

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
ВЫСШЕГО ОБРАЗОВАНИЯ  
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

# ОСНОВЫ СЕТЕВЫХ ТЕХНОЛОГИЙ

*Методические рекомендации к лабораторным работам  
для студентов специальности  
1-28 01 02 «Электронный маркетинг»  
дневной и заочной форм обучения*



УДК 004.383.2  
ББК 32.973.202-04  
075

Рекомендовано к изданию  
учебно-методическим отделом  
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»  
«17» марта 2020 г., протокол № 8

Составители: ст. преподаватель В. Т. Садовский;  
ассистент Н. П. Скрылев

Рецензент Ю. С. Романович

Методические рекомендации предназначены к лабораторным работам для студентов специальности 1-28 01 02 «Электронный маркетинг» дневной и заочной форм обучения.

Учебно-методическое издание

## ОСНОВЫ СЕТЕВЫХ ТЕХНОЛОГИЙ

Ответственный за выпуск	А. И. Якимов
Корректор	А. А. Подошевка
Компьютерная верстка	Е. В. Ковалевская

Подписано в печать 17.06.2020 .Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Печать трафаретная. Усл. печ. л. 2,79 .Уч.-изд. л. 3,0 . Тираж 31 экз. Заказ № 279.

Издатель и полиграфическое исполнение:  
Межгосударственное образовательное учреждение высшего образования  
«Белорусско-Российский университет».  
Свидетельство о государственной регистрации издателя,  
изготовителя, распространителя печатных изданий  
№ 1/156 от 07.03.2019.  
Пр-т Мира, 43, 212022, Могилев.

© Белорусско-Российский  
университет, 2020

## Содержание

Введение.....	4
1 Лабораторная работа № 1. Топология компьютерной сети.....	6
2 Лабораторная работа № 2. Моделирование различных топологий с использованием Packet Tracer.....	11
3 Лабораторная работа № 3. Изучение протоколов доступа к среде передачи LAN .....	13
4 Лабораторная работа № 4. Базовые настройки коммутатора CISCO .....	17
5 Лабораторная работа № 5. Изучение алгоритма STA коммутаторов CISCO .....	19
6 Лабораторная работа № 6. Изучение виртуальных локальных сетей (VLAN) .....	22
7 Лабораторная работа № 7. Изучение правил адресации сетевого уровня .....	25
8 Лабораторная работа № 8. Изучение принципов статической маршрутизации IP-сетей .....	27
9 Лабораторная работа № 9. Изучение принципов динамической маршрутизации IP-сетей .....	32
10 Лабораторная работа № 10. Изучение сетевых утилит командной строки Windows .....	36
11 Лабораторная работа № 11. Изучение текстовых протоколов высших уровней модели OSI .....	41
12 Лабораторная работа № 12. Изучение протоколов электронной почты .....	45
Список литературы .....	48

## Введение

Целью изучения учебной дисциплины «Основы сетевых технологий», является подготовка студентов в области современных сетевых технологий по специальности «Электронный маркетинг».

В инфраструктуре электронного бизнеса вообще и электронного маркетинга в частности системам телекоммуникаций и вычислительным сетям сегодня отводится ключевая роль. Сетевые технологии позволяют получать и передавать различного рода информацию для успешного ведения бизнеса, а именно: нормативную информацию, информацию о конъюнктуре рынка, создаваемую конкурентами, потребителями, поставщиками, о финансовой деятельности (в государственные органы, кредиторам, инвесторам), маркетинговую информацию (потребителям, партнерам, поставщикам) и т. д. Большой объем информации накапливается, обрабатывается и циркулирует внутри предприятий и служит основой для прогнозирования развития экономической системы, корректировки целей в хозяйственной деятельности.

В связи с этим подготовка студентов по специальности 1-28 01 02 «Электронный маркетинг» в области современных сетевых технологий является актуальной.

Задачи изучения учебной дисциплины получение знаний по основам компьютерных сетей:

- приобретение базовых теоретических знаний в области современных сетевых технологий;

- приобретение навыков практической работы в области современных сетевых технологий, включая их разработку, эксплуатацию и администрирование;

- изучение функциональных возможностей, основных методов конфигурирования сетевых устройств и программного обеспечения, таких как коммутаторы локальных вычислительных сетей, маршрутизаторы, серверные операционные системы и рабочие станции.

В результате освоения учебной дисциплины обучающийся узнает:

- основные концепции построения локальных и глобальных сетей; методы объединения компьютеров и устройств в сети;

- основные протоколы, методы организации, способы объединения компьютеров в сети;

- виды топологий сети и основные реализуемые алгоритмы взаимодействия узлов;

- способы передачи, методы кодирования и защиты данных.

Для лучшего освоения дисциплины:

- изучить основные теоретические положения, сделав необходимые выписки в конспект;

- выполнить в полном объеме лабораторные работы и практические задания к ним, согласно данных методических рекомендациях;
- оформить отчет.

### ***Содержание отчета***

1 Титульный лист.

2 Цели и задачи лабораторной работы.

3 Перечень использованного оборудования и программного обеспечения.

4 Постановка задачи.

4 Схема топологии, алгоритм, исходный код задачи лабораторной работы.

5 Методика выполнения работы.

6 Анализ результатов и выводы по работе.

Отчет оформляют вместе с рисунками Screen Shot's в письменном виде, либо с помощью графического редактора.

# 1 Лабораторная работа № 1. Топология компьютерной сети

**Цель работы:** изучение топологий вычислительных сетей.

## *Основные теоретические положения*

**Сетевая топология** – способ описания конфигурации сети: схема расположения и соединения сетевых устройств или/и схема прохождения электрических сигналов, или/и описание направления потоков информации.

Сетевая топология может быть:

- **физической** – описывает реальное расположение и связи между узлами сети – способ физического соединения компьютеров с помощью среды передачи, например, участками кабеля;
- **логической** – описывает пути и направление передачи потоков данных между сетевыми устройствами в рамках физической топологии;
- **информационной** – описывает направление потоков информации, передаваемых по сети;
- **управление обменом** – это принцип передачи права на пользование сетью.

Выделяют три базовых топологии и ряд дополнительных производных типовых сетевых топологий, объединяющих компьютеры в единую сеть (таблица 1.1).

Таблица 1.1 – Типы сетевых топологий

Базовый тип сетевых топологий	Производный (дополнительный) тип сетевых топологий	
Шина	Дерево	Двойное кольцо
Звезда	Ячеистая топология	Fat Tree
Кольцо	Полносвязная	Решётка

Дополнительные способы являются комбинациями базовых.

### *Топология «шина»*

В топологии «шина» используется один кабель, именуемый сегментом, вдоль которого подключены все компьютеры сети (рисунок 1.1).

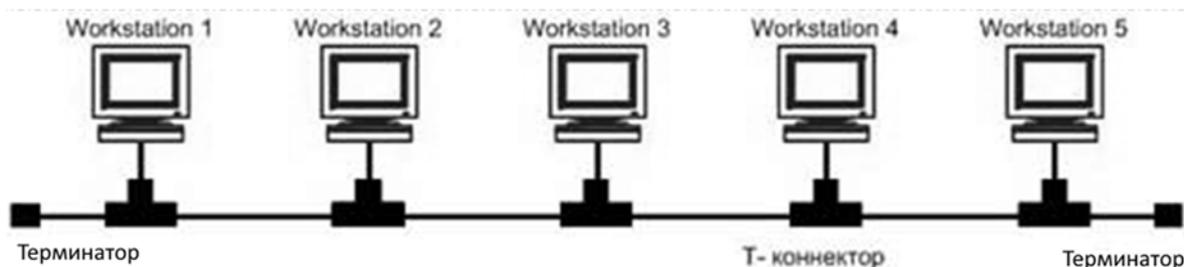
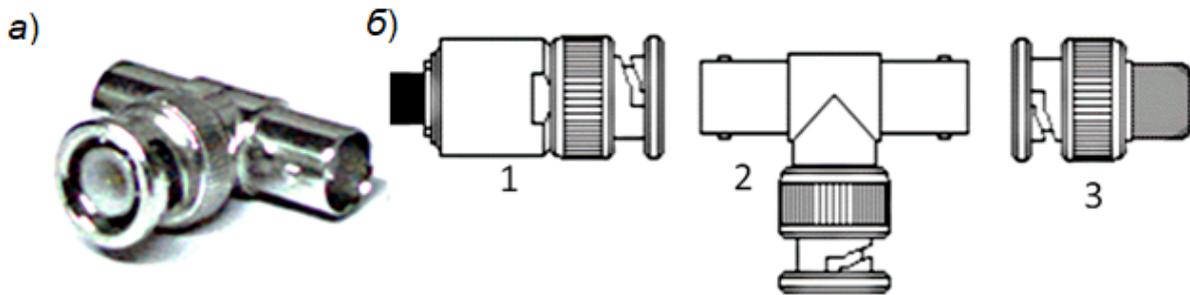


Рисунок 1.1 – Сетевая топология «шина»

Топология «шина» применяется в локальных сетях с архитектурой Ethernet (классы 10Base-5 и 10Base-2 для толстого и тонкого коаксиального кабеля соответственно), режим обмена полудуплекс (half duplex). В настоящее время толстый коаксиальный кабель для Ethernet не применяется. Что касается Ethernet 10Base-2, применяется тонкий коаксиальный кабель (0,2 дюйма).

Каждый компьютер подключается к коаксиальному кабелю с помощью Т-разъема (Т-коннектор), т. е. Т-коннектор включается в сетевую карту компьютера, а к свободным концам коннектора подключаются куски кабеля от предыдущего и последующего компьютера. К первому компьютеру и к последнему на свободные концы Т-коннектора устанавливаются «терминаторы» (рисунок 1.2).



1 – соединитель BNC; 2 – Т-коннектор; 3 – терминатор

Рисунок 1.2 – Общий вид Т-коннектора (а) и элементы BNC разъема (б)

Терминаторы используются для гашения сигналов, которые достигают концов канала передачи данных. Информация поступает на все узлы одновременно, но принимается только тем узлом, у которого MAC-адрес совпадает с адресом в информационном кадре. Описанная выше топология относится к физической и логической топологии «шина».

### ***Топология «звезда»***

Звезда – это топология сети с явно выделенным центром, к которому подключаются все остальные абоненты (рисунок 1.3)

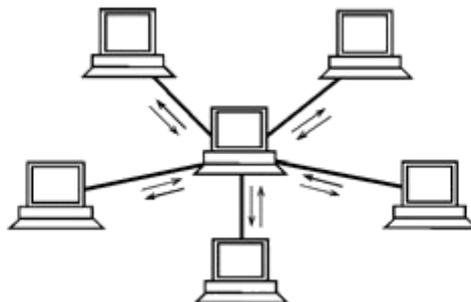


Рисунок 1.3 – Топология «звезда» (активная звезда)

Обмен информацией идет исключительно через центральный узел: компьютер, коммутатор, маршрутизатор. Выход из строя периферийного компьютера или его сетевого оборудования, линии связи никак не отражается на функционировании оставшейся части сети, зато любой отказ центрального компьютера делает сеть полностью неработоспособной.

Для соединения периферийного компьютера с центральным используются две линии связи (в одном кабеле), каждая из которых передает информацию в одном направлении, т. е. на каждой линии связи имеется только один приемник и один передатчик. Это так называемая дуплексная передача точка-точка.

Серьезный недостаток топологии «звезда» состоит в жестком ограничении количества абонентов, обычно не более 848 периферийных абонентов. В звезде допустимо подключение еще одного центрального абонента.

Звезда, показанная на рисунок 1.3, носит название активной или истинной звезды. Существует также топология, называемая «пассивная звезда», которая внешне похожа на звезду (рисунок 1.4), но логика работы у неё другая.

В центре сети с данной топологией помещается специальное устройство (концентратор) – хаб (hub), – которое выполняет ту же функцию, что и повторитель, т. е. восстанавливает входящие сигналы и пересылает их на все линии связи, всем компьютерам одновременно (см. рисунки 1.4 и 1.5).

Информация поступает на все рабочие станции одновременно, но обрабатывается только узлом назначения. Логическая топология пассивной звезды является **логической шиной**.

Данная топология применяется в локальных сетях с 10Base-T Ethernet.

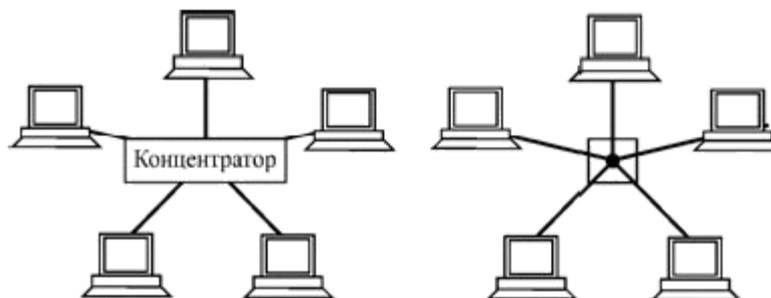


Рисунок 1.4 – Топология «пассивная звезда» и ее эквивалентная схема

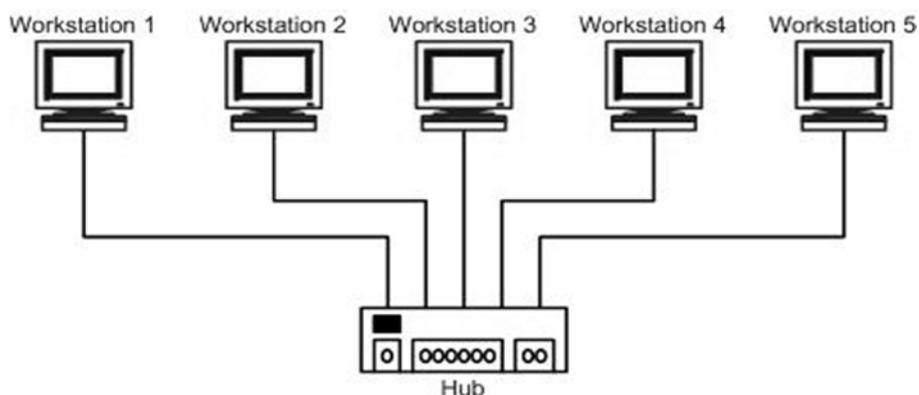


Рисунок 1.5 – Подключение ПК к концентратору («звезда», «шина»)

### **Топология «кольцо»**

Кольцо – это топология, в которой каждый компьютер соединен линиями связи с двумя другими: от одного он получает информацию, а другому передает. На каждой линии связи, как и в случае звезды, работает только один передатчик и один приемник (связь типа «точка – точка») (рисунок 1.6).

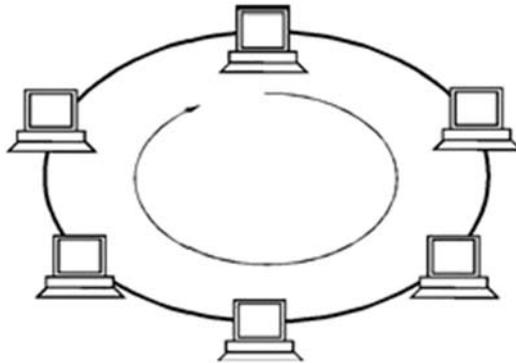


Рисунок 1.6 – Сетевая топология «кольцо».

Важная особенность кольца состоит в том, что каждый компьютер ретранслирует (восстанавливает, усиливает) проходящий к нему сигнал, т. е. выступает в роли репитера. Поэтому предельная длина кольца может достигать  $N \cdot L_{np}$  (где  $N$  – количество компьютеров в кольце;  $L_{np}$  – предельная длина кабеля, ограниченная затуханием). Кольцо в этом отношении существенно превосходит любые другие топологии. В топологии кольцо право на передачу (или, как еще говорят, на захват сети) переходит последовательно от одного компьютера к следующему по кругу.

Подключение новых абонентов в кольцо требует обязательной остановки работы всей сети на время подключения. Максимальное количество абонентов в кольце может быть довольно велико (до тысячи и больше). Кольцевая топология обычно обладает высокой устойчивостью к перегрузкам, обеспечивает уверенную работу с большими потоками передаваемой по сети информации.

Выход из строя хотя бы одного из компьютеров (или же его сетевого оборудования, линии связи) нарушает работу сети в целом. Это существенный недостаток кольца.

Из трех рассмотренных топологий кольцо наиболее уязвимо к повреждениям кабеля, поэтому в случае топологии «кольцо» обычно предусматривают прокладку двух (или более) параллельных линий связи, одна из которых может находиться в резерве (рисунок 1.7), или же предназначается для увеличения (в идеале – вдвое) скорости передачи информации.

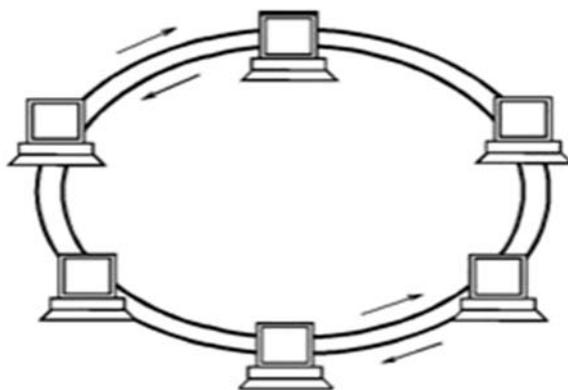


Рисунок 1.7 – Сеть с двумя кольцами

### ***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Зарисовать расположение компьютеров в классе и определить тип топологии локальной сети.

3 Описать преимущества и недостатки каждого из базовых типов топологии.

4 Оформить отчет.

### ***Контрольные вопросы***

1 Что означает топология, Абонент, Сервер, Клиент?

2 Назовите базовые и дополнительные типы топологий.

3 Какие типы кабеля применяются при шинной топологии?

4 Преимущества и недостатки шинной топологии.

5 Топология «звезда». Преимущества и недостатки данной топологии.

6 Активная и пассивная топология «звезда».

7 Предельная длина сети с применением топологии «звезда».

8 Топология «кольцо». Преимущества и недостатки данной топологии.

## 2 Лабораторная работа № 2. Моделирование различных топологий с использованием Packet Tracer

**Цель работы:** изучение принципов построения компьютерных сетей базовой топологии базовым набором аппаратных средств, с применением программного обеспечения – симулятора сетей Packet Tracer.

### *Основные теоретические положения*

Для построения моделей сетей используют симуляторы.

Программный продукт **Packet Tracer (PT)** – является симулятором сети передачи данных, разработанный фирмой Cisco Systems. *Packet Tracer* позволяет имитировать работу различных сетевых устройств: маршрутизаторов, коммутаторов, точек беспроводного доступа, персональных компьютеров, сетевых принтеров, IP-телефонов и т. д. С помощью *Packet Tracer* можно создавать работоспособные модели сети любой сложности.

Компьютерная сеть состоит из компонентов аппаратного и программного обеспечения, которые должны работать согласованно.

Аппаратное обеспечение можно разделить на три основные группы.

#### 1 Абонентские системы:

- компьютеры (сервера, рабочие станции, или клиенты, терминалы);
- принтеры;
- сканеры и др.

#### 2 Сетевое оборудование:

- сетевые адаптеры;
- концентраторы (хабы);
- мосты;
- коммутаторы;
- маршрутизаторы и др.

#### 3 Коммуникационные каналы:

- кабели;
- разъемы;
- устройства передачи и приема данных в беспроводных технологиях;
- устройства передачи и приема данных по стандартным каналам связи (аналоговые и цифровые системы телефонной связи).

### *Порядок выполнения работы*

1 Изучить основные теоретические положения.

2 Необходимо смоделировать локальную сеть (LAN) с использованием коммутатора в качестве центрального коммутирующего сетевого устройства и концентратора, как устройства для расширения сети (рисунок 2.1).

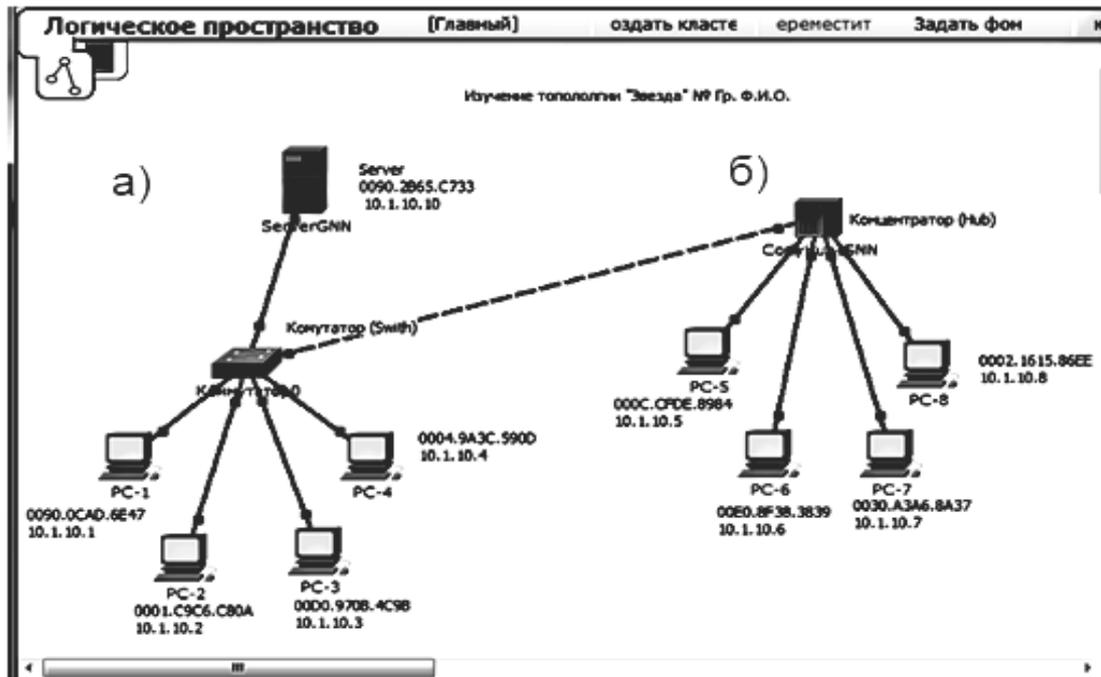


Рисунок 2.1 – Модель сети с использованием коммутатора и концентратора

3 В качестве вариантов следует выполнять следующее правило: при вводе IP-адресов для компьютеров PC-1... PC-N – 192.GGG.NN.1...192.GGG.NN.N, для сервера вводим IP-адрес: 192.GGG.NN.10, где GGG – номер группы (только цифры); NN – порядковый номер по журналу группы; маска 255.255.255.0.

4 Запустить программу Packet Tracer с созданной моделью сети в режиме симуляции, отправить пакеты с компьютеров на сервер. Сравнить трассы прохождения, объяснить причину.

5 Оформить отчет.

### **Контрольные вопросы**

- 1 Какие сетевые устройства может имитировать Packet Tracer (PT)?
- 2 Назовите три основные группы сетевого аппаратного обеспечения.
- 3 Что относится к абонентским системам?
- 4 Какое оборудование является сетевым?
- 5 Какие типы оборудования входят в группу коммуникационных каналов?

### 3 Лабораторная работа № 3. Изучение протоколов доступа к среде передачи LAN

**Цель работы:** изучение протоколов доступа к среде передачи LAN. Метод случайного доступа, детерминированный доступ. Канальный уровень, подуровни LLC и MAC, MAC- адрес. Формат кадра Ethernet.

#### *Основные теоретические положения*

По **порядку доступа** среда передачи может быть **индивидуальной** или **разделяемой**.

Разделяемой средой передачи (shared media) называется линия, которая используется попеременно несколькими (более чем двумя) устройствами, подключенными к ней. Индивидуальной средой передачи является линия связи, к каждому окончанию которой подключено только одно устройство.

Примерами связи компьютеров посредством разделяемой среды являются системы с топологией «общая шина» и «кольцо», а также сети Wi-Fi. Примером связи компьютеров с индивидуальной средой, является топология сетей LAN-«звезда», а также соединение двух компьютеров «точка – точка».

Существуют два основных метода доступа к разделяемой физической среде:

- 1) метод случайного доступа;
- 2) детерминированный доступ.

**Метод случайного доступа** является одним из основных методов захвата разделяемой среды. Он основан на том, что узел, у которого есть кадр для передачи, пытается его отправить без согласования использования разделяемой среды с другими узлами сети.

**Детерминированный доступ** – это другой подход доступа к разделяемой среде. В данном случае максимальное время ожидания доступа к среде всегда известно. Алгоритмы детерминированного доступа используют два механизма – *передачу токена и опрос*.

Для случайного доступа применяются два типа алгоритма **CSMA/CA** и **CSMA/CD**.

**CSMA/CA (Carrier Sense Multiple Access/Collision Avoidance)** – множественный доступ с прослушиванием несущей и избеганием коллизий. Узел, готовый послать кадр, прослушивает линию. При отсутствии несущей он посылает короткий сигнал запроса на передачу (RTS) и определенное время ожидает ответа (CTS) от адресата назначения. При отсутствии ответа (подразумевается возможность коллизии) попытка передачи откладывается, при получении ответа в линию посылается кадр. Метод не позволяет полностью избежать коллизий, но они обрабатываются на вышестоящих уровнях протокола.

**CSMA/CD (Carrier Sense Multiple Access/Collision Detect)** – множественный доступ с прослушиванием несущей и обнаружением коллизий. Узел прослушивает линию. При отсутствии несущей он начинает передачу кадра, одновременно контролируя состояние линии. При обнаружении коллизии передача прекращается, и повторная попытка откладывается на случайное время. Коллизии – нормальное, хотя и не очень частое, явление для CSMA/CD. Их частота связана с количеством подключенных узлов.

Упрощенный алгоритм метода CSMA/CD представлен на рисунке 3.1.

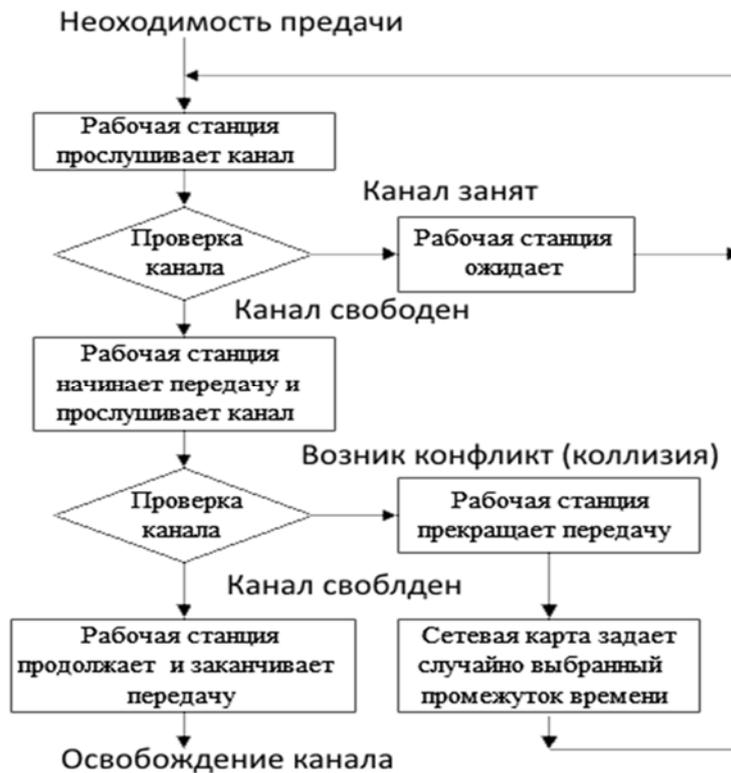


Рисунок 3.1 – Упрощенный алгоритм метода доступа CSMA/CD

Алгоритм состоит из следующих этапов.

1 Для передачи кадра интерфейс-отправитель должен убедиться, что разделяемая среда свободна.

2 Если канал занят – рабочая станция ждет.

3 Если канал свободен – рабочая станция начинает передачу и одновременно прослушивает канал.

4 При возникновении коллизии рабочая станция прекращает передачу и ждет случайно выбранный промежуток времени –  $T = Lx$  (интервал отсрочки). Интервал отсрочки равен 512 битовым интервалам:  $L$  – целое число, выбранное с равной вероятностью из диапазона  $[0, 2N]$  (где  $N$  – номер повторной попытки передачи данного кадра: 1, 2, ..., 10).

5 При отсутствии коллизии – передача продолжается до конца кадра.

Метод управления обменом CSMA/CD используется как в обычных сетях типа Ethernet, так и в высокоскоростных сетях (Fast Ethernet, Gigabit Ethernet). Но с применением коммутаторов в сети, дуплексного режима и особенно построения сегментов сети Ethernet высоких скоростей по топологии «звезда» необходимость в данном методе отпала. В стандарте 10 Gigsbit Ethernet (10 GE) он уже не применяется.

### Форматы кадров технологии Ethernet

В сетях Ethernet на канальном уровне используются кадры четырех различных форматов (типов). На практике, чаще всего применяется кадр, который носит название Ethernet DIX, или Ethernet II (рисунок 3.2).

Кадр Ethernet DIX (II)				
6	6	2	46-1500	4
DA	SA	T	Данные	FCS

Рисунок 3.2 – Формат кадра Ethernet DIX, или Ethernet II

Кадр Ethernet DIX (Ethernet II) представляет собой структуру из пяти полей: DA (Destination Address) – MAC-адрес назначения; SA (Source Address) – MAC-адрес источника; Данные – поле данных; FCS (Frame Check Sequence) – поле контрольной последовательности кадра.

Кадр Ethernet формируется подуровнем MAC-канального уровня и широко применяется в современных локальных вычислительных сетях.

#### Детерминированные методы доступа

**TPMA** (Token Passing Multiple Access) – множественный доступ с передачей полномочия, или метод с передачей маркера.

*Метод с передачей маркера* – это метод доступа к среде, в котором от рабочей станции к рабочей станции передается маркер, дающий разрешение на передачу сообщения. При получении маркера рабочая станция может передавать сообщение, присоединяя его к маркеру, который переносит это сообщение по сети. Каждая станция видит это сообщение, но принимает его только станция-адресат. При этом она создает новый маркер.

Широко применяется в сетях с топологией «кольцо».

**TDMA** (Time Division Multiple Access) – множественный доступ с разделением во времени, основан на распределении времени работы канала между системами (рисунок 3.3).

Доступ TDMA основан на использовании тактового генератора. Этот генератор делит время канала на повторяющиеся циклы. Каждый из циклов начинается сигналом-разграничителем (сигнал синхронизации). Цикл включает  $n$  (обычно 30) пронумерованных временных интервалов, называемых ячейками. Интервалы предоставляются для загрузки в них блоков данных.



Рисунок 3.3 – Метод доступ с разделением во времени

Данный способ позволяет организовать передачу данных с коммутацией пакетов и с коммутацией каналов.

**FDMA (Frequency Division Multiple Access)** или множественный доступ с разделением длины волны (**Wavelength Division Multiple Access – WDMA**). Доступ FDMA основан на разделении полосы пропускания канала на группу полос частот (рисунок 3.4), образующих *логические каналы*. Широкая полоса канала делится на ряд узких полос, разделенных защитными полосами.

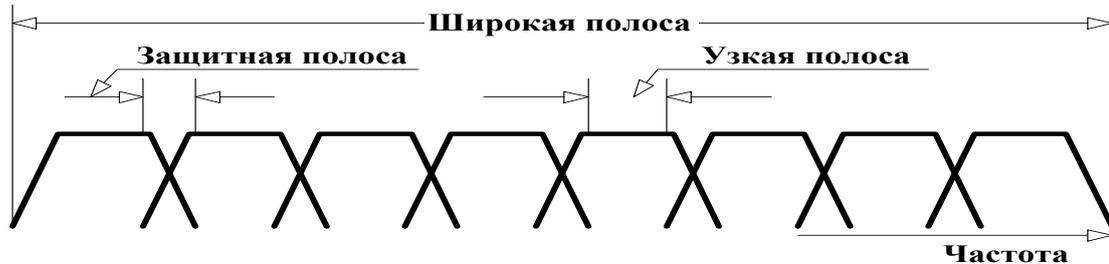


Рисунок 3.4 – Схема выделения логических каналов (метод доступа FDMA)

### ***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Определить максимальное количество кадров минимальной и максимальной длины, проходящих по стандартной сети Ethernet, –10Base-5.

3 Определить максимальную пропускную способность сегмента 10Base-5 (10 Мбит/с) для кадров максимальной, усредненной и минимальной длины.

4 Сделать выводы по результатам исследований.

5 Оформить отчет.

### ***Контрольные вопросы***

1 Какие методы доступа относятся к случайным, а какие к детерминированным?

2 Подробно опишите алгоритм CSMA/CD.

3 Как обрабатывается коллизия методами CSMA/CD и CSMA/CA?

4 Какие поля входят в формат кадров технологии Ethernet?

5 В чем суть метода с передачей маркера TPMA?

6 Как осуществляется передача сигнала методом TDMA?

7 Как формируются каналы методом FDMA?

## 4 Лабораторная работа № 4. Базовые настройки коммутатора CISCO

**Цель работы:** изучение принципов и алгоритмов построения локальных сетей (LAN) с использованием коммутаторов Ethernet на основе симулятора сети передачи Cisco Packet Tracer. Изучение основных методов настройки коммутаторов локальных сетей Ethernet на примере коммутаторов CISCO.

### *Основные теоретические положения*

Применение разделяемой физической среды в локальных сетях Ethernet привело к необходимости создания устройств, позволяющих разбить большую сеть на сегменты и микросегменты, с целью уменьшения количества коллизий и тем самым повысить общую производительность сети. Такими устройствами стали мосты и коммутаторы Ethernet, которые используют «**Алгоритм прозрачного моста IEEE 802.1D**» в продвижении кадра к узлу назначения. Работа коммутатора (моста) заключается в том, что он принимает целый кадр в буфер, анализирует его и только потом перенаправляет кадр на соответствующий порт.

Коммутатор – многофункциональное и интеллектуальное устройство, выполняет множество функций и работает в различных режимах. Коммутатор принимает кадр в буфер порта, анализирует адрес источника и помещает запись формата *<MAC адрес источника – номер порта>* **в адресную таблицу (таблицу коммутации)**. Таким образом, коммутатор строит свою адресную таблицу на основании пассивного наблюдения за трафиком. **По адресу источника кадра коммутатор делает вывод о принадлежности узла – источника тому или иному сегменту сети.**

Заполнение адресной таблицы называется **режимом обучения**.

Буферизация разрывает логику работы всех сегментов как единой разделяемой среды, позволяя уменьшить домены коллизий. Разбивая локальную сеть на мелкие сегменты, мы приходим к тому, что к каждому порту может подключиться один узел сети (рабочая станция, сервер, концентратор, коммутатор и т. д.). Эту особенность коммутатора называют **микросегментацией**. В этом случае и при применении полудуплексного режима домен коллизий сужается до области «коммутатор – компьютер». Если используется режим полного дуплекса, то домен коллизий как таковой не существует.

К порту коммутатора можно подключить концентратор, к портам которого подключены узлы сети. В этом случае домен коллизий определяется сегментами сети «концентратор – компьютеры», подключенные к портам концентратора, а также «коммутатор – концентратор».

## Порядок выполнения работы

1 Изучить основные теоретические положения.

2 С помощью симулятора Packet Tracer смоделировать локальную сеть (LAN) с использованием коммутатора и концентратора (рисунок 4.1).

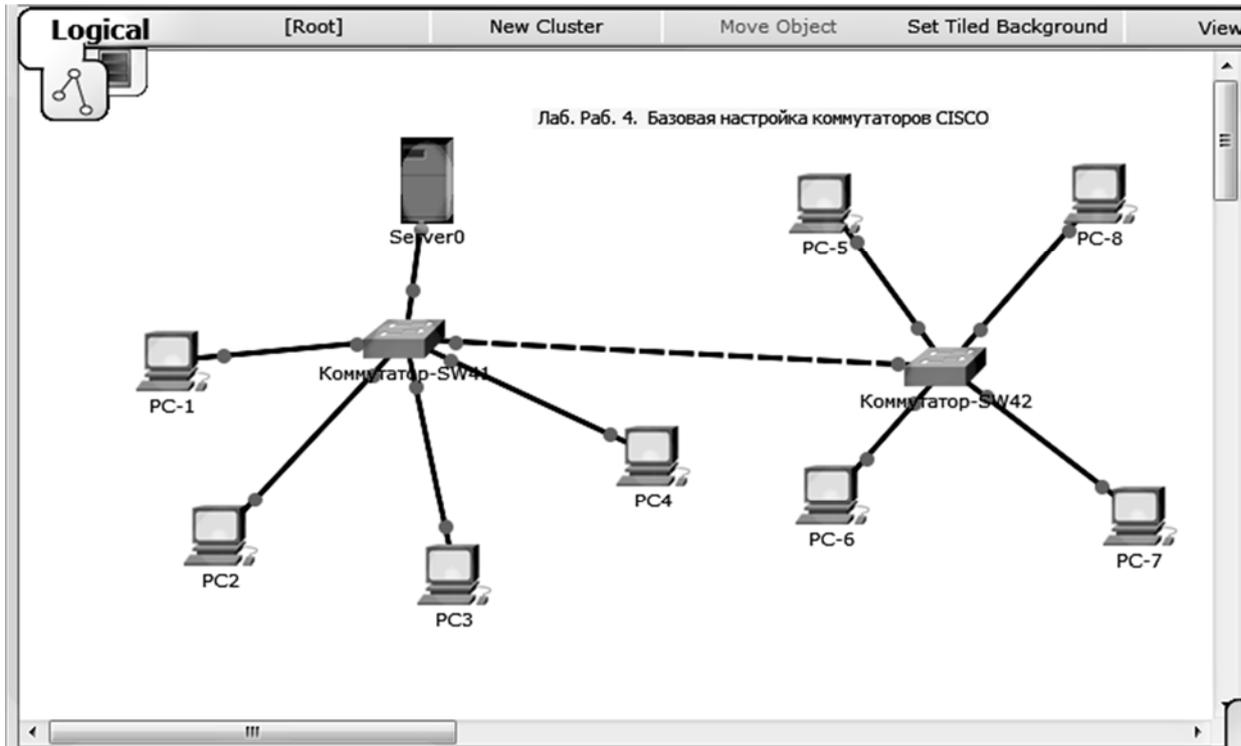


Рисунок 4.1 – Модель сети с использованием коммутаторов CISCO 2960

3 Используя интерфейс командной строки CLI, выполнить базовую настройку коммутатора CISCO Catalyst 2960. Выполнить следующие команды.

3.1 Заходим в привилегированный режим privileged EXEC:

**Switch>enable**

**Switch#**

3.2 Переключаемся в режим глобальной конфигурации и задаем имя устройству, для этого вводим команды:

**Switch#configure terminal**

**Switch(config)#hostname Switch-41**

На втором **Switch(config)#hostname Switch-42**

3.3 Переключаем порты, к которым подключены узлы, в положение «Up» (поднимаем порты), например:

**Switch-41(config)#interface fa0/1**

**Switch-41(config)#no shutdown**

4 Далее проводим тестирование. Для тестирования переводим Packet Tracer в режим симуляции и с одного компьютера посылаем тестовые пакеты на любой другой компьютер. Сохраняем Screen's Shot для отчета.

### ***Контрольные вопросы***

- 1 Назовите назначение коммутаторов и мостов Ethernet.
- 2 Основной алгоритм работы коммутатора.
- 3 Как называется режим коммутатора при построении таблицы коммутации?
- 4 С помощью каких команд можно войти в привилегированный режим и режим глобальной конфигурации?
- 5 Могут ли возникать коллизии в случае подключения концентратора к порту коммутатора и в каком сегменте сети?
- 6 С помощью каких команд поднимаются порты коммутатора?

## **5 Лабораторная работа № 5. Изучение алгоритма STA коммутаторов CISCO**

**Цель работы:** изучение алгоритма покрывающего дерева (Spanning Tree Algorithm, STA). Построение логической схемы сети.

### ***Основные теоретические положения***

Локальные сети, построенные на коммутаторах и разделенные на логические сегменты, называют **коммутируемыми локальными сетями**.

Коммутаторы локальных сетей обрабатывают кадры на основе алгоритма прозрачного моста (стандарт IEEE 802.1D), который позволяет значительно снизить вероятность коллизий при полудуплексном режиме, а в режиме дуплексной передачи вообще избавиться от них.

При приеме широковещательных кадров коммутатор «обязан» передавать кадры на все порты согласно алгоритму прозрачного моста, что ограничивает выбор топологии построения сети, а именно: для построения локальной сети на коммутаторах нельзя использовать параллельные соединения коммутаторов между собой, а также петлеобразные соединения в цепочке соединённых последовательно коммутаторов, т. к. в этом случае возникает закливание сигналов, что приводит к «широковещательному шторму». При построении сети с помощью коммутаторов можно использовать только топологию «звезда» или «дерево» с корневым коммутатором. При построении масштабных сетей, часто для повышения надежности сети, используются альтернативные дополнительные линии связи параллельно с основными.

Для избежание «широковещательных штормов» в современных коммутаторах применяется «Алгоритм покрывающего дерева» (Spanning Tree Algorithm, STA), который блокирует альтернативные связи и держит их в горячем резерве, до момента выхода из строя основной связи, корневого коммутатора или изменения в топологии сети.

## Практическое выполнение задания

1 Запустите программу Cisco Packet Tracer.

2 В области «Логическое пространство» создайте иерархическое дерево сети, аналогичное дереву на рисунке 5.1. Выберите тип коммутатора CISCO 2950.

3 Установите IP-адрес  $192.100+G.NN.H$  для всех конечных устройств (Host), (где G – последняя цифра номера группы; NN – порядковый номер в журнале; H – порядковый номер компьютера), маску  $255.255.255.0$ .

4 Проверить прохождение пакета от каждого компьютера к любому другому, сделать Screen Shot's и отразить в отчете.

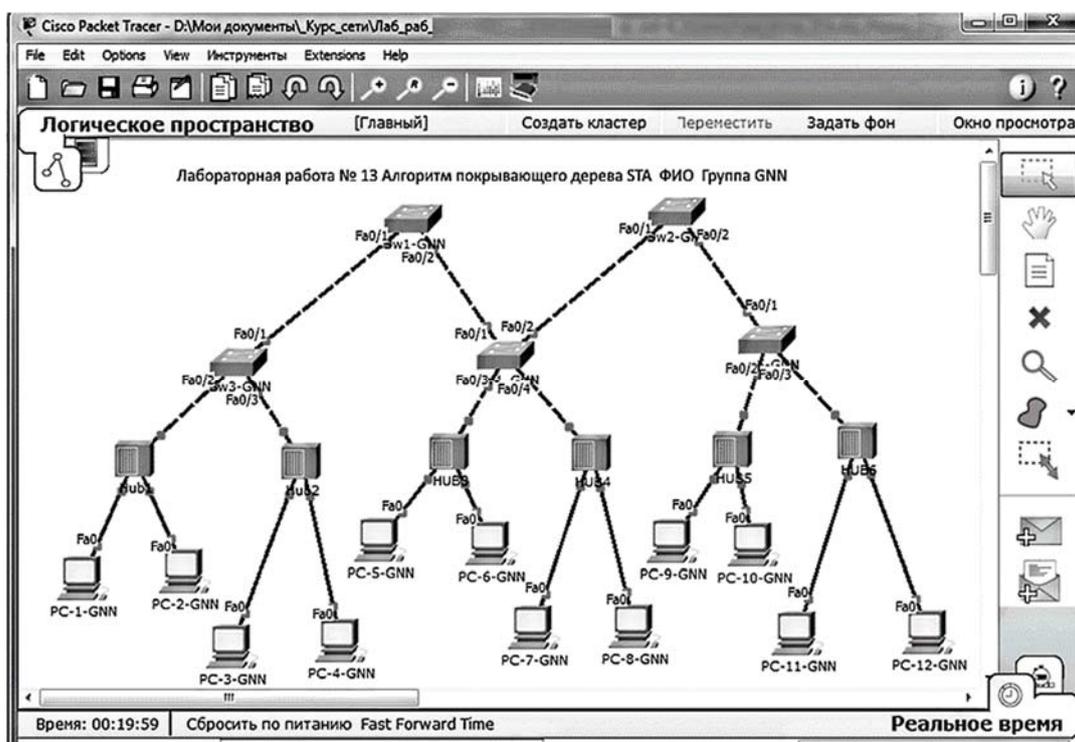


Рисунок 5.1 – Топология сети иерархическое дерево

## Моделирование работы протокола STP

1 Соедините оба коммутатора первого уровня SW1 и SW2 со всеми коммутаторами второго уровня (рисунок 5.2), создавая избыточные связи.

2 Через некоторое время индикаторы на одном из портов коммутатора первого уровня загорятся оранжевым цветом. Это означает, что пакеты будут проходить по сети, минуя данные порты этого коммутатора.

3 Пропуская испытательные пакеты «простого PDU» от одного компьютера к другому, убедитесь в том, что пакеты проходят, минуя альтернативные связи. Зафиксируйте данный этап работы сети в Screen Shot's.

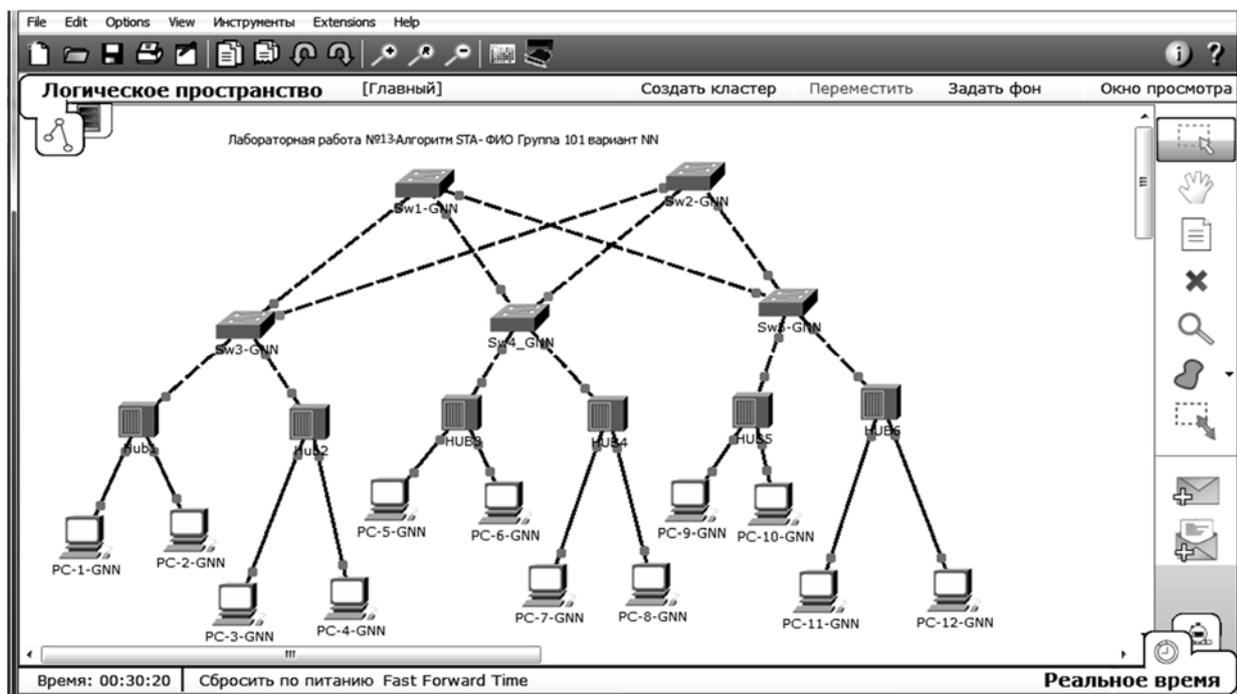


Рисунок 5.2 – Создание избыточных резервных связей

### ***Порядок выполнения работы***

- 1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.
- 2 Выполнить типовое задание.
- 3 Сделать выводы по результатам исследований.
- 4 Оформить отчет.

### ***Контрольные вопросы***

- 1 Что такое коммутируемые локальные сети?
- 2 При помощи какой команды можно просмотреть таблицу MAC-адресов?
- 3 В чем суть микросегментации коммутатора?
- 4 Что такое «режим обучения»?
- 5 В чем суть явления «широковещательный шторм»?
- 6 Что такое сегмент? Какие устройства он включает?
- 7 В чем суть алгоритма STA?
- 8 Что представляет собой идентификатор коммутатора?
- 9 При помощи какой команды можно просмотреть таблицу MAC-адресов?
- 10 Как выбирается root switch?

## **6 Лабораторная работа № 6. Изучение виртуальных локальных сетей (VLAN)**

**Цель работы:** изучение принципов работы технологии виртуальных локальных сетей VLAN при помощи симулятора CISCO Packet Tracer и освоение методов конфигурирования коммутаторов Ethernet.

### ***Основные теоретические положения***

Виртуальной локальной сетью (Virtual LAN, VLAN) называется группа узлов сети, трафик которой, в том числе широковещательный, на канальном уровне полностью изолирован от трафика других узлов.

Основное назначение VLAN – ограничить распространение широковещательного трафика по всем узлам локальной сети.

Сеть, построенная на коммутаторах (устройства второго уровня модели OSI), даже при значительно разветвленной топологии обязана пропускать широковещательный трафик (MAC-адреса FF:FF:FF:FF:FF:FF) ко всем компьютерам разных сегментов сети. Производительность сети в данные моменты значительно снижается. Создание VLAN означает разбиение сети на коммутаторах на несколько широковещательных доменов. Одна и та же VLAN на разных коммутаторах образует один широковещательный домен.

VLAN также позволяет гибко разделить узлы компьютерной сети на группы, с учетом их принадлежности к разным IP-сетям (подсетям), независимо от места положения.

С помощью VLAN, упрощается задача применения политик и правил безопасности. Политики безопасности с помощью VLAN можно применять к целым подсетям, а не только к отдельному устройству. Кроме того, переход из одной VLAN в другую предполагает прохождение через устройство третьего уровня, на котором, как правило, применяются политики, разрешающие или запрещающие доступ из одной VLAN в другую, например, листы доступа – ACL.

Построение VLAN-сетей в основном осуществляется тремя способами:

- 1) VLAN на базе портов;
- 2) VLAN на основе меток в дополнительном поле кадра – стандарт IEEE 802.1Q;
- 3) VLAN на базе MAC-адресов.

### ***Практическое выполнение задания***

#### **VLAN на базе портов на одном коммутаторе.**

При использовании VLAN на базе портов, каждый порт назначается в определенную VLAN, независимо от того, какой пользователь или компьютер, или Hub подключены к этому порту. Это означает, что все пользователи, подключенные к этому порту, будут членами одной VLAN.

### Этапы создания модели.

1 С помощью симулятора Packet Tracer (PT) создадим модель фрагмента сети (рисунок 6.1).

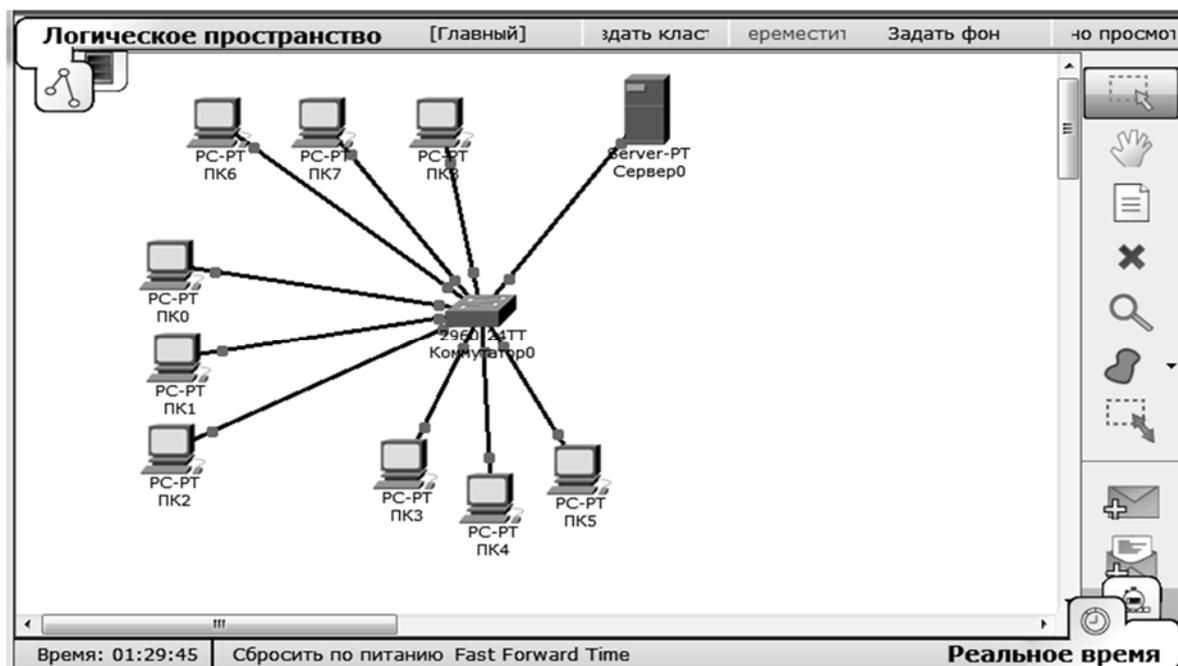


Рисунок 6.1 – Топология сети с одним коммутатором

2 Компьютеры ПК0 ...ПК9 соединяем с портами коммутатора и назначаем IP-адреса согласно таблице 6.1 (в отчёте создать аналогичную таблицу с конкретными данными согласно вариантам).

Таблица 6.1 – Конфигурация компьютеров и VLAN

Отдел	Компьютер	IP- адрес	Номер порта коммутатора	ID VLAN
Бухгалтерия (Buh)	ПК-0	10.10.NN.G01	Fa 0/1	121
	ПК-1	10.10.NN.G02	Fa 0/2	
	ПК-2	10.10.NN.G03	Fa 0/3	
Отдел продаж (Sales)	ПК-3	10.10.NN.G04	Fa 0/4	122
	ПК-4	10.10.NN.G105	Fa 0/5	
	ПК-5	10.10.NN.G06	Fa 0/6	
Отдел маркетинга (Market)	ПК-6	10.10.NN.G07	Fa 0/7	123
	ПК-7	10.10.NN.G08	Fa 0/8	
	ПК-8	10.10.NN.G09	Fa 0/9	
	SServer	10.10.NN.G10	Fa 0/10	

3 Данная сеть является сетью одного адресного пространства IP-адресов и одного широковещательного домена. Проверим данное утверждение, посылая простой пакет PDU от каждого компьютера к серверу и друг другу.

4 Создадим три VLAN для одного коммутатора, используя статическое конфигурирование и интерфейс командной строки – CLI:

– создаём первую VLAN в режиме глобального конфигурирования:

**Правило для выбора VLAN:  $IDVLAN = NN * 10 + NumVLAN$ ,  
NN-Ваш порядковый номер по журналу**

***Switch(config)#vlan 121***

и присваиваем ей имя:

***Switch(config-vlan)#name Buh***

– аналогично создаем вторую и третью VLAN;

– после создания VLAN-сети назначаем ей порт или диапазон портов, используя команды `switchport mode access` и `switchport access vlan №`, согласно таблице 6.1:

***Switch(config)#int range fa0/1-3***

***Switch(config-if-range)#switchport mode access***

***Switch(config-if-range)#switchport access vlan 121***

***Switch(config-if-range)#end***

– выполним дальнейшую конфигурацию для остальных портов (команды CLI отобразить в отчете в виде Screen Shot's);

– с помощью команды `ping` проверим «связанность» компьютеров в одной и в разных VLAN, результат представить Screen Shot's;

– в режиме симуляции воспроизвести продвижение широковещательного пакета, просмотреть «Информацию PDU» на устройствах: коммутатор, компьютер, проанализируйте, результат Screen Shot's сохраните в отчете.

### ***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Выполнить типовое задание. Продемонстрировать распространение широковещательного трафика до назначения VLAN и после.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

### ***Контрольные вопросы***

1 Кратко объясните основные алгоритмы коммутаторов Ethernet.

2 Что такое «широковещательный шторм» и в каких ситуациях возникает?

3 Что представляет собой виртуальная локальная сеть VLAN?

4 Назначение, функции и преимущества сетей VLAN?

5 Рассказать подробно создание виртуальных сетей на базе одного коммутатора.

6 Показать на примере конфигурацию VLAN на базе портов одного коммутатора.

## **7 Лабораторная работа № 7. Изучение правил адресации сетевого уровня**

**Цель работы:** изучить систему адресации протокола сетевого уровня, принципы распределения адресного пространства, методы назначения IP-адресов между участниками сети передачи данных и общие функциональные особенности сетевого уровня модели OSI на примере IP-протокола.

### ***Основные теоретические положения***

Сетевой уровень (network layer) служит для образования единой транспортной системы, объединяющей несколько сетей LAN (разных технологий Ethernet, Token Ring, FDDI), PAN, MAN, Wi-Fi, просто отдельных компьютеров (host) в единую сеть – называемой составной сетью. Доставка пакетов сетевого уровня осуществляется с помощью специальных устройств – маршрутизаторов, которые на физическом уровне имеют специфические интерфейсы для каждого типа сетей и линий связи. Для того чтобы протоколы сетевого уровня могли доставлять пакеты любому узлу составной сети, эти узлы должны иметь адреса, уникальные в пределах данной составной сети. Такие адреса называются сетевыми, или глобальными. В связи с необходимостью перенаправлять пакеты из одной подсети сети в другую, сетевые адреса должны удовлетворять следующим требованиям:

- адреса должны быть уникальны. В сети не может быть нескольких участников с одинаковыми адресами во избежание неоднозначности;
- сетевой адрес должен содержать информацию о том, как достичь получателя по составной сети, т. е. он должен иметь адрес подсети и адрес получателя.

### ***Протокол IP (Internet Protocol)***

Архитектуру сетевого уровня удобно рассматривать на примере сетевого протокола IP – самого распространенного в настоящее время, основного протокола сети Интернет. Термин «стек протоколов TCP/IP» означает «набор протоколов, связанных с IP и TCP (протоколом транспортного уровня)».

Архитектура протоколов TCP/IP предназначена для объединенной сети, состоящей из соединенных друг с другом шлюзами отдельных разнородных пакетных подсетей, к которым подключаются разнородные машины.

Каждая из подсетей работает в соответствии со своими специфическими требованиями и имеет свою природу средств связи. Однако предполагается, что каждая подсеть может принять пакет информации (данные с соответствующим сетевым заголовком) и доставить его по указанному адресу в этой конкретной подсети.

Не требуется, чтобы подсеть гарантировала обязательную доставку пакетов и имела надежный сквозной протокол.

Таким образом, две машины, подключенные к одной подсети, могут обмениваться пакетами.

Когда необходимо передать пакет между машинами, подключенными к разным подсетям, то машина-отправитель посылает пакет в соответствующий шлюз (шлюз подключен к подсети также, как обычный узел). Оттуда пакет направляется по определенному маршруту через систему шлюзов и подсетей, пока не достигнет шлюза, подключенного к той подсети, что и машина-получатель: там пакет направляется к получателю.

Таким образом, адрес получателя должен содержать в себе:

- номер (адрес) подсети;
- номер (адрес) участника (хоста) внутри подсети.

IP-адреса (IPv4) представляют собой 32-разрядные двоичные числа. Для удобства их записывают в виде четырех десятичных чисел, разделенных точками 192.168.200.47. Каждое число является десятичным эквивалентом соответствующего байта адреса.

В двоичном формате этот адрес представляет собой 11000000.10101000.11001000.00101111 (для удобства каждый байт отделен точками, как и в десятичном формате).

Каждый IP-адрес содержит адрес сети и адрес компьютера (хоста) в данной сети, но специального разграничительного знака между номером сети и номером узла не предусмотрено. Для разделения IP-адреса на номер сети и номер узла используются несколько вариантов.

Первоначально использовался простой способ, а именно: адрес фиксировано, жестко разбивался на две части (RFC 760), но из-за неэффективного использования адресного пространства не нашел широкого применения.

Второй подход, распространенный до недавнего времени, заключается в использовании классов адресов (RFC 791). Вводится пять классов адресов: А, В, С, D, Е. Три из них (А, В и С) используются для адресации сетей, а два (D и Е) имеют специальное назначение. Для каждого класса сетевых адресов определено собственное положение границы между номером сети и номером узла.

Третий способ (RFC 950, RFC 1518) основан на использовании маски, которая позволяет максимально гибко устанавливать границу между номером сети и номером узла, а также создавать множество сетей разного размера. Для этих целей наряду с IP-адресом введено такое понятие как маска.

Маска сети также является 32-разрядным двоичным числом. Разряды маски имеют следующий смысл:

- 1) если разряд маски равен 1, то соответствующий разряд адреса является разрядом адреса подсети;
- 2) если разряд маски равен 0, то соответствующий разряд адреса является разрядом хоста внутри подсети.

Все единичные разряды маски (если они есть) находятся в старшей (левой) части маски, а нулевые – в правой (младшей).

### ***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Выполнить типовое задание.

Напишите компьютерную программу со следующими функциями:

– прямое и обратное преобразования 32-битовых чисел в точечное десятичное представление;

– определение к какому классу А, В, С, D, Е относится введенный IP-адрес;

– определение и отдельное отображение адреса сети, адреса хоста и маски сети по введенному IP-адресу. (автоматическое определение корректности введенных значений);

– прямое и обратное преобразование из системы обозначений CIDR с косой чертой в эквивалентное точечное десятичное представление.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

### ***Контрольные вопросы***

1 Для чего нужен сетевой уровень, за что отвечает, как обобщённо называются технологии, обеспечивающие сетевой уровень?

2 Какова структура IP-адреса?

3 Что такое маска подсети?

4 Как определяется размер подсети?

5 Как определить диапазон адресов в подсети?

6 Можно ли использовать в качестве сетевого MAC-адрес?

7 Маршрутизаторы – назначение, функции, физические интерфейсы?

## **8 Лабораторная работа № 8. Изучение принципов статической маршрутизации IP-сетей**

**Цель работы:** изучение принципов маршрутизации IP-сетей на примере протоколов статической маршрутизации с использованием программного обеспечения построения виртуальных сетей – Packet Tracer.

### ***Основные теоретические положения***

Сетевой протокол IP является маршрутизируемым. Для передачи данных от компьютера одной локальной сети к компьютеру другой локальной сети, могут использоваться различные маршруты и маршрутизаторы. **Маршрутизация (routing)** – процесс определения маршрута следования информации в сетях связи. Задача маршрутизации состоит в определении последовательности транзитных узлов для передачи пакета от источника до адресата. Каждый маршрутизатор имеет от двух и более сетевых интерфейсов, к которым

подключены локальные сети, либо маршрутизаторы соседних сетей. Выбор маршрута или другими словами интерфейса, маршрутизатор осуществляет на основе таблицы маршрутизации. Таблицы маршрутизации содержат информацию о сетях, подключенных локально (непосредственно), и сведения о маршрутах или путях, по которым маршрутизатор связывается с удаленными сетями.

Эти маршруты могут назначаться администратором статически или определяться динамически при помощи программного протокола маршрутизации.

**Маршрутизатор (router, роутер)** – сетевое устройство третьего уровня модели OSI, обладающее как минимум двумя сетевыми интерфейсами, которые находятся в разных сетях. Причем в сетях могут использовать различные технологии физического и канального уровней. Маршрутизатор может иметь интерфейсы: для работы по медному кабелю, оптическому кабелю, так и по беспроводным «линиям» связи.

### *Средства маршрутизации*

Каждый маршрутизатор принимает решения о направлении пересылки пакетов на основании таблицы маршрутизации. Таблица маршрутизации содержит набор правил. Каждое правило в наборе описывает шлюз или интерфейс, используемый маршрутизатором для доступа к определенной сети.

Маршрут состоит из пяти основных компонентов (полей записи):

- 1) значение получателя (адрес сети назначения);
- 2) маска;
- 3) интерфейс (порт);
- 4) адрес шлюза;
- 5) стоимость маршрута или метрика маршрута.

Чтобы переслать сообщение получателю, маршрутизатор извлекает IP-адрес получателя из пакета и находит соответствующее правило в таблице маршрутизации. При обнаружении совпадающего адреса пакет пересылается на соответствующий интерфейс или к соответствующему шлюзу.

### *Протоколы маршрутизации*

Протокол маршрутизации – это сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной сети.

**Статическая маршрутизация** – вид маршрутизации, при котором информация о маршрутах заносится в таблицы маршрутизации каждого маршрутизатора вручную администратором сети. Статические маршруты не изменяются до тех пор, пока администратор не перенастроит их вручную. В таблице маршрутизации эти маршруты обозначаются буквой S. Символом C в таблице маршрутизации помечены непосредственно присоединенные к маршрутизатору сети.

Данный вид маршрутизации имеет ряд **недостатков**:

- плохая масштабируемость, т. к. при добавлении  $N$  сети потребуется сделать  $2(N+1)$  записей о маршрутах;
- отсутствует возможность адекватно отреагировать на ошибки и отказы оборудования канального и сетевого уровня;
- ввод всей информации вручную является весьма трудоемкой задачей и влечет за собой необходимость документирования этих маршрутов;
- при изменении топологии сети требуется вручную менять правила маршрутизации, т. е. переконфигурировать таблицу маршрутизатора.

**Положительные качества:**

- легкость конфигурации. Метод статической маршрутизации является довольно простым для понимания и настройки;
- отсутствует обмен служебной информацией между маршрутизаторами о топологии сетей, таким образом и дополнительная нагрузка на сеть в виде широковещательного служебного трафика;
- при использовании статических записей процессор маршрутизатора наименее нагружен при определении маршрутов.

**Статическая маршрутизация продолжает успешно использоваться:**

- при организации работы компьютерных сетей небольшого размера (1–3 маршрутизатора);
- на компьютерах (рабочих станциях) внутри сети. В таком случае обычно задается маршрут шлюза по умолчанию;
- в целях безопасности, когда необходимо скрыть некоторые части составной корпоративной сети.

**Статическая маршрутизация по умолчанию** означает, что если пакет предназначен для сети, которая не перечислена в таблице маршрутизации, то маршрутизатор отправит пакет по заданному по умолчанию маршруту.

### **Практическое выполнение задания**

Для выполнения лабораторной работы используется Packet Tracer.

**Базовая настройка маршрутизаторов и устройств сети:** в области «Логическое пространство» создайте фрагмент составной (корпоративной) сети аналогично рисунку 8.1.

**Варианты задания.** В качестве вариантов используется следующее правило обозначения сетевых устройств (коммутаторов, маршрутизаторов, компьютеров) и IP-адресов, например, для коммутаторов –  $SW-I-GNN$ ,  $G$  – номер группы (первая цифра),  $NN$  – порядковый номер в журнале группы (ведущий ноль в данном случае пишется).

**Установка дополнительного модуля.** Для соединения маршрутизаторов между собой, устанавливается дополнительный модуль интерфейсов NM-4A/S и линия связи DTE/DCE цифровых каналов на каждом маршрутизаторе.

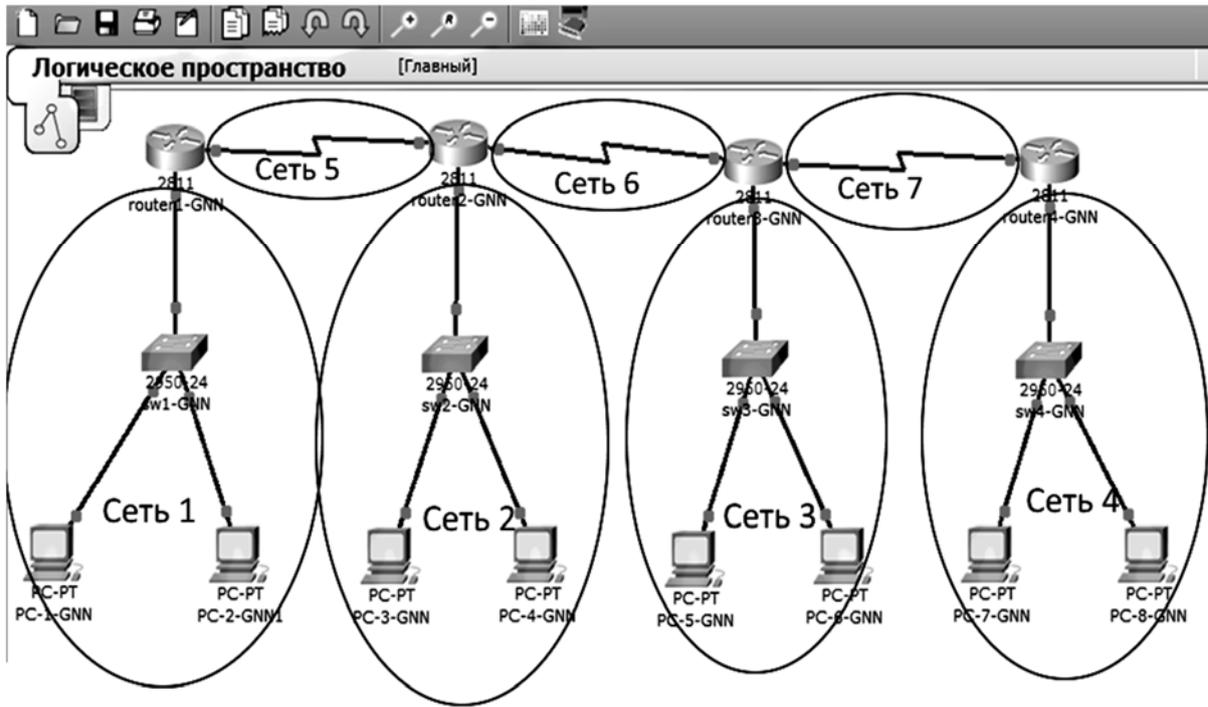


Рисунок 8.1 – Топология составной сети из четырех маршрутизаторов

**Назначаем IP-адреса интерфейсам маршрутизаторов** согласно таблице 8.1 и в соответствии с рисунком 8.1.

При назначении IP-адреса интерфейсу маршрутизатора вводим команды:

```
R1-GNN(config-if)# interface f0/0  
R1-GNN(config-if)# ip address 192.102.12.1 255.255.255.0
```

(пример для студента со второй группы с номером NN равным 12; в примере конфигурация интерфейса **Fast Ethernet 0/0.**)

Таблица 8.1 – Адреса сетей и интерфейсов маршрутизаторов

Сеть	IP-адрес сети	Интерфейс	IP-адрес интерфейса
1	192.100+G.NN.0/24	F0/0 R1-GNN	192.100+G.NN.1
2	192.100+G.10+NN.0/24	F0/0 R2-GNN	192.100+G.10+NN.1
3	192.100+G.20+NN.0/24	F0/0 R3-GNN	192.100+G.20+NN.1
4	192.100+G.30+NN.0/24	F0/0 R4-GNN	192.100+G.30+NN.1
5	200.50.50.0/30	S1/1 R1-GNN	200.50.50.9
6		S1/2 R2-GNN	200.50.50.10
7	200.60.60.0/30	S1/1 R2-GNN	200.60.60.9
8		S1/2 R3-GNN	200.60.60.10
9	200.70.70.0/30	S1/1 R3-GNN	200.70.70.9
10		S1/2 R4-GNN	200.70.70.10

Для конфигурации интерфейсов между маршрутизаторами (сети 5–7), поступаем аналогичным образом исходя из таблицы 8.1 и рисунка 8.1.

```
R1-GNN(config)# interface S1/1
R1-GNN(config-if)# ip address 200.50.50.9 255.255.255.252
R2-GNN(config)# interface S1/2
R2-GNN(config-if)# ip address 200.50.50.10 255.255.255.252
```

**Назначаем IP-адреса компьютеров сетей 1–4** исходя из таблицы 8.2. «Кликнуть» конфигурируемый узел, например, PC1-GNN. Во всплывшем окне выбрать опцию Desktop, затем IP Configuration и в новом окне установить IP-адрес узла, маску подсети и адрес шлюза в соответствии с таблицей 8.2.

Таблица 8.2 – Адреса составной сети

Сеть	Адрес сети	Шлюз по умолчанию	IP-адрес узла 1	IP-адрес узла 2
1	192.100+G.NN.0/24	192.100+G.NN.1	192.100+G.NN.2	192.100+G.NN.3
2	192.100+G.10+NN.0/24	192.100+G.10+NN.1	192.100+G.10+NN.2	192.100+G.10+NN.3
3	192.100+G.20+NN.0/24	192.100+G.20+NN.1	192.100+G.20+NN.2	192.100+G.20+NN.3
4	192.100+G.30+NN.0/24	192.100+G.30+NN.1	192.100+G.30+NN.2	192.100+G.30+NN.3

**Конфигурирование статической маршрутизации.** Адрес входного интерфейса следующего маршрутизатора на пути к адресату также называют шлюзом по умолчанию. Например, для пакетов, попавших в маршрутизатор R2-GNN, шлюзами по умолчанию будут:

- 1) интерфейс s1/1 маршрутизатора R1-GNN с адресом 200.50.50.11;
- 2) интерфейс s1/2 маршрутизатора R3-GNN с адресом 200.60.60.8.

**Формирование таблицы статической маршрутизации.** Маршрутизатор R3-GNN является транзитным, поэтому в него вводим информацию о всех сетях, которые напрямую не связаны с данным маршрутизатором. В таблице 8.3 приведен пример для  $G = 2$  и  $NN = 12$ . (Конфигурация производится согласно своему варианту).

Таблица 8.3 – Адреса составной сети

Сеть	Адрес сети	Адрес шлюза
1	192.102.12.0	200.60.60.11
2	192.102.22.0	200.60.60.11
4	192.102.42.0	200.70.70.12

Таблицу маршрутизации формируем с помощью команд, используя CLI:  
– для сети 1

```
R3(config)# ip route 192.168.12.0 255.255.255.0 200.60.60.11;
```

– для сети 2

```
R3(config)# ip route 192.168.22.0 255.255.255.0 200.60.60.11;
```

– для сети 4

```
R3(config)# ip route 192.168.42.0 255.255.255.0 200.60.60.12.
```

Просмотрите таблицу маршрутизации с помощью команды ***show ip route***.

Аналогично сконфигурируйте маршрутизаторы R1, R2, R4.

Маршрутизаторы R1 и R4 являются тупиковыми, для них можно сконфигурировать таблицу маршрутизации одной командой – маршрут по умолчанию:

```
R1(config)# ip route 0.0.0.0 0.0.0.0 200.50.50.10
```

```
R4(config)# ip route 0.0.0.0 0.0.0.0 200.70.70.9
```

***Проверьте связанность сетей*** с помощью прохождения тестовых пакетов из одной сети в другую.

### ***Контрольные вопросы***

- 1 В чем заключается задача маршрутизации?
- 2 Что такое маршрутизатор?
- 3 Перечислите основные компоненты маршрута.
- 4 С помощью каких команд можно сконфигурировать маршрут?
- 5 Что такое статическая маршрутизация? Что обозначается буквами C и S?
- 6 Назовите недостатки статической маршрутизации.
- 7 Назовите три положительных признака статической маршрутизации.
- 8 В каких случаях применяется статическая маршрутизация?
- 9 Для чего необходим маршрут по умолчанию, как его установить?

## **9 Лабораторная работа № 9. Изучение принципов динамической маршрутизации IP-сетей**

**Цель работы:** изучение принципов динамической маршрутизации IP-сетей на примере протоколов маршрутизации RIP и OSPF с использованием программного обеспечения построения виртуальных сетей Packet Tracer 6.2.

### ***Основные теоретические положения***

Сетевой протокол IP является маршрутизируемым.

**Маршрутизация (routing)** – процесс определения маршрута следования информации в сетях связи. **Задача маршрутизации** состоит в определении последовательности транзитных узлов для передачи пакета от источника до адресата. Определение маршрута следования и продвижение IP-пакетов выполняют специализированные сетевые устройства – **маршрутизаторы**.

Процесс маршрутизации (определение оптимального маршрута) осуществляется на основе таблиц маршрутизации.

Таблица маршрутизации может составляться двумя способами:

1) **статическая маршрутизация**, когда записи в таблице вводятся и изменяются вручную администратором. Такой способ требует минимума вычислительных ресурсов, но в больших сетях слишком трудоемкий;

2) **динамическая маршрутизация** (dynamic routing) – или адаптивная маршрутизация, когда записи в таблице обновляются автоматически при помощи одного или нескольких протоколов маршрутизации (**RIP, OSPF, IGRP, EIGRP, IS-IS, BGP** и др.). Кроме того, маршрутизатор строит таблицу оптимальных путей к сетям назначения на основе различных критериев: количества промежуточных узлов, пропускной способности каналов, задержки передачи данных.

### ***Протокол маршрутизации RIP***

Протокол RIP (Routing Information Protocol – протокол маршрутной информации), является внутренним протоколом маршрутизации дистанционно-векторного типа.

Будучи простым в реализации, этот протокол чаще всего используется в небольших сетях. Для IP имеются две версии RIP – RIPv1 и RIPv2. Протокол RIPv1 не поддерживает масок. Протокол RIPv2 передает информацию о масках сетей и в большей степени соответствует требованиям сегодняшнего дня.

Протокол RIP строит таблицы маршрутизации за три этапа.

*Этап 1.* В исходном состоянии создается минимальная таблица, включающая необходимые параметры на основании непосредственно подключенных локальных сетей, на каждом маршрутизаторе составной сети.

*Этап 2.* После инициализации каждый маршрутизатор начинает посылать своим соседям сообщения протокола RIP – минимальную таблицу.

*Этап 3.* Получение RIP-сообщений от соседей и обработка полученной информации. Маршрутизатор вычисляет наиболее эффективные маршруты, выбирая записи из альтернативных маршрутов, а остальные удаляет.

### ***Протокол маршрутизации OSPF***

**OSPF (Open Shortest Path First)** – протокол динамической маршрутизации, основанный на технологии отслеживания состояния канала (link-state technology).

В основе работы протокола OSPF лежит **Алгоритм Дейкстры** или алгоритм поиска кратчайшего пути, отсюда и название SPF (shortest path first).

#### **Принцип работы протокола OSPF.**

1 Маршрутизаторы обмениваются маленькими HELLO-пакетами. Обменявшись пакетами, они устанавливают соседские отношения, добавляя каждый друг друга в свою локальную таблицу соседей.

2 Маршрутизаторы формируют пакет, называемый LSA (Link State Advertisement): состояние всех связей с соседями, идентификатор ID Маршрутизатора и ID соседа, префикс сети между ними, тип сети, «стоимость» канала связи – метрику.

3 Маршрутизатор рассылает LSA своим соседям.

4 Каждый маршрутизатор, получивший LSA, добавляет в свою локальную таблицу LSDB (Link State Database) информацию из LSA.

5 В LSDB скапливается информация обо всех парах, соединённых в сети маршрутизаторов, т. е. каждая строка таблицы – это информация вида: «Маршрутизатор А соединен с маршрутизатором В, тип связи между ними».

6 После обмена LSA каждый маршрутизатор знает про все «линки». На основании пар строится полная карта сети, включающая все маршрутизаторы и все связи между ними.

7 На основании этой карты каждый маршрутизатор индивидуально ищет кратчайшие с точки зрения метрики маршруты во все сети и добавляет их в таблицу маршрутизации.

Алгоритм протокола OSPF более сложный, требующий более производительных маршрутизаторов, но тем не менее его чаще применяют в сетях среднего и большого масштаба, т. к. он обладает лучшими характеристиками по производительности, сходимости таблиц, отсутствием ошибочных петлеобразных маршрутов.

### ***Порядок выполнения работы***

1 Изучить основные теоретические положения, сделав необходимые выписки в конспект.

2 Выполнить практическое задание.

3 Сделать выводы по результатам исследований.

4 Оформить отчет

### ***Практическое выполнение задания***

1 С помощью симулятора Packet Tracer создайте модель составной сети с маршрутизаторами, изображенной на рисунке 9.1.

На рисунке 9.1 цифрами 1–4 и 5–7 обозначены IP-сети независимых LAN-сетей, цифрами 8–15 – подсети IP-WAN для линий связи между маршрутизаторами.

2 Настройте IP-интерфейсы маршрутизаторов, подключенных к подсетям LAN с хостами в соответствии с таблицей 9.1.

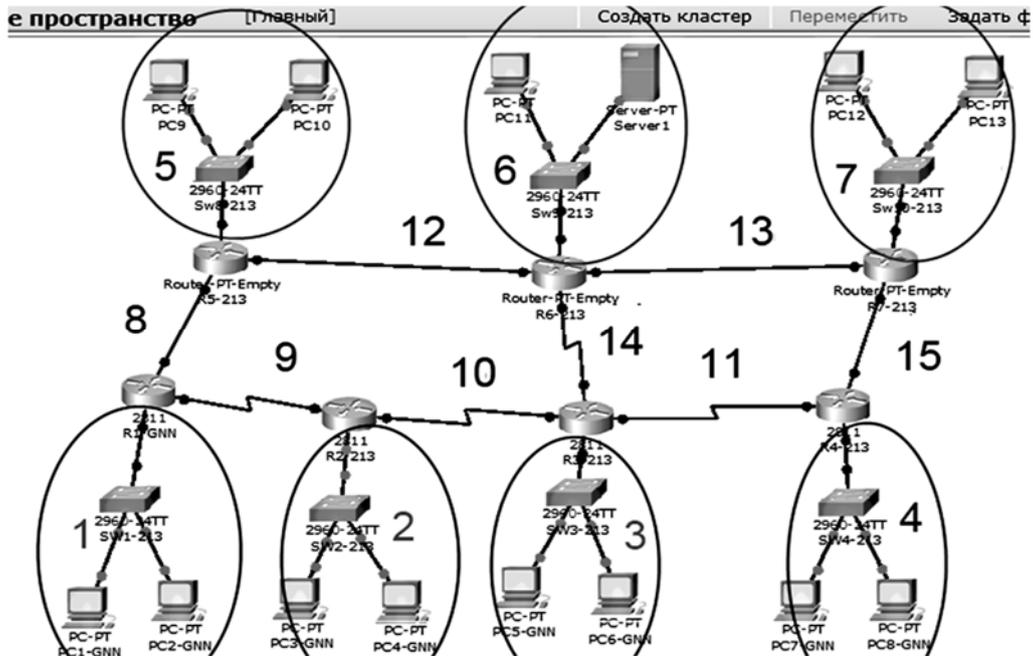


Рисунок 9.1 – Топология составной сети

Таблица 9.1 – Адреса сетей LAN и интерфейсов маршрутизаторов

Сеть	IP-адрес сети	Интерфейс	IP-адрес интерфейса
1	192.100 + G.NN.0/24	F0/0 R1-GNN	192.100+G.NN.1
2	192.100 + G.10 + NN.0/24	F0/0 R2-GNN	192.100+G.10+NN.1
3	192.100 + G.20 + NN.0/24	F0/0 R3-GNN	192.100+G.20+NN.1
4	192.100 + G.30 + NN.0/24	F0/0 R4-GNN	192.100+G.30+NN.1
5	192.100 + G.40 + NN.0/24	F2/0 R5-GNN	192.100+G.40+NN.1
6	192.100 + G.50 + NN.0/24	F3/0 R6-GNN	192.100+G.50+NN.1
7	192.100 + G.60 + NN.0/24	F2/0 R7-GNN	192.100+G.60+NN.1

*Примечание – G – номер группы; NN – порядковый номер в журнале группы (ведущий ноль в данном случае пишется), например, G-2, NN-02*

### Конфигурирование протокола RIP

1 Для инициализации и запуска работы протокола RIP вводим команды с подключением соседних сетей:

```

R5-213(config) #route rip
R5-213(config-router) #network 200.8.8.0
R5-213(config-router) #network 200.12.12.0
R5-213(config-router) #network 192.102.53.0
R5-213(config-router) #exit

```

2 Аналогично вводим команды для остальных маршрутизаторов, проверяем работоспособность сети, делаем подробные Screen Shot's, сохраняем файл макета под именем LR#18-RIP-GNN для отчета.

## **Конфигурирование протокола OSPF**

1 Открываем созданный файл LR#18-RIP-GNN.pkt и удаляем созданные маршруты RIP с помощью команды: `no router rip` на всех маршрутизаторах.

2 На каждом роутере вводим команду ***router ospf ID 1.***

3 Идентифицируем адреса непосредственно подключенных сетей и AS, с помощью следующих команд:

```
R3-213(config) #router ospf 1
R3-213(config-router) #network 192.102.33.0 0.0.0.255 area 0
R3-213(config-router) #network 200.10.10.0 0.0.0.3 area 0
R3-213(config-router) #network 200.11.11.0 0.0.0.3 area 0
R3-213(config-router) #network 200.14.14.0 0.0.0.3 area 0
R3-213(config-router) #end
```

4 Вводим аналогичные команды для остальных маршрутизаторов, проверяем работоспособность сети, делаем подробные Screen Shot's, сохраняем файл под именем LR#18-OSPF-GNN для предоставления вместе с отчетом.

### **Контрольные вопросы**

- 1 Указать принцип работы дистанционно-векторного протокола RIP.
- 2 Что входит в поля таблицы маршрутизации протокола RIP?
- 3 На каком алгоритме основан протокол OSPF?
- 4 Поэтапное описание работы протокола OSPF.
- 5 На основе каких данных создают таблицы протоколы Link-state?
- 6 Как делятся по типам маршрутизаторы зон?
- 7 Что является метрикой OSPF и как она вычисляется?
- 8 В чем заключается преимущество протокола OSPF и RIP?

## **10 Лабораторная работа № 10. Изучение сетевых утилит командной строки Windows**

**Цель работы:** изучить утилиты командной строки Windows, предназначенные для контроля и мониторинга сетей, построенных на базе стека протоколов TCP/IP.

### **Основные теоретические положения**

Сетевая операционная система Windows содержит набор утилит, полезных при диагностике сети. Основные задачи этих программ:

- 1) определение работоспособности сети;
- 2) определение параметров и характеристик сети;
- 3) в случае неправильного функционирования сети – локализация службы или сервиса, вызывающих неисправность.

Главными параметрами сетевых подключений являются их каналные и сетевые адреса и параметры, влияющие на работу сетевого уровня.

Единственным параметром канального уровня, являются MAC-адреса сетевых адаптеров. Для их просмотра можно воспользоваться утилитами **IPCONFIG** или **ROUTE PRINT**, которые покажут MAC-адреса для каждого адаптера. Для изменения MAC-адресов следует воспользоваться драйверами соответствующих сетевых адаптеров.

#### 1 Утилита IPCONFIG.

Утилита IPCONFIG используется для отображения текущих настроек протокола TCP/IP и для обновления некоторых параметров, задаваемых при автоматическом конфигурировании сетевых интерфейсов.

#### *Синтаксис:*

```
ipconfig [/allcompartments] [/all] [/renew[Adapter]] [/release[Adapter]]
[/renew6[Adapter]] [/release6[Adapter]] [/flushdns] [/displaydns] [/registerdns]
[/showclassidAdapter] [/setclassidAdapter [ClassID]].
```

#### *Параметры:*

**/?** – отобразить справку по использованию IPCONFIG;

**/all** – отобразить полную конфигурацию настроек TCP/IP для всех сетевых адаптеров. Отображение выполняется как для физических интерфейсов, так и для логических, как, например, dialup или VPN подключения;

**/allcompartments** – вывести полную информацию о конфигурации TCP/IP для всех секций. Применимо для Windows Vista/Windows 7;

**/displaydns** – отобразить содержимое кэш-службы DNS-клиент;

**/flushdns** – сбросить содержимое кэш-службы DNS-клиент;

**/registerdns** – инициировать регистрацию записей ресурсов DNS для всех адаптеров данного компьютера. Этот параметр используется для изменения настроек DNS сетевых подключений без перезагрузки компьютера;

**/release[Adapter]** – используется для отмены автоматических настроек сетевого адаптера, полученных от сервера DHCP. Если имя адаптера не указано, то отмена настроек выполняется для всех адаптеров;

**/release6[Adapter]** – отмена автоматических настроек для протокола IPv6;

**/renew[Adapter]** – обновить конфигурацию для сетевого адаптера, настроенного на получение настроек от сервера DHCP. Если имя адаптера не указано, то обновление выполняется для всех адаптеров;

**/renew6[Adapter]** – как и в предыдущем случае, но для протокола IPv6;

**/showclassid Adapter** и **/setclassid Adapter [ClassID]** – эти параметры применимы для Windows Vista / Windows 7 и используются для просмотра или изменения идентификатора Class ID, если он получен от DHCP-сервера при конфигурировании сетевых настроек.

#### *Примеры использования:*

**ipconfig** – отобразить базовые сетевые настройки для всех сетевых адаптеров;

**ipconfig /all** – отобразить все сетевые настройки для всех сетевых адаптеров;

**ipconfig /renew «LAN-2»** – обновить сетевые настройки, полученные от DHCP-сервера, только для адаптера с именем «LAN-2»;

**ipconfig /displaydns** – вывести на экран содержимое кэш-службы разрешения имен DNS.

## 2 Утилита ARP.EXE.

Утилита ARP позволяет просматривать и изменять записи в кэш ARP (Address Resolution Protocol – протокол разрешения адресов), который представляет собой таблицу соответствия IP-адресов аппаратным адресам сетевых устройств.

*Синтаксис ARP.EXE:*

**Arp** [-a [InetAddr] [-NIfaceAddr]] [-g [InetAddr] [-NIfaceAddr]] [-d InetAddr [IfaceAddr]] [-s InetAddr EtherAddr [IfaceAddr]];

**-a [ InetAddr] [ -NIfaceAddr]** – ключ -a, отображает текущую таблицу ARP для всех интерфейсов. Для отображения записи конкретного IP-адреса используется ключ -a с параметром InetAdd – IP-адрес;

**-g [ InetAddr] [ -NIfaceAddr]** – ключ -g идентичен ключу -a;

**-d InetAddr [ IfaceAddr]** – используется для удаления записей из ARP-кэш. Возможно удаление по выбранному IP или полная очистка ARP-кэш. Для удаления всех записей, вместо адреса используется символ «\*». Можно выполнить для конкретного интерфейса, указав в поле IfaceAddr его IP-адрес;

**-s InetAddr EtherAddr [ IfaceAddr]** – используется для добавления статических записей в таблицу ARP. Статические записи хранятся в ARP-кэш постоянно. Обычно добавление статических записей используется для сетевых устройств, не поддерживающих протокол ARP или не имеющих возможности ответить на ARP-запрос;

**/?** – получение справки по использованию arp.exe.

### **Примеры использования ARP:**

**arp -a** – отобразить все записи таблицы ARP;

**arp -a 192.168.0.9** – отобразить запись, соответствующую IP-адресу 192.168.0.9;

**arp -a 192.168.1.158 -N 192.168.1.1** – отобразить таблицу ARP для адреса 192.168.1.158 на сетевом интерфейсе 192.168.1.1;

**arp -a -N 10.164.250.148** – отобразить все записи таблицы ARP на сетевом интерфейсе 10.164.250.148;

**arp -s 192.168.0.1 00-22-15-15-88-15** – добавить в таблицу ARP статическую запись, задающую соответствие IP-адреса 192.168.0.1 и MAC-адреса 00-22-15-15-88-15;

**arp -d 192.168.1.1 192.168.1.56** – удаление записи из таблицы ARP для IP-адреса 192.168.1.1 на сетевом интерфейсе 192.168.1.56;

**arp -d \*** – полная очистка таблицы ARP. Аналогично – **arp -d** без параметров. Если имеется несколько сетевых интерфейсов, то очистка может быть выполнена только для одного из них - **arp -d \* 192.168.0.56**.

## 3 Утилита PING.

Утилита **ping** чаще всего используется для определения достижимости заданного адреса.

*Принцип работы:* посылает адресату пакет заданного размера, который при приеме получателем посылается обратно. Программа проверяет возможность доставки пакета, показывает время между отправкой и приемом пакета, а также

минимальное, среднее и максимальное время доставки пакета, тем самым позволяет оценить скорость передачи и определить среднюю пропускную способность линии связи (рисунок 10.1).

```

Обработчик команд Windows
Microsoft Windows [Version 6.1.7601]
(c) Корпорация Майкрософт (Microsoft Corp.), 2009. Все права защищены.
C:\Windows\System32>ping bru.by

Обмен пакетами с bru.by [82.209.221.3] с 32 байтами данных:
Ответ от 82.209.221.3: число байт=32 время=12мс TTL=120
Ответ от 82.209.221.3: число байт=32 время=11мс TTL=120
Ответ от 82.209.221.3: число байт=32 время=10мс TTL=120
Ответ от 82.209.221.3: число байт=32 время=11мс TTL=120

Статистика Ping для 82.209.221.3:
    Пакетов: отправлено = 4, получено = 4, потеряно = 0
    (0% потерь)
Приблизительное время приема-передачи в мс:
    Минимальное = 10мсек, Максимальное = 12 мсек, Среднее = 11 мсек
  
```

Рисунок 10.1 – Определения достижимости узла bru.by командой ping

*Синтаксис:*

**ping** [-t] [-a] [-n число] [-l размер] [-f] [-i TTL] [-v TOS] [-r число] [-s число] [[-j список Узлов] | [-k список Узлов]] [-w таймаут] конечное Имя.

*Параметры:*

**-t** – отправка пакетов на указанный узел до команды прерывания. Для прекращения нажимается <Ctrl>+<C>;

**-a** – определение адресов по именам узлов;

**-n** – число отправляемых запросов;

**-l** – размер буфера отправки;

**-f** – установка флага, запрещающего фрагментацию пакета;

**-i** –TTL – задание срока жизни пакета (поле «Time To Live»);

**-v** –TOS – задание типа службы (поле «Type Of Service»);

**-r** – запись маршрута для указанного числа переходов;

**-s** – штамп времени для указанного числа переходов;

**-j** – свободный выбор маршрута по списку узлов;

**-k** – жесткий выбор маршрута по списку узлов;

**-w** – тайм-аут каждого ответа в миллисекундах.

4 Утилита TRACERT.

Утилита **tracert** позволяет определить маршрут прохождения пакета TCP/IP через маршрутизаторы и шлюзы. Программа измеряет и показывает время между отправкой пакета и получением ответа.

*Синтаксис:*

**tracert** [-d] [-h максЧисло] [-j списокУзлов] [-w интервал] имя.

*Параметры:*

**-d** – без разрешения в имена узлов;

**-h** – максимальное число прыжков при поиске узла;

- j – свободный выбор маршрута по списку узлов;
- w – интервал ожидания каждого ответа в миллисекундах.

На рисунке 10.2 представлено тестирование маршрута прохождения к узлу google.by.

```
C:\Windows\System32>tracert google.com

Трассировка маршрута к google.com [216.58.209.14]
с максимальным числом прыжков 30:

  1  <1 ms    <1 ms    <1 ms    192.168.100.1
  2   3 ms     2 ms     2 ms     100.83.0.1
  3   3 ms     3 ms     3 ms     93.84.80.153
  4   5 ms     2 ms     2 ms     10.0.61.69
  5  16 ms    15 ms    15 ms    core1.net.belpak.by [93.85.253.193]
  6  11 ms     7 ms     7 ms     ie2.net.belpak.by [93.85.80.42]
  7   7 ms     7 ms     7 ms     asbr9.net.belpak.by [93.85.80.242]
  8  24 ms    24 ms    23 ms    74.125.146.96
  9  20 ms     *        *        108.170.250.209
 10  23 ms    19 ms    19 ms    172.253.68.31
 11  18 ms    17 ms    18 ms    waw02s18-in-f14.1e100.net [216.58.209.14]

Трассировка завершена.
```

Рисунок 10.2 – Определение и тестирование маршрута с помощью команды tracert

### ***Практическое выполнение задания***

1 Используя утилиту PING определить пропускную способность сети до адресов 10.7.0.120, 10.219.0.1, 10.239.1.1 и 10.7.15.15. Объясните разницу в результатах.

2 С помощью утилиты IPCONFIG получите текущие настройки протокола TCP/IP компьютера в классе. Объясните полученный результат.

3 Передайте пакеты участникам сети напрямую и через шлюз. Объясните полученные записи в таблице ARP.

4 Используя утилиту TRACERT и IP-адреса, постройте схему фрагмента сети университета, обозначив шлюзы и узлы.

Список IP-адресов: 10.203.0.175, 10.203.22.130, 10.202.213.107, 10.202.213.150, 10.1.0.1, 10.239.0.100, 10.239.0.1, 10.7.200.10

### ***Контрольные вопросы***

- 1 Назначение и применение утилиты PING с различными ключами?
- 2 Как с помощью утилиты PING оценить пропускную способность сети?
- 3 Какие параметры протокола TCP/IP выводит утилита IPCONFIG?
- 4 Зачем нужна команда и таблица ARP?
- 5 Объясните разницу во времени между обращениями к одному и тому же хосту по имени и IP-адресу.
- 6 Как с помощью команды TRACERT определить маршрут к узлу?

## 11 Лабораторная работа № 11. Изучение текстовых протоколов высших уровней модели OSI

**Цель работы:** ознакомиться с принципами работы текстовых протоколов высших уровней на примере протоколов Telnet, FTP и HTTP.

### *Основные теоретические положения*

Большинство протоколов высших уровней **текстовые** – запросы и ответы передаются в виде текста, т. е. в запросах и ответах могут присутствовать только печатные символы.

Во многих протоколах ответы начинаются со специальной строки, состоящей из трехзначного числа и, возможно, текстового описания типа ответа. Трехзначное число разделяется на две части: первый символ рассматривается как код класса сообщений; два последних – как тип сообщения данной важности.

Коды классов следующие.

1 **Информационное сообщение.** Обычно игнорируется программными клиентами.

2 **Удачное завершение запроса.** Рассматривается программами-клиентами как успех обработки запроса и обычно игнорируется.

Часто программы-серверы не различают сообщения первого и второго типа, т. е. информационное сообщение проходит по второй категории.

3 **Сообщение об удачной обработке запроса, но требующее дополнительных действий клиента.**

4 **Ошибка со стороны клиента,** т. е. клиент послал запрос, который не может обработать сервер вследствие ошибочности или недостаточности данных.

5 **Ошибка со стороны сервера.** Клиент послал правильный запрос, но сервер не смог его выполнить в силу каких-то причин.

Трехзначные коды ответов очень удобны для программного распознавания, нет необходимости распознавать текст ответа, который, в общем случае, может прийти на разных языках, достаточно распознать только три цифры.

Для работы с текстовыми протоколами используют программу TELNET, входящую в состав многих операционных систем. Эта программа предназначена для работы с протоколом TELNET, задачей которого является обмен информацией между клиентом и сервером без каких-либо преобразований, т. е. организация прозрачного канала между клиентом и сервером.

### **Протокол TELNET.**

Протокол TELNET (от слов telecommunication network – телекоммуникационная сеть) обеспечивает возможность входа в удаленную систему. Он позволяет пользователю одного компьютера зарегистрироваться на удаленном компьютере, расположенном в другой части сети. При этом пользователю кажется, что он работает за терминалом удаленного компьютера.

Работу TELNET обеспечивает специальная программа (сервер), запущенная на компьютере, к которому вы подключаетесь, и обрабатывающая поступающие запросы. На вашем компьютере выполняется программа TELNET, которая

обращается к серверу. В процессе установления соединения компьютеры договариваются о режиме эмуляции терминала в данном сеансе работы.

Для начала сеанса работы TELNET необходимо ввести доменное имя или IP-адрес удаленного компьютера. После установления соединения удаленная система обычно запрашивает имя пользователя и пароль, хотя это зависит от типа операционной системы и программного обеспечения TELNET, установленных на удаленном компьютере.

После установления соединения ваш компьютер играет роль терминала удаленной машины. Все вводимые вами команды выполняются на удаленном компьютере. TELNET лишь осуществляет обработку и передачу ваших команд на удаленный компьютер. TELNET автоматически заканчивает свою работу, когда вы выходите из удаленной системы.

*Синтаксис* команды TELNET следующий:

***TELNET адрес\_сервера [порт]***

Если порт не указан, используется 23-стандартный порт TELNET.

### **Протокол FTP.**

В отличие от TELNET, протокол FTP (File Transfer Protocol) предназначен не для работы на удаленном компьютере, а для передачи файлов между подключенными к сети компьютерами. Так же, как и TELNET, сервис FTP основан на совместном использовании двух программ – программы-сервера, которая выполняется постоянно в фоновом режиме на удаленном компьютере, и программы-клиента, которую вы должны запустить на своем компьютере, чтобы начать сеанс работы по протоколу FTP. Программа-сервер занимается обработкой всех запросов, приходящих к ней от программы-клиент, поэтому если программа-сервер не предоставляет каких-либо возможностей вроде докачки и т. д., то каким бы навороченным клиентом вы ни пользовались все равно данные возможности так и останутся недоступными для вас. Протокол FTP позволяет передавать файлы как в текстовом, так и в двоичном формате между совершенно различными платформами.

### **Протокол HTTP.**

Протокол HTTP (Hyper Text Transfer Protocol – «протокол передачи гипертекста») изначально использовался для передачи гипертекстовых документов в формате HTML. В настоящий момент используется для передачи произвольных данных, т. е. данных разного представления (текст, изображения, видео, звук), в сети Интернет для получения информации с веб-сайтов. Обычно этот протокол работает на 80-м порту.

HTTP используется также в качестве «транспорта» для других протоколов прикладного уровня, таких как SOAP, XML-RPC, WebDAV.

### ***Структура протокола HTTP***

Каждое HTTP-сообщение состоит из трёх частей, которые передаются в указанном порядке.

1 Стартовая строка (Starting line) – определяет тип сообщения.

2 Заголовки (Headers) – характеризуют тело сообщения, параметры передачи и прочие сведения.

3 Тело сообщения (Message Body) – непосредственно данные сообщения. Обязательно должно отделяться от заголовков пустой строкой.

Стартовая строка и заголовок являются обязательными элементами. Для версии протокола 1.1 в запросе обязательно должен быть заголовок Host.

Стартовые строки различаются для запроса и ответа. Строка запроса выглядит так:

***GET URI – для версии протокола 0.9;  
Метод URI HTTP/Версия – для остальных версий,***

где Метод (Method) – тип запроса, одно слово заглавными буквами. В версии HTTP 0.9 использовался только метод GET, для версии HTTP 1.1 существует девять обязательных методов;

URI – определяет путь к запрашиваемому документу;

Версия (Version) – пара разделённых точкой цифр. Например: 1.0.

Чтобы запросить страницу Википедии, клиент должен передать строку (задан всего один заголовок):

***GET /wiki/HTTP HTTP/1.0  
Host: ru.wikipedia.org***

### ***Практическое выполнение задания***

#### **Протокол FTP.**

Протокол FTP, как уже указывалось выше, служит для обмена файлами. Это клиент-серверный протокол. Чтобы войти на сервер FTP можно использовать TELNET, встроенную программу ftp клиента, или браузер.

Войдем на FTP-сервер через браузер. Открываем браузер и в адресной строке указываем: ***ftp://10.203.0.201***.

Откроется окно для ввода логина и пароля (логин и пароль узнайте у своего преподавателя). Вводим требуемые данные (рисунок 11.1).



Рисунок 11.1 – Вход на сервер FTR через браузер

Вошли. Откроется стандартное окно проводника Windows (см. рисунок 11.1). Теперь зайдем на FTP-сервер через командную строку, используя ftp клиента.

***ftp 10.203.0.201***

Залогинемся – введем имя пользователя и пароль (*пароль при вводе остается невидимым*).

При удачном входе сервер ответит строкой «230 userftp1 logged in» (рисунок 11.2).

```

C:\WINDOWS\system32\cmd.exe - ftp 10.203.0.201
Microsoft Windows [Version 10.0.15063]
(c) Корпорация Майкрософт (Microsoft Corporation), 2017. Все права защищены.

D:\Users\aso1141>ftp 10.203.0.201
Связь с 10.203.0.201.
220 Microsoft FTP Service
530 Please login with USER and PASS.
Пользователь (10.203.0.201:(none)): userftp1
331 Password required for userftp1.
Пароль:
230 User userftp1 logged in.
ftp>
  
```

Рисунок 11.2 – Вход на сервер FTP через командную строку, используя ftp клиента

Введем команду LS либо DIR, чтобы отразить список файлов и папок в удаленном каталоге.

Получим файл MyDoc.txt, установив перед этим режим передачи файлов в двоичном формате с помощью команды BINARY (рисунок 11.3):

*binary*

«Ответ сервера на команду» **200 Type set to I  
get MyDoc.txt**

```

C:\WINDOWS\system32\cmd.exe - ftp
220-Microsoft FTP Service
220 FTP-Server-ASU
ftp> binary
200 Type set to I.
ftp> get MyDoc.txt
200 PORT command successful.
150 Opening BINARY mode data connection for MyDoc.txt(50 bytes).
226 Transfer complete.
ftp: 50 байт получено за 0.00 (сек) со скоростью 50000.00 (КБ/сек).
ftp>
  
```

Рисунок 11.3 – Скачивание файла MyDoc.txt с сервера FTP

По умолчанию файл скопирован в папку C:\DocumentsandSettings\%user% на компьютер клиента.

*Все действия фиксировать с помощью скриншотов и предоставить в отчете.*

### **Контрольные вопросы**

- 1 Почему протоколы FTP, TELNET, HTTP называются протоколами высших уровней?
- 2 Что такое TELNET и для чего он предназначен?
- 3 Назначение протокола FTP?
- 4 Назовите основные команды протокола FTP.
- 5 В каких случаях применяется протокол HTTP?
- 6 Опишите структуру протокола HTTP.
- 7 Приведите команды стартового запроса к серверу HTTP.

## 12 Лабораторная работа № 12. Изучение протоколов электронной почты

**Цель работы:** изучить принципы организации взаимодействия электронной почты с помощью протоколов SMTP и POP3.

### *Основные теоретические положения*

Существует несколько типов служб электронной почты, базирующихся на различных протоколах обмена: X.400, UUCP, SMTP, POP3 и др.

На рисунке 12.1 представлена упрощенная схема отправки и получения электронного письма.

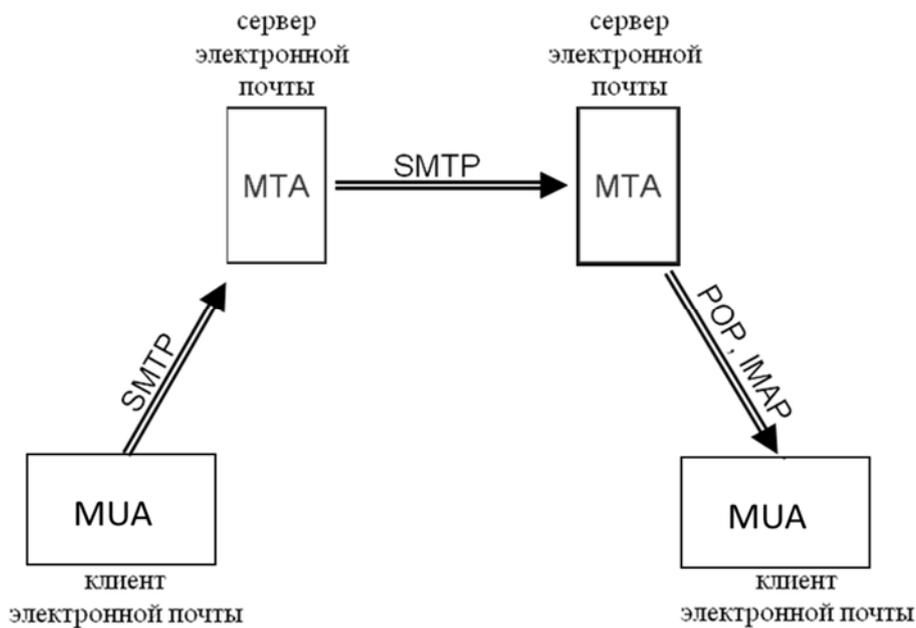


Рисунок 12.1 – Схема взаимодействия с прикладными почтовыми протоколами

Для пересылки электронных писем используются специальные сервера электронной почты, которые называются агентами пересылки почтовых сообщений (mail transfer agent, MTA). Электронное письмо может пройти через множество транзитных серверов, прежде чем достигнет сервера, к которому подключен абонент назначения. Пользователь для отправки и приема писем использует почтовый пользовательский агент – MUA (mail user agent). Алгоритм работы MTA и MUA определен стандартами ITU (X.400), UNIX (UUCP) и Интернет (RFC).

В настоящее время в основном используют протоколы стандартов Интернет: RFC821 и RFC1939.

Стандарт RFC821 описывает протокол SMTP (Simple Mail Transfer Protocol – простой протокол передачи почты), а именно как MUA общается с MTA и как общаются друг с другом MTA-агенты по TCP-соединению при передаче почты, а RFC1939 описывает протокол POP3 (Post Office Protocol – протокол почтового отделения), то есть процесс доставки и команды взаимодействия с сервером POP. Протоколы SMTP и POP3 разработаны давно,

но активно используются и в настоящее время. Эти протоколы используют в запросах текстовые команды, а в ответах присутствуют специальные строки, состоящие из трехзначного числа и, возможно, текстового описания.

Трехзначное число разделяется на две части: первый символ рассматривается как код класса сообщения, два последних – как тип сообщения данной важности.

Коды классов следующие.

- 1 *Информационное сообщение.*
- 2 *Удачное завершение запроса.*
- 3 *Сообщение требующее дополнительных действий клиента.*
- 4 *Ошибка со стороны клиента.*
- 5 *Ошибка со стороны сервера.*

Для изучения протоколов SMTP и POP3 можно воспользоваться программой TELNET.

### *Практическое выполнение задания*

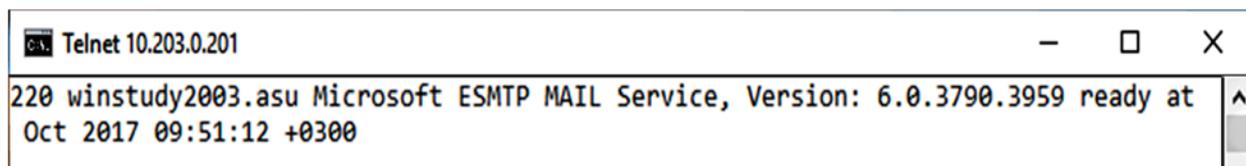
Рассмотрим работу протокола SMTP. Подключимся к тестовому серверу SMTP с помощью команды telnet: **TELNET адрес\_сервера [порт]**.

Протокол SMTP использует по умолчанию 25-й порт.

В командной строке даем команду на подключение:

**telnet 10.203.0.201 25**

Получаем ответ (рисунок 12.2).



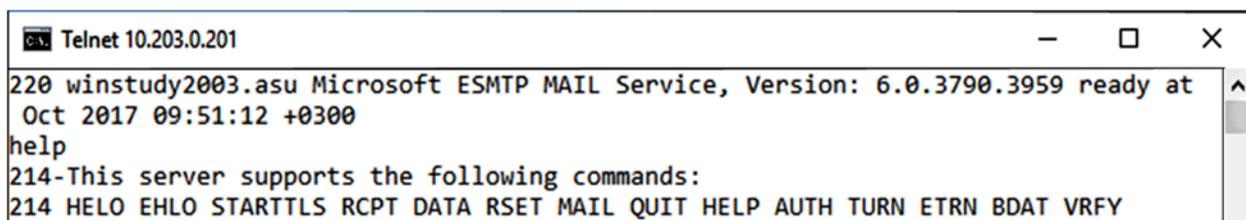
```
Telnet 10.203.0.201
220 winstudy2003.asu Microsoft ESMTMP MAIL Service, Version: 6.0.3790.3959 ready at
Oct 2017 09:51:12 +0300
```

Рисунок 12.2 – Подключение к серверу SMTP с помощью команды TELNET

Обратите внимание на число 220 в начале строки ответа. Это «положительный» ответ, сервер ответил на наш запрос на подключение.

Многие серверы поддерживают команду help. Введем команду **help**.

Сервер покажет список поддерживаемых команд (рисунок 12.3).



```
Telnet 10.203.0.201
220 winstudy2003.asu Microsoft ESMTMP MAIL Service, Version: 6.0.3790.3959 ready at
Oct 2017 09:51:12 +0300
help
214-This server supports the following commands:
214 HELO EHLO STARTTLS RCPT DATA RSET MAIL QUIT HELP AUTH TURN ETRN BDAT VRFY
```

Рисунок 12.3 – Ответ сервера на команду help

*Список команд*, которые поддерживает данный сервер:

HELO – открывает приглашение от клиента;

EHLO (устаревшая HELO) – открывает приглашение от клиента;

STARTTLS – позволяет создать зашифрованное соединение;

MAIL – определяет отправителя сообщения;  
 RCPT – определяет получателей сообщения;  
 DATA – определяет начало сообщения;  
 RSET – сброс SMTP-соединения;  
 ETRN – борьба с проблемами безопасности команды TURN;  
 VRFY – проверяет имя пользователя системы;  
 HELP – запрашивает список команд;  
 QUIT – остановить сеанс SMTP;  
 TURN – реверс ролей в SMTP (клиент становится сервером);  
 AUTH – показывает серверу механизм аутентификации.

Попробуем написать письмо. На почтовом сервере с помощью преподавателя создается тестовый пользователь.

Перед работой с почтовым сервером нужно «поздороваться» с ним при помощи команды `helo` указав свой IP-адрес.

В данном примере команда выглядит следующим образом:

***helo 10.203.0.161***

Укажем отправителя письма:

***mail from: testuser@study130.asu***

Укажем получателя письма:

***rcpt to: usertest@study130.asu***

Перейдем в режим ввода письма – пошлем команду `data`:

***data***

Ответ сервера – ***354 Start Mail input, end with <CRLF>.<CRLF>***

Обратите внимание на код ответа 354. Это нормальное завершение, но требуются дополнительные данные – само письмо, которое, как видно, должно заканчиваться строкой, состоящей из одной точки.

Создадим само письмо. Формат письма описан стандартами RFC822. Рассмотрим наиболее важные служебные строки:

*Дата создания по GMT и часовой пояс:*

***Date: Tue, 22 Nov 2017 12:55:07 +0200***

***From: User testuser@study130.asu – От кого***

***Reply-To: User user1@home.my – Кому отвечать***

***To: user2@home.my – Кому***

***Subject: Test – Тема письма***

***MIME-Version: 1.0***

***Content-Type: text/plain; charset=us-ascii***

***Content-Transfer-Encoding: 7bit***

Последние две строки – это информация программе почтовому клиенту, т. е. шрифт текста письма, (как закодировано письмо). С помощью этих строк почтовая программа-клиент сможет реализовать шестой уровень – представить информацию пользователю в читабельном виде.

Вводим текст сообщения и не забываем про точку в конце (рисунок 12.4).

***Subject: HELLO!***

***Its ASOI test message!***

```

Telnet 10.203.0.201
Oct 2017 14:36:35 +0300
helo 10.203.0.161250 winstudy2003.asu Hello [10.203.0.161]
mail from: testuser@asu
250 2.1.0 testuser@asu...Sender OK
rcpt to: usertest@asu
250 2.1.5 usertest@asu
data
354 Start mail input; end with <CRLF>.<CRLF>
subject: hello!
Its ASOI test message!
.
250 2.6.0 <WINSTUDY2003MxjhZey00000002@winstudy2003.asu> Queued mail for delivery

```

Рисунок 12.4 – Ввод заголовка, текста письма и его отправка

Письмо принято!

### *Контрольные вопросы*

- 1 Какие протоколы электронной почты относятся к стандартам Интернета?
- 2 Опишите процесс передачи письма на основе модели с MTA и MUA.
- 3 Что означают аббревиатуры SMTP и POP3?
- 4 Опишите синтаксис команды TELNET?
- 5 Какой порт используется для протокола SMTP?
- 6 Какую команду вводят при подключении к серверу SMTP?
- 7 Назовите основные команды заголовка письма при взаимодействии с сервером.
- 8 Назовите команду, после которой отправляется «тело» письма (текст).
- 9 Назовите команду окончания письма.

### Список литературы

- 1 **Олифер, В. Г.** Компьютерные сети. Принципы, технологии, протоколы: учебное пособие / В. Г. Олифер, Н. А. Олифер. – 5-е изд. – Санкт-Петербург: Питер, 2016. – 992 с. ил.
- 2 **Новиков, В. А.** Информационные системы и сети: учебное пособие / В. А. Новиков, А. В. Новиков, В. В. Матвеев. – Минск: Изд-во Гревцова, 2014. – 448 с.
- 3 **Бройдо, В. Л.** Архитектура ЭВМ и систем: учебник для вузов / В. Л. Бройдо, О. П. Ильина. – Санкт-Петербург: Питер, 2009. – 720 с.
- 4 **Пескова, С. А.** Сети и телекоммуникации: учебное пособие / С. А. Пескова, А. В. Кузин, А. Н. Волков. – 2-е изд., стер. – Москва: Академия, 2007. – 352 с.
- 5 **Бройдо, В. Л.** Вычислительные системы, сети и телекоммуникации: учебное пособие / В. Л. Бройдо, О. П. Ильина. – 4-е изд. – Санкт-Петербург: Питер, 2011. – 560 с.
- 6 **Чекмарев, А. Н.** Windows Server 2008. Настольная книга администратора / А. Н. Чекмарев. – Санкт-Петербург: БХВ-Петербург, 2013. – 512 с.