ПРИДНЕСТРОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Т.Г. ШЕВЧЕНКО

Инженерно-технический институт

Кафедра информационных технологий и автоматизированного управления производственными процессами

КУРСОВОЕ ПРОЕКТИРОВАНИЕ ПО ДИСЦИПЛИНЕ «СЕТИ ЭВМ И ТЕЛЕКОММУНИКАЦИИ»

Методические указания

Тирасполь, 2015

УДК 004.915 ББК 32.973.202

Составители:

М.В. Нижегородова, доцент каф. программного обеспечения вычислительной техники и автоматизированных систем

С.Т. Иванова, преподаватель каф. информационных технологий и автоматизированного управления производственными процессами

Рецензенты:

Т.Д. Бордя, доцент каф. информационных технологий и автоматизированного управления производственными процессами

О.С. Белоконь, ст. преп. каф. программного обеспечения вычислительной техники и автоматизированных систем

Курсовое проектирование по дисциплине «Сети ЭВМ и телекоммуникации»: Методические указания /Сост: М.В. Нижегородова, С.Т. Иванова.– Тирасполь, 2015.-62 с.

Методические указания имеют своей целью предоставить студентам материал, касающийся выполнения заданий курсовой работы по дисциплине «Сети ЭВМ и телекоммуникации» и составления пояснительной записки.

Изложенный материал позволяет студентам очной и заочной формы обучения выполнить задачу по проектированию и настройки параметров компьютерной сети, созданию сетевого программного приложения.

Предназначено для студентов ИТИ ПГУ, дневной и заочной форм обучения обучающихся по направлениям «Информатика и вычислительная техника», «Информационные системы и технологии», а так же специальности «Вычислительные машины, комплексы, системы и сети, выполняющих курсовую работу по дисциплине «Сети ЭВМ и телекоммуникации».

> УДК 004.915 ББК 32.973.202

Рекомендовано Научно-методическим Советом ПГУ им. Т.Г. Шевченко

© М.В. Нижегородова, С.Т. Иванова, составление, 2015

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ 1 ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ И ЭТАПЫ РЕАЛЕЗЦИИ 2 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОМУ ПРОЕКТУ	4 5 9
З СТРУКТУРА И ПРИМЕРНОЕ СОДЕРЖАНИЕ КУРСОВОГО ПРОЕТА	12
3.1 Описание и анализ предметной области	12
3.2 Проектирование компьютерной сети	16
3.3 Теоретико - расчетная часть проекта компьютерной сети	42
3.4 Планирование безопасности в проекте компьютерной сети	46
3.5 Проектирование и реализация сетевого приложения	49
ПРИЛОЖЕНИЕ А – Пример схемы здания, для которого необходимо	60
спроектировать компьютерную сеть	
ПРИЛОЖЕНИЕ Б – Расчетные таблицы для конфигурации Ethernet	61
ПРИЛОЖЕНИЕ В – Расчетные таблицы для конфигурации FastEthernet	62

введение

В наши дни компьютерная сеть является необходимым элементом любого предприятия, любой фирмы. Почти невозможно представить себе офис, который не использует в своей работе компьютеры, будь то строительная компания или стоматологическая поликлиника.

На сегодняшний день компьютерные сети должны присутствовать в каждом заведение, компании или больнице, потому что именно они относятся к одному из факторов достижения успеха в сфере бизнеса. Компьютерные сети способствуют соединению всех компьютеров, которые находятся на фирме, в единую сеть с выходом в Интернет. В современном мире первоначальной задачей при организации любой компании или фирмы становится монтаж сетей.

Монтаж компьютерных сетей предполагает соединение между собой нескольких компьютерных аксессуаров и стационарных персональных компьютеров в общую систему. Серверы, сканеры, принтеры, факсы и прочее подразумеваются под компьютерными аксессуарами. Благодаря передовым технологиям множество клиник, которые находятся в разных зданиях, на разных этажах или в разных точках, могут быть связаны в одну систему. Монтаж компьютерных сетей может осуществляться с помощью самого современного оборудования с использованием новых достижений и технологий. Гибкость и эффективность компьютерных сетей значительно повысилась, так как в настоящее время информацию можно передавать не только посредством проводов, но и с помощью радиоканала. При хорошем администрировании и правильном монтаже сетей можно с легкостью найти нужную информацию в любом компьютере, который соединён в сеть.

Эффективность и долговечность системы будет обеспечиваться при условии вложения на начальном этапе в компьютерную сеть необходимого количества денежных средств. На содержание и эксплуатацию подобной сети понадобится немного затрат – спустя очень короткое время все непременно окупится.

4

1 ЗАДАЧИ КУРСОВОГО ПРОЕКТИРОВАНИЯ И ЭТАПЫ РЕАЛЕЗАЦИИ

Курсовая работа предусматривает решение профессиональных задач: сбор и анализ исходных данных для проектирования компьютерной сети; проектирование сетевых программных и аппаратных средств, баз данных в соответствии с техническим заданием с использованием средств автоматизации проектирования; разработка и оформление технической документации; контроль соответствия разрабатываемых проектов и технической документации стандартам компьютерных сетей; задач, связанных с проектно-технологической деятельностью: применение современных инструментальных средств при разработке сетевых приложений; применение *Web*-технологий при реализации удаленного доступа в системах клиент/сервер; освоение программно-методических комплексов исследования процессов в компьютерной сети и автоматизированного проектирования компьютерных сетей.

Таким образом, курсовая работа по дисциплине «Сети ЭВМ и телекоммуникации» позволяет сформировать следующие компетенции: способность инсталлировать программное и аппаратное обеспечение для информационных и автоматизированных систем; способность осваивать методики использования программных средств для решения практических задач; способность разрабатывать бизнес-планы и технические задания на оснащение отделов, лабораторий, офисов компьютерным и сетевым оборудованием; способность решать стандартные задачи профессиональной деятельности с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; способность обосновывать принимаемые проектные решения, осуществлять постановку и выполнять эксперименты по проверке их корректности и эффективности.

Курсовая работа по дисциплине «Сети ЭВМ и телекоммуникации» выполняется студентами очного обучения параллельно с изучением данной дисциплины. Выдача заданий курсовой работы осуществляется в течение двух недель после начала занятий в семестре. Выполнение курсовой работы разбивается на этапы:

- 1) проектирование структуры приложения;
- 2) проектирование структуры базы данных;
- 3) реализация сетевого приложения;
- 4) планирование проекта компьютерной сети;
- 5) реализация проекта в среде Packet Tracer;
- 6) реализация маршрутизации в среде Packet Tracer;

7) реализация проекта в среде моделирования процессов компьютерной сети *NetCracker*;

8) теоретико – аналитический расчет параметров сети;

- 9) анализ параметров безопасности компьютерной сети;
- 10) оформление пояснительной записки;
- 11) подготовка презентации;
- 12) защита курсовой работы.

Защита курсовой работы студентами очного обучения включает в себя:

1) предоставление руководителю пояснительной записки, выполненной в соответствии с требованиями в печатном виде;

- 2) доклад студента с использованием презентации;
- 3) демонстрацию студентом проектов сети;
- 4) обоснование выбора аппаратного обеспечения;
- 5) обоснование структуры компьютерной сети;

6) демонстрацию студентом сетевого приложения, которое должно содержать несколько страниц, элементы навигации, возможность поиска информации в базе данных для пользователя с ограниченными правами доступа и возможность добавления, удаления и редактирования информации в базе данных для пользователя с правами администратора.

Курсовая работа по дисциплине «Сети ЭВМ и телекоммуникации» выполняется студентами заочного обучения после изучения данной дисциплины. Выдача заданий курсовой работы осуществляется в период проведения экзаменационного контроля по дисциплине. Студенты заочного обучения имеют возможность выбрать тематику курсовой работы согласно направлению их профессиональной деятельности. Тематика курсовых работ для студентов заочного отделения:

 – создание проекта компьютерной сети определенной организации;

 – создание сетевого приложения учета деятельности определенной организации или сетевого обучающего ресурса разделам дисциплин.

Этапы выполнения курсовой работы по созданию проекта компьютерной сети студентами заочной формы обучения:

1) планирование проекта компьютерной сети;

- 2) реализация проекта в среде Packet Tracer;
- 3) реализация маршрутизации в среде Packet Tracer;

4) реализация проекта в среде моделирования процессов компьютерной сети *NetCracker*;

- 5) теоретико аналитический расчет параметров сети;
- 6) анализ параметров безопасности компьютерной сети;
- 7) оформление пояснительной записки;
- 8) подготовка презентации;
- 9) защита курсовой работы.

Защита курсовой работы по созданию проекта компьютерной сети студентами заочной формы обучения включает в себя:

1) предоставление руководителю пояснительной записки, выполненной в соответствии с требованиями в печатном виде;

- 2) доклад студента с использованием презентации;
- 3) демонстрацию студентом проектов сети;
- 4) обоснование выбора аппаратного обеспечения;
- 5) обоснование структуры компьютерной сети.

Этапы выполнения курсовой работы по созданию сетевого приложения студентами заочной формы обучения:

- 1) анализ средств разработки сетевых приложений;
- 2) проектирование структуры приложения;
- 3) проектирование структуры базы данных;
- 4) реализация сетевого приложения;
- 5) размещение ресурса на веб сервере;
- 6) оформление пояснительной записки;
- 7) подготовка презентации;
- 8) защита курсовой работы.

Защита курсовой работы по созданию сетевого приложения студентами заочной формы обучения включает в себя:

1) предоставление руководителю пояснительной записки, выполненной в соответствии с требованиями в печатном виде;

2) доклад студента с использованием презентации;

3) демонстрацию студентом сетевого приложения, которое должно содержать несколько страниц, элементы навигации, возможность поиска информации в базе данных для пользователя с ограниченными правами доступа и возможность добавления, удаления и редактирования информации в базе данных для пользователя с правами администратора.

Оценивание результатов выполнения курсовой работы студентов очного и заочного обучения проводится комиссией из двух и более преподавателей.

2 СОДЕРЖАНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ К КУРСОВОМУ ПРОЕКТУ

При выполнении курсовой работы студент должен:

- провести проектирование локальной вычислительной сети (ЛВС);
- разработать сетевое приложение, взаимодействующее с базой данных;
- оформить пояснительную записку.
 - Проектирование ЛВС в общем случае предполагает:

- сравнительный анализ различных вариантов архитектуры ЛВС с системных позиций по основным параметрам: производительность (быстродействие), надежность, расширяемость, масштабируемость, управляемость, защищенность (информационная безопасность), стоимость.

– разработку структурной схемы ЛВС, структуры аппаратного и состава программного обеспечения для предоставления пользователям заданного перечня услуг (сервисов), включая услуги глобальной вычислительной сети.

При создании ЛВС перед разработчиком стоит проблема: при известных данных о назначении, перечне функций ЛВС и основных требованиях к комплексу технических и программных средств ЛВС построить сеть для информационной системы в заданной предметной области.

Методика проектирования ЛВС в общем случае включает следующие этапы:

- системное проектирование (технико-экономическое обоснование разработки);

- разработку конфигурации;
- разработку архитектуры;
- планирование информационной безопасности;
- расчет экономической эффективности.

Системное проектирование (технико-экономическое обоснование разработки) ЛВС включает: анализ предметной области, обоснование потребности проектирования вычислительной сети и определение перечня функций и соответствующих услуг (сервисов), предоставляемых пользователям в вычислительной сети.

Анализ предметной области предполагает обследование организационной структуры предприятия и выявлении функций подразделений. Определяются перечни задач, решаемых отделами, и информационные массивы (базы данных, группы файлов), используемые для решения этих задач. Выполнение функций пользователей основано на использовании сетевых служб (серверов). Как правило, используются следующие службы: файл-сервер, сервер печати, сервер базы данных, электронная почта, *Web*-сервер.

Пояснительная записка к курсовой работе студентов *очного* обучения должна содержать:

Титульный лист

Задание на курсовую работу Введение

1 Описание и анализ предметной области

- 1.1. Описание предметной области
- 1.2. Анализ потоков информации.
- 1.3. Анализ функций сетевого приложения
- 1.4. Аргументация выбора средств разработки

2 Проектирование компьютерной сети

- 2.1. Планирование сети;
- 2.2. Спецификация локальной вычислительной сети
- 2.3. Схема соединения сетевого оборудования
- 2.4. Теоретико расчетная часть
- 2.5. Планирование безопасности

3 Проектирование и реализация сетевого приложения

- 3.1. Структура, архитектура сетевого приложения
- 3.2. Описание схемы базы данных
- 3.3. Функциональная схема и реализация сетевого приложения

Заключение

Список литературы

Приложение А. Руководство пользователя.

Приложение В. Руководство системного администратора (какие настройки необходимо провести при организации подключения к базе данных, как разместить на *web*-сервере, как поместить ссылку на приложение в поисковых системах).

Пояснительная записка к курсовой работе по теме проектирование компьютерной сети студентов *заочного* обучения должна содержать:

Титульный лист Задание на курсовую работу Введение 1 Описание и анализ предметной области

- 1.1. Описание предметной области
- 1.2. Анализ потоков информации.
- 1.3. Аргументация выбора средств разработки

2 Проектирование компьютерной сети

- 2.1. Планирование сети
- 2.2. Спецификация локальной вычислительной сети

3 Проектирование и реализация проекта компьютерной сети

- 3.1 Схема соединения сетевого оборудования
- 3.2 Теоретико расчетная часть
- 3.3 Планирование безопасности

Заключение

Список литературы

Приложение А. План здания и таблица с указаниями размеров **Приложение Б.** Таблица технических средств вычислительной сети **Приложение В.** Таблица расчетов себестоимости проектирования сети

Пояснительная записка к курсовой работе по теме создание сетевого приложения студентов за*очного* обучения должна содержать:

Титульный лист

Задание на курсовую работу

Введение

1 Описание и анализ предметной области

- 1.1. Описание предметной области
- 1.2. Анализ потоков информации.
- 1.3. Анализ функций сетевого приложения

2 Проектирование сетевого приложения

2.1. Сравнительный анализ средств разработки сетевых приложений

- 2.2. Аргументация выбора средств разработки
- 2.3. Структура, архитектура сетевого приложения

3 Реализация сетевого приложения

3.1. Описание схемы базы данных

3.2. Функциональная схема и реализация сетевого приложения

Заключение

Список литературы

Приложение А. Руководство пользователя

Приложение В. Руководство системного администратора (какие настройки необходимо провести при организации подключения к базе данных, как разместить на *web*-сервере, как поместить ссылку на приложение в поисковых системах)

3 СТРУКТУРА И ПРИМЕРНОЕ СОДЕРЖАНИЕ КУРСОВОГО ПРОЕКТА

3.1 Описание и анализ предметной области

На первом этапе проектирования информационной системы необходимо выполнить анализ предметной области, т.е. определить объекты предметной области и связи между объектами.

При выборе состава и структуры предметной области возможны два подхода: функциональный и предметный.

Функциональный подход реализует принцип движения «от задач» и применяется, когда определен комплекс задач, для обслуживания которых создается информационная система. В этом случае можно выделить минимальный необходимый набор объектов предметной области, которые должны быть описаны.

В предметном подходе объекты предметной области определяются с таким расчетом, чтобы их можно было использовать при решении множества разнообразных, заранее не определенных задач. Чаще всего используется комбинация этих двух подходов.

В данном разделе представить особенности рассматриваемой предметной области.

Например: предметная область «институт». Примерное содержание описания предметной области. Институт – учреждение, которое предоставляет образовательные услуги, содержит следующие административные структуры: дирекция, методический отдел, кафедры и др. Институт расположен в двух зданиях. Первое здание содержит три этажа, общая площадь каждого из этажей примерно 80 кв.м. Содержит 23 аудитории, аудитории делятся на лекционные и лаборатории. Лаборатории оснащены компьютерной сетью, содержат по 12 узлов. Деятельность института связана с подготовкой студентов по разным специальностям. Работа института организованна следующим образом: учебный год начинается в сентябре месяце, осуществляется набор новых групп по разным специальностям, при поступлении студентам оформляется входная документация: студенческие билеты, зачетные книжки. Студенты обучаются в течение семестра, в конце каждого семестра они проходят итоговый контроль в форме экзамена или зачета. Документация для проведения итогового контроля формируется сотрудниками методического отдела. По завершению сессии заведующие кафедр и ответственные лица анализируют результаты сессии, формируют отчеты об успеваемости студентов, проводится анализ количества пропущенных занятий, формируются

графики пересдач. Доступ к этой информации имеют как студенты, так и сотрудники института. Деятельность института регулируется служебной документацией: приказами, распоряжениями, которые формируются дирекцией и должны быть распространены между всеми структурными подразделениями института и т.д.

Таким образом, при описании предметной области должны быть проанализированы структурные составляющие, организационные составляющие и процессы.

Анализ потоков информации. Из описания предметной области необходимо выделить тип передаваемой информации по проектируемой сети.

Например: по завершению сессии заведующие кафедр и ответственные лица анализируют результаты сессии, формируются отчеты об успеваемости студентов, проводится анализ количества пропущенных занятий, формируются графики пересдач. Доступ к этой информации имеют, как студенты, так и сотрудники института. Из этого следует, что между структурными подразделениями института происходит обмен текстовыми документами. В том числе студенты имеют возможность получить информацию через *web*-сайт. Таким образом, осуществляется обмен электронными документами, изображениями. Электронный документооборот института предполагает использование обращений к базе данных, в которой хранятся сведения о студентах и их успеваемости.

Таким образом, при планировании сети необходимо:

- организовать передачу данных между структурными подразделениями института;

- разграничить доступ к информации, которой обмениваются подразделения;
- предусмотреть возможность подключения к сети Интернет;
- предусмотреть возможность подключения к серверу баз данных.

В результате в данном разделе должны быть проанализированы типы передаваемой информации, кто с кем обменивается информацией, к чему необходимо обеспечить доступ, какие разграничения доступа должны быть обеспечены. Анализ функций сетевого приложения. При описание функций сетевого приложения необходимо отметить какие функции будет выполнять приложение, например: добавление информации о сотрудниках будет осуществляться только пользователем с правами администратора; редактирование информации о сотрудниках будет осуществляться с использованием вызова пользовательской функции, в теле которой *SQL*-запрос в условии которого будет обращение к элементам *web*-страницы *Page1.html*.

В результате должны быть рассмотрены все функции приложения, осуществляемые проверки переменных, ввода в текстовые поля, выбора пользователем определенных значений в элементе управления и т.д.

Должны быть указаны имена страниц, в которых будут выполняться рассматриваемые функции.

Описание необходимо завершить графическим представлением функциональной схемы приложения.

Аргументация выбора средств разработки. Описание возможностей программных средств, например, *Packet Tracer*, *Net-Cracker*, *web*-сервера, языка *PHP*, средств разработки *web*-страниц.

Необходимо провести сравнительный анализ существующих средств разработки, выделить их достоинства и недостатки и аргументировать их выбор для реализации задачи курсовой работы.

3.2 Проектирование компьютерной сети

Планирование сети. В данном разделе необходимо рассмотреть план здания, если оно содержит несколько этажей, значит план каждого этажа с указанием либо на рисунке, либо в прилагаемой таблице размеров (длину и высоту). Цветными линиями показать план размещения кабелей вдоль стен, точками или другими геометрическими примитивами обозначить места сетевых розеток и другого оборудования.

Пример представления плана здания изображен в приложение А.

Пример оформления размеров помещения представлен в таблице:

<i>L1,</i>	<i>Н1,</i>	<i>D1,</i>	<i>L11,</i>	<i>L12,</i>	<i>Н2,</i>	<i>D2,</i>	<i>L21,</i>	<i>L22,</i>	Этажность	Этажность
м	м	м	м	м	м	м	м	м	здания 1	здания 2
max	9	60	15	30	8	150	30	15	3	2

Так же необходимо указать какое и в каком количестве будет использовано сетевое оборудование.

Рассчитать длину кабеля. Для расчет длины кабеля рабочего места на основе витой пары проводников, используемые в конфигурации многопользовательской телекоммуникационной розетки и мебельных систем открытого офиса определяют по формуле 1.

$$L_1 = \frac{102 - L_r}{1 + \Pi.K.} \tag{1}$$

где,

 L_{I} – длина кабеля горизонтальной подсистемы, м,

П.К – поправочный коэффициент коммутационных шнуров:

0,2 – для *UTP/FTP/ScTP/SFTP* с проводниками калибра 24 *AWG*,

0,5 – для *FTP/ScTP/SFTP* с проводниками калибра 26 *AWG*. *L*_r принимает значения 70,75,80,85,90, обязательно <= 100 м.

Спецификация локальной вычислительной сети. В данном разделе необходимо проанализировать цены на сетевое оборудование любого из поставщиков, например организации торгующей сетевым оборудованием на территории республики или за ее пределами, и представить их в табличном виде. По результатам проектирования составляется спецификация ЛВС. Пример спецификации ЛВС показан в таблице:

№ п/п	Тип ТС	Наименование ТС	Цена ТС,	Кол-во ТС,	Стоимость ТС,
			y.e.	ШТ.	y.e.
1	Сервер	Компьютер Pentium	783	1	783
		166 MMX, RAM 16			
		sdram, HDD 1.6 Gb,			
		VM 14"			
2	Сетевой адаптер	Ethernet 3 COM	95	50	475
3	Линия связи	Кабель коаксиаль-	0.6 за 1 м	700 м	420
		ный <i>RG 58</i>			
4	Активные	HUB 8-port	175	4	740
	концентраторы	10Mbs			
5	Разъемы	BNCconnector	2	100	200
6	Разъемы	T connector	2	50	100
7	Сетевой принтер	HP Laser Jet 5	610	1	610
		Программное об	беспечение		
8	Сетевая	MS Windows NT	4176	1	4176
	операционная	4 Server			
	система				
9	Интегрированная	4176MS	407	1	407
	офисная система	OFFICE'97 RUS			

Таблица 2 - Технические средства (ТС) вычислительной сети.

Схема соединения сетевого оборудования. В данном разделе должны быть представлены:

– описание схемы соединения активного оборудования и реализация ее в *Packet Tracer*;

– назначение *IP* адресов хостам, формирование масок подсетей, назначение *IP* адресов портам маршрутизаторов;

- формирование таблиц маршрутизации (в *Packet Tracer*);

– структура вертикальной и горизонтальной кабельной сети;

– схема размещения и соединения рабочих станций, серверов, принтеров, повторителей, концентраторов и другого оборудования, включая подключение к удаленной ЛВС и к *Internet*. (Схемы из *Net Cracker*).

Далее представлены примеры функциональной реализации указанных пунктов. Рассмотрим описание схемы соединения активного оборудования на примере проекта «Мебельный магазин» и реализация его в инструментальной среде проектирования Net Cracker.

Мебельный магазин состоит из двух этажей, что бы сделать это в проекте *Net Cracker*, необходимо по пиктограмме «здание» кликнуть правой кнопкой мыши, из меню выбрать команду *Exspand*. Откроется новое окно, его название будет «Мебельный магазин». Далее в это окно из браузера устройств выбираем «*Bildings, campuses and LAN workgroups*» из палитры в нижнем окне браузера, зажав левую клавишу мыши, перетаскиваем объект «*Floor*». Данную операцию проделаем дважды, т.к. у нас в проекте два этажа. Результат представлен на рисунке:



Рисунок 1 – Проект «Мебельный магазин»

Для проектирования первого этажа необходимо кликнут по его иконке правой клавишей, выбрать команду *Exspand*. Появится новое пустое окно с заголовком «Этаж 1». По проекту на первом этаже 8 комнат: серверная, кладовая, сервисное обслуживание, менеджеры, два рабочих помещения, охрана, туалет. В браузере устройств выбираем *«Bildings, campuses and LAN workgroups»* из палитры в нижнем окне браузера, зажав левую клавишу мыши необходимо перетащить объект *«Room»*(комната) на окно проекта. Далее даем название каждой комнате. Результат представлен на рисунке:



Рисунок 2 – Проект первого этажа мебельного магазина

По проекту на первом этаже 6 компьютеров: 1 ПК в кабинете охраны, 1 ПК в комнате сервисного обслуживания, 3 ПК у менеджеров, по одному в каждом рабочем помещение. Далее каждую комнату необходимо открыть командой *Expand*, поместить в нее нужное количество компьютеров. Далее необходимо выбрать и переместить рабочую станцию *Ethernet Workstation*. Далее в каждую рабочую станции поместить платы *Fast Ethernet adapter* из раздела *LAN adapters* \rightarrow *Ethernet*. На рисунке можно представлена комната менеджеров, 3 ПК в ней и конфигурация компьютеров:



Рисунок 3 – Комната менеджеров и конфигурация компьютеров

В оставшихся комнатах нужно проделать аналогичные действия, за исключение серверной. Для серверной комнаты применить команду Espand, из браузера устройств выбираем «Bildings, campuses and LAN workgroups» из палитры в нижнем окне браузезажав левую клавишу мыши перетащить «Clopa, set»(телекоммуникационный шкаф). К нему так же применить команду *Espand*, затем в браузере во вкладке «LAN servers» выбирать Ethernet Server. Далее на сервер необходимо установить программное обеспечение (драйвера). Для этого в браузере устройств прокрутить до вкладки «Network and enterprise software» («Сеть и программное обеспечение предприятия») и развернуть ее, нажав на знак плюс (+). Нажать на «Server software» («Программное

обеспечение Сервера»). В сервер поместить: Small office database server, FTP server, File server, E-mail server.

Далее необходимо выбрать *switch*. В данном проекте будет использоваться один *swich* в целях экономии и увеличения быстродействия сети. Идея самой сети заключается в том, что все компьютеры будут соединяться с одним *swich*. По проекту сеть нужно строить, так, что бы потом была возможность ее расширить, тогда необходимо выбрать 32 портовый *swich* из браузера устройств: *Switch* – *Ethernet* – *Hitachi Data System* - *HiSpeed* 32-port Fast *Ethernet* Workgroup Switch 150-10-32.

Соединить *swich* и *server* можно следующим образом: на панели инструментов щелкнув по кнопке связь (*Link devices*), затем кликнув по серверу и свичу, появится окно соединения, в котором выбирать порт соединения, затем кликнув по кнопке *Link* и установить расстояние между сервером и свичом. Для сохранения связи нажать *Enter*. Результат действий показан на рисунке:

ink Assistant			
Device #1			Device #2
	Ethernet, Fast Ethernet	n 10BaseT	
	СОМ	TOBaseT 💭	
Ethernet Server		UBaseT 10BaseT	HiSpeed 32-port Fast
Device Properties	Port Properties	Ret Properties	Ethernet Workgroup
Device Properties	Port Properties		
		10BaseT	
Link settings		Cable Length	
Protocol		2 m 🔻	Link
Ethernet 10BASE-T	•		Unlink
Media	Bandwidth		Close
Twisted Pair	▼ 10 Mbit/s	•	
			нер

Рисунок 4 – Соединение swich и server

Далее необходимо приступить к созданию проекта второго этажа. На нем находится 7 комнат: отдел кадров 1 ПК, директор 1 ПК, секретарь 1 ПК, зам директора 1 ПК, столовая, охрана 1 ПК, бухгалтерия 2 ПК. Комнаты и компьютеры в них создаем аналогично первому этажу.

Далее необходимо соединить все компьютеры в единую сеть. Для этого нужно открыть окно «Мебельный магазин» и со второго этажа на первый «проложим» сеть. Так как на втором этаже 7 компьютеров, то линий связи со второго этажа на первый тоже должно быть 7 – по числу компьютеров. Связь создается той же кнопкой *Link devices* с панели инструментов. В окне первого этаже появятся 7 значков связи (выделены красным). Результат действий представлен на следующем рисунке:



Рисунок 5 – Соединение компьютеров в единую сеть

Далее от каждого значка связи необходимо протянуть линии связи к серверной. Связь нарисована пунктиром, это говорит о том, что она еще не активна.



Рисунок 6 – Соединение сервера и компьютеров

Затем необходимо открыть серверную комнату и от значков связи так же протянуть линии связи к телекоммуникационному шкафу, линии связи так же будут обозначены пунктиром. Результат на рисунке:



Рисунок 7 - Соединение компьютеров в телекоммуникационном шкафу

Далее необходимо открыть телекоммуникационный шкаф и соединить значки связи с *swich*. Линии связи до сих пор являются не доделанными, что бы их доделать, нужно открыть окно второго этажа, заметить, что там появились значки дополнительной связи, а если навести на них мышь, то можно увидеть, что это значки связи со *swich*. Далее необходимо создать связь от каждой комнаты к значкам дополнительной связи. Результат предыдущих действий показан на рисунке:



Рисунок 8 - Соединение компьютеров

Далее требуется создать связи непосредственно в каждой комнате. На рисунке представлен пример связи в комнате зам директора:

🖉 Зам. Директора			
зам. директора			
LINK Assistant			
Device #1	1999-1999		Device #2
	10BaseT	Ethernet, Fast Ethernet	
Hispand 22 pert East	UBaseT	Fast Ethernet	
Ethernet Workgroup	💌 💭 🛛 10BaseT		зам. директора
Device Properties	Port Properties	Port Properties	Device Properties
10BaseT			
Link settings		_ L ongth	
Protocol			Link
Ethernet 10BASE-T	-	J34 Im 💌	Unlink
Media	Bandwidth		
Twisted Pair	▼ 10 Mbit/s	•	Close
		-	Help

Рисунок 9 – Связи в комнате директора

Аналогично поступить с другими комнатами. После того, как были соединены компьютеры в каждой комнате, линии связи стали синего цвета, что означает, что они активны. Результат подключения второго этажа к серверной представлен на рисунке:



Рисунок 10 – Схема подключения второго этажа

Далее необходимо подключить первый этаж, это делается аналогично тому, что было проделано.

Два этажа соединены сетью между собой, осталось соединить здание с внешней сетью *Workgroup*. Чтобы это сделать, необходимо перейти в окно *Top*, линией связи соединить здание и *Workgroup*, после этого в окне «Мебельный магазин» появился значок \mathbf{M} , который соединить с первым этажом. Далее открыть первый этаж и появившийся значок соединить с «Серверной», затем открыв «Серверную», значок соединить с «Телекоммуникационным шкафом», а в нем значок соединяем со *swich*, связь готова. Результат представлен на рисунке:



Рисунок 11 – Соединение сетью двух этажей

Чтобы проверить работоспособность сети необходимо создать трафик. На втором этаже открыть комнаты «Секретарь» и «Директор». Установить на каждый из ПК из браузера устройств, вкладки «Network and enterprise software» - «Server software» -Small office database server. Задать трафик между ПК секретарь и директор кнопкой Set Traffics, откроется окно в нем кликнуть по Small Office peer-to-per трафик между этими двумя рабочими ПК, нажать Assign. Повторить операции, но уже в окне «Profiles» выбрать «Small office database server». Нажать на кнопку Start на инструментальной панели Control. Трафик от рабочих станций появится и «потечет» сквозь коммутатор. По такому принципу можно соединить остальные ПК. Например: на каждый ПК устанавливаем из вкладки «Network and enterprise software» - «Server software» - E-mail server, соединить их кнопкой Set Traffics, в окне «Profiles» выбрать «E-mail», получается, что все ПК могут обмениваться сообщениями.

Пример назначения *IP* адресов хостам, формирование масок подсетей, назначение *IP* адресов портам маршрутизаторов в среде *Pacet Tracer*.

Пусть необходимо создать следующую схему сети, представленную на рисунке:



Рисунок 12 – Схема сети

Настройка сети происходит следующим образом:

– *Server1*–*DNS* и *Web* сервер;

– Server2 – *DHCP* сервер;

– Компьютер ПК1 получает параметры протокола *TCP/IP* с *DHCP* сервера и открывает сайт *www.rambler.ru* на *Server1*.

Этап 1. Задание параметров протокола *TCP/IP* на ПК1 и серверах.

Вход в конфигурацию ПК1 и установка настройки *IP* через *DHCP* сервер показаны на рисунке:

изическое пространство	Конфигурация	Рабочий стол	Software/Service	S			
ОБЩЕЕ	FastEthernet						
астройки алгорити: Сост	Состояние порта						
ИНТЕРФЕЙС Прог	пускная способное	СТЬ		🔽 Авто			
0 1	0 M6/c	@ 1	00 M6/c				
Дуп	лекс Іолный дуплекс		Полулуплекс	🔽 Авто			
MAC	-алрес	60 2EC6 6BC8					
Ha	стройка IP НСР						
© C	татический						
IP-a	дрес						
Mac	ка подсети						
На Адр	стройка IPv6 ес локального кан	ала:					
0	HCP						
A (0)	втоконфигурация						
• C	татический						
		III	101				

Рисунок 13 – Настройка ІР на ПК1

Задание в конфигурации серверов следующих настроек ІР:

- Server1: IP адрес 10.0.0.1, маска подсети 255.0.0.0
- Server2: IP адрес 10.0.0.2, маска подсети 255.0.0.0

Этап 2. Настройка службы DNS на Server1.

В конфигурации *Server1* перейти на вкладку *DNS* и задать две ресурсные записи в прямой зоне *DNS*:

 в ресурсной записи типа А связать доменное имя компьютера с его *IP* адресом рисунок 14 и нажать кнопку «Добавить»:

изическое пространст	во Конфигураци	я Рабочий стол	Software/Services
OFILIEE A			
Настройки		DNS	
астройки алгоритма		Bra	Выкл
СЛУЖБЫ			
HTTP Re	esource Records		
DHCP	Ma serve	r1.rambler.ru Ти	A Record
TFTP			
DNS	nec 10.0.0.1		
SYSLOG	apec		
AAA	Добавить	Сохранить	Удалить
NTP	19 Имя	Тип	Данные
			Harming
EMAIL			
EMAIL			
ЕМАІL FTP ИНТЕРФЕЙС			

Рисунок 14 – Ввод ресурсной записи типа А

в ресурсной записи типа *CNAME* связать псевдоним сайта с компьютером:

🖗 Server1		1		
Физическое простран	нство	Конфигурация	Рабочий стол	Software/Services
ОБЩЕЕ ^ Настройки			DNS	
астройки алгоритма	Служ	ба DNS	Вкл	🔘 Выкл
НТТР	Resou	irce Records		
DHCP	Имя	www.ran	nbler.ru Tu	CNAME -
DNS	Имя	узла server1.ran	nbler.ru	
		Добавить	Сохранить	Удалить
EMAIL	Nº	Имя	Тип	Данные
FTP MHTEPΦEЙC FastEthernet	1	server1.rambler.ru	A Record	10.0.0.1

Рисунок 15 – Ввод ресурсной записи типа СNAME

В конфигурации *Server1* войти на вкладку *HTTP* и задать стартовую страницу сайта *www.rambler.ru*:

Server1				
Физическое простра	анство	Конфигурация	Рабочий стол	Software/Services
ОБЩЕЕ ^ Настройки			нттр	
астройки алгорити: СЛУЖБЫ	HTTF)	HTTP	S
HTTP	BI	кл 🔘 Выкл	Вк	л 🔘 Выкл
DHCP				
TFTP	Имя ф	айла: index.html		
DNS	<html< td=""><td>></td><td></td><td></td></html<>	>		
SYSLOG	<cent< td=""><td>er><font <="" size="+2" td=""><td>color='blue'>Cisc</td><td>o Packet</td></td></cent<>	er> <font <="" size="+2" td=""><td>color='blue'>Cisc</td><td>o Packet</td>	color='blue'>Cisc	o Packet
AAA	Trace	er <td>></td> <td></td>	>	
NTP	www	V.RAMBLER.RU		
EMAIL				
FTP	Serve	r1		
ИНТЕРФЕЙС				
FastEthernet	<td>11></td> <td></td> <td></td>	11>		

Рисунок 16 – Стартовая страница сайта

Включить командную строку на Server1 и проверить работу службы DNS. Для проверки прямой зоны DNS сервера необходим ввод команды: SERVER>nslookup www.rambler.ru Если все правильно, то получится отклик, представленный на рисунке 17, с указанием полного доменного имени *DNS* сервера в сети и его *IP* адрес.

SERVER>nslookup www.rambler.ru
Server: [10.0.0.1] Address: 10.0.0.1
Non-authoritative answer: Name: server1.rambler.ru Address: 10.0.0.1
Aliases: server1.rambler.ru
SERVER>

Рисунок 17 – Проверка прямой зоны DNS

Этап 3. Настройка DHCP службы на Server2. Для этого необходимо войти в конфигурацию Server2 и на вкладке DHCP настроить службу (рисунок 18):

изическое простра	нство Конфигур	Рабочий ст	стол Software/Services					
ОБЩЕЕ ^			DHCP					
астройки алгоритма СЛУЖБЫ	Служба) Вкл		🔘 Вь	кл		
HTTP	Имя пула	server	Pool					
DHCP	Основной шлюз	3 0 0 0 0						
TFTP	DNS-cepsep	epBep 10.0.0.1						
DNS		10.0.0.1		10	0	0	10	
SYSLOG	начальный ір-ад	tpec:		10	U	U	10	
AAA	Маска подсети:			255	0	0	0	
NTP	Максимальное к	ссимальное кол-во						
EMAIL	пользователей:		5					
FTP	TETD-congen:	0.0.0	0					
ИНТЕРФЕЙС	пп сервер.	0.0.0.	0					
FastEthernet	Добавить		Сохранит	гь		Удали	ІТЬ	
	Имя пу Основно	ой IDN	S-cer Началі	ьный Ма	аска п	Макс.	к HTTP-	
	serv 0.0.0.0	10.	0.0.1 10.0.0.	10 255	5.0	5	0.0.0.0	
	•						+	

Рисунок 18 – Настройка *DHCP* сервера

Этап 4. Проверка работы клиента.

Войти в конфигурации хоста ПК1 на рабочий стол и в командной строке сконфигурировать протокол *TCP/IP*. Командой *PC>ipconfig* /*release* сбросить старые параметры *IP* адреса, а командой: *PC>ipconfig* /*renew* получить новые параметры с *DHCP* сервера:

PC>ipconfig /release	
IP Address	0.0.0.0
Subnet Mask	0.0.0.0
Default Gateway:	0.0.0.0
DNS Server	0.0.0.0
PC>ipconfig /renew	
IP Address:	10.0.0.10
Subnet Mask	255.0.0.0
Default Gateway:	0.0.0.0
DNS Server:	10.0.0.1
PC>	

Рисунок 19 – Конфигурация протокол ТСР/ІР клиента

Открыть сайт www.rambler.ru в браузере на клиенте:

TIK1		A		
Физическое пространство	Конфигурация	Рабочий стол	Software/Serv	vices
Веб-браузер				X
C > URL http://www.ra	mbler.ru		Перейти	Стоп
WWW.RAMBLER.RU	Cisco Packe	t Tracer		*
Server1				

Рисунок 20 – Проверка работы клиента

Решение задачи по формирование таблиц маршрутизации в проекте компьютерной сети. Протоколы маршрутизации – это правила, по которым осуществляется обмен информации о путях передачи пакетов между маршрутизаторами. Протоколы характеризуются временем сходимости, потерями и масштабируемостью. В настоящее время используется несколько протоколов маршрутизации *RIP*, *OSPF*, *EIGP* и др. Одна из главных задач маршрутизатора состоит в определении наилучшего пути к заданному адресату. Маршрутизатор определяет пути (маршруты) к адресатам или из статической конфигурации, введённой администратором, или динамически на основании маршрутной информации, полученной от других маршрутизаторов. Маршрутизаторы обмениваются маршрутной информацией с помощью протоколов маршрутизации.

Маршрутизатор хранит таблицы маршрутов в оперативной памяти. Таблица маршрутов это список наилучших известных доступных маршрутов. Маршрутизатор использует эту таблицу для принятия решения - куда направлять пакет. В случае статической маршрутизации администратор вручную определяет маршруты к сетям назначения. В случае динамической маршрутизации – маршрутизаторы следуют правилам, определяемым протоколами маршрутизации для обмена информацией о маршрутах и выбора лучшего пути.

Статические маршруты не меняются самим маршрутизатором. Динамические маршруты изменяются самим маршрутизатором автоматически при получении информации о смене маршрутов от «соседних» маршрутизаторов. Статическая маршрутизация потребляет мало вычислительных ресурсов и полезна в сетях, которые не имеют нескольких путей к адресату назначения. Если от маршрутизатора к маршрутизатору есть только один путь, то часто используют статическую маршрутизацию.

Пусть необходимо произвести настройку статической маршрутизации с помощью графических мастеров интерфейса *Cisco Packet Tracer* для следующей схемы сети, представленной на рисунке 21:



Рисунок 21 – Схема сети в среде Packet Tracer

На данной схеме представлена корпоративная сеть, состоящая из следующих компонентов: Сеть 1 – на *Switch1* замыкается сеть первой организации:

Таблица 3 - Сеть первой организации

Компьютер	IP адрес	Функции
Comp2	192.168.1.2/24	DNS и HTTP сервер
Comp3	192.168.1.3/24	<i>DHCP</i> сервер
Comp4	Получен с <i>DHCP</i> сервера	Клиент сети

В данной сети на *Comp2* установлен *DNS* и *Web* сервер с сайтом организации.

На *Сотр3* установлен *DHCP* сервер. Компьютер *Сотр4* получает с *DHCP* сервера *IP* адрес, адрес *DNS* сервера провайдера (сервер *Provider*) и шлюз. Шлюз в сети – 192.168.1.1/24.

Сеть 2 – на *Switch2* замыкается сеть второй организации (таблица 4):

Таблица 4 - Сеть второй организации

Компьютер	<i>IP</i> адрес	Функции
Comp5	10.0.0.5/8	DNS и HTTP сервер
Сотрб	10.0.0.6/8	<i>DHCP</i> сервер
Comp7	Получен с <i>DHCP</i> сервера	Клиент сети

В данной сети на *Comp5* установлен *DNS* и *Web* сервер с сайтом организации.

На *Comp4* установлен *DHCP* сервер. Компьютер *Comp7* получает с *DHCP* сервера *IP* адрес, адрес *DNS* сервера провайдера (сервер *Provider*) и шлюз. Шлюз в сети – 10.0.0.1/8.

Сеть 3 – на *Hub1* замыкается городская сеть 200.200.200.0/24. В сети установлен *DNS* сервер провайдера (компьютер *Provaider* с *IP* адресом -200.200.200.10/24), содержащий данные по всем сайтам сети (*Comp2*, *Comp5*, *Comp8*).

Сеть 4 – маршрутизатор *Router3* выводит городскую сеть в интернет через коммутатор *Switch3* (сеть 210.210.210.0/24). На *Comp8* (*IP* адрес 210.210.210.8/24, шлюз 210.210.210.3/24.) установлен *DNS* и *Web* сервер с сайтом.

Маршрутизаторы имеют по два интерфейса:

- *Router1* 192.168.1.1/24 и 200.200.200.1/24.
- *Router2* 10.0.0.1/8 и 200.200.2/24.
- *Router3* 210.210.210.3/24 и 200.200.200.3/24.
 Рассмотри пример решения задачи, в которой требуется
- настроить сети организаций;
- настроить DNS сервер провайдера;
- настроить статические таблицы маршрутизации на роутерах;

– проверить работу сети – на каждом из компьютеров - *Comp4*, *Comp7* и *Comp8*. С каждого из них должны открываться все три сайта корпоративной сети.

Приступим к настройке статической маршрутизации на роутерах. Поскольку на представленной схеме четыре сети, то таблицы маршрутизации как минимум должны содержать записи к каждой из этих сетей – т.е. четыре записи. На роутерах *Cisco* в таблицах маршрутизации, как правило, не прописываются пути к сетям, к которым подсоединены интерфейсы роутера. Поэтому на каждом роутере необходимо внести по две записи.

Для настройки первого роутера необходимо войти в конфигурацию маршрутизатора и в интерфейсах установите *IP* адрес и маску подсети. Затем в разделе «Маршрутизация» открыть вкладку «Статическая», внести данные и нажать кнопку «Добавить»:

ОБЩЕЕ Настройки настройки алгоритма Сеть Маршрутизация Статическая RIP	Стати	ческие	маршруты
астройки алгорити: Сеть маршрутизация Маска Статическая Следую RIP			10.0.0.0
МАРШРУТИЗАЦИЯ Маска Статическая Следую RIP			
Статическая Следую RIP			255.0.0.0
RIP	Следующий переход		200.200.200.2
			Добавить
КОММУТАЦИЯ			
База данных VLAN Сетево	й адрес		
ИНТЕРФЕЙС			
FastEthernet0/0			
FastEthernet0/1			

Рисунок 22 – Данные для сети 10.0.0/8

В результате у вас должны появиться две записи в таблице маршрутизации:

Ризическое простра	нство Конфигурация С		
ОБЩЕЕ *	Статич	ческие маршруты	
астройки алгоритиа	Сеть	210.210.210.0	
МАРШРУТИЗАЦИЯ	Маска	255.255.255.0	
Статическая	Следующий переход	200.200.200.3	
RIP		Добавить	
коммутация			
База данных VLAN	Сетевой адрес		
ИНТЕРФЕЙС	10.0.0.0/8 via 200.200.200.2		
FastEthernet0/0	210.210.210.0/24 via 200.200.200.3		
FastEthernet0/1	210.210.210.0/24 Via 200.200.200.3		

Рисунок 23 – Формирование статической таблицы маршрутизации.

Чтобы посмотреть полную настройку таблицы маршрутизации, выберите в боковом графическом меню инструмент «Проверка» (пиктограмма лупы), щелкните в схеме на роутере и выберите в раскрывающемся меню пункт «Таблица маршрутизации».

После настройки всех роутеров в вашей сети станут доступны *IP* адреса любого компьютера, и вы сможете открыть любой сайт с компьютеров *Comp4*, *Comp7* и *Comp8*.

Динамическая маршрутизация. Статическая маршрутизация не подходит для больших, сложных сетей потому, что обычно сети включают избыточные связи, многие протоколы и смешанные топологии.

Маршрутизаторы в сложных сетях должны быстро адаптироваться к изменениям топологии и выбирать лучший маршрут из многих кандидатов.

IP сети имеют иерархическую структуру. С точки зрения маршрутизации сеть рассматривается как совокупность автономных систем. В автономных подсистемах больших сетей для маршрутизации на остальные автономные системы широко используются маршруты по умолчанию.

Динамическая маршрутизация может быть осуществлена с использованием одного и более протоколов. Эти протоколы часто группируются согласно того, где они используются. Протоколы для работы внутри автономных систем называют внутренними протоколами шлюзов (*interior gateway protocols* (*IGP*)), а протоколы для работы между автономными системами называют внешними протоколами шлюзов (*exterior gateway protocols* (*EGP*)). К протоколам *IGP* относятся *RIP*, *RIP v2*, *IGRP*, *EIGRP*, *OSPF* и *IS-IS*. Протоколы *EGP3* и *BGP4* относятся к *EGP*. Все эти протоколы могут быть разделены на два класса: дистанционно-векторные протоколы и протоколы состояния связи.

Пример настройки протокола *RIP*. Пусть необходимо создать схему, представленную на рисунке:



Рисунок 24 – Схема сети

На схеме представлены следующие три сети:

- *Switch1* сеть 10.11.0.0/16
- Switch2 сеть 10.12.0.0/16.
- Сеть для роутеров 10.10.0.0/16.

Введите на устройствах следующую адресацию: маршрутизаторы имеют по два интерфейса:

- *Router1* 10.11.0.1/16 и 10.10.0.1/16.
- *Router2* 10.10.0.2/16 и 10.12.0.1/16.
- ПК11 10.11.0.11/16.
- ПК12 10.12.0.12/16.

Для проведения настройки протокола *RIP* на маршрутизаторе *Router1* необходимо войти в конфигурации в консоль роутера и выполнить следующие настройки:

- ввод в привилегированный режим: *Router1>en*;
- ввод в режим конфигурации: *Router1*>#*conf t*;
- ввод в режим конфигурирования протокола *RIP*: *Router1(config)*#router rip;

– подключение клиентской сети к poyrepy: *Router1(config-router)#network 10.11.0.0*;

– подключение второй сети к роутеру: *Router1(config-router)#network 10.10.0.0.*

Пример настройки использования второй версии протокол *RIP*: *Router1(config-router)*#version 2:

- выход из режима конфигурирования протокола *RIP*: *Rout*erl(config-router)#exit;

- выход из консоли настроек: *Router1(config)#exit*;

- сохранение настройки в память маршрутизатора: *Router1>#write memory*.

Аналогично проводится настройка протокола *RIP* на маршрутизаторе *Router2*.

Проверяется связь между компьютерами ПК11 и ПК12 командой *ping*. Если связь есть – все настройки сделаны, верно.

Пример настройки протокола RIP для проекта корпоративной сети. Пусть необходимо создать схему, представленную на рисунке 25.



Рисунок 25 – Схема сети

В четырех сетях: 11.0.0.0/8, 12.0.0.0/8, 13.0.0.0/8 и 14.0.0.0/8 установлены компьютеры с адресами:

- *Comp1* 11.0.0.11, маска 255.0.0.0.
- Сотр2-12.0.0.12, маска 255.0.0.0.
- Сотр3-13.0.0.13, маска 255.0.0.0.
- Сотр4-14.0.0.14, маска 255.0.0.0.

Между ними находится корпоративная сеть с шестью маршрутизаторами.

На маршрутизаторах заданы следующие интерфейсы, представленные в таблице:

Маршрутизатор	Интерфейс 1	Интерфейс 2	Интерфейс 3
<i>Router1</i>	11.0.0.1/8	21.0.0.1/8	31.0.0.1/8
Router2	21.0.0.2/8	51.0.0.2/8	
Router3	12.0.0.3/8	61.0.0.3/8	51.0.0.3/8
Router4	31.0.0.4/8	81.0.0.4/8	13.0.0.4/8
Router6	61.0.0.6/8	81.0.0.6/8	14.0.0.6/8

Таблица 5 – Интерфейсы маршрутизаторов

Пусть необходимо настроить маршрутизацию по протоколу *RIP* на каждом из роутеров. Чтобы убедиться в том, что маршрутизатор действительно правильно сконфигурирован и работает корректно, необходимо просмотреть таблицу *RIP* роутера, используя команду *show* следующим образом: *Router*#show ip route rip.

Например, для шестого маршрутизатора *Router6* таблица будет иметь следующий вид, представленный на рисунке:

```
Router6>en
Router6#show ip route rip
     11.0.0.0/8 [120/2] via 81.0.0.4, 00:00:08, FastEthernet0/1
R
R
    12.0.0.0/8 [120/1] via 61.0.0.3, 00:00:08, Ethernet0/0/0
    13.0.0.0/8 [120/1] via 81.0.0.4, 00:00:08, FastEthernet0/1
R
R
     21.0.0.0/8 [120/2] via 61.0.0.3, 00:00:08, Ethernet0/0/0
                [120/2] via 81.0.0.4, 00:00:08, FastEthernet0/1
     31.0.0.0/8 [120/1] via 81.0.0.4, 00:00:08, FastEthernet0/1
R
     51.0.0.0/8 [120/1] via 61.0.0.3, 00:00:08, Ethernet0/0/0
R
Router6#
```

Рисунок 26 – Таблица маршрутизации RIP

Данная таблица показывает, что к сети 21.0.0.0 есть два пути: через *Router4* (сеть 81.0.0.0) и через *Router3* (сеть 61.0.0.0).

Пример настройки протокола OSPF. Пусть необходимо использовать схему сети, представленную на рисунке 24, и провести настройку протокола OSPF на маршрутизаторе Router1.

Для этого необходимо войти в конфигурации в консоль роутера и выполнить следующие настройки:

- вход в привилегированный режим: *Switch>en*;
- вход в режим конфигурации: *Switch1*#*conf t*;
- вход в режим конфигурирования протокола OSPF: Router1(config)#router ospf 1.

В команде *router ospf* <идентификатор_процесса> под идентификатором процесса понимается уникальное числовое значение для каждого процесса роутинга на маршрутизаторе. Данное значение должно быть больше в интервале от 1 до 65535. В *OSPF* процессам на роутерах одной зоны принято присваивать один и тот же идентификатор.

Подключение клиентской сети к poytepy: Router1(configrouter)#network 10.11.0.0.

Подключение второй сети к роутеру: *Router1(config-router)*#network 10.10.0.0.

Задание использования второй версии протокол OSPF: Router1(config-router)#version 2.

Выход из режима конфигурирования протокола OSPF: Router1(config-router)#exit.

Выход из консоли настроек: *Router1(config)*#exit.

Сохранение настройки в память маршрутизатора: *Switch1*#*write memory*.

Список команд для настройки *NAT*:

– обозначение Интернет интерфейса: *interface FastEthernet0/0 ip nat outside*;

- обозначение локального интерфейса: *interface Vlan1 ip nat in-side*;

– создание списка IP, имеющего доступ к NAT: ip access-list extended NAT permit ip host 192.168.?????? any;

– включение *NAT* на внешнем интерфейсе: *ip nat inside source list NAT interface FastEthernet0/0 overload*.

Посмотреть существующие трансляции можно командой «show ip nat translations».

Отладка запускается командой «debug ip nat».

Настройка Static NAT: router(config)#ip nat inside source static <local-ip> <global-ip> *router(config)*#*interface fa0/4* router(config-if)#ip nat inside router(config)#interface fa0/4 router(config-if)#exit *router(config)*#*interface s0* router(config-if)#ip nat outside Настройка Dynamic NAT: router(config)#ip nat pool name start-ip end-ip {netmask netmask | *prefix-length prefix-length*} router(config)#access-list <acl-number> permit <source-IP> [source-wildcard] router(config)#ip nat inside source list <acl-number> pool <name> router(config)#interface fa0/4 *router(config-if)*#*ip nat inside* router(config-if)#exit router(config)#interface s0 router(config-if)#ip nat outside. Настройка Overloading: router(config)#access-list acl-number permit source-IP sourcewildcard router(config)#ip nat inside source list acl-number interface interface overload router(config)#interface fa0/4 router(config-if)#ip nat inside router(config-if)#exit router(config)#interface s0 router(config-if)#ip nat outside

Пример преобразования сетевых адресов NAT в проекте компьютерной сети. В данном примере необходимо решить задачу вывода компьютеров локальной сети организации в интернет. Локальная сеть настроена в частной адресации – в сети 10.0.0.0, адреса которой не имеют выхода в интернет. Пусть необходимо настроить службу *NAT* для следующей схемы сети, представленной на рисунке 27.



Рисунок 27 – Схема сети.

Необходимо сконфигурировать *NAT* на маршрутизаторе *Router1*. Для настройки *NAT* на роутере нам необходимо будет выполнить следующие шаги:

- зайти в настройки *Router1*, во вкладку *CLI*;

- для входа в режим администратора ввести команду *enable (en) Router>en*;

– для входа в режим настройки вводим команду *config tRouter#config t*.

Интерфейс *FastEthernet* 0/0 это внутренний интерфейс, к которому подключены рабочие станции. Для настройки *NAT* на роутере необходимо это обозначить в настройках. Это можно сделать при помощи следующих команд:

– войти в настройки интерфейса: *Router(config)#int FastEthernet* 0/0;

- объявить интерфейс внутренним интерфейсом: *Router(config-if)#ip nat inside*;

- осуществить выход из настроек интерфейса: Router(configif)#exit.

Аналогично настроить интерфейс *FastEthernet* 0/1, который подключен к сети провайдера, лишь с тем различием, что он будет являться внешним интерфейсом *NAT*:

- войти в настройки интерфейса: *Router(config)#int FastEthernet* 0/1;

– объявить интерфейс внешним интерфейсом NAT: Router(config-if)#ip nat outside;

- выйти из настроек интерфейса: Router(config-if)#exit.

Задаем пул внешних адресов, которые В будут транслироваться внутренние адреса. Для задания пула, содержащего только один адрес – адрес внешнего интерфейса роутера - необходимо ввести команду: Router(config)#ip nat pool natpool 11.0.0.0 11.0.0.1 netmask 255.0.0.0.

При задании пула адресов необходимо указать первый и последний адреса из входящей в пул последовательности адресов. Если в пуле один адрес необходимо указать его два раза.

Задать список доступа: *Router(config)#access-list 34 permit any*, где 34 – число от 1 до 99 обозначает № списка доступа и задается администратором. *Апу* – ключевое слово, означает, что список доступа будет разрешать пакеты с любым адресом отправителя.

Далее ввод последней команды, которая, и включает NAT на *Router0*. Команда, является основной, но без задания всех предыдущих параметров она работать не будет: *Router(config)*#*ip nat inside source list 34 pool natpool overload*.

Данная команда говорит роутеру, что у всех пакетов, полученных на внутренний интерфейс и разрешенных списком доступа номер 34, адрес отправителя будет транслирован в адрес из *NAT* пула «*natpool*». Ключ *overload* указывает, что трансляции будут перегружены, позволяя нескольким внутренним узлам транслирован в адрес.

3.3 Теоретико - расчетная часть проекта компьютерной сети

Расчет времени двойного оборота сигнала в конфигурации *Ethernet*. Модель, применяемая для оценки конфигурации *Ethernet*, основана на подсчете временных характеристик данной конфигурации. В ней применяется две системы расчетов: одна предполагает вычисление двойного (кругового) времени прохождения сигнала по сети, а другая – проверку допустимости получаемого (межкадрового) временного интервала. При этом расчеты в обеих системах расчетов ведутся для наихудшего случая.

При первой системе расчетов используются такие понятия, как «начальный сегмент», «промежуточный сегмент» и «конечный сегмент». Необходимо отметить, что промежуточных сегментов может быть несколько, а начальный и конечный сегменты при разных расчетах могут меняться местами. Для расчетов используются величины задержек, представленные в приложение Б.

Примечание. Задержки даны в битовых интервалах.

Расчет сводится к следующему:

- в сети выделяется путь наибольшей длины;

- если длина сегмента не максимальна, то рассчитывается двойное (круговое) время прохождения в каждом сегменте выделенного пути по формуле 2:

$$t_s = L^* t_1 + t_0 \tag{2}$$

где

L – длина сегмента в метрах (при этом надо учитывать тип сегмента: начальный, промежуточный или конечный);

- если длина сегмента максимальна, то из таблицы для него берется величина задержки *t_m*;

- суммарная величина задержек всех сегментов выделенного пути не должна превышать 575 битовых интервалов;

- затем необходимо проделать те же действия для обратного направления выбранного пути (то есть, считая конечный сегмент начальным, и наоборот);

- если задержки в обоих случаях не превышают 575 битовых интервалов, то сеть работоспособна.

Если в выбранной вами конфигурации сети путь наибольшей длины не столь очевиден, то подобные расчеты необходимо произвести для всех путей, претендующих на наибольшую задержку сигнала. В любом случае двойное время прохождения в соответствии со стандартом недостаточно, чтобы сделать окончательный вывод о работоспособности сети.

Расчет сокращения межкадрового интервала в конфигурации *Ethernet*. Чтобы признать конфигурацию сети корректной, нужно рассчитать также уменьшение межкадрового интервала репитерами (репитерными концентраторами).

Эта величина не должна быть меньше, чем 49 битовых интервалов. Для вычислений здесь также используются понятия начального сегмента и промежуточного сегмента (конечный сегмент не вносит вклада в сокращение межкадрового интервала, так как пакет доходит по нему до принимающего компьютера без прохождения репитеров и репитерных концентраторов).

Для расчета сокращения межкадрового интервала можно воспользоваться значениями максимальных величин уменьшения межкадрового интервала при прохождении репитеров (репитерных концентраторов) различных физических сред приведенными в Таблице 2, приложения Б. Суммируя величины сокращений межкадрового интервала для наибольшего пути в выбранной конфигурации и сравнивая сумму с предельной величиной в 49 битовых интервалов, делаем вывод о работоспособности сети. Такие же вычисления проводятся и для обратного направления по этому же пути.

Для определения работоспособности сети Fast Ethernet стандарт IEEE 802.3 предлагает две модели, называемые Transmission System Model 1 и Transmission System Model 2. При этом первая модель основана на несложных правилах, а вторая использует систему расчетов.

В соответствии с первой моделью, при выборе конфигурации надо руководствоваться следующими принципами:

- сегменты, выполненные на электрических кабелях (витая пара), не должны быть длиннее 100 м;

- сегменты, выполненные на оптоволоконных кабелях, не должны быть длиннее 412 м;

- если используются трансиверы, то трансиверные кабели не должны быть длиннее 50 см.

При выполнении этих правил надо руководствоваться таблицей 3, приложения B, определяющей максимальные размеры (в метрах) зоны конфликта (т.е. максимальное расстояние между абонентами сети, не разделенными коммутаторами). При этом в двух последних столбцах таблицы, относящихся к случаю использования смешанных сред передачи (как витых пар, так и оптоволоконных кабелей), предполагается, что длина витой пары составляет 100 м, применяется только один оптоволоконный кабель. Первая строка относится к соединению двух компьютеров без применения репитера. Нереализуемые ситуации отмечены в таблице прочерками.

Вторая модель основана на вычислениях суммарного двойного времени прохождения сигнала по сети.

Для расчетов в соответствии со второй моделью сначала надо выделить в сети путь с максимальным двойным временем прохождения и максимальным числом репитеров (концентраторов) между компьютерами. Если таких путей несколько, то расчет должен производиться для каждого из них. Расчет в данном случае ведется на основании таблицы 4, приложения В.

Для вычисления полного двойного (кругового) времени прохождения для сегмента сети необходимо умножить длину сегмента на величину задержки на метр, взятую из второго столбца таблицы 2. Если сегмент имеет максимально возможную длину, то можно взять величину максимальной задержки для данного сегмента из третьего столбца таблицы. Затем задержки сегментов, входящих в путь максимальной длины, надо просуммировать и прибавить к этой сумме величину задержки для двух абонентов (три верхние строчки таблицы) и величины задержек для всех репитеров (концентраторов), входящих в данный путь. Суммарная задержка должна быть меньше, чем 512 битовых интервалов.

Задержки в кабеле могут отличаться от тех, которые приведены в таблице 4, приложения В.

Для более точного расчета следует использовать временные характеристики конкретного кабеля, применяемого в сети. Производители кабелей иногда указывают величину задержки на метр длины, а иногда – скорость распространения сигнала относительно скорости света (или *NVP – Nominal Velocity of Propagation*). Связанны эти две величины формулой:

$$t_{3} = \frac{1}{3*10^{10}*NVP}$$
(3)

где

*t*₃ – величина задержки на метр кабеля.

Например, если NVP = 0,4 (40%) от скорости света, то задержка t_3 будет равна 8,34 нс/м или 0,834 битовых интервала. Для вычисления двойного (кругового) времени прохождения нужно удвоенное значение t_3 умножить на длину кабеля. В таблице 6 даны величины NVP для некоторых типов кабелей.

Таблица 6 – Величины NVP для	некоторых типов кабелей
------------------------------	-------------------------

	1		
Фирма	Марка	Категория	NVP
AT&T	1010	3	0,67
AT&T	1041	4	0,70
AT&T	1061	5	0,70
AT&T	2010	3	0,70
AT&T	2041	4	0,75
AT&T	2061	5	0,75
Belden	1229A	3	0,69
Belden	1455A	4	0,72
Belden	1583A	5	0,72
Belden	1245A2	3	0,69
Belden	1457A	4	0,75
Belden	1585A	5	0,75

Для некоторых репитеров и концентраторов изготовители указывают меньшие величины задержек, чем приведенные в таблице 6, что также надо учитывать при выборе конфигурации сети.

3.4 Планирование безопасности в проекте компьютерной сети

Основной критерий для выбора уровня защиты – важность информации. Как правило, рекомендуется централизованная защита на основе сервера. Часто выборочно наделяют пользователей правами доступа к ресурсам, создавая группы с соответствующим уровнем доступа.

Сетевые серверы должны быть расположены в защищенном месте. На каждой рабочей станции должен быть предусмотрен уникальный входной пароль с периодическим обновлением. Для регистрации попыток обращения пользователей к ресурсам следует регулярно проводить аудит сети. С целью предотвращения возможности несанкционированного копирования данных следует использовать бездисковые рабочие станции. Наконец, с этой же целью можно предусмотреть шифрование данных.

Для защиты от безвозвратного разрушения данных обычно предусматривают резервное копирование по составленному плану, а также применение резервных источников бесперебойного питания.

Как правило, при разработке сети необходимо предусмотреть взаимодействие ее с *Internet*. При этом необходимо выбрать соответствующие средства. Прежде всего, это касается выбора канала связи – аналогового или цифрового. Иногда предусматривается использование для этих целей нескольких каналов. Этот выбор зависит от того, какого рода трафик будет передаваться по этим каналам:

- речь;

- электронная почта;

- небольшие файлы;

- большие файлы;

– обмен запросами и ответами с базой данных архитектуры «клиент – сервер» (сравнительно небольшой трафик);

- информация из файлов базы данных, которые хранятся на удаленном сервере и обрабатываются на компьютере – клиенте (как правило, очень интенсивный трафик).

Основываясь на этой информации, определяют приблизительную пропускную способность, которая необходима для разрабатываемой сети.

Название группы	Внутренние	Уровни доступа к	Доступ в Internet и
	ресурсы	внутренним ресур-	электронная почта
		сам	-
Администраторы	Все сетевые	Права администра-	Все сетевые ресурсы
	ресурсы	тора в каталогах, в	
		том числе измене-	
		ние уровня и прав	
		доступа	
Разработчики	Базы данных разра-	Создание, чтение	Все сетевые ресурсы
	батываемых доку-	файлов, запись в	
	ментов	файл, создание под-	
		каталогов и файлов,	
		удаление каталогов,	
		поиск файлов, изме-	
		нение каталогов	
Сотрудники в офисе	Вся информация	Ограничение досту-	Ограничение по IP-
	предприятия	па к папкам (по	адресу (адресата и
	(учреждения)	необходимости)	источника), ограни-
			чение по содержа-
			нию (входящей и
			исходящей корре-
			спонденции)
Сотрудники вне	Вся информация	Ограничение досту-	Ограничение по <i>IP</i> -
офиса	предприятия	па к папкам (по	адресу (адресата и
	(учреждения)	необходимости)	источника), ограни-
			чение по содержа-
			нию (входящей и
			исходящей корре-
			спонденции), аутен-
			тификация удален-
			ного пользователя
			перед осуществ-
			лением доступа
Поставщики, дело-	Специальные ката-	Доступ только к	Ограничение по IP-
вые партнеры, кли-	логи и папки для	специально отве-	адресу (адресата и
енты	производителей,	денным областям	источника). Иденти-
	партнеров и клиен-		фикация и аутенти-
	тов		фикация удаленного
			пользователя
Потенциальные кли-	Специальные ката-	Просмотр объектов	При открытом дос-
енты	логи и папки для	(чтение и поиск	тупе Интрасеть
	клиентов	файлов)	должна быть изоли-
			рована; идентифи-
			кация пользователя
			не требуется

Таблица 7 - Права доступа для групп пользователей

Далее следует выяснить, какие каналы доступа к глобальным сетям существуют в регионе расположения проектируемой сети (их тип и пропускная способность) и какой провайдер предлагает лучшее соотношение цена – производительность. На основе существующих

тарифных планов аргументировать выбор пакета и тип подключения к сети Интернет.

В проектируемой вычислительной сети нужно выборочно наделить пользователей правами доступа к каталогам и создать группы для предоставления доступа к общим сетевым ресурсам. Пример определения прав доступа для групп пользователей показан в таблице 7.

3.5 Проектирование и реализация сетевого приложения

Структура, архитектура сетевого приложения. В данном разделе необходимо представить описание структуры *web*страницы, к какому типу относится (линейная, иерархическая и т.д.), привести схематичное изображение структуры *web*приложения, стрелочками указать вызываемые страницы.

Описать взаимодействие *web*-страниц, указать имена скриптов, которые вызываются при определенных действиях пользователя на данной странице (по проверке вводимых данных, смены изображения, открытия новой страницы и т.д.). Привести фрагменты программного кода.

Описать структуру взаимодействия сетевого приложения с *web*-браузером и *web*-сервером, с базой данных. Например: вызов серверных скриптов, подключение к базе данных, получение данных из базы данных, отправка данных в базу данных. Привести фрагменты программного кода.

Описание схемы базы данных. При описании базы данных выделить объекты и процессы информация, о которых содержится в базе данных. Рассмотреть структуру каждой из таблиц с пояснением, какая информация и о чем в ней хранится, аргументировать выбор первичного ключа.

Определить и аргументировать тип связей между всеми таблицами. Оформить в виде *ER*-диаграммы, в которой сущности (таблицы) обозначить прямоугольником, внутри которого имя таблицы, тип связи обозначаются, как указано в таблице 8.

Пример *ER*-диаграммы показан на рисунке:



Рисунок 28 – Пример *ER*-диаграммы связи между сущностями «Студенты» и «Экзамены»

Таблица 8 – Пример обозначений типов связи

Тип связи	Обозначение
Один к одному	← →
Многие к одному	↔ →
Один ко многим	<>>>
Многие ко многим	↔ → →

Привести скриншоты таблиц в режиме конструктора с указанием названия полей и типов данных полей.

Привести общую схему данных формируемую СУБД, в которой реализуется база данных для взаимодействия с сетевым приложением.

Далее приведены примеры разработки *web*-сайта и взаимодействия с *MySQL*.

Базы данных дают возможность создавать динамические *web*сайты с большим объёмом информации.

Пример, как организовать соединение с сервером баз данных в *web* приложении.

Сначала необходимо получить доступ к серверу, где находится ваша БД *MySQL*. Это делается функцией *mysql_connect* с таким синтаксисом:

mysql connect(cepвep, username, password)

Сначала вы записываете местоположение БД (сервер), а затем указываете *username* и *password*. Если у вас имеется собственный *web* -сайт, вы должны прочесть о размещении вашего *MySQL*сервера на страницах поддержки. *Username* и *password* - часто те же, что для *FTP*-доступа.

Пример *MySQL*-соединения (установки по умолчанию):

mysql_connect("localhost", "root", "") or die (mysql_error());

В этом примере добавлено *or die(mysql_error())*, этот оператор, прерывает работу скрипта, и записывают ошибку, если соединение не удалось.

Правильно будет закрывать БД всякий раз после окончания запрашивания и обновления данных. Это делается функцией *mysql close*.

Существует два способа создания БД и таблиц. Первый, как это делается в *PHP*, второй – с помощью утилиты *PhpMyAdmin*, которая является стандартной на большинстве *web* -хостов. Например, при создании БД используется *SQL*-запрос *CREATE DATABASE имя БД*

mysql_connect("mysql.myhost.com", "user", "sesame") or die(mysql_error());

mysql_query("CREATE DATABASE mydatabase") or die(mysql_error());

mysql_close();

При создании таблиц мы используем *SQL*-запрос *CREATE TABLE* с таким синтаксисом:

СREATE TABLE имя таблицы (имя_столбца1 ТИП_ДАННЫХ, имя_столбца2 ТИП_ДАННЫХ, имя_столбца3 ТИП_ДАННЫХ, ...)

Имя_таблицы и имя_столбца это, соответствующие имена. ТИП_ДАННЫХ используется для специфицирования типа данных, вставляемых в столбец. Вот наиболее распространённые типы:

- *Int* Целые числа.
- *Decimal* Десятичные числа.
- *Char* Краткий текст до 255 символов длиной.
- *Text* Обычный текст до 65,535 символов.
- Longtext Длинные куски текста до 4,294,967,295 символов.
- *Date* Даты в формате ГГГГ-ММ-ДД.
- Тіте Время в формате ЧЧ:ММ:СС.
- *DATETIME* Время и дата в формате ГГГГ-ММ-ДД ЧЧ:ММ:СС.

Пример создания базы данных с использованием *phpMyAdmin*.

Скриншот ниже показывает, как создавать базу данных и таблицы с применением *phpMyAdmin*.

Начинаем со входа в *phpMyAdmin*. В адресной строке набираем *http://localhost/Tools/phpMyAdmin/*

Когда вы вошли, введите имя базы данных, и нажмите кнопку «*Create*»:

Firefox *	host / localhost php +
+> 🔔 http://localho	st/phpmyadmin/
phpMyAdmin Constraints and the second secon	Iocalhost Databases SQL Status Variables Charsets Engines Privileges Actions MySQL localhost ereate new database mydatabase Collation Create mydatabase Collation Create Interface Clanguage (): English English Custom color: English Custom color: Font size: 82%

Рисунок 29 – Осуществление входа в *phpMyAdmin*

Для создания таблицы щёлкните вкладку «Databases» и выберите базу данных, щёлкнув на ней:

Firefox * Allocalhost / localhost php +					
+> Attp://localhost/phpmyadmin/					
phpMyAdmin	gi loca @Dat	alhost tabases 🛛 👼 SQL 🛛 🏶 tabases	Status 👔 Variables		
 information_schema (37) mydatabase mysql (24) 		Database 🔺	50		
 performance_schema (1/) phpmyadmin (8) test webauth (1) 		information_schema	50		
		mydatabase mysol	50		
Please select a database		performance_schema	50		
		phpmyadmin	50		
		test	50		
		webauth	50		
	t	Check All / Uncheck A	II With selected: 🛛 🔀		

Рисунок 30 - Создание таблицы

Затем появится текстовое поле «*Create new table in database*», где вы вводите имя таблицы и количество полей и нажимаете «*Go*».

Пусть необходимо создать таблицу с именем *MyTable*, которая содержит пять полей:

😭 Server: localhost 🕨 👜 Database: MyDataBase 🕨 🏢 Table: MyTable				
Field	Туре 🕜	Length/Values ¹		
id	INT			
FirstName	CHAR	255		
LastName	CHAR	255		
Phone	INT			
BirthDate	DATE			

Рисунок 31 – Конструктор таблицы

Для поля «*id*» устанавливаем в столбце *Index* значение *PRIMARY KEY*.

Для того чтобы значения поля *id* автоматически наращивалось (поле типа Счетчик), можно в столбце конструктора *AU*-*TO_INCREMENT* (*A_I*) поставить галочку.

Примечание: Название полей не должно содержать пробелов. Закончив создание структуры таблицы, нажмите Save. Для заполнения данными таблицы выберите на панели Insert.

	Struc	ture 🛛 🖓 SQL	🔎 Search	n F ilnsert	Export	
Заполните данными в столбце, вводя их в столбец Value.						
Field	Туре	Functio	n N	Vull		Value
id	int(11)		-	112		
FirstName	char(255)		-	Серге	й	
LastName	char(255)		-	Петро	в	
Phone	int(11)		-	65478	;	
BirthDate	date		-	1986-	12-04 📃	
						Go

Рисунок 32 – Окно для ввода данных в таблицу

Нажмите *Go*. По умолчанию после вставки записи, вы перейдете к предыдущей странице конструктора, для того, чтобы после вставки записи, загружалась страница для ввода новой записи, измените настройки внизу страницы.

Insert as new row 🖵	and then	Insert another new row
1		GoReset

Рисунок 33 – Окно настроек ввода данных в таблицу

Рассмотрим, как вставлять данные в базу данных непосредственно из *PHP*-скриптов.

Для этого используется *SQL* запрос со следующим синтаксисом: *INSERT INTO* ИмяТаблицы (столбец1, столбец2, ...) *VALUES*(значение1, значение2, ...)

Пример запроса, который вставит запись сотрудника имени Василий Уткин, номер телефона 12345 и дата рождения - 1964-04-20.

Oператор SQL может выглядеть так: \$strSQL = "INSERT INTO MyTable(id,FirstName,LastName,Phone,BirthDate) VALUES(246,'Bacuлuŭ','Уткин','12345 ','1964-04-20')";

Пример следующего программного кода *PHP* реализует добавление записи в таблицу базы данных.

```
<?php
//подключение к MySQL
mysql_connect("localhost", "root", "") or die(mysql_error());
// подключение к базе данных с именем MyDataBase
mysql_select_db("MyDataBase") or die(mysql_error());
//формирование строки запроса
$strSQL = "INSERT INTO MyTable(id,FirstName,LastName,Phone,BirthDate)
VALUES(246,'Bacилий','Уткин','12345 ','1964-04-20')";
// команда - выполнить запрос
mysql_query($strSQL) or die(mysql_error());
// закрытие сеанса работы с сервером MySQL
mysql_close ();
echo "Запись успешно добавлена.";
?>
```

Рисунок 34 – Пример содержимого файла РНР

Запустите страницу в браузере, затем перейдите на страницу *phpMyAdmin* и убедитесь (вкладка *Browse*), что в таблице добавлена данная запись.

Осуществим вывод фамилии и имени таблицы на страницу. Для этого используется цикл *while*. Если в результате выполнения запроса была выбрана хотя бы одна строка, то логическое значение переменной *\$row* будет *TRUE*. На первом шаге цикла значение условного выражения в скобках будет *TRUE*, поэтому начнет выполняться блок операторов. Данные массива отображаются на странице, затем делается попытка получить следующую строку.

```
<?php
mysql_connect("localhost", "root", "") or die(mysql_error());
mysql_select_db("MyDataBase") or die(mysql_error());
$sqlAll = "select* from MyTable";
$resultsAll = mysql_query($sqlAll);
$row = mysql_fetch_array($resultsAll);
while ($row) {
    echo $row["id"]." -- ".$row["FirstName"]." -- ".$row["LastName"]."<br/>br />";
$row = mysql_fetch_array($resultsAll);
}
mysql_close ();
```

Рисунок 35 – Содержимое файла с кодом РНР

Пример создания страницы *html* для ввода данных в таблицу и вывода из таблицы в странице *pнp*.

```
<html>
<head>
<title>Форма для добавления</title></head>
<body>
<h1>Введите данные:</h1>
<form action="lab6.php" method="POST">
Taбельный номер: <input type="text" name="id" /><br />
Имя: <input type="text" name="Fname" /><br />
Фамилия: <input type="text" name="Lname" /><br />
Номер телефона: <input type="text" name="Phone" /><br />
Дата рождения в формате " год - месяц -день: <input type="text"
name="BDate" /><br />
<input type="submit" value="Coxpaнumь" />
</form>
</body> </html>
```

<u>Структура PHP страницы:</u> <?php mysql_connect("localhost", "root", "") or die(mysql_error()); mysql_select_db("MyDataBase") or die(mysql_error());

\$strSQL = "INSERT INTO
MyTable(id,FirstName,LastName,Phone,BirthDate)
VALUES('''.\$_POST["id"].''', '''.\$_POST["Fname"].''',
'''.\$_POST["Lname"].''','''.\$_POST["Phone"].''','''.

\$_POST["BDate"]."')";
mysql_query(\$strSQL) or die(mysql_error());

```
$sqlAll = "select* from MyTable";
	$resultsAll = mysql_query($sqlAll);
	$row = mysql_fetch_array($resultsAll);
		while ($row) {
		echo 		$row["id"]." -- ".$row["FirstName"]." --
".$row["LastName"]."<br />";
```

```
$row = mysql_fetch_array($resultsAll);
}
mysql_close ();
```

?>

Рассмотрим пример с использованием дополнительных элементов управления, таких как: поле со списком, переключатель. Рассмотрим пример программного кода, который в зависимости от выбранного типа товара и указания, есть ли дисконтная карта, рассчитывает стоимость покупки.

Код *НТМL*- формы:

```
<html>
<head>
<title> Pacyer покупки</title>
</head>
<body>
<form action ="pr.php" metod="GET">
<select name="Tovar">
<option > Блок питания
<option > Карта памяти
<option > Манипулятор "мышь"
</select><br><br>
Наличие дисконтной карты: <input type="checkbox" name ="DK" ><br/>br/>>br/>
<input type ="text" name = "K">
<input type="submit" value="Paccчитать">
</form>
</body>
</html>
```

Рисунок 36 – Пример структуры файла с расширением НТМL

Пример скрипта РНР.

```
<?
switch($ GET['Tovar'][0])
{
case "B":
$sum = $ GET['K']*20;
case "K":
$sum = $ GET['K']*15;
case "M":
$sum = $ GET['K']*10;
if ($_GET['DK']=="on")
{$sum=$sum-$sum*0.05;
echo "Вы получаете скидку в размере 5 %";
}
2>
<html>
<head>
</head>
<body>
 Сумма покупки: <? echo $sum;?> y.e. 
</body>
</html>
```

Рисунок 37 – Пример структуры файла с расширением РНР

Использование сессии в сетевом приложении. При посещении сайта вы выполняете различные действия. Переходите с одной страницы на другую. Возможно, заполняете форму или покупаете что-то. Возникает необходимость в сохранении значении переменных, состояния объектов в момент просмотра пользователем страницы или в момент перемещения пользователя по страницам сайта. РНР-сессии дают возможность работать с информацией о пользовательской сессии. Вы можете создавать приложения, которые идентифицируют и собирают информацию о пользователях. Скажем, 50 пользователей просматривают страницы одного сайта, например, интернет магазина. Информацию о том, что у каждого посетителя в корзине, лучше всего сохранить в сессии. Чтобы идентифицировать пользователей, сервер использует уникальные пользовательские идентификаторы /user ID, которые хранятся в cookie . Следовательно, сессии часто требуют поддержки cookie в браузерах пользователей.

Рассмотрим пример, в котором будем сохранять имя пользователя, вошедшего на сайт, и будем осуществлять подсчет количества посещений страницы, которую может просматривать только пользователь, который ввел правильный логин и пароль. Сессии могут начинаться разными способами. Мы рассмотрим вариант, когда сессия начинается с сохранения значения. Сессия заканчивается, если пользователь не запрашивает страниц в течение какого-то времени (стандартное значение – 20 минут) или пользователь закрывает окно браузера. В любой момент можно принудительно закончить сессию вашем скрипте, используя функцию session destroy();

Пример использования сессий в сетевом приложении. Следующая *html* страница содержит текстовые поля для аутентификации пользователей.

```
<html>
<head>
<title> Аутентификация</title>
</head>
<body>
<form action="login2.php" method="post">
Логин:<input type="text" name="user_name"><br>
Пароль:<input type="text" name="user_name"><br>
сinput type="password" name="user_pass"><br>
<input type="submit" name="Submit">
</form>
</body>
</html>
```

Рисунок 38 – Пример структуры файла формы РНР

Следующий программный код содержит сессионную переменную с именем *count*, значение которой увеличивается на единицу при новом посещении страницы пользователем:

```
<?
    session start();
    $count++;
    session_register("count");
    // проверяем данные на правильность
    if(($_POST['user_name']=="asd")&&($_POST['user_pass']=="123"))
      $logged user = $ POST['user name'];
      // запоминаем имя пользователя
      session register("logged user");
      // и переправляем его на <секретную> страницу...
      echo "<a href='dokument.php'>Ссылка на защищённый файл</a>";
      exit:
 2>
<html><body>
Вы ввели неверный пароль!
</body></html>
```

Рисунок 39 – Пример структуры файла РНР

Пример страницы *document.php*, которая открывается только для зарегистрированного пользователя.

```
<?php
session_start();
echo " Приветсвуем " . $_SESSION['logged_user'].
 "! Вы зарегистрированный пользователь! " ."</br>;
echo "Вы посетили данную страницу ". $_SESSION['count']." раз.";
$_SESSION['count']++;
?>
```

Рисунок 40 – Пример файла РНР с приветствием зарегистрированного пользователя

Регистрация новой сессионной переменной происходит при вызове функции *session_register()*, которая принимает одно или несколько текстовых переменных, значения которых – название будущей сессионной переменной.

Чтобы обратиться к сессионной переменной, необходимо использовать суперглобальный массив *\$_SESSION*.

Теперь решим задачу подсчета персональных посещений пользователей с помощью *cookie*.

Использование Cookie в сетевом приложении. Cookie это небольшой текстовый файл, в котором сайт может хранить различную информацию. Cookie сохраняются на жёстком диске пользователей, а не на сервере. Cookie записывается сервером на компьютер посетителя. В дальнейшем эту информацию можно извлечь, причем сделать это может только тот сервер, который записал ее туда. Информация ограничена размером четыре килобайта. Есть ограничения по времени существования cookie, по их количеству и т.д. Срок годности большинства Cookie истекает после предопределённого периода времени (они самоудаляются), и этот период может варьироваться от одной минуты до нескольких лет. Но пользователь и сам может удалять любые Cookie на своём компьютере.

Многие сайты очень активно используют ее для сбора информации о своих клиентах, их предпочтениях, настройках и т.д. Для примера представьте себе форум, в котором нужно обязательно указывать имя, адрес или другую информацию. Обычно все это приходится каждый раз вводить заново, а с помощью *cookie* все данные можно запомнить на компьютере клиента и при следующем посещении форума их извлечь и воспроизвести в форме. Еще проще – один раз запомнить имя посетителя, и каждый раз приветствовать его по имени. Самый простой способ установить *cookie* с помощью *PHP* – это использовать специальный оператор: *setcookie('name', '12345')*;

Теперь, пока не будет закрыто окно браузера пользователем, переменную *\$name* со значением равным числу 12345 можно считать с помощью другого оператора: *isset (\$ COOKIE["name"]);*

Часто делают сразу проверку, установлена или нет соответствующая переменная, и на основе полученных результатов принимают решение, что делать дальше:

If (isset (\$_COOKIE["name"]);)) { ... установлена ... } else { ... нет ... }

Надо помнить, что установленный таким образом *cookie* живет только до тех пор, пока не закрыто окно браузера, его породившее.

Как только пользователь закрывает окно, установленная переменная уже не доступна. Это так называемый сессионный *cookie*. Впрочем, совсем не сложно продлить срок жизни *cookie* до нужного времени. Делается это с помощью третьего параметра oператора *setcookie*, который указывает дату истечения срока действия *cookie* либо срок его действия, если число меньше, чем дата установки:

\$y = mktime(0,0,0,1,1,2012);
setcookie(\$_COOKIE["name"]);, 'bret', \$y);

Перед тем, как задать третий параметр, его нужно сформировать специальной функцией. Дело в том, что в *PHP* этот параметр указывает на количество секунд, прошедшее после 1 января 1970 года. Функция *mktime()* именно для этого и предназначена. Ей нужно указать час, минуту, секунду, месяц, день и год в такой же последовательности. На выходе получим нужное значение в секундах, прошедших с 1.01.1970. В примере *cookie* будет жизнеспособен (а значит, и поддастся считыванию) до 1 января 2012 года.

Переустановить или удалить *cookie* очень просто – достаточно указать либо новое значение, либо не указать его вовсе. И под конец одно существенное ограничение, работа с *cookie* должна вестись до какого-либо вывода на экран (в браузер пользователя) любого текста, в том числе и тегов *HTML*.

```
То есть вот так уже не сработает:

<html>

... работаем с cookie ...

Только так:

... работаем с cookie ...

<html>

... любые другие теги и текст ...

<?

$x = mktime(0,0,0,1,1,2012);

if (isset ($kol)) { setcookie('kol', '0', $x); }

else { $_COOKIE["kol"]++; setcookie('kol', $_COOKIE["kol"], $x); }

?>
```

Рассмотрим пример, если *cookie* с именем *kol* установлен, его значение считывается и увеличивается на единицу. Если нет – записывается в *cookie* ноль. В дальнейшем он будет считан и учтен. В любом месте страницы теперь можно вывести результаты: *echo* "<*p*> Персональный счетчик — ".\$ COOKIE["kol"]." </*p*>";

Следующий программный код демонстрирует пример использования cookie в сетевом приложении.

```
Рисунок 41 – Фрагмент кода из файла php
```

По умолчанию *cookie* хранится, пока не закроется браузер, но это легко изменить, добавив другой параметр - установив срок годности:

```
<?php
// Установка cookie
setcookie("Name", "C. Wing", time()+3600);
setcookie("Interests", "plane sporting", time()+3600);
?>
```

"Time()+3600" означает, что *cookie* должна действовать 3600 секунд (60 минут) с данного момента. Для получения значения *cookie* используется \$_*COOKIE*. Например, если вам необходима информация из предыдущего примера, то:

<?php // Запросить значение cookie \$strName = \$_COOKIE["Name"]; \$strInterest = \$_COOKIE["Interests"]; // Вывести на страницу echo "". strName. "" echo "Bac интересует." strInterest. "" ?>

При запуске этого кода браузер получает *cookie* с именем BG, значением «green» сроком хранения час после установки *cookie*. <?php

\$color = "green";
setcookie("BG", \$color, time()+3600),
?>

Пример отображения цвета фона страницы, сохраненного в *cookie*.

```
<html>
<hody_bacolor=<`
```

```
<body bgcolor= <?=$_COOKIE['BG'] ?> >
<h3>Hello Cookie!</h3>
</body> </html>
```

Сначала необходимо запустить скрипт, чтобы он установил *cookie*, а затем уже просмотреть результат.

Вставить картинку в *PHP* скрипт можно несколькими способами:

Способ № 1 – Вставка картинки в *РНР* с помощью функции вывода *echo*.

```
<?php

// ... здесь какой-то код PHP ...

echo "<img src='http://lphp.ru/images/dalmation.gif'>";

// ... здесь какой-то код PHP ...

?>

Способ № 2 – Вывод картинки через переменную

<?php

// ... здесь какой-то код PHP ...

// добавляем картинку в переменную $a, после чего её

// можно вывести в любом месте php кода страницы

$a = "<img src='http://lphp.ru/images/dalmation.gif'>";

// ... здесь какой-то код PHP ... ?>
```

Функциональная схема и реализация сетевого приложения. Рассмотреть каждую из страниц *web*-приложения на предмет выполняемых функций. Например: проверка вводимых данных, всплывающее меню, изменение цвета элементов управления, смена картинок, структурирование информации на странице, вызов серверных скриптов и т.д. Для каждой из рассмотренных функций привести программный код с пояснением. Программный код не может занимать более трети страницы, если код превышает, то его необходимо разбить на части словесными комментариями с описанием «что делает данный код». Пример схемы здания, для которого необходимо спроектировать компьютерную сеть



ПРИЛОЖЕНИЕ Б

Тип	Макс.	Начал	ьный	Промеж	уточный	Коне	чный	Задержка
сегмента	длина,	сегм	ент	сеги	иент	сеги	лент	на метр
Ethernet	М							длины
		t0	tm	t0	tm	t0	tm	tl
10BASE5	500	11,8	55,0	46,5	89,8	169,5	212,8	0,0866
10BASE2	185	11,8	30,8	46,5	65,5	169,5	188,5	0,1026
10BASE-T	100	15,3	26,6	42,0	53,3	165,0	176,3	0,1130
10BASE-FL	2000	12,3	212,3	33,5	233,5	156,5	356,5	0,1000
FOIRL	1000	7,8	107,8	29,0	129,0	152,0	252,0	0,1000
<i>AUI</i> (>2 м)	2+48=50	0	5,1	0	5,1	0	5,1	0,1026

Таблица 1 – Справочные данные для расчета конфигурации сети *Ethernet*

Таблица 2 – Справочные данные для расчета сокращения межкадрового интервала для конфигурации сети *Ethernet*

Тип сегмента	Начальный сегмент	Промежуточный сегмент
10BASE5	16	11
10BASE2	16	11
10BASE-T	10,5	8
10BASE-FL	10,5	8

ПРИЛОЖЕНИЕ В

Тип репитера	Витая пара	Оптоволоконный	T4 и FX	ТХ и FХ
(концентратора)		кабель		
Без репитера	100	412	-	-
(два абонента)				
Один репитер	200	272	231	260,8
класса I				
Один репитер	200	320	-	308,8
класса II				
Два репитера	205	228	-	216,2
класса II				

Таблица 3 – Справочные данные для расчета конфигурации сети FastEthernet

Таблица 4 – Справочные данные для расчета сокращения межкадрового интервала для конфигурации сети *FastEthernet*

Тип сегмента	Задержка на метр	Максимальная задержка
	(битовый интервал)	(битовый интервал)
Два абонента <i>ТХ/FX</i>	-	100
Два абонента <i>Т4</i>	-	138
Один абонент Т4	-	127
и один <i>ТХ/FX</i>		
Сегмент на кабеле катего-	1,14	114 (100 м)
рии 3		
Сегмент на кабеле катего-	1,14	114 (100 м)
рии 4		
Сегмент на кабеле катего-	1,112	111,2 (100 м)
рии 5		
Экранированная витая пара	1,112	111,2 (100 м)
Оптоволоконный кабель	1,0	412 (412 м)
Репитер (концентратор)	-	140
класса I		
Репитер (концентратор)	-	92
класса II с портами ТХ/FХ		
Репитер (концентратор)	-	67
класса II с портами <i>Т4</i>		