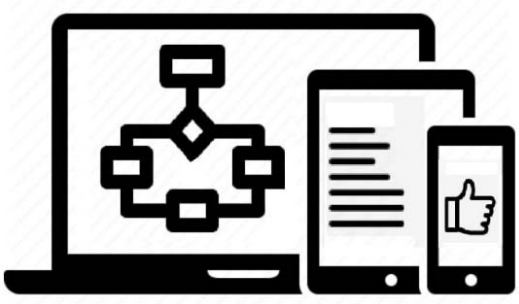
МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

СЕТЕВЫЕ ТЕХНОЛОГИИ

Методические рекомендации к лабораторным работам для студентов направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» очной формы обучения



Могилев 2024

УДК 004.71
ББК 32.973-018.2
С33

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «30» августа 2024 г., протокол № 1

Составители: канд. техн. наук, доц. А. В. Кушнер;
канд. техн. наук, доц. В. В. Кутузов

Рецензент канд. техн. наук, доц. И. В. Лесковец

В методических рекомендациях кратко изложены теоретические сведения, необходимые для выполнения лабораторных работ. Рекомендации составлены в соответствии с рабочей программой по дисциплине «Сетевые технологии» для направлений подготовки 09.03.01 «Информатика и вычислительная техника» и 09.03.04 «Программная инженерия» очной формы обучения.

Учебное издание

СЕТЕВЫЕ ТЕХНОЛОГИИ

Ответственный за выпуск	В. В. Кутузов
Корректор	И. В. Голубцова
Компьютерная верстка	М. М. Дударева

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 16 экз. Заказ №

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, г. Могилев.

Содержание

Введение

Целью преподавания данной учебной дисциплины является предоставление студентам фундаментальные знаний и практических навыков, необходимых для понимания, разработки, управления и поддержки компьютерных сетей.

В соответствии с учебной программой дисциплины студент:

изучит:

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

– основные функции и режимы взаимодействия компьютеров, аппаратное и программное обеспечение 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;

– перспективные направления развития компьютерных сетей и сетевых технологий, методы использования сетей и сетевых технологий в профессиональной деятельности;

научится:

– анализировать уровень эффективности сетевых решений;

– эффективно использовать операционные системы и предлагать сетевые решения для разрабатываемых прикладных задач;

– использовать различные протоколы при разработке программных средств;

овладеет:

– методами разработки и обоснования конфигурации сети, оценки трафика в сегментах, выбора сетевого оборудования и программного обеспечения;

– техникой конфигурирования локальных сетей, реализации сетевых протоколов с помощью программных средств;

– методиками постановки и решения задачи проектирования или модернизации локальной, корпоративной вычислительной сети;

– навыками работы с информацией в локальных и глобальных компьютерных сетях.

Целью методических рекомендаций является формирование умений и навыков разработки, управления и поддержки компьютерных сетей.

Результатом выполнения лабораторной работы является отчет в виде электронного документа, представляющего собой скрины выполнения задания по компьютерным сетям на персональном компьютере.

Студент, выполнивший лабораторную работу, должен предъявить файл с выполненным заданием преподавателю для оценки в конце каждого занятия. Качество выполнения задания оценивается в баллах модульно-рейтинговой системы.

1 Проектирование локальной сети

Цель работы: изучить основные виды, преимущества и недостатки сетевых топологий, их наиболее распространенные типы сетей, виды и методы доступа к среде передачи данных, сетевые архитектуры.

1.1 Краткие теоретические сведения

При организации компьютерной сети исключительно важным является выбор *топологии,* т. е. *компоновки сетевых устройств и кабельной инфраструктуры.* Нужно выбрать такую топологию, которая обеспечила бы надежную и эффективную работу сети, удобное управление потоками сетевых данных. В топологии **Шина (Bus)** все компьютеры соединяются друг с другом *одним кабелем.* Посланные в такую сеть данные передаются всем компьютерам, но обрабатывает их только тот компьютер, аппаратный адрес сетевого адаптера которого записан в кадре как адрес получателя. В топологии **Кольцо (Ring)** каждый из компьютеров соединяется с двумя другими так, чтобы от одного он получал информацию, а второму – передавал ее. Последний компьютер подключается к первому и кольцо *замыкается.* В конфигурации **Звезда (Active Star)** все потоки данных идут исключительно через центральный компьютер; он же полностью отвечает за управление информационным обменом между всеми участниками сети.

Звезда – Шина (Star Bus), **или «пассивная звезда»**.

Здесь периферийные компьютеры подключаются не к центральному компьютеру, а к пассивному *концентратору*, или *хабу* (*hub*). Последний, в отличие от центрального компьютера, никак не отвечает за управление обменом данными, а выполняет те же функции, что и повторитель, т. е. восстанавливает приходящие сигналы и пересылает их всем остальным подключенным к нему компьютерам и устройствам. Именно поэтому данная топология, хотя физически и выглядит как «звезда», логически является топологией «шина».

Однако особо следует выделить топологию **Дерево (tree)**, которую можно рассматривать как объединение нескольких «звезд». Именно эта топология сегодня является наиболее популярной при построении локальных сетей. Наконец, следует упомянуть о **сетчатой**, или **сеточной (mesh)** топологии, в которой все либо многие компьютеры и другие устройства соединены друг с другом напрямую.

Чтобы компьютеры могли взаимодействовать, необходима какая-либо среда, обеспечивающая возможность передачи сигналов на физическом уровне.

Наиболее часто в компьютерных сетях применяются кабельные соединения, выступающие в качестве среды передачи электрических или оптических сигналов между компьютерами и другими сетевыми устройствами. При этом используются следующие типы кабеля:

– коаксиальный кабель (coaxial cable);

– витая пара (twisted pair): неэкранированная (unshielded, UTP) и экранированная (shielded);

– волоконно-оптический, или оптоволоконный, кабель (fiber optic).

1.2 Порядок выполнения работы

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

2 Получить задание у преподавателя, выполнить типовые задания.

3 Сделать выводы по результатам исследований.

4 Оформить отчет.

Контрольные вопросы

1 В чем заключается различие между физическими и логическими связями?

2 Каковы преимущества и недостатки конфигурации «звезда»? В каких локальных сетях она применяется?

3 Каковы преимущества и недостатки топологии «кольцо»? В каких локальных сетях она применяется?

4 Каковы преимущества и недостатки конфигурации «шина»? В каких локальных сетях она применяется?

5 Какие гибридные топологии вам известны?

6 Какие факторы необходимо учитывать при планировании сети?

7 К какой категории относится кабель из неэкранированной витой пары, способный передавать данные со скоростью до 10 Мбит/с?

8 Какие именно проводники используются в коаксиальном кабеле?

9 Зачем в кабеле «витая пара» используется несколько пар проводников?

10 Какой разъем используется для подключения кабеля «витая пара» к компьютерам?

11 Какие вы знаете разновидности архитектуры Ethernet? Чем они различаются?

2 Изучение сетевого уровня модели OSI на примере протокола IP

Цель работы: изучить структуру модели OSI и ее отличие от модели TCP/IP; ознакомиться с функциями основных протоколов; на практике изучить инкапсуляцию данных.

2.1 Краткие теоретические сведения

Стек протоколов Transmission Control Protocol/Intemet Protocol (TCP/IP) – это набор взаимодействующих протоколов разных уровней. Каждый уровень взаимодействует с соседним и производит обмен данными в сети, т. е. протокол – это набор правил, согласно которым происходит обмен данными. Протоколы разделены по уровням функциональности, что делает работу

сетевого оборудования и программного обеспечения гораздо проще и прозрачнее, а также позволяет выполнять определенные задачи.

Для разделения набора протоколов по уровням была разработана модель OSI (Open Systems Interconnection Basic Reference Model, 1978 г.), которая является базовой эталонной моделью взаимодействия открытых систем и представляет собой обобщенные стандарты для разработчиков программ. При помощи данных стандартов любой компьютер может расшифровать информацию, полученную с другого компьютера.

Сетевой протокол – это правила и технические процедуры, позволяющие компьютерам одной сети осуществлять соединение и обмен данными. Группа протоколов, объединенных общей конечной целью, называется «стек протоколов». Модель OSI имеет семь уровней.

Уровень 7. Уровень приложений (Application Layer).

Уровень приложений является ближайшим к пользователю и предоставляет службы приложениям внутри уровня. От других уровней он отличается тем, что не предоставляет служб другим уровням; вместо этого он предоставляет службы только приложениям, которые находятся вне рамок эталонной модели OSI. Примерами таких приложений могут служить электронные таблицы (например, программа Excel) или текстовые процессоры (например, программа Word). Уровень приложений определяет доступность партнеров по сеансу связи друг для друга, а также синхронизирует связь и устанавливает соглашение о процедурах восстановления данных в случае ошибок и процедурах контроля целостности данных. Примерами приложений седьмого уровня могут служить протоколы Telnet и HTTP.

Уровень 6. Уровень представления данных (Presentation Layer).

Задача уровня представления данных состоит в том, чтобы информация уровня приложений, которую посылает отправитель, могла быть прочитана уровнем приложений получателя. При необходимости уровень представления преобразует данные в один из многочисленных существующих форматов, который поддерживается обеими системами. Другая важная задача этого уровня – шифрование и расшифровывание данных. Типовыми графическими стандартами шестого уровня являются PICT, TIFF и JPEG, примерами стандартов шестого уровня эталонной модели, описывающих формат представления звука и видео, – MIDI и MPEG.

Уровень 5. Сеансовый уровень (Session Layer).

Как показывает само название, сеансовый уровень устанавливает сеанс связи между двумя рабочими станциями, управляет им и разрывает его. Сеансовый уровень предоставляет свои службы уровню представления данных. Он также синхронизирует диалог между уровнями представления двух систем и управляет обменом данными. Кроме своей основной постоянной функции управления, уровень сеанса связи обеспечивает эффективную передачу данных, требуемый класс обслуживания и рассылку экстренных сообщений о наличии проблем на сеансовом уровне, уровне представления данных или уровне приложений.

Уровень 4. Транспортный уровень (Transport Layer).

Транспортный уровень сегментирует данные передающей станции и вновь собирает их в одно целое на принимающей стороне. Границу между транспортным уровнем и уровнем сеанса связи можно рассматривать как границу между протоколами приложений и протоколами передачи данных. В то время как уровни приложений, представления данных и сеанса связи отвечают за аспекты коммуникаций, которые связаны с работой приложений, нижние четыре уровня «решают» вопросы транспортировки данных по сети. Транспортный уровень обеспечивает службу передачи данных таким образом, чтобы скрыть от верхних уровней детали процесса передачи. В частности, задачей уровня является обеспечение надежности передачи данных между двумя рабочими станциями.

Для выполнения службы связи транспортный уровень устанавливает, поддерживает и соответствующим образом ликвидирует виртуальные каналы. Выявление ошибок при передаче данных и управление информационными потоками – основные средства обеспечения надежности транспортной службы. К четвертому уровню относятся протокол управления передачей TCP и протокол пользовательских дейтаграмм UDP.

Уровень 3. Сетевой уровень (Network Layer).

Сетевой уровень является комплексным уровнем, обеспечивающим выбор маршрута и соединение между собой двух рабочих станций, которые могут быть расположены в географически удаленных друг от друга сетях. Кроме того, он «решает» вопросы логической адресации. К третьему уровню относится Internet-протокол (IP).

Уровень 2. Канальный уровень (Data Link Layer).

Канальный уровень обеспечивает надежную передачу данных по физическому каналу. При этом он «решает» задачи физической (в противоположность логической) адресации, анализа сетевой топологии, доступа к сети, уведомления об ошибках, упорядоченной доставки фреймов и управления потоками. Примерами протоколов являются Ethernet, ARP.

Уровень 1. Физический уровень (Physical Layer).

Физический уровень определяет электрические, процедурные и функциональные спецификации для активизации, поддержки и отключения физических каналов между конечными системами. Спецификациями физического уровня определяются уровни напряжений, синхронизация изменений напряжения, физическая скорость передачи данных, максимальная дальность передачи, физические соединения и другие аналогичные параметры. К первому уровню относятся протоколы, основанные на принципах кодирования информации.

Инкапсуляция – это процесс передачи данных с верхнего уровня приложений вниз (по стеку протоколов) к физическому уровню для дальнейшей передачи по сетевой физической среде (витая пара, оптическое волокно, Wi-Fi и др.). Причем на каждом уровне различные протоколы добавляют к передающимся данным свою информацию.

Таблица 2.1 – Стек протоколов моделей OSI и TCP/IP

<table>
<tr><th>Уровень модели OSI</th><th>Уровень модели TCP/IP</th><th>Протокол</th><th>Представление информации</th></tr>
<tr><td>Прикладной</td><td rowspan="3">Прикладной</td><td rowspan="3">HTTP, Telnet, SMTP, POP, FTP, DNS</td><td rowspan="3">Данные</td></tr>
<tr><td>Представления</td></tr>
<tr><td>Сеансовый</td></tr>
<tr><td>Транспортный</td><td>Транспортный</td><td>UDP, TCP</td><td>Сегменты</td></tr>
<tr><td>Сетевой</td><td>Сетевой</td><td>IP, ICMP</td><td>Пакеты</td></tr>
<tr><td>Канальный</td><td rowspan="2">Доступа к среде</td><td>ARP</td><td>Кадры</td></tr>
<tr><td>Физический</td><td>Ethernet</td><td>Биты</td></tr>
</table>

TCP (Transmission Control Protocol, протокол управления передачей) – один из основных протоколов передачи данных в сети Интернет, предназначенный для управления передачей данных, отдельными сеансами связи между серверами и клиентами. TCP делит сообщения HTTP на более мелкие части, называемые сегментами. Эти сегменты передаются между вебсервером и клиентскими процессами, запущенными на узле назначения. TCP также отвечает за управление размером и скоростью, с которой происходит обмен сообщениями между сервером и клиентом.

UDP (User Datagram Protocol – протокол пользовательских дейтаграмм) – использует простую модель передачи без обеспечения надежности, упорядочивания или целостности данных. Дейтаграммы могут прийти не по порядку, дублироваться или пропасть. Подразумевается, что проверка ошибок и их исправление либо не нужны, либо должны исполняться в приложении.

Протокол IP (Internet Protocol – межсетевой протокол) – отвечает за прием форматированных сегментов от TCP, инкапсуляцию их в пакеты, присвоение им соответствующих адресов и доставку по наилучшему пути к узлу назначения.

Общий сценарий работы протокола на каком-либо узле сети, принимающем дейтаграмму, следующий:

1) с одного из интерфейсов уровня доступа к среде передачи (например, Ethernet) поступает дейтаграмма;

2) модуль IP анализирует ее заголовок;

3) если пунктом назначения дейтаграммы является данный компьютер:

– если дейтаграмма является фрагментом большей дейтаграммы, ожидаются остальные фрагменты, после чего из них собирается исходная большая дейтаграмма;

– из дейтаграммы извлекаются данные и направляются на обработку одному из протоколов вышележащего уровня;

4) если дейтаграмма не направлена ни на один из IP-адресов данного узла, то дальнейшие действия зависят от того, разрешена или запрещена ретрансляция (forwarding) дейтаграмм:

– если ретрансляция разрешена, то определяются следующий узел

сети, на который должна быть переправлена дейтаграмма для доставки ее по назначению, и интерфейс нижнего уровня, после чего дейтаграмма передается на нижний уровень этому интерфейсу для отправки (при необходимости может быть произведена фрагментация дейтаграммы);

– если же дейтаграмма ошибочна или по каким-либо причинам не может быть доставлена, она уничтожается, при этом, как правило, отправителю дейтаграммы отсылается ICMP-сообщение об ошибке.

При получении данных от вышестоящего уровня для отправки их по сети IP-модуль формирует дейтаграмму с этими данными, в заголовок которой заносятся адреса отправителя и получателя (также полученные от транспортного уровня) и другая информация, после чего выполняются следующие шаги:

– если дейтаграмма предназначена этому же узлу, из нее извлекаются данные и направляются на обработку одному из протоколов транспортного уровня;

– если дейтаграмма не направлена ни на один из IP-адресов данного узла, то определяются следующий узел сети, на который должна быть переправлена дейтаграмма для доставки ее по назначению, и интерфейс нижнего уровня, после чего дейтаграмма передается на нижний уровень этому интерфейсу для отправки (при необходимости может быть произведена фрагментация дейтаграммы);

– если же дейтаграмма ошибочна или по каким-либо причинам не может быть доставлена, она уничтожается.

IP-адрес является уникальным 32-битовым идентификатором.

IP-интерфейса в сети Интернет. IP-адреса принято записывать разбивкой всего адреса по октетам, каждый октет записывается в виде десятичного числа, цифры разделяются точками. Например, адрес 10100000010100010000010110000011 записывается как 10100000.01010001.00000101.10000011 = 160.81.5.131.

IP-адрес состоит из номера IP-сети, который занимает старшую область адреса, и номера устройства в этой сети, занимающего младшую часть.

HTTP (HyperText Transfer Protocol – протокол передачи гипертекста) – протокол прикладного уровня определяет, каким образом взаимодействуют веб-сервер и веб-клиент. HTTP определяет содержание и формат запросов и ответов, которыми обмениваются клиент и сервер. HTTP реализует программное обеспечение и веб-клиента, и веб-сервера как часть приложения. Для управления процессом передачи сообщений между клиентом и сервером HTTP обращается к другим протоколам.

SMTP (Simple Mail Transfer Protocol - простой протокол передачи почты) – это широко используемый сетевой протокол, предназначенный для передачи электронной почты.

DNS (Domain Name System - система доменных имен) – компьютерная распределенная система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), а также информации о маршрутизации почты, обслуживающих узлах для протоколов в домене.

ARP (Address Resolution Protocol – протокол определения адреса) –

протокол в компьютерных сетях, предназначенный для определения MAC-адреса при наличии IP-адреса другого компьютера. Протокол ARP выполняет две основные функции: сопоставление адресов IPv4 и MAC-адресов, сохранение таблицы сопоставлений.

POP (Post Office Protocol – протокол почтового отделения) – стандартный интернет-протокол прикладного уровня, используемый клиентами электронной почты для получения почты с удаленного сервера по TCP-соединению.

Протокол Telnet – обеспечивает передачу потока байтов между процессами, а также между процессом и терминалом. Наиболее часто этот протокол используется для эмуляции терминала удаленной ЭВМ.

FTP (File Transfer Protocol – протокол пересылки файлов) – реализует удаленный доступ к файлу. Для того чтобы обеспечить надежную передачу, FTP использует в качестве транспорта протокол с установлением соединений – TCP. Кроме пересылки файлов, протокол FTP предлагает и другие услуги. Так, пользователю предоставляется возможность интерактивной работы с удаленной машиной, например, он может распечатать содержимое ее каталогов. FTP позволяет пользователю указывать тип и формат запоминаемых данных. Наконец, FTP выполняет аутентификацию пользователей. Прежде чем получить доступ к файлу, в соответствии с протоколом пользователи должны сообщить свое имя и пароль. Приложения, которым не требуются все возможности FTP, могут использовать другой, более экономичный протокол – **TFTP** (Trivial File Transfer Protocol – простейший протокол пересылки файлов). Этот протокол реализует только передачу файлов, причем в качестве транспорта используется более простой, чем TCP, протокол без установления соединения – UDP.

Для обмена служебной и диагностической информацией в сети используется специальный протокол управляющих сообщений **ICMP** (Internet Control Message Protocol). Команда ping позволяет выполнить отправку управляющего сообщения типа Echo Request (тип равен восьми и указывается в заголовке ICMP-сообщения) адресуемому узлу и интерпретировать полученный от него ответ в удобном для анализа виде. В поле данных отправляемого ICMP-пакета обычно содержатся символы английского алфавита. В ответ на такой запрос опрашиваемый узел должен отправить ICMP- пакет с теми же данными, которые были приняты, и типом сообщения EchoReply (код типа в ICMP-заголовке равен нулю). Если при обмене ICMP-сообщениями возникает какая-либо проблема, то утилита ping выведет информацию для ее диагностики.

Одним из примеров протокола сетевого доступа является **Ethernet.** Такие протоколы управления каналами передачи данных принимают пакеты от протокола IP и форматируют их для передачи в среде. Стандарты и протоколы физической передачи данных управляют отправлением сигналов и их интерпретацией клиентами при получении.

На сегодняшний день Ethernet является наиболее широко используемой технологией локальных сетей, функционирующей на канальном и физическом уровнях. Это семейство сетевых технологий, которые регламентируются

стандартами IEEE 802.2 и 802.3. Технология Ethernet поддерживает передачу данных на скоростях 10 Мбит/с, 100 Мбит/с, 1000 Мбит/с (1 Гбит/с), 10 000 Мбит/с (10 Гбит/с), 40 000 Мбит/с (40 Гбит/с), 100 000 Мбит/с (100 Гбит/с). Стандарты Ethernet регламентируют как протоколы второго уровня, так и технологии первого уровня. Для протоколов второго уровня, как и в случае со всеми стандартами группы IEEE 802, технология Ethernet полагается на работу этих двух отдельных подуровней канального уровня, а также на подуровни управления логическим каналом (LLC) и MAC (рисунок 2.1).

Подуровень LLC технологии Ethernet обеспечивает связь между верхними и нижними уровнями. Как правило, это происходит между сетевым программным обеспечением и аппаратным обеспечением устройства. Подуровень LLC использует данные сетевых протоколов, которые обычно представлены в виде пакета IPv4, и добавляет управляющую информацию, чтобы помочь доставить пакет к узлу назначения. LLC используется для связи с верхними уровнями приложений и перемещает пакет для доставки на нижние уровни. LLC реализован в программном обеспечении, и его применение не зависит от оборудования. LLC для компьютера можно рассматривать как программное обеспечение драйвера сетевой платы (NIC).

Драйвер сетевой платы – это программа, которая непосредственно взаимодействует с аппаратными средствами компьютера на сетевой интерфейсной плате для передачи данных между подуровнем MAC и физической средой.

MAC представляет собой более низкий подуровень канального уровня и реализуется на сетевой интерфейсной плате устройства. Спецификации содержатся в стандартах IEEE 802.3.

<table>
<tr><td rowspan="2">Канальный</td><td>Подуровень LLC</td><td rowspan="3">Ethernet</td><td colspan="3">IEEE 802.2</td></tr>
<tr><td>Подуровень MAC</td><td rowspan="2">IEEE 802.3 Ethernet</td><td rowspan="2">IEEE 802.3u FastEthemet</td><td rowspan="2">IEEE 802.3z Gigabit-Ethemet</td></tr>
<tr><td colspan="2">Физический</td></tr>
</table>

Рисунок 2.1 – Подуровни канального уровня

Далее приведено описание основных полей кадра Ethernet, представленных на рисунке 2.2.

Поля «Преамбула» (7 байт) и «Начало разделителя кадра (SFD)» (1 байт) (также называется «Начало кадра») используются для синхронизации отправляющих и получающих устройств. Эти первые 8 байт кадра необходимы для привлечения внимания получающих узлов. По существу, первые несколько байтов сообщают получателям о необходимости приготовиться к поступлению нового кадра.

Поле «MAC-адрес назначения» (6 байт) является идентификатором для предполагаемого получателя. Этот адрес используется вторым уровнем, чтобы помочь устройствам определить, адресован ли кадр именно им. Адрес в кадре сравнивается с MAC-адресом в устройстве. В случае совпадения устройство принимает кадр.

Поле «MAC-адрес источника» (6 байт) определяет сетевую плату или интерфейс, с которого был отправлен кадр.

7 Преамбула	1 Начало разделителя кадра	6 Адрес источника	6 Адрес назначения	2 Длина	46-1500 Заголовок и данные	4 Контрольная последовательность кадра

Рисунок 2.2 – Формат Ethemet-кадра по IEEE 802.3

Поле «Длина» в любом стандарте IEEE 802.3, используемом до 1997 г., определяет точную длину поля данных кадра. Позже оно применялось как часть контрольной последовательности кадра (FCS), чтобы обеспечить правильность получения сообщения. В других случаях это поле используется с целью описать, какой протокол более высокого уровня присутствует. Если 2-октетное значение равно или превышает шестнадцатеричный формат 0*0600 или десятичное число 1536, то содержимое поля «Данные» декодируется в соответствии с указанным протоколом EtherType. Если же значение равно или меньше шестнадцатеричного формата 0*05DC или десятичного числа 1500, то поле «Длина» позволяет обозначить использование формата кадра IEEE 802.3.

Поле «Данные» (46…1500 байт) содержит инкапсулированные данные из более высокого уровня, который является универсальным PDU третьего уровня, пакет IPv4. Длина всех кадров должна быть не менее 64 байт. В случае инкапсуляции небольшого пакета используются дополнительные биты, которые называются символами-заполнителями, для увеличения размера кадра до этого минимального значения.

Поле «Контрольная последовательность кадра» (4 байта) используется для обнаружения ошибок в кадре. В нем применяется циклический контроль избыточности (CRC). Отправляющее устройство включает в себя результаты циклического контроля избыточности в поле FCS кадра. Получающее устройство принимает кадр и создает CRC для поиска ошибок. Если расчеты совпадают, ошибки отсутствуют. Несовпадение расчетов означает изменение данных, следовательно, кадр сбрасывается. Данные могут измениться в результате нарушения электрических сигналов, которые представляют последовательность битов.

2.2 Порядок выполнения работы

1 Открыть файл lab1-2.pkt Cisco Packet Tracer, сохраненный в лабораторной работе № 1. Задания данной лабораторной работы выполнять в пространстве логической топологии.

2 Открыть командную строку на любом компьютере и выполнить команду arp -d, чтобы очистить таблицу ARP. Командой arp -a проверить, что таблица чистая.

3 Перейти в режим симуляции времени (рисунок 2.3, кнопка 1). Отправить ICMP-пакет с любого компьютера, используя команду ping. Будут созданы два пакета PDU (см. рисунок 2.3). При нажатии на пакет ARP (см. рисунок 2.3, кнопка 2) появляется окно содержания данных (PDU), в котором представлена информация по инкапсуляции данных на каждом уровне модели OSI (см. рисунок 2.3, область 3). Проследить прохождение пакета, нажимая клавишу «Capture/Forward», и заполнить таблицу 2.2. Сделать вывод о необходимости отправки ARP-пакета перед отправкой ICMP-пакета.

4 Проверить таблицу ARP командой arp -a, зафиксировать в отчете ее содержимое.

5 Открыть командную строку на любом компьютере и выполнить команду arp -d, чтобы очистить таблицу ARP. Командой arp -a проверить, что таблица чистая. Перейти в режим симуляции времени. Отправить ICMP-пакет с любого компьютера, используя команду ping. Используя данные окна «PDU Information» для ARP-запроса на закладке «Outbound PDU Details» (см. рисунок 2.3, закладка 4), отобразить вид передаваемых кадров в таблицах 2.3 и 2.4. Проделать аналогичные действия для ICMP-пакета (заполнить таблицы 2.5 и 2.6).

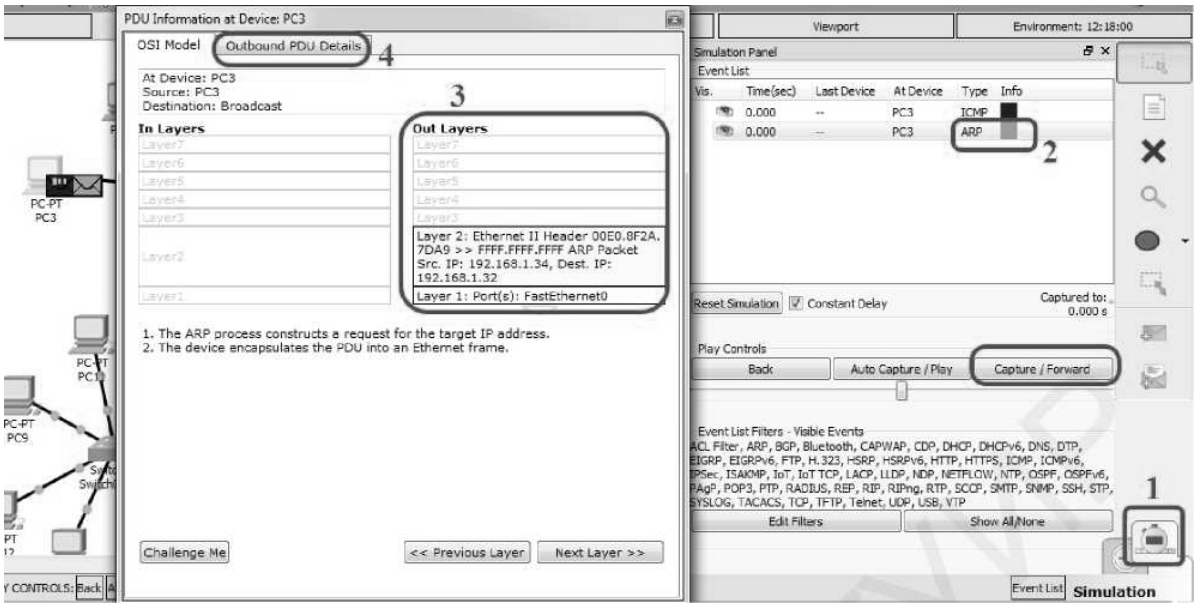


Рисунок 2.3 – Просмотр содержимого пакета в режиме симуляции времени

Таблица 2.2 – Последовательность действий в смоделированной сети с коммутаторами при отправке ping-запроса

Номер шага	Отправитель	Получатель	Содержание уровня	Описание действий
1				
2				
3				

Таблица 2.3 – Формат кадра Ethernet для ARP-запроса

Количество байтов							
Тип поля							
Содержание							

Таблица 2.4

Количество битов									
Тип поля									
Содержание									

Таблица 2.5 – Формат кадра IP для ICMP-запроса

Количество байтов							
Тип поля							
Содержание							

Таблица 2.6 – Формат ICMP-запроса

Количество битов									
Тип поля									
Содержание									

6 Заполнить таблицы 2.3–2.6 для пришедшего ответа ARP и ICMP. Сравнить структуру полученных и отправленных пакетов, отобразить различия и особенности в выводе лабораторной работы.

Контрольные вопросы

1 Перечислить уровни моделей OSI и TCP/IP, назвать основные отличия.

2 Привести описания уровней модели OSL.

3 Объяснить назначение протоколов HTTP, Telnet, SMTP, POP, FTP, DNS, UDP, TCP, IP, ICMP, ARP, Ethernet.

4 Какие существуют подуровни канального уровня? В чем заключается их сущность?

5 Представить формат Ethemet-кадра и назначения его полей.

3 Изучение маршрутизации IP

Цель работы: изучить принципы маршрутизации IP-сетей на примере протоколов статической и динамической маршрутизации с использованием программного обеспечения построения виртуальных сетей – Packet Tracer.

3.1 Краткие теоретические сведения

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), 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, 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:

1) плохая масштабируемость, т. к. при добавлении **N** сети потребуется сделать 2(7V + 1) записей о маршрутах;

2) отсутствует возможность адекватно отреагировать на ошибки и отказы оборудования канального и сетевого уровня;

3) ввод всей информации вручную является весьма трудоемкой задачей и влечет за собой необходимость документирования этих маршрутов;

4) при изменении топологии сети требуется вручную менять правила маршрутизации, т. е. переконфигурировать таблицу маршрутизатора.

Положительные качества:

1) легкость конфигурации. Метод статической маршрутизации является довольно простым для понимания и настройки;

2) отсутствует обмен служебной информацией между маршрутизаторами о топологии сетей, также и дополнительная нагрузка на сеть в виде широковещательного служебного трафика;

3) при использовании статических записей процессор маршрутизатора наименее нагружен при определении маршрутов.

Статическая маршрутизация продолжает успешно использоваться:

1) при организации работы компьютерных сетей небольшого размера (один-три маршрутизатора);

2) на компьютерах (рабочих станциях) внутри сети. В таком случае обычно задается маршрут шлюза по умолчанию;

3) 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.

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: RIPvl и RIPv2. Протокол RIPvl 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 следующий.

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 строчка 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.

3.2 Порядок выполнения работы

1 С помощью симулятора Packet Tracer создать модель составной сети с маршрутизаторами, изображенной на рисунке 3.1.

На рисунке 3.1 цифрами 1–7 обозначены IP-сети независимых LAN-сетей, цифрами 8–15 – подсети IP-WAN для линий связи между маршрутизаторами.

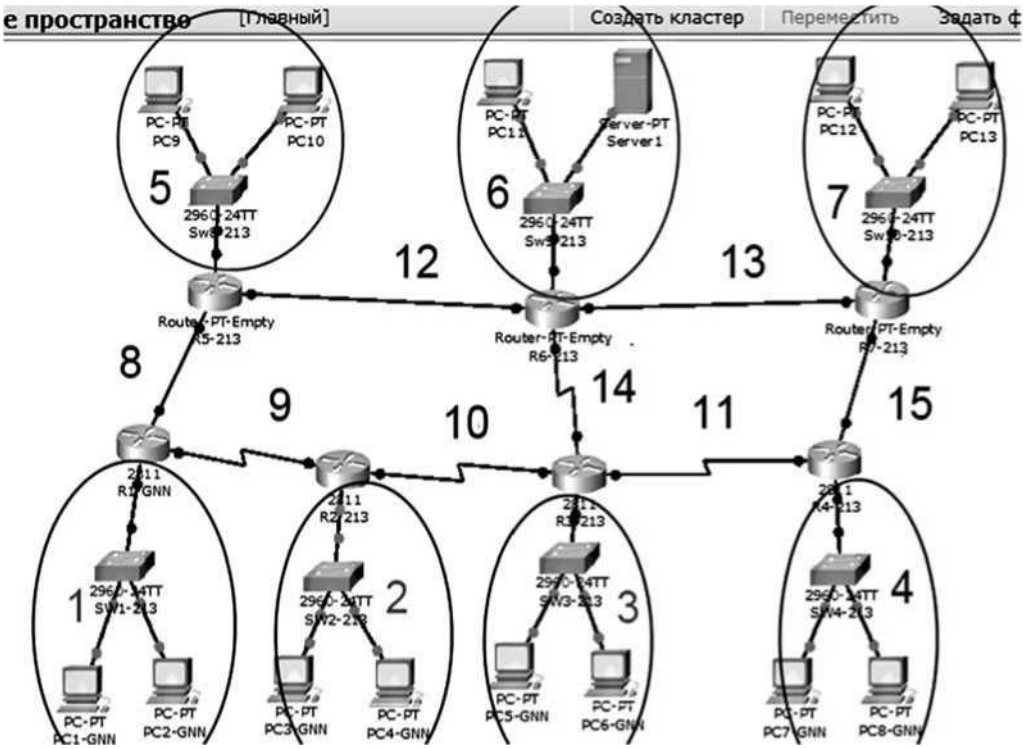


Рисунок 3.1 – Топология составной сети

2 Настроить IP-интерфейсы маршрутизаторов, подключенных к подсетям LAN с хостами в соответствии с таблицей 3.1.

Конфигурирование протокола RIP.

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

R5-213 (config) ftroute rip
R5-213(config-router) ttnetwork 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) ttexit

Таблица 3.1 – 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			

2 Аналогично ввести команды для остальных маршрутизаторов, проверить работоспособность сети, сделать подробные Screen Shot’s, сохранить файл макета под именем LR&18-RIP-GNN для отчета.

Конфигурирование протокола OSPF.

1 Открыть созданный файл LR#18-RIP-GNN.pkt и удалить созданные маршруты RIP с помощью команды **no router rip** на всех маршрутизаторах.

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

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

R3-213(config) ttrouter 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) ttnetwork 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 Ввести аналогичные команды для остальных маршрутизаторов, проверить работоспособность сети, сделать подробные ScreenShot’s, сохранить файл под именем LR#18-OSPF-GNN для предоставления вместе с отчетом.

Контрольные вопросы

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

4 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.

4.1 Краткие теоретические сведения

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 канальные 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]] [/renew 6[Adapter]] [/released[Adapter]] [/flushdns] [/displaydns] [/registerdns] [/showclassidAdapter] [/setclassidAdapter [ClasslD],

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 измерения настроек DNS сетевых подключений без перезагрузки компьютера;

/release[Adapter] – отмена 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;

Ishowclassid 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 /dysplaydns – 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] [-NlfaceAddr]]]-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 1P-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 4.1).

```
 Обработчик команд Windows                                                    | o_|| в || й |
Microsoft Windows [Version 6.1.7601 ]                                                    *
(c) Корпорация Майкрософт (Microsoft Corp.). 2009. Все права защищены, д

C:\Windows\System32>pmg 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 ТП.=120
Ответ от 82.209.221.3: число байт=32 время=11мс ТП_=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 = Юмсек, Максимальное = 12 мсек. Среднее = 11 мсек
```

Рисунок 4.1 – Определение достижимости узла bru.by командой ping

Использование, синтаксис:

ping [-t] [-a] [-п число] [-1 размер] [-f] [-i TTL] [-v TOS] [-г число] [-s число] [[-j список Узлов] | [-k список Узлов]] [—w таймаут] конечноеИмя.

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;

I – 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.

Пример – Тестирование прохождения сигнала к узлу bru.by с помощью команды **ping bru.by** (см. рисунок 4.1).

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 4.2 представлено тестирование маршрута прохождения к узлу google.by.

```
C:\Wi ndows\System32>tracert google.com
Трассировка маршрута к google.com [216.58.209.14]
с максимальным числом прыжков 30:

  1    <1 мс    <1 MC    <1 M    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   corel.net.belpak.by [93.85.253.193]
  6    11 ms     7 ms     7 ms   ie2.net.be!pak.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     A        A      108.170.250.209
 10    23 ms    19 ms    19 ms   172.253.68.31
 11    18 ms    17 ms    18 ms   waw02sl8-in-fl4.lel00.net [216.58.209.14

Трассировка завершена
```

Рисунок 4.2 – Определение и тестирование маршрута с помощью команды tracert

4.2 Порядок выполнения работы

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. Объяснить разницу в результатах.

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

3 Передать пакеты участникам сети напрямую и через шлюз. Объяснить полученные записи в таблице ARP.

4 Используя утилиту TRACERT и нижеприведенные IP-адреса, построить 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 нужны команда и таблица ARP?
5 Объяснить 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?

5 Моделирование топологий с использованием Packet Tracer (PT)

Цель работы: изучить принципы построения компьютерных сетей базовой топологии с базовым набором аппаратных средств с помощью программного обеспечения – симулятора сетей Packet Tracer.

5.1 Общие сведения

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;
– аналоговые и цифровые каналы телефонной связи.

5.2 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 5.1).

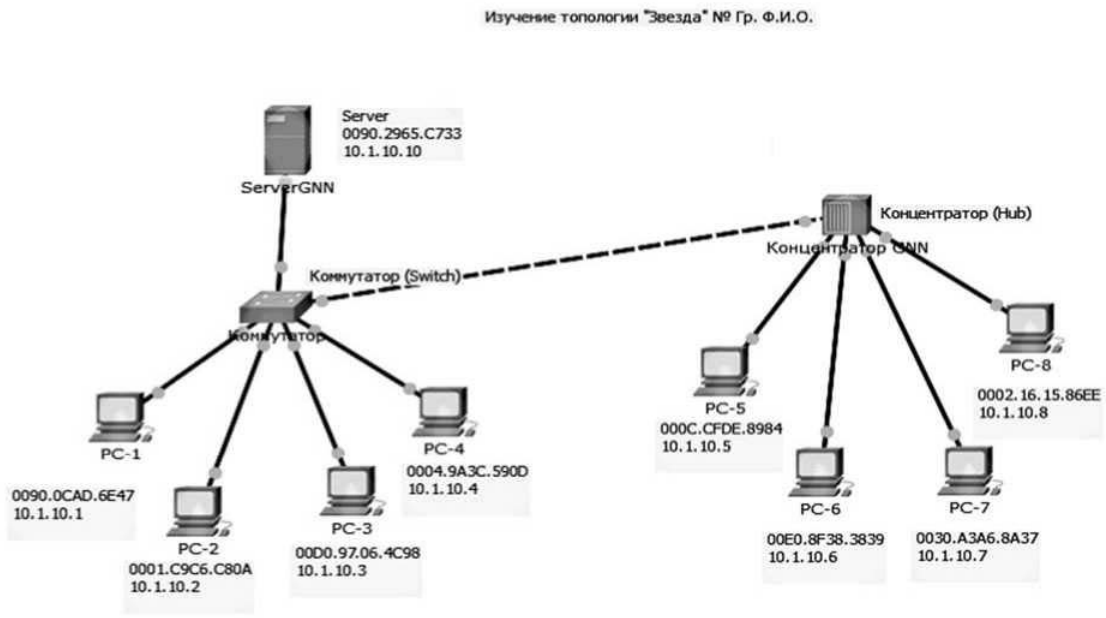


Рисунок 5.1 – Модель сети с использованием коммутатора и концентратора

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 Назвать 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?

6 Изучение виртуальных локальных сетей (VLAN)

Цель работы: изучить структуру и особенности применения виртуальных локальных компьютерных сетей.

6.1 Общие сведения

VLAN (Virtual Local Area Network) – виртуальная локальная компьютерная сеть из группы хостов с общим набором требований. VLAN позволяют хостам группироваться или дистанцироваться между собой. Устройства, в пределах одной VLAN, могут общаться, а узлы, находящиеся в разных VLAN, невидимы друг для друга. Они взаимодействуют так, как если бы они были подключены к широковещательному домену независимо от их физического местонахождения. Проектирование локальных сетей кампуса с использованием большего количества VLAN, в каждом из которых используется минимальное количество коммутационного оборудования, часто помогает улучшить локальную сеть во многих отношениях. Например, широковещательная передача, отправленная одним узлом во VLAN1, будет приниматься и обрабатываться всеми другими узлами этого VLAN 1, но не узлами из другого VLAN. Чем меньше посторонних узлов в сети получают широковещательные кадры, тем выше безопасность локальной сети.

В следующем списке перечислены наиболее распространенные причины, по которым следует создавать VLAN: чтобы уменьшить нагрузку на процессор на каждом устройстве; повышение производительности узла путем уменьшения числа устройств, которые принимают каждый широковещательный кадр; повышение безопасности хостов за счет применения различных политик безопасности для каждого VLAN; создание подразделений, группирующих пользователей по отделам или группам, которые работают вместе, а не по физическому местоположению; уменьшение нагрузки для протокола связующего дерева (STP) путем ограничения VLAN одним коммутатором доступа.

6.2 Порядок выполнения работы

Используя два VLAN, организовать две сети, что изображены на рисунке 6.1, создать два широковещательных домена с помощью одного коммутатора. С VLAN коммутатор может настроить некоторые интерфейсы в один широковещательный домен, а некоторые – в другой, создавая несколько широковещательных доменов. Эти отдельные широковещательные домены, созданные коммутатором, называются виртуальными локальными сетями (VLAN).

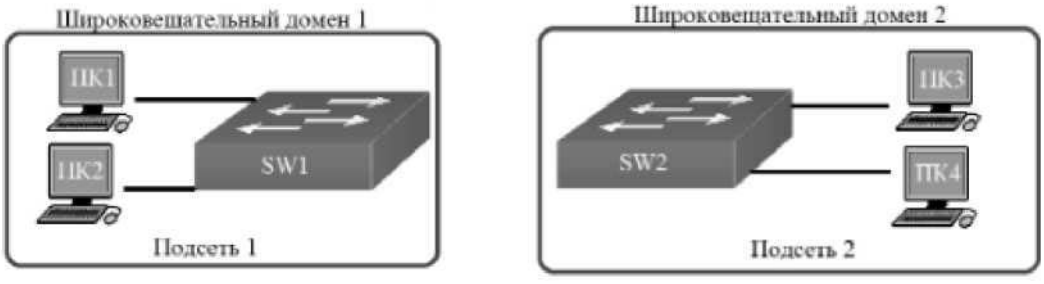


Рисунок 6.1 – Две VLAN-сети

Использовать один коммутатор для нескольких широковещательных доменов (рисунок 6.2). Из широковещательного домена 1 (подсеть 1) две системы ПК1 и ПК2 подключены к коммутатору SW1. Из широковещательного домена 2 (подсеть 2) к коммутатору SW1 подключены две системы ПК3 и ПК4. Из широковещательного домена 1 (подсеть 1) две системы ПК1 и ПК2 подключены к коммутатору SW1. Из широковещательного домена 2 (подсеть 2) к коммутатору SW1 подключены две системы ПК3 и ПК4.

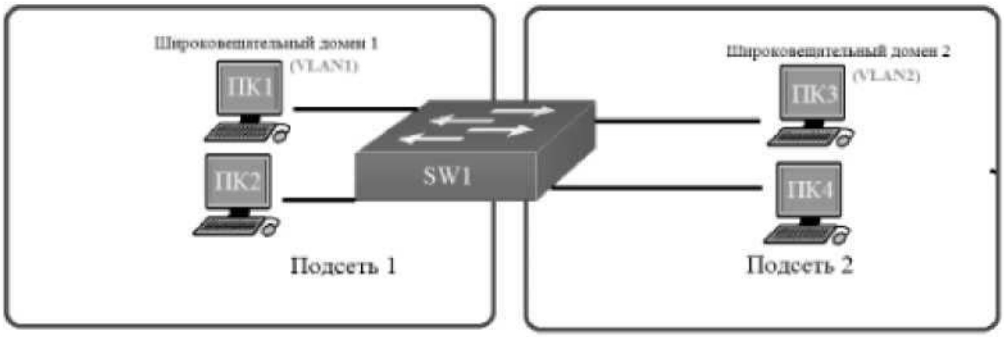


Рисунок 6.2 – Две VLAN-сети с одним коммутатором

Контрольные вопросы

1 Что такое VLAN?
2 Преимущества VLAN.
3 Свойства VLAN.

Список литературы

1 **Белоусова, Е. С.** Основы построения локальных сетей. Лабораторный практикум: учеб.-метод. пособие / Е. С. Белоусова. – Минск: БГУИР, 2020. – 103 с.

2 Cisco Packet Tracer – мощный инструмент моделирования сетей. – RL: https://www.youtube.com/watch?v=fyz- yejEHRc (дата обращения: 20.09.2024).

3 **Кении, А.** Самоучитель системного администратора / А. Кении. – СПб.: БХВ-Петербург, 2012. – 512 с.

4 Microsoft Windows Server 2012. Полное руководство / Р. Моримото [и др.]. – М.: Вильямс, 2013. – 1456 с.

5 **Поляк-Брагинский, А.** Администрирование сети на примерах / А. Поляк-Брагинский. – СПб.: БХВ-Петербург, 2012. – 432 с.

6 **Олифер, В. Г.** Компьютерные сети. Принципы, технологии, протоколы: учеб. пособие / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб.: Питер, 2013. – 944 с. : ил.

7 **Новиков, В. А.** Информационные системы и сети: учеб. пособие / В. А. Новиков, А. В. Новиков, В. В. Матвеенко. – М.: Изд-во Гревцова, 2014. – 448 с.

8 **Бройдо, О. П.** Вычислительные системы, сети и телекоммуникации: учебник / О. П. Бройдо, В. Л. Бройдо, О. П. Ильина. – 4-е изд. – СПб.: Питер, 2011. – 560 с.