения носят более частный характер и основываются на предположениях 1 – 7.

### 2.3. Математическое описание элементов сложной системы

В работах [5,6] современные компьютерные сети с развитой топологией рассматриваются как являются гетерогенные системы со множеством компонент, выполняющих различные функции. Несмотря на кажущееся многообразие устройств и выполняемых ими задач, возможно выделить несколько характерных признаков, присущих любой функциональной составляющей информационно-вычислительной сети. Одним из таких

признаков является отношение компонент системы к потокам данных в сети. Поясним сказанное. В сеть подается информация и извлекается из нее. В реальности прикладные процессы заказывают и используют сетевые сервисы. Это соответствует процессам генерации, передачи и поглощения информации. Отсюда следует, что каждая компонента информационной сети обладает по крайней мере одной из ниже перечисленных способностей.

1. Генерация (создание) данных.
2. Передача (транзит) данных.
3. Прием (поглощение) данных.

В компьютерных сетях используется как специализированное оборудование, выполняющее только одну из указанных функций, так и устройства с возможностью функций всех трех типов (например, шлюз сети).

В модели компоненты обмениваются информацией с внешней средой посредством соответствующих входов и выходов, которые мы назовем *контактами*.

Контакт  $P_{ik}$  компоненты  $C_i$  представляет собой упорядоченную тройку

$$P_{ik} = (V_t, k, \varepsilon_t), \quad (2.4)$$

где  $V_t = V_t[P_{ik}] \subseteq T_t$  – содержимое контакта,  $k \in \mathbf{N}$  – номер контакта, уникальный для соответствующей компоненты,  $\varepsilon_t = \varepsilon_t[P_{ik}] \in \mathbf{N} \cup \{0\}$  – пропускная способность контакта. Смысл приведенных здесь понятий будет раскрыт ниже.

В случаях, когда не важно о контакте какой компоненты идет речь, индексы не будут указываться.

Множество входных контактов компоненты  $C_i$  обозначим  $Q_i$ , а множество выходных контактов –  $R_i$ .

*Предположение 8.* В общем случае любой контакт компоненты в каждый момент модельного времени может передать несколько следов информационных сообщений.

Каждому контакту системы поставим в однозначное соответствие неотрицательное целое число – его *пропускную способность*.

Если воспользоваться терминологией, применяемой в компьютерных сетях, то пропускная способность – это количество единиц информации, передаваемой устройством в единицу времени. В работе в указанное понятие вкладывается такой же смысл, в следствии чего пропускная способность измеряется в тех же единицах, что и мера информационных сообщений.

Пусть  $V_t[P]$  – множество следов передаваемых через контакт  $P$  в момент времени  $t$  – *содержимое контакта*. Тогда следующее предположение определяет основное свойство пропускной способности.

*Предположение 9.* Для каждого момента времени  $t$  суммарная мера информации передаваемой через контакт  $P$  не может быть больше его пропускной способности  $\varepsilon$ . Т.е. выполняется неравенство

$$m(V_t) \leq \varepsilon_t, \quad (2.5)$$

где  $m(\cdot)$  – мощность множества,  $V_t = V_t[P]$  – содержимое контакта  $P$ ,  $\varepsilon_t = \varepsilon_t[P]$  – пропускная способность контакта  $P$ .

С каждой компонентой  $C_k$  системы  $S$  связано некоторое множество следов  $S_t[C_k] = \{(x_i, k_{n_i})\} \subseteq T_t$ , где  $k_{n_i} \in \{1, \dots, \mu(x_i)\}$ ,  $n_i \leq \mu(x_i)$ ,  $x_i \in X_t$ . Это множество называется *хранилищем сообщений*. При получении следа он помещается в хранилище компоненты, а при отправке – удаляется из него.

При приеме следов через контакт  $P_{ik}$  они удаляются из множества  $V_t[P_{ik}]$  и помещаются в хранилище компоненты  $C_i$ . При отправке следов посредством контакта  $P_{il}$  они удаляются из хранилища компоненты и добавляются во множество содержимого контакта  $V_t[P_{il}]$ .

*Предположение 10.* Каждая компонента системы может создавать новое и удалять существующие информационные сообщения.

При создании информационного сообщения  $x$  в хранилище соответствующей компоненты помещаются следы  $(x, k)$ ,  $k=1, \dots, \mu(x)$ , а во множества  $X_t$  и  $T_t$  добавляются новое сообщение  $x$  и новые следы системы соответственно.

*Предположение 11.* Информационное сообщение может быть удалено из системы только в том случае если оно целиком находится в хранилище одной

компоненты. Т.е. компонента  $C_k$  может удалить информационное сообщение  $x$ , если справедливо равенство

$$S_i[C_k] \cap T(x) = T(x), \quad (2.6)$$

где  $T(x) = \bigcup_{k=1}^{\mu(x)} (x, k)$  - множество всех следов сообщения  $x$ .

При удалении информационного сообщения соответствующие следы удаляются из хранилища, а сообщение  $x$  – из множества сообщений системы  $X_i$ .

Как уже отмечалось выше, элемент компьютерной сети представлен в виде динамической системы.

Чтобы учесть внешние причины, необходимо рассмотреть взаимодействие динамической системы с внешней средой. Оно осуществляется посредством воздействия внешней среды на систему и системы на внешнюю среду. Такого рода воздействия осуществляются при помощи информационных сообщений. Следовательно процесс взаимодействия динамической системы с внешней средой можно рассматривать в рамках обмена сообщениями.

Динамическая система как математический объект содержит в своем описании следующие механизмы:

1. механизм изменения состояний под действием внутренних причин (без вмешательства внешней среды);
2. механизм приема входного информационного сообщения и изменения состояния под действием этого сообщения;
3. механизм формирования выходного сообщения (как реакция динамической системы на внутренние и внешние причины изменения состояний).

Однако такое задание динамической системы оказывается слишком общим и недостаточно определенным. Имея в виду цели машинного моделирования СС, можно обойтись частным случаем описания упомянутых механизмов.

Для этого выясним от чего зависит состояние элемента СС в каждый момент времени  $t$ .



В реальной компьютерной сети данные передаются не непрерывно, а своеобразными порциями – блоками информации. В зависимости от того, на каком уровне модели OSI рассматривается обмен данными в сети меняется и название этих блоков (см. табл. 1.1). Наиболее распространенный термин, употребляемый для обозначения «единицы» информации, передаваемой по сети – *пакет*. Пакет – это блок данных, в сети передачи данных, имеющий строго определенную структуру, включающую заголовок и поле данных. Настройки сетевого оборудования и сетевого программного обеспечения влияют на содержимое заголовка пакетов и их длину. Обратим внимание на то, что любые действия с блоком информации, выполняемые устройствами сети, происходят не мгновенно, а в течении некоторого промежутка времени. Длина этого промежутка в общем случае определяется следующими факторами: 1) содержимым пакета, а именно информацией находящейся в заголовке пакета и блоке данных; 2) характеристиками обрабатывающего устройства; 3) текущими значениями параметров; 4) состоянием устройства в предыдущие моменты времени. Эти же факторы влияют и на состояние устройства в текущий момент времени  $t$ .

Т.о. можно заключить что:

*Предположение 12.* Множество  $Z_t[C_i]$  описывающее состояние компоненты  $C_i$  в любой момент времени  $t$  является конечным.

Множество состояний компоненты может не содержать элементов.

Расчет состояния  $Z_{t+1}[C_i]$  компоненты  $C_i$  проводится по формуле

$$Z_{t+1} = \mathfrak{R}_i(S_t, Z_t, \dots, Z_{t-n}, t), \quad (2.7)$$

где  $\mathfrak{R}_i$  – оператор перехода компоненты  $C_i$  в новое состояние,  $Z_t = Z_t[C_i]$ ,  $Z_{t-1} = Z_{t-1}[C_i]$ , ...,  $Z_{t-n} = Z_{t-n}[C_i]$  – состояния компоненты  $C_i$  в моменты времени  $t, t-1, \dots, t-n$  ( $n < t$ ),  $S_t = S_t[C_i]$  – хранилище компоненты  $C_i$ ,  $t$  – текущее модельное время.

С помощью аналогичных рассуждений можно показать, что пропускные способности входных контактов компоненты  $C_i$  определяются при помощи выражения

$$\varepsilon_{t+1}[P_{ij}] = \mathfrak{R}_i(Z_t, j), \quad (2.8)$$

где  $\varepsilon_t[P_{ij}]$  – пропускная способность контакта  $P_{ij} \in Q[C_i]$ ,  $\mathfrak{X}_i$  – оператор пропускных способностей входов компоненты  $C_i$ ,  $j$  – номер контакта компоненты  $C_i$ .

Содержимое контакта  $P_{ij} \in R_i$  компоненты  $C_i$  определяется из соотношения

$$V_t[P_{ij}] = \wp_i(Z_t, j), \quad (2.9)$$

где  $V_t[P_{ij}]$  – содержимое контакта  $P_{ij} \in R[C_i]$ ,  $\wp_i$  – оператор выходов компоненты  $C_i$ ,  $j$  – номер контакта компоненты  $C_i$ .

В пункте 2.5 будет описано по каким законам устанавливаются пропускные способности входов компонент.

Примеры дальнейшей детализации операторов  $\mathfrak{X}_i$ ,  $\wp_i$  и  $\mathfrak{R}_i$  будет приведены в следующей главе данной работы. Заметим, что вид и свойства указанных операторов зависят от математического аппарата (математических схем) используемых для создания математической модели компонент СС.

Наибольшей популярностью в сфере системотехники, исследования операций и системного анализа пользуются следующие *типичные математические схемы*: обыкновенные дифференциальные уравнения, конечные автоматы, вероятностные автоматы, системы и сети массового обслуживания, а также сети Петри.

Для создания математической модели компонента сети может быть применена любая из указанных схем. В работе будут использованы алгоритмические модели основанные на теории конечных и вероятностных автоматов. Базовые алгоритмы функционирования некоторых устройств сети будут рассмотрены ниже в данной работе.

Подводя итоги сказанного в этом пункте, приведем общее математическое описание компонента системы, которое представим в виде следующего соотношения

$$C_i = (Q_i, R_i, S_t, Z_t, \mathfrak{X}_i, \mathfrak{R}_i, \wp_i), \quad (2.10)$$

где  $Q_i$  – множество входных контактов компоненты  $C_i$ ,  $R_i$  – множество выходных контактов компоненты  $C_i$ ,  $S_t = S_t[C_i]$  – хранилище компоненты  $C_i$ ,  $Z_t = Z_t[C_i]$  – текущее состояние компоненты  $C_i$ ,  $\mathfrak{X}_i$  – оператор перехода компоненты

$C_i$  в новое состояние,  $\mathfrak{X}_i$  – оператор установки пропускных способностей контактов компоненты  $C_i$ ,  $\wp_i$  – оператор содержимого контактов компоненты  $C_i$ .

#### 2.4. Разработка модели сопряжения компонент системы

Помимо уже сформулированных, для построения модели сопряжения компонент СС требуется ввести еще некоторые предположения.

*Предположение 13.* В частности, будем считать, что информационные сообщения передаются в системе независимо друг от друга по *идеальным каналам*, соединяющим соответствующие контакты компонент. Термин *идеальный канал* означает среду передачи информации, обладающую такими характеристиками: 1) бесконечная пропускная способность; 2) нулевая задержка перемещения 3) отсутствие искажения данных при передаче.

Обмен данными между компонентами возможен лишь в том случае, если в моделируемой СС фактически имеется (реализован) соответствующий канал.

*Предположение 14.* К каждому контакту подключается не более чем один канал. Т. о. каждой точке доступа одной компоненты может соответствовать не более одной точки доступа другой компоненты и наоборот.

Следовательно связи  $N$  компонент в системе можно задать при помощи биективного отображения вида

$$l: Q \rightarrow R, \quad (2.11)$$

где  $Q=\{Q_i\}$  и  $R=\{R_i\}$  – множества входных и выходных контактов компонент системы соответственно,  $i=1, \dots, N$ .

Отображение  $l$  назовем *функцией связей* контактов.

С помощью идеального канала. информационные сообщения в системе передаются от выходного контакта одной компоненты к связанному входному контакту другой компоненты. Допускаются соединения компоненты с собой.

Последнее предположение выступает как ограничение на структуру сети идеальных каналов СС. Если допустить, что к контакту некоторой компоненты модели будет подключено несколько каналов от различных источников, то поведение этого элемента будет неопределенным из-за влияния нескольких

сообщений, поступивших из разных источников. Кроме того, это предположение предусматривает, что дублирование при передаче данных исключается.

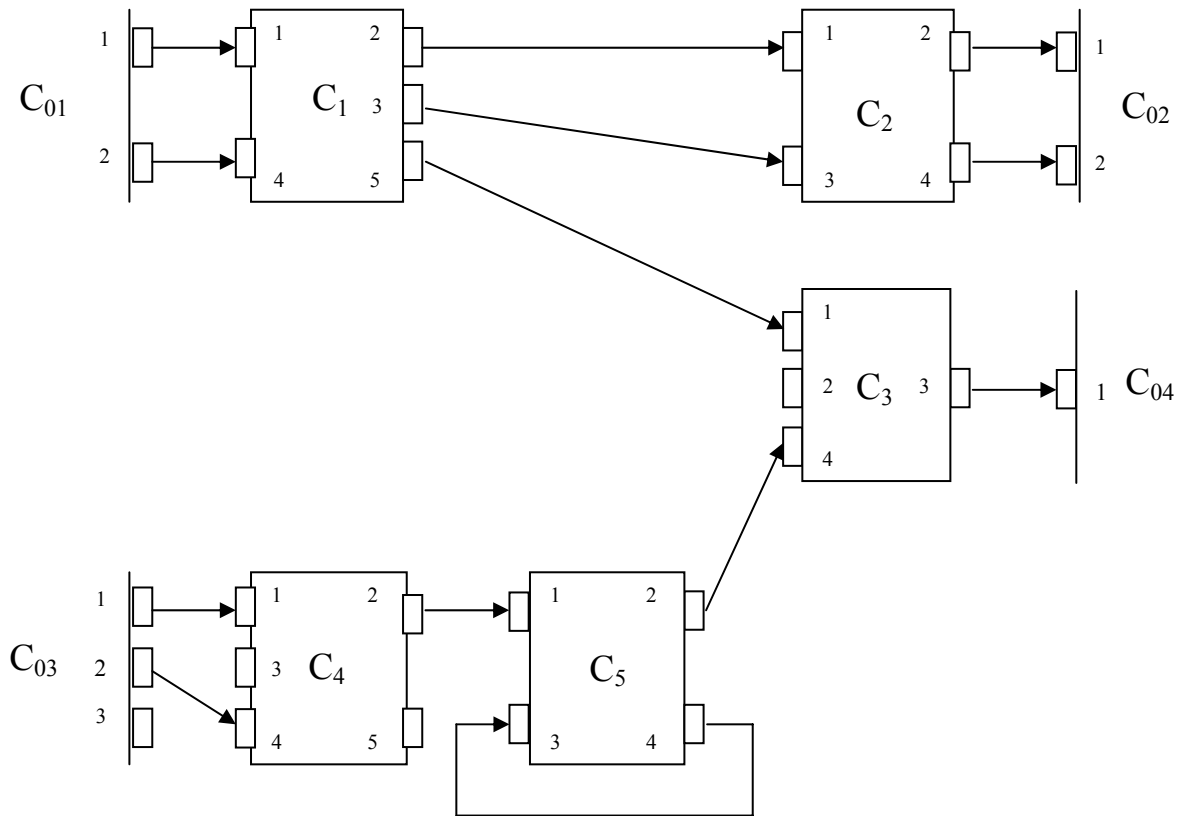
Введенные здесь предположения не всегда выполняются в реальных объектах, рассматриваемых как СС, и на первый взгляд представляются стеснительными. Предлагаемая формализация сопряжения элементов требует соответствующей структуры наборов контактов и сети идеальных каналов. Причиной неадекватности реальной системы и математической модели может оказаться не идентичность форматов входных и выходных данных. Такого рода причины можно устранить в рамках предложенной схемы сопряжения элементов СС. Достаточно ввести единый формат сообщений и соответствующих им контактов. Применительно к компьютерным сетям существуют стандарты ISO и документы RFC, определяющие форматы кадров, дейтаграмм, пакетов передачи данных, а также структуры протоколов разных уровней модели OSI. В некоторых случаях упомянутая не идентичность не может быть устранена с помощью использования единых форматов передаваемых данных. Тогда реально существующие *устройства сопряжения* (например, в сетях с интерфейсом ISDN – параллельный код и телефонной линией связи – последовательный код) должны рассматриваться как самостоятельные элементы СС, моделируемые соответствующими компонентами макромоделей. Т.о. разрабатываемая модель позволяет адекватно представить широкий класс реальных объектов.

Для построения математической модели сопряжения элементов СС необходимо также иметь в виду и взаимодействие ее с внешней средой. Оно рассматривается как процесс обмена сообщениями между объектами внешней среды и элементами системы. Объекты внешней среды можно представить в виде элементов, аналогичных элементам системы.

На рис. 2.1 представлен пример системы состоящей из элементов  $C_1, C_2, C_3, C_4$  и  $C_5$  взаимодействующей с внешней средой, представленной объектами  $C_{01}, C_{02}, C_{03}$  и  $C_{04}$ . Функция связей для изображенной системы может быть

задана в виде такого множества:  $l = \{(P_{01,1}, P_{1,1}), (P_{01,2}, P_{1,4}), (P_{1,2}, P_{2,1}), (P_{1,3}, P_{2,3}), (P_{1,5}, P_{3,1}), (P_{2,2}, P_{02,1}), (P_{2,4}, P_{02,2}), (P_{3,3}, P_{04,1}), (P_{03,1}, P_{4,1}), (P_{03,2}, P_{4,4}), (P_{4,2}, P_{5,1}), (P_{5,2}, P_{3,4}), (P_{5,4}, P_{5,3})\}$ .

Рис. 2.1 Одноуровневая схема сопряжения



Одноуровневая схема сопряжения является основным средством задания структуры связей между элементами СС в том случае, когда сама система оказывается одноуровневой: ее элементы входят непосредственно в систему и не распределены по подсистемам. Однако не трудно показать, что полученная одноуровневая схема может служить эквивалентной схемой сопряжения для СС, описанной схемой сопряжения с любым числом уровней.

Отметим некоторую схожесть данной модели со способом сопряжения компонент СС, предложенного в [9], однако ниже будет показан ряд отличий обусловленных методами моделирования.

## 2.5. Разработка модели взаимодействия компонент системы

Эта модель основывается на ряде предположений, относящихся к строению входных и выходных сигналов динамических систем, а также к порядку их передачи по каналам связи.

Под *взаимодействием* элементов СС мы понимаем режим совместного функционирования компонент, при котором поведения или свойства одного элемента в общем случае зависят от условий, определенных поведением или свойствами других элементов. Взаимодействие представляется как результат влияний, оказываемых элементами СС друг на друга.

Пусть рассматривается система  $S$ , состоящая из элементов  $C_1, C_2, \dots, C_N$ . Влияние компоненты  $C_j$  системы  $S$  на компоненту  $C_k$  той же системы определяется сообщениями, поступающими от элемента  $C_j$  к элементу  $C_k$ . Данные от элемента  $C_j$ , сформированные с учетом условий функционирования этого элемента передаются без искажений по идеальному каналу связи и поступает к элементу  $C_k$  в качестве входной информации, вызывая изменение в поведении этого элемента.

Для обеспечения требуемой диакоптическим подходом общности схемы взаимодействия элементов СС и адекватности модели сети необходимо учесть особенности циркуляции данных в информационно-вычислительных сетях. Это можно сделать введя в модель ряд предположений о характере передачи данных между компонентами.

Рассмотрим две компоненты  $C_i$  и  $C_j$  системы  $S$  и два связанных контакта  $P_{iq} \in Q_i$  и  $P_{jr} \in R_j$ ,  $r = l(q)$ ,  $l$  – функция связей системы.

*Предположение 14.* Сообщения поступают на вход  $P_{iq}$  в такой же последовательности, в какой они были отправлены на выход  $P_{jr}$ .

При таком подходе к моделированию процессов взаимодействия элементов СС, возможны ситуации, когда на какой-нибудь вход компоненты поступит информации больше, чем он может поглотить в этот момент времени. Это означает потерю не востребовавшей информации.

Например, при заполнении входного буфера сетевой интерфейсной карты (Network Interface Card – NIC) все поступающие кадры отбрасываются до тех пор, пока в буфере не освободится место хотя бы для одного поступающего кадра. Такие потери информации не представляют собой угрозы для нормальной работы сети, если они не имеют массовый и постоянный характер. Однако в любом случае желательно иметь возможность получить информацию о количестве передаваемых данных до начала транспортировки. Поэтому в модель взаимодействия компонент также следует включить механизм синхронизации объемов передаваемой выходом и принимаемой связанным входом информации. Такое согласование неявно выполняется и в компьютерных сетях. Например, в протоколе ТСР используется принцип "скользящего окна" (*sliding window*), который заключается в том, что каждая сторона может отправлять партнеру максимум столько байт, сколько партнер указал в поле "размер окна" заголовка ТСР-пакета, подтверждающего получение предыдущих данных. Этот принцип обеспечивает "опережающую" посылку данных с "отложенным" их подтверждением. Размер окна, как правило, определяется объемом свободного места в буферах принимающего ТСР-модуля.

Следующее предположение предотвращает потерю информации при передаче между контактами компонент.

*Предположение 15.* Пропускная способность выходного контакта не может быть больше пропускной способности связанного с ним входного контакта. Т.е. для контактов  $P_{iq} \in Q_i$  и  $P_{jr} \in R_j$ , таких что  $r = l(q)$  должно выполняться неравенство

$$\varepsilon_i[P_{iq}] \leq \varepsilon_i[P_{jr}], \quad (2.11)$$

где  $\varepsilon_i[P_{iq}]$  и  $\varepsilon_i[P_{jr}]$  – пропускные способности контактов  $P_{iq}$  и  $P_{jr}$  соответственно.

Т.к. в модели элемента СС, описанной в пункте 2.3, не определен механизм установки пропускных способностей входных контактов компонент, то в дальнейшем будем считать, что  $\varepsilon_i[P_{iq}] = \varepsilon_i[P_{jr}]$ .

При формализации взаимодействия элементов СС обычно придерживаются предположения, согласно которому взаимодействие достаточно полно и точно описывается в рамках механизма обмена сообщениями. Поэтому для формального описания взаимодействия элементов СС достаточно иметь следующие модели: 1) сопряжения элементов системы сетью каналов связи, обеспечивающих передачу сигналов между ними; 2) процессов генерации, транзита и приема сигнала элементами системы во времени.

Первая из перечисленных здесь моделей нам уже знакома по материалам предыдущего пункта. Эта модель отражает структурный аспект взаимодействия элементов СС в процессе ее функционирования. С практической же точки зрения цель построения модели сопряжения состоит в том, чтобы каждому сигналу, поступающему из внешней среды или возникающий как выходной сигнал элемента системы, указать адрес, по которому он должен поступить как входной сигнал элемента системы или объекта внешней среды.

В этом пункте составлена вторая модель – модель процесса распространения сигнала от элемента к элементу во времени, воспроизводящая передачу данных между устройствами системы.



### 3. РАЗРАБОТКА АЛГОРИТМИЧЕСКОЙ МОДЕЛИ КОМПЬЮТЕРНОЙ СЕТИ

#### 3.1. Разработка концептуальной модели мониторинга системы

Для создания имитационной модели информационно-вычислительной сети необходимо разработать алгоритм управления процессом функционирования системы.

В дальнейшем указанный алгоритм будем называть *монитором системы*.

Так как разработка модели сети ведется с применением диакоптики, то монитор должен обладать некоторым минимальным набором свойств или функций необходимых для управления системой. Частично эти свойства заложены уже в определении: под *монитором* системы будем понимать некоторый механизм, не входящий в систему, осуществляющий синхронизацию и взаимодействие функциональных компонент и обеспечивающий доступ компонент к ресурсам, выделенным для использования системой.

*Ресурсами* системы будем называть логическую или физическую часть системы которая может быть выделена для использования компонентами.

Итак, предполагается, что монитор обеспечивает

1. Взаимодействие компонент, т.е. перемещение информационных сообщений между связанными контактами.
2. Квазипараллельное выполнение моделирующих программ. Приставка «квази» в данном случае отражает последовательный характер моделирования компонент, одновременно функционирующих в реальной системе.
3. Доступ моделирующей программы к данным о моделируемой компоненте. В общем случае одна модель может «обслуживать» несколько компонент, каждой из которых имеет свой набор данных о текущем состоянии, содержимом хранилища, контактах и т.д.

4. Возможность доступа к некоторым данным о состоянии системы, которое характеризуется текущим временем, информационными сообщениями циркулирующими в системе и распределением следов. Часть из этой информации недоступна компонентам, однако может использоваться подпрограммами обработки и анализа результатов моделирования.
5. Реализацию стандартных действий, выполняемых компонентами. В качестве примеров таких действий можно привести отправку и получение следов информационных сообщений, установку пропускных способностей и т. д.

Из всего вышеперечисленного можно сделать вывод о том, что полноценное управление системой невозможно без взаимодействия монитора и функциональных компонент.

На концептуальном уровне моделирования взаимодействие монитора и компонент можно представить в виде обмена сообщениями и запросами. *Сообщения* передаются от монитора к компоненте и содержат информацию необходимую моделирующим программам. *Запросы* посылаются компонентами монитору и содержат требования о выполнении некоторых стандартных действий или о получении информации о компоненте. Методы и способы при помощи которых осуществляется коммуникационных процесс между монитором системы и компонентами будут рассмотрены ниже.

### **3.2. Разработка алгоритмической модели монитора системы**

Базируясь на концептуальной модели составим алгоритм функционирования монитора системы. Для этого рассмотрим фрагмент модели – два связанных друг с другом контакта.

Введем следующие обозначения: контакт компоненты представим в виде цилиндра. Если через контакт передаются данные, то цилиндр будет заполнен (полностью или частично). Степень заполнения определяется количеством передаваемых следов. Оставшаяся незаполненная область соответствует свободной пропускной способности контакта.

Рис. 3.1 Схема функционирования фрагмента системы

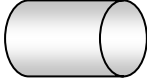

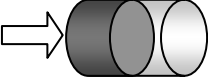

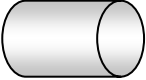
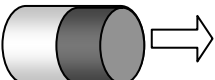
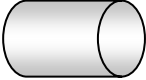


Этап	Выходной контакт	Входной контакт
1		
↓	Выполнение моделирующих программ. Отправка данных компонентами.	
2		
↓	Переход системы в новое состояние. Перемещение данных между связанными контактами.	
3		
↓	Выполнение моделирующих программ. Прием данных компонентами.	
4		
↓	Выполнение моделирующих программ. Установка компонентами будущих пропускных способностей.	
5		
↓	Изменение пропускных способностей связанных контактов компонент.	
6(1)		

Схема на рис. 3.1 иллюстрирует все этапы процесса передачи данных между связанными контактами.

Если какой-либо выход компоненты не имеет связанного с ним входа, то все данные поступившие на такой контакт удаляются при переходе системы в новое состояние.

Прежде чем перейти к составлению алгоритма управления системой определимся, каким способом реализуется взаимодействие компонент и монитора. Как было определено в предыдущем пункте, от монитора к компонентам поступают сообщения, а в обратном направлении – запросы.

Передачу сообщений можно осуществлять следующим образом. Все алгоритмические модели компонент представляют собой подпрограммы, имитирующие работу элемента реальной компьютерной сети. В качестве входных данных алгоритма используется информация о текущей моделируемой компоненте. Аналогично при вызове моделирующей процедуры ей передаются данные сообщения через список входных параметров. Представляется целесообразным использовать сообщения информирующие о 1) моделируемой компоненте; 2) текущем времени; 3) времени завершения моделирования; 4) причине вызова моделирующей программы (отправка или прием данных, установка пропускных способностей и т.д.). Т.о. вызов моделирующей программы приобретает такой вид:

*имя\_подпрограммы( данные\_о\_системе, информация\_о\_компоненте, причина\_вызова, текущее\_время, время\_завершения ).*

Что касается запросов, то они могут быть реализованы в виде функций, которые будут вызываться в моделирующих программах.

В таблице 3.1 приведен один из вариантов списка запросов, используемых в алгоритмических моделях компонент.

Каждая из указанных в таблице функций имеет характерный набор параметров определяемых назначением соответствующего запроса. В случае отсутствия ошибок функция выполняет соответствующие действия и/или возвращает запрашиваемую информацию. Если во время выполнения запроса

произошла ошибка, то причину неудачного завершения можно узнать с помощью специально предназначенного для этих целей запроса.

Таблица 3.1 – Список запросов

Имя	Назначение
LastSysimError	Получить последнюю ошибку при выполнении запроса
ComponentInfo	Получить информацию о текущей компоненте
ContactInfo	Получить информацию о контакте
GetCapacity	Получить пропускную способность контакта
SetCapacity	Установить будущую пропускную способность контакта
RecvTrail	Получить след из контакта и поместить в хранилище
SendTrail	Отправить след из хранилища в контакт
CopyTrail	Отправить копию следа из хранилища в контакт
FirstTrail	Получить первый след в хранилище
NextTrail	Получить следующий след в хранилище
NewTrailExtra	Установить дополнительную информацию о следе
GetTrailExtra	Получить дополнительную информацию о следе
StorageGetCount	Получить количество следов в хранилище
ContactGetCount	Получить количество следов в контакте
CreatePacket	Создать информационное сообщение
DestroyPacket	Уничтожить информационное сообщение
GetPacketInfo	Получить данные об информационном сообщении
CreateParam	Создать параметр
IsParam	Проверить существует ли параметр
DeleteParam	Удалить параметр
SetParam	Установить значение параметра
GetParam	Получить значение параметра

В первой колонке таблицы указаны имена функций реализующих соответствующий запрос, во второй колонке – действия выполняемые при вызове подпрограмм.

Для мониторинга всей системы в целом необходимо выполнять указанные на рис. 3.1. этапы для всех связанных контактов в системе до тех пор, пока не закончится время моделирования. Также к алгоритму управления системой следует добавить вызов подпрограмм, выполняющих сбор и анализ данных полученных в результате моделирования.

Руководствуясь вышесказанным, приведем описание алгоритма мониторинга системы на С-подобном псевдоязыке.

СИСТЕМА Система;

КОМПОНЕНТА Компоненты[КоличествоКомпонент];

ВРЕМЯ ТекущееВремя, ВремяЗавершения;

ИНДЕКС i;

СООБЩЕНИЕ ПричинаВызова;

ТекущееВремя = 0

while( 1 ){

*Изменение пропускных способностей связанных контактов;*

ПричинаВызова = ОтправкаДанных;

for( i = 0; i < КоличествоКомпонент; i++ )

    Компонента[i].Модель( Система, Компонента[i], ПричинаВызова,  
    ТекущееВремя, ВремяЗавершения );

ТекущееВремя++;

if( ТекущееВремя == ВремяЗавершения )

    break;

*Перемещение данных между связанными контактами;*

ПричинаВызова = ПриемДанных;

for( i = 0; i < КоличествоКомпонент; i++ )

Компонента[i].Модель( Система, Компонента[i], ПричинаВызова,  
ТекущееВремя, ВремяЗавершения );

*Вызов подпрограмм сбора и анализа данных;*

}

Дадим некоторые пояснения. Большими буквами обозначены используемые типы данных. Тип КОМПОНЕНТА является структурой содержащей информацию о компоненте. Поле Модель этой структуры содержит имя моделирующей программы. Переменная «Компоненты» обозначает массив содержащий данные о компонентах системы. Т.о. цикл for по очереди перебирает все компоненты системы. В переменной «Система» хранятся данные о моделируемой системе. Курсивом отмечены участки управления системой.

Заметим, что в приведенном алгоритме присутствуют только два этапа вызова компонент, вместо трех отмеченных на схеме (рис. 3.1). Это связано с тем, что установку будущих пропускных способностей необязательно проводить в виде отдельного этапа. Для этих целей можно использовать стадии приема и/или отправки следов информационных сообщений.

### **3.3 Разработка алгоритмических моделей компонент**

Выделим и охарактеризуем основные устройства компьютерной сети.

Линия связи или канал – физическая среда передачи данных в компьютерных сетях. Основные характеристики: 1) скорость передачи или пропускная способность – число единиц передаваемой информации в единицу времени; 2) задержка перемещения - время перемещения единицы информации между точками инциденции; 3) вероятность ошибки передачи единицы информации между точками инциденции.

Концентратор называемый также распределителем, или хабом – коммутирующее устройство в сетях звездной архитектуры. Концентраторы предназначены для объединения в сеть многих узлов. Параметры концентраторов аналогичны характеристикам линий связи.

Узел сети – компьютер занимающийся как обслуживанием сети и передачей данных, так и выполнением пользовательских программ. Некоторые характеристики узлов сети: 1) среднее время обработки пакетов 2) размеры входных и выходных буферов для хранения данных.

Приведем примеры базовых алгоритмов функционирования перечисленных устройств.

Составим алгоритм моделирования линия связи. Предположим, что пропускная способность канала задана в виде входного параметра. Передача данных осуществляется без искажений в обоих направлениях. Минимальная задержка перемещения равна одному промежутку модельного времени. У дуплексной линии существует два входных и два выходных контакта. Пропускная способность входов вычисляется по формуле

$$\varepsilon_{t+1}[P_{ij}] = (C - m(S_t[P_{ij}]))/2, \quad (3.1)$$

где  $\varepsilon_{t+1}[P_{ij}]$  – будущая пропускная способность входного контакта,  $S_t[P_{ij}]$  – хранилище линии,  $j=1,2$ ,  $m(\cdot)$  – мощность множества,  $C$  – пропускная способность линии.

В начальный момент времени  $\varepsilon_0[P_{ij}] = C$ ,  $S_0 = \emptyset$ .

Опишем алгоритм работы многопортового концентратора. Предположим, что имеется  $N$  входов и такое же количество выходов. Все порты имеют одинаковую максимальную пропускную способность равную  $C$ . Для упрощения алгоритма будем считать, что в каждый момент модельного времени для приема сообщений доступен только один входной контакт  $P_{iq}$ . Пропускная способность входных контактов определяется из выражения

$$\varepsilon_{t+1}[P_{il}] = \begin{cases} \min(C, \varepsilon_t[P_{i1}], \dots, \varepsilon_t[P_{iN-1}]), & \text{если } l = q \\ 0, & \text{если } l \neq q \end{cases}, \quad (3.2)$$

где  $P_{ij} \in R_i$ ,  $k=1, 2, \dots, N-1$ ,  $j_k \neq q$ .

В начальный момент времени  $\varepsilon_0[P_{i1}] = C$ ,  $\varepsilon_0[P_{ij}] = 0$ ,  $P_{ij} \in Q_i$ ,  $j=2, \dots, N$ ,  $S_0 = \emptyset$ .



В следующий момент времени данные будут принимать другой контакт с номером  $q+1$ . Все данные поступившие на  $q$ -й вход хаба дублируются и посылаются на все выходы кроме  $q$ -го. Размер хранилища не ограничивается.

Один из простейших алгоритмов функционирования узла заключается в следующем. Узел имеет один входной и один выходной контакт. Пропускная способность входа задается в качестве параметра и не изменяется со временем., т.е.

$$\varepsilon_t[P_{i1}] = C, \quad (3.3)$$

где  $C$  – заданная пропускная способность,  $P_{i1}$  – входной контакт узла  $C_i$ .  
Формула (3.3) справедлива для любого момента времени  $t$ .

Все принятые информационные сообщения удаляются. Узел может генерировать информационные сообщения, которые затем отправляются через выходной контакт во внешнюю среду. Если все следы созданного сообщения отправлены, то узел генерирует новое сообщение.

### **3.4. Получение и анализ результатов моделирования**

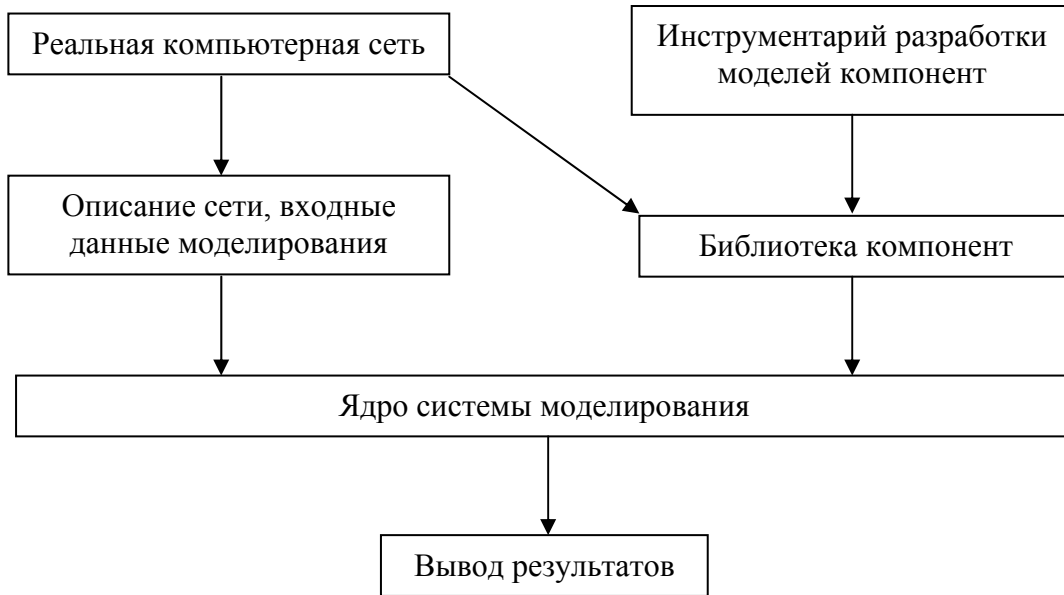
В предыдущих разделах данной работы были созданы математическая и алгоритмическая модели функционирования вычислительной сети. На основании проведенных исследований был разработан пакет прикладных программ (ППП) для имитационного моделирования компьютерных сетей.

В состав пакета входят:

- 1) инструментальные средства для создания моделей компонент сети;
- 2) основная программа, выполняющая разбор входного описания исследуемой сети, мониторинг системы, анализ и вывод результатов моделирования;
- 3) тестовая библиотека моделей компонент, содержащая подпрограмму имитирующую узел сети.

Процесс исследования компьютерной сети при помощи разработанного ППП показан на рис. 3.2.

Рис. 3.2. Использование ППП для имитационного моделирования компьютерных сетей



При помощи созданного комплекса программ можно получать статистические данные о времени передачи информационных сообщений между компонентами моделируемой сети. В каждый момент модельного времени пользователь имеет возможность просмотреть распределение следов в хранилище компонент, количество и размеры сообщений, циркулирующих в системе.

В качестве примера применения пакета создадим модель сети из состоящую из двух связанных узлов. Для этого составим файл описания системы на специальном входном языке. Образец такого описания представлен на рис 3.3.

Рис. 3.3. Пример файла описания системы

```

[system]
Count=2
Library=net.dll
Time=300

[components]
t:_Host
c:LineOut=50
c:LineIn=50
  
```

```

i:MaxPacket=100
n:Client

t:_Host
c:LineOut=150
c:LineIn=150
i:MaxPacket=400
n:Server

[links]
Client:LineOut=Server:LineIn
Server:LineOut=Client:LineIn

```

Затем запускаем программу выполняющую моделирование. В командной строке нужно указать имя файла содержащего описание системы.

Фрагмент вывода состояния системы и результаты процесса моделирования показаны на рис. 3.4.

Рис. 3.4. Результаты моделирования

```

Message/Size Client Server
   175    71    71    -
   174    43     -    -
   173   341   200   91
Packets count: 3
Message/Size Client Server
   175    71    21    -
   173   341   250   41
Packets count: 2
Message/Size Client Server
   175    71     -   50
   173   341   300    -
Packets count: 2
Summary:
During 300 clock period(s) 173 packet(s) were transmitted.
Packets sizes.
  Maximum = 393, Minimum = 2, Average = 95
Packets ages.
  Maximum = 9, Minimum = 2, Average = 3

```

С помощью созданного ППП не составляет особого труда поэкспериментировать с моделью. Изменяя такие входные данные в описании системы как пропускные способности, максимальные размеры пакетов и время моделирования можно проследить тенденции в изменении времен передачи информационных сообщений.

## ВЫВОДЫ

Результатом данной работы является алгоритм моделирования процесса функционирования компьютерной сети. В качестве инструмента для исследования системы выбрано имитационное моделирование. Полученные моделирующие алгоритмы позволяют по исходным данным, содержащим сведения о начальном состоянии системы, и фактическим значениям параметров сети отобразить реальные явления в системе и получить сведения о возможном поведении сети для данной конкретной ситуации. На основании этой информации исследователь может принять соответствующие решения.

Использование диакоптического метода построения модели выдвинуло на первый план требования о взаимной независимости функциональных компонент модели и инвариантности алгоритма мониторинга системы. Как следствие разработанная модель обладает свойством расширяемости, т. е. возможностью включения в ее состав новых компонент без изменения алгоритмов функционирования макромодели. В итоге появляется возможность создать ПО с открытой архитектурой, использующее для моделирования модули разработанные сторонними производителями. Эти компоненты могут представлять собой не только модели оборудования, но и средства для визуализации и анализа процессов, протекающих в период функционирования сети. У пользователя такой программной системы не возникает потребности импортировать данные, полученные в результате работы модели, в другие приложения для выполнения необходимых расчетов. Также появляется возможность создать набор компонент моделирующих протоколы и сервисы разных уровней модели OSI. Включая или исключая требуемые компоненты из системы пользователь может менять состав стека протоколов и степень детализации модели сети.

Полученную в работе модель можно рассматривать как оболочку или среду для взаимодействия компонент. Естественно, для того, чтобы работать в составе модели компоненты должны удовлетворять определенным правилам, касающихся способов обмена информацией с другими составляющими

системы и взаимодействия с монитором. Однако эти правила не ограничивают сферу применения разработанных моделирующих алгоритмов, но способствуют тому, что их можно применять для моделирования и анализа широкого класса компьютерных сетей с разнообразными типами аппаратного и программного обеспечения.

## СПИСОК ССЫЛОК

1. Кульгин М. Технологии корпоративных сетей. Энциклопедия. – СПб: Питер, 2000. – 704с.: ил.
2. Жожикашвили В. А., Вишнеvский В. М. Сети массового обслуживания. Теория и применение к сетям ЭВМ. – М.: Радио и связь, 1988. – 192 с.
3. Шеннон Р. Имитационное моделирование систем. – Искусство и наука – М.: Мир, 1978. – 417 с.
4. Шолл Ф. В. Азбука планирования нагрузки. – Lan Magazine. Русское издание, 1996. Т. 2, № 8.
5. Jander M. Design tools take a step closer to the real world – Data Communications. 1995. V. 24. № 6. – P. 99-104.
6. Jones J.G. Modelling and Simulation with COMNET III: Course Notes – CACI Producers Company, December 1994.
7. Стернс Т. Учимся моделировать. – Сети. 1998, №5. – С. 130-135.
8. Крон Г. Исследование сложных систем по частям (диакоптика). – М.: Наука, 1972. – 544 с.
9. Борю, С. Ю., Ермолаев, В. А., Толок, В. А. Диакоптический подход к моделированию процессов в многофункциональных информационных системах. – ЗГУ, Запорожье, 1999.
10. Бусленко Н. П. Автоматизация имитационного моделирования сложных систем. – М.: Наука, 1977. – 240 с.: ил.