Министерство образования Республики Беларусь

Учреждение образования
«Полоцкий государственный университет»

С. В. Калинцев

КОМПЬЮТЕРНЫЕ СЕТИ

Методические указания
по выполнению лабораторных работ для студентов специальности
1-40 05 01 «Информационные системы и технологии (в экономике)»

Текстовое электронное издание

Новополоцк
Полоцкий государственный университет
2022

Об издании – 1, 2

1 – дополнительный титульный экран – сведения об издании

УДК 004.7

Рекомендовано к изданию
методической комиссией факультета информационных технологий
в качестве методических указаний
(протокол № 11 от 27.12.2021)

РЕЦЕНЗЕНТ:

канд. техн. наук, доц., заведующий кафедрой математики и компьютерной безопасности Полоцкого государственного университета *И. Б. БУРАЧЁНОК*

2 – дополнительный титульный экран – производственно-технические сведения

Для создания текстового электронного издания «Компьютерные сети. Методические указания по выполнению лабораторных работ для студентов специальности 1-40 05 01 «Информационные системы и технологии (в экономике)» С. В. Калинцева использованы текстовый процессор Microsoft Office Word и программа Adobe Acrobat XI Pro для создания и просмотра электронных публикаций в формате PDF.

Технические требования:
1 оптический диск.
Системные требования:
PC с процессором не ниже Core 2 Duo;
2 Gb RAM; свободное место на HDD 3,5 Mb;
Windows XP/7/8/8.1/10
привод CD-ROM/DVD-ROM;
мышь.

Редактор С. Е. Рясова

Подписано к использованию 12.03.2022.
Объем издания: 3,1 Мб. Заказ 137.

Свидетельство о государственной регистрации
издателя, изготовителя, распространителя печатных изданий
№ 1/305 от 22.04.2014.

211440, Ул. Блохина, 29,
г. Новополоцк,
Тел. 8 (0214) 59-95-41, 59-95-44
http://www.psu.by

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Данное пособие рассчитано на студентов специальности 1-40 05 01 «Информационные системы и технологии (в экономике)».

В ходе выполнения лабораторных заданий студенты знакомятся с основными функциями и процедурами канального и сетевого уровня эталонной модели ISO/OSI на примере эмуляции виртуальных локальных вычислительных сетей. Виртуальная сеть передачи данных строится с помощью активных мультипликационных элементов средствами графического интерфейса пользователя. Элементы виртуальной сети представляют собой абстрактное изображение конкретного класса оборудования сети передачи данных. Доступ и управление оборудованием осуществляется посредством интерфейса командной оболочки, поддерживающей ряд утилит и встроенных команд (*ipconfig*, *arp*, *route* и т.д.), подобных утилитам операционных систем Windows и GNU/Linux, а также семейства BSD. Узлы виртуальной сети объединяются посредством экземпляра протокола, аналогичного по функциональности и принципам работы протоколу Ethernet.

Одним из основных преимуществ использования пакета «CPT 5.0» в курсе лабораторных занятий является применение прозрачной модели сетевого взаимодействия, небольшие системные требования и строгое соответствие специфике изучаемой дисциплины.

Рассматриваемое программное обеспечение способно максимально точно воспроизвести работу реальных сетей с помощью их виртуальных аналогов. Это позволяет в короткие сроки ознакомить студентов с принципами построения современных компьютерных сетей передачи данных, служебными протоколами и стандартным сетевым оборудованием.

Данные методические указания содержат краткое изложение теоретических сведений, необходимых для выполнения лабораторных работ; описание самих лабораторных работ.

ЛАБОРАТОРНАЯ РАБОТА № 1
Основные сетевые команды Windows

Цель работы: научиться осуществлять диагностику сетевых подключений с помощью встроенных команд операционной системы Windows.

Задачи:

– изучить основные сетевые команды операционной системы Windows;

– научиться применять команды для диагностики сетевого подключения операционной системы Windows.

Краткие теоретические сведения

Компьютерная сеть – это два или более компьютера, обменивающихся информацией по линиям связи.

Компьютерная сеть позволяет передавать информацию с одного компьютера на другой, а значит, совместно использовать ресурсы, например, принтеры, модемы и устройства хранения информации.

Сети бывают:

– *локальные* – объединяют компьютеры, находящиеся недалеко друг от друга, например, стоящие в соседних комнатах, в одном здании;

– *глобальные* – компьютеры могут находиться в разных городах и странах. Глобальные сети, как правило, объединяют несколько локальных сетей.

Для взаимодействия между собой программ в Internet используют *протоколы* (таблица 1)

Протокол – это набор правил и соглашений, используемых при передаче данных.

Таким образом, каждая программа, работающая в сети, должна следовать определенным правилам для приема и передачи данных.

Основополагающим протоколом сети Internet является *протокол TCP/IP*. TCP/IP – это два различных протокола, тесно связанных между собой. *TCP (Transmission Control Protocol)* – протокол управления передачей. Он определяет, каким образом информация должна быть разбита на пакеты и отправлена по каналам связи. TCP располагает пакеты в нужном порядке, а также проверяет каждый пакет на наличие ошибок при передаче.

Таблица 1. – Основные протоколы глобальных сетей

Название протокола	Расшифровка	Назначение
HTTP	Hyper Text Transfer Protocol	Протокол передачи гипертекста
FTP	File Transfer Protocol	Протокол передачи файлов
SMTP	Simple Mail Transfer Protocol	Простой протокол отправки электронных писем
POP3	Post Office Protocol 3	Протокол получения электронных писем
NNTP	News Net Transfer Protocol	Протокол телеконференций

Чтобы информация безошибочно могла передаваться с одного компьютера на другой, необходимо наличие уникальных адресов, с помощью которых можно однозначно определить (идентифицировать) получателя информации. В сети Internet информационные пакеты доставляются по адресам и в адресе указываются номера сетей, к которым подключен компьютер-получатель и номера самих компьютеров в этих сетях.

Каждый компьютер, подключенный к сети Internet, имеет свой уникальный *IP-адрес*.

IP-адрес (от англ. IP – Internet Protocol) – это уникальный номер, однозначно идентифицирующий компьютер в Internet. IP-адрес представляет собой четыре числа (октета), разделенные точками, например, *194.67.67.97* (после последнего числа точка не ставится).

Расшифровка такого адреса ведется слева направо. Первое число – номер наиболее крупной сети в составе Internet, последнее – номер конкретного компьютера. Второе и третье число обозначают участки сети, например, региональную и локальную сеть.

Каждое число может быть в интервале от 0 до 255, что соответствует информационному объему в 1 байт (8 бит). Таким образом, IP-адрес – это 4 байта или 32 бита. Если с помощью одного байта можно передать 2^8 = 256 вариантов, то с помощью 4-х байтов можно передать 2^{32} = 4 млрд вариантов, следовательно, к сети Internet может быть максимально подключено 4 млрд пользователей. Поскольку в настоящее время наблюдается стремительный рост пользователей Internet, а, кроме того, современные технические достижения позволяют подключать к сети Internet не только компьютеры, но и сотовые телефоны, телевизоры и даже холодильники, то это пространство адресов становится очень тесным. Для его расширения предполагается перевести Internet на 128-битный IP-адрес (максимум пользователей 2^{128}).

Сетевой адаптер – NIC (от англ. NIC – Network Interface Card), является важным аппаратным компонентом, используемым для обеспечения сетевых подключений.

В зависимости от способа доступа сетевого адаптера к сети, существуют проводной и беспроводной сетевые адаптеры. Проводной сетевой адаптер подключает узел к сети с помощью кабеля (например, Ethernet или оптоволоконный кабель). Беспроводной сетевой адаптер обычно поставляется с небольшой антенной, использующей радиоволны для связи с точкой доступа по беспроводной сети (например, Bluetooth или Wi-Fi).

MAC-адрес (от англ. Media Access Control – надзор за доступом к среде, также Hardware Address, также физический адрес) – уникальный идентификатор, присваиваемый каждой единице активного оборудования или некоторым их интерфейсам в компьютерных сетях Ethernet.

При проектировании стандарта Ethernet было предусмотрено, что каждая сетевая карта должна иметь уникальный шестибайтный номер (MAC-адрес), запрограммированный в ней при изготовлении.

Уникальность MAC-адресов достигается тем, что каждый производитель получает в координирующем комитете IEEE Registration Authority диапазон из 16 777 216 (2^{24}) адресов и, по мере исчерпания выделенных адресов, может запросить новый диапазон. Поэтому по трем старшим байтам MAC-адреса можно определить производителя. Существуют таблицы, позволяющие определить производителя по MAC-адресу.

Маска подсети – битовая маска для определения по IP-адресу адреса подсети и адреса узла этой подсети. В отличие от IP-адреса маска подсети не является частью IP-пакета.

Благодаря маске можно узнать, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети.

Пример. Узел с IP-адресом 12.34.56.78 и маской подсети 255.255.255.0, с длиной префикса 24 бита (/24), находится в сети 12.34.56.0.

Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию поразрядной конъюнкции (побитовое И).

Пример. В случае более сложной маски:

- IP-адрес: 11000000 10101000 00000001 00000010 (192.168.1.2);
- маска подсети: 11111111 11111111 11111110 00000000 (255.255.254.0);
- адрес сети: 11000000 10101000 00000000 00000000 (192.168.0.0).

Основной шлюз в локальной сети может представлять собой либо отдельное устройство – маршрутизатор, либо программное обеспечение, которое синхронизирует работу всех сетевых компьютеров.

Компьютеры при этом могут использовать разные протоколы связи (например, локальные и глобальные), которые предоставляют доступ к локальной или глобальной сети соответственно.

Основное назначение шлюза в сети заключается в конвертации данных. Кроме того, основной шлюз в сети – это указатель, необходимый для обмена информацией между компьютерами из разных сегментов сети.

При этом формирование IP-адреса маршрутизатора (или выполняющего его роль ПО) напрямую зависит от адреса сетевого шлюза.

Таким образом, адрес основного шлюза фактически представляет собой IP-адрес интерфейса устройства, с помощью которого осуществляется подключение компьютера к локальной сети

DNS (Domain Name System) – это технология, позволяющая браузеру найти запрошенный пользователем сайт по его имени.

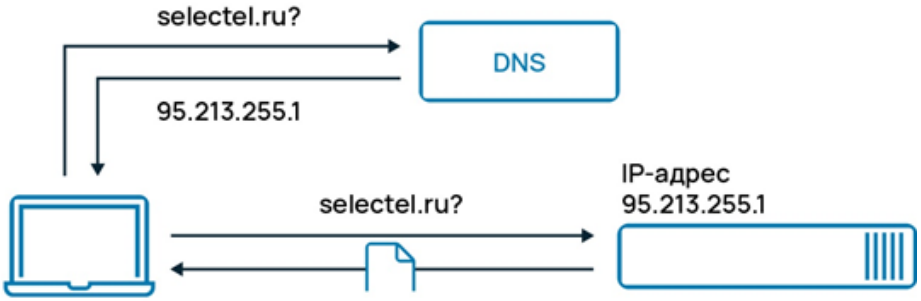

Рисунок 1. – Принцип работы DNS

DNS-сервер – это специализированный компьютер (или группа компьютеров), который хранит IP-адреса сайтов, привязанные к именам сайтов, и обрабатывает запросы пользователя. В сети Internet много DNS-серверов, они есть у каждого провайдера и обслуживают их пользователей.

Основное предназначение DNS-серверов – хранение информации о доменах и ее предоставление по запросу пользователей, а также кэширование DNS-записей других серверов.

Доменное имя – символьное имя, служащее для идентификации областей, которые являются единицами административной автономии в сети Internet, в составе вышестоящей по иерархии такой области. Каждая из таких областей называется *доменом*. Общее пространство имен Internet функционирует благодаря DNS – системе доменных имен. Доменные имена дают

возможность адресации интернет-узлов и расположенным на них сетевым ресурсам (веб-сайтам, серверам электронной почты, другим службам) быть представленными в удобной для человека форме.

Отдельно взятый DNS-сервер не может хранить вообще всю информацию об адресах сайтов и связанных с ними IP-адресах. Есть исключения – корневые DNS-серверы. При обращении к сайту компьютера пользователя браузер первым делом проверяет локальный файл настроек DNS – файл *hosts*. Если там нет нужного адреса, запрос направляется дальше – на локальный DNS-сервер интернет-провайдера пользователя.

Локальный DNS-сервер в большинстве случаев взаимодействует с другими DNS-серверами из региона, в котором находится запрошенный сайт. После нескольких обращений к таким серверам локальный DNS-сервер получает искомые данные и отправляет их в браузер – запрошенный сайт открывается. Полученные данные сохраняются на локальном сервере, что значительно ускоряет его работу, поскольку, единожды определив IP-адрес сайта, запрошенного пользователем, локальный DNS сохраняет эту информацию. Процесс сохранения полученных ранее данных и называется *кэшированием*.

Если пользователь обратится к ранее запрошенному сайту еще раз, то сайт откроется быстрее, поскольку используется сохраненная информация. Хранится кэш не постоянно, время хранения зависит от настроек самого сервера.

IP-адрес сайта может измениться, например, при перенаправлении на другой хостинг или сервер в рамках прежнего хостинга. В этом случае обращения пользователей к сайту, чей IP-адрес изменился, некоторое время обрабатываются по-старому, то есть перенаправление идет на прежний IP-адрес. И лишь через определенное время кэш локальных серверов обновляется, после чего обращение к сайту идет уже по новому IP-адресу.

DHCP (англ. Dynamic Host Configuration Protocol – протокол динамической настройки узла) – прикладной протокол, позволяющий сетевым устройствам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации компьютер-клиент на этапе конфигурации сетевого устройства обращается к так называемому серверу DHCP и получает от него нужные параметры. Сетевой администратор может задать диапазон адресов, распределяемых сервером среди компьютеров. Это позволяет избежать ручной настройки компьютеров сети и уменьшает количество ошибок. Протокол DHCP используется в большинстве сетей TCP/IP.

Основные сетевые команды Windows

1. Команда *IPCONFIG* служит для отображения параметров TCP/IP.

Команда *ipconfig* служит для управления сетевыми интерфейсами и отображения всех текущих параметров сети TCP/IP, а также обновления параметров DHCP и DNS в операционных системах Windows. При вызове команды *ipconfig* без параметров выводится только IP-адрес, маска подсети и основной шлюз для каждого сетевого адаптера.

Параметры утилиты *ipconfig*:

ipconfig [/all] [/renew [адаптер]] [/release [адаптер]] [/flushdns] [/displaydns] [/registerdns] [/showclassid адаптер] [/setclassid адаптер [код_класса]], где:

– */all* – вывод полной конфигурации TCP/IP для всех адаптеров. Без этого параметра команда *ipconfig* выводит только IP-адреса, маску подсети и основной шлюз для каждого адаптера. Адаптеры могут представлять собой физические интерфейсы, такие как установленные сетевые адаптеры, или логические интерфейсы, такие как подключения удаленного доступа;

– */renew [адаптер]* – обновление конфигурации DHCP для всех адаптеров (если адаптер не задан) или для заданного адаптера. Данный параметр доступен только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов. Чтобы указать адаптер, введите без параметров имя, выводимое командой *ipconfig*;

– */release [адаптер]* – отправка сообщения DHCPRELEASE серверу DHCP для освобождения текущей конфигурации DHCP и удаление конфигурации IP-адресов для всех адаптеров (если адаптер не задан) или для заданного адаптера. Этот адаптер отключает протокол TCP/IP для адаптеров, настроенных для автоматического получения IP-адресов. Чтобы указать адаптер, введите без параметров имя, выводимое командой *ipconfig*;

– */flushdns* – сброс и очистка содержимого кэша сопоставления имен DNS-клиента. Во время устранения неполадок DNS эту процедуру используют для удаления из кэша записей отрицательных попыток сопоставления и других динамически добавляемых записей;

– */displaydns* – отображение содержимого кэша сопоставления имен DNS-клиента, включающего записи, предварительно загруженные из локального файла Hosts, а также последние полученные записи ресурсов для запросов на сопоставление имен. Эта информация используется службой DNS-клиента для быстрого сопоставления часто встречаемых имен без обращения к указанным в конфигурации DNS-серверам;

– */registerdns* – динамическая регистрация вручную имен DNS и IP-адресов, настроенных на компьютере. Этот параметр полезен при устранении неполадок в случае отказа в регистрации имени DNS или при выяснении причин неполадок динамического обновления между клиентом и DNS-сервером без перезагрузки клиента. Имена, зарегистрированные в DNS, определяются параметрами DNS в дополнительных свойствах протокола TCP/IP;

– */showclassid адаптер* – отображение кода класса DHCP для указанного адаптера. Чтобы просмотреть код класса DHCP для всех адаптеров, вместо параметра *адаптер* укажите звездочку (*). Данный параметр доступен только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов;

– */setclassid адаптер [код_класса]* – задание кода класса DHCP для указанного адаптера. Чтобы задать код класса DHCP для всех адаптеров, вместо параметра адаптер укажите звездочку (*). Данный параметр доступен только на компьютерах с адаптерами, настроенными для автоматического получения IP-адресов. Если код класса DHCP не задан, текущий код класса удаляется.

– */?* – отображение справки в командной строке.

2. *PING* – основная утилита командной строки Windows для проверки сетевых соединений в Windows.

Синтаксис, параметры, важные ключи команды *ping*:
ping [-t] [-a] [-n счетчик] [-l размер] [-f] [-i TTL] [-v mun] [-r счетчик] [-s счетчик] [{-j список_узлов | -k список_узлов}] [-w интервал] [имя_конечного_компьютера], где:

– *-t* – задает для команды *ping* отправку сообщений с эхо-запросом к точке назначения до тех пор, пока команда не будет прервана. Для прерывания команды и вывода статистики нажмите комбинацию CTRL+BREAK. Для прерывания команды *ping* и выхода из нее нажмите клавиши CTRL+C;

– *-a* – задает разрешение обратного имени по IP-адресу назначения. В случае успешного выполнения выводится имя соответствующего узла;

– *-n* счетчик – задает число отправляемых сообщений с эхо-запросом. По умолчанию – 4;

– *-l размер* – задает длину (в байтах) поля данных в отправленных сообщениях с эхо-запросом. По умолчанию – 32 байта. Максимальный размер – 65 500;

– *-f* – задает отправку сообщений с эхо-запросом с флагом «Don't Fragment» в IP-заголовке, установленном на 1. Сообщения с эхо-запросом не фрагментируются маршрутизаторами на пути к месту назначения. Этот параметр полезен для устранения проблем, возникающих с максимальным блоком данных для канала (Maximum Transmission Unit).

– *-i TTL* – задает значение поля TTL в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию берется значение TTL, заданное по умолчанию для узла. Для узлов Windows XP это значение обычно равно 128. Максимальное значение TTL – 255.

– *-v тип* – задает значение поля типа службы (TOS) в IP-заголовке для отправляемых сообщений с эхо-запросом. По умолчанию это значение равно 0, тип – это десятичное значение от 0 до 255.

– *-r счетчик* – задает параметр записи маршрута (Record Route) в IP-заголовке для записи пути, по которому проходит сообщение с эхо-запросом и соответствующее ему сообщение с эхо-ответом. Каждый переход в пути использует параметр записи маршрута. По возможности значение счетчика задается равным или большим, чем количество переходов между источником и местом назначения. Параметр *счетчик* имеет значение от 1 до 9.

– *-s счетчик* – указывает вариант штампа времени Internet (Internet Timestamp) в заголовке IP для записи времени прибытия сообщения с эхо-запросом и соответствующего ему сообщения с эхо-ответом для каждого перехода. Параметр *счетчик* имеет значение от 1 до 4.

– *-j список_узлов* – указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в *списке_узлов*. При свободной маршрутизации последовательные промежуточные точки назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число адресов или имен в списке узлов – 9. *Список узлов* – это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами.

– *-k список_узлов* – указывает для сообщений с эхо-запросом использование параметра строгой маршрутизации в IP-заголовке с набором промежуточных точек назначения, указанным в *списке_узлов*. При строгой маршрутизации следующая промежуточная точка назначения должна быть доступной напрямую (она должна быть соседней в интерфейсе маршрутизатора). Максимальное число адресов или имен в списке узлов равно 9. *Список узлов* – это набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами.

– *-w интервал* – определяет в миллисекундах время ожидания получения сообщения с эхо-ответом, которое соответствует сообщению с эхо-запросом. Если сообщение с эхо-ответом не получено в пределах заданного интервала, то выдается сообщение об ошибке «Request timed out». Интервал по умолчанию равен 4000 (4 секунды).

– *имя_конечного_компьютера* – задает точку назначения, идентифицированную IP-адресом или именем узла.

3. Команда *TRACERT* показывает трассировку маршрута до указанного удаленного хоста. В ходе трассировки будет показан весь маршрут прохождения пакетов. Также эта команда командной строки показывает в мс задержку пакетов от каждого узла на пути каждого маршрутизатора. Эта задержка позволяет определить, на каком промежуточном участке происходит потеря пакетов (потери обозначаются *).

Параметры и ключи утилиты *tracert*:
tracert [-d] [-h максимальное_число_переходов] [-j список_узлов] [-w интервал [имя_конечного_компьютера], где:

– *-d* – предотвращает попытки команды *tracert* разрешения IP-адресов промежуточных маршрутизаторов в имена. Увеличивает скорость вывода результатов команды *tracert*;

– *-h максимальное_число_переходов* – задает максимальное количество переходов на пути при поиске конечного объекта. Значение по умолчанию равно 30;

– *-j список_узов* – указывает для сообщений с эхо-запросом использование параметра свободной маршрутизации в заголовке IP с набором промежуточных мест назначения, указанных в *списке_узлов*. При свободной маршрутизации успешные промежуточные места назначения могут быть разделены одним или несколькими маршрутизаторами. Максимальное число адресов или имен в списке – 9. *Список_адресов* представляет набор IP-адресов (в точечно-десятичной нотации), разделенных пробелами;

– *-w интервал* – определяет в миллисекундах время ожидания для получения эхо-ответов протокола ICMP или ICMP-сообщений об истечении времени, соответствующих данному сообщению эхо-запроса. Если сообщение не получено в течение заданного времени, выводится звездочка (*). Таймаут по умолчанию 4000 (4 секунды);

– *имя_конечного_компьютера* – задает точку назначения, указанную IP-адресом или именем узла;

– *-?* – отображает справку в командной строке по утилите *tracert*.

Утилита командной строки *NSLOOKUP* является, пожалуй, наиболее полезным инструментом для поиска и устранения проблем, связанных с клиентами DNS. Информация, которую она позволяет получать, оказывает неоценимую помощь в понимании проблем, связанных с DNS. В самом общем случае утилита *NSLOOKUP* связывается со стандартным DNS-сервером клиента и пытается преобразовать введенное имя.

Ход работы

1. Войти в интерпретатор командной строки (рисунок 2), сочетание клавиш «Windows + R».

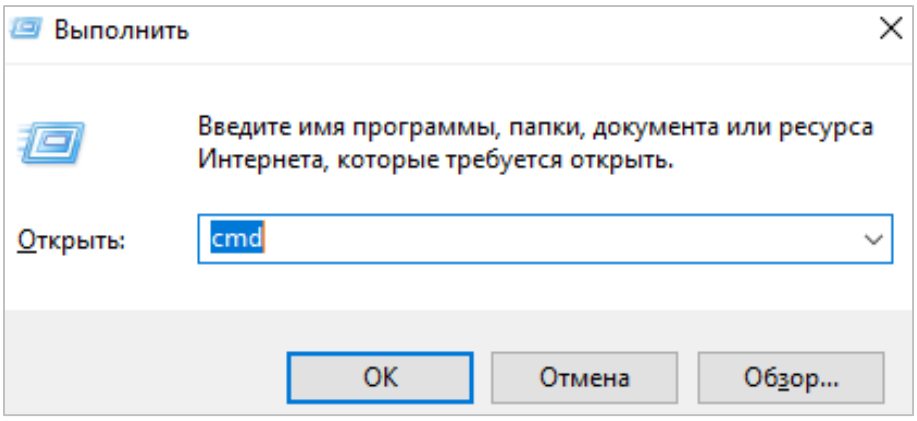

Рисунок 2. – Интерпретатор командной строки

2. С помощью утилиты *ipconfig* определить и записать в отчет следующую информацию:

– название сетевого подключения;
– тип используемого адаптера;
– MAC-адрес адаптера;
– IP-адрес сетевого подключения;
– сетевую маску;
– основной шлюз;
– IP-адрес DNS-сервера;
– IP-адрес DHCP-сервера.

Скопировать вывод утилиты *ipconfig* в отчет.

Щелкнуть в любом месте окна *cmd* правой кнопкой мыши и выбрать «Пометить» (рисунок 3). Выделить нужный фрагмент текста и нажать «Enter».

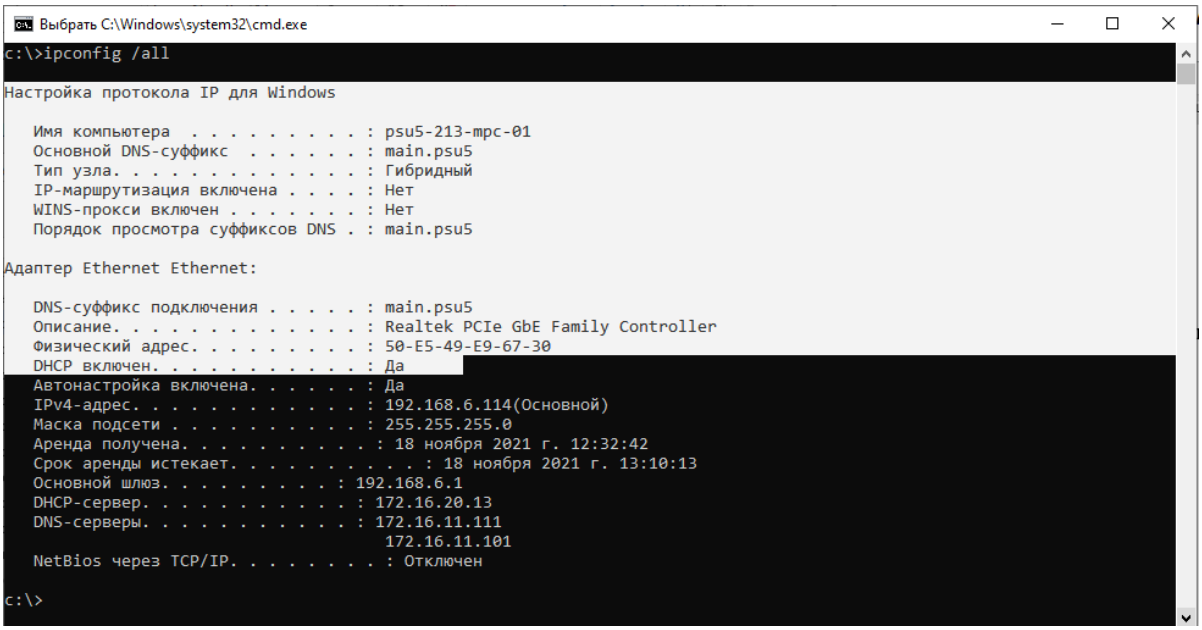

Рисунок 3. – Копирование информации из окна командной строки

3. С помощью утилиты *ping* проверить доступность следующих устройств:

– основой шлюз;

– сервер DHCP;

– сервер DNS;

– информационный ресурс www.psu.by.

Используя дополнительные ключи, сделать так, чтобы количество посылаемых эхо-запросов равнялось номеру компьютера + 10.

Скопировать результаты выполнения *ping* в отчет.

Для каждого устройства и информационного ресурса записать в отчет следующую информацию:

– процент потерь;

– среднее время приема передачи.

4. С помощью утилиты *tracert* проверить доступность следующих устройств:

– основой шлюз;

– информационный ресурс www.psu.by.

Используя дополнительные ключи, сделать так, чтобы утилита не определяла DNS-имена промежуточных устройств.

Скопировать результаты выполнения *tracert* в отчет.

Для каждого устройства и информационного ресурса записать в отчет следующую информацию:

– количество промежуточных устройств;

– IP-адреса всех промежуточных устройств;

– определить IP-адрес шлюза.

Невозможность провести трассировку до информационного ресурса www.psu.by косвенно указывает на наличие в сети другого устройства, которое управляет доступом в Internet. Обычно таким устройством является прокси-сервер. В данном случае прокси-сервер имеет имя *proxy5.main.psu5*.

5. Определить сетевой адрес и маршрут до прокси-сервера. Скопировать полученные результаты в отчет.

6. С помощью утилиты *nslookup* (используя команду *ls*) получить сетевые адреса устройств, зарегистрированных в сети в данный момент.

Скопировать полученный результат в отчет.

Для любых 3 компьютеров из списка записать следующую информацию:

– доменное имя компьютера;

– сетевой адрес компьютера.

С помощью утилиты *ping* определить доступность выбранных 3 компьютеров.

Скопировать полученный результат в отчет.

7. С помощью команды *arp* определить и записать в отчет MAC-адреса следующих устройств:

– основной шлюз;

– компьютеры, выбранные в п. 6.

Скопировать результаты выполнения команды *arp* в отчет.

8. С помощью сервиса Smart-Whois (http://www.who.is) определить и записать в отчет регистрационные данные для информационного ресурса www.psu.by.

9. С помощью сервиса Smart-Whois (http://www.all-nettools.com/toolbox/smartwhois.php) определить и записать в отчет регистрационные данные для сетевого адреса, принадлежащего информационному ресурсу www.psu.by.

10. С помощью сервиса Traceroute (http://www.all-nettools.com/toolbox/traceroute.php) осуществить трассировку информационного ресурса www.psu.by. Скопировать полученный результат в отчет.

11. С помощью утилиты Smart-Whois выяснить и записать в отчет следующую информацию о промежуточных устройствах:

– название организации, владеющей сетевым адресом устройства;

– страна и город, в котором предположительно находится данное устройство.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Ход работы в виде скриншотов экрана с пояснениями.
5. Выводы.

Контрольные вопросы

1. За что отвечает команда *ping*?
2. Как отрегулировать количество посылок команды *ping*?
3. Какой протокол отвечает за работу команды *ping*?
4. Для чего предназначена команда *ipconfig/all*?
5. Как работает утилита *nslookup*?
6. Для чего нужна команда *traceroute*?

ЛАБОРАТОРНАЯ РАБОТА № 2
Построение простейших сетей при помощи эмулятора работы локальной вычислительной сети Cisco Packet Tracer

Цель работы: изучение среды проектирования сетей Cisco Packet Tracer.

Задачи:

– изучить интерфейс Packet Tracer 5.0;

– провести эмуляцию простейших локальных сетей.

Краткие теоретические сведения

Программные продукты Packet Tracer дают возможность создавать сетевые топологии из широкого спектра маршрутизаторов и коммутаторов компании Cisco, рабочих станций и сетевых соединений типа Ethernet, Serial, ISDN, Frame Relay. Эта функция может быть выполнена как для обучения, так и для работы. Например, чтобы сделать настройку сети еще на этапе планирования или чтобы создать копию рабочей сети с целью устранения неисправности.

Для запуска Cisco Packet Tracer необходимо вызвать исполняемый файл, PacketTracer52.exe. Общий вид программы можно увидеть на рисунке 4.

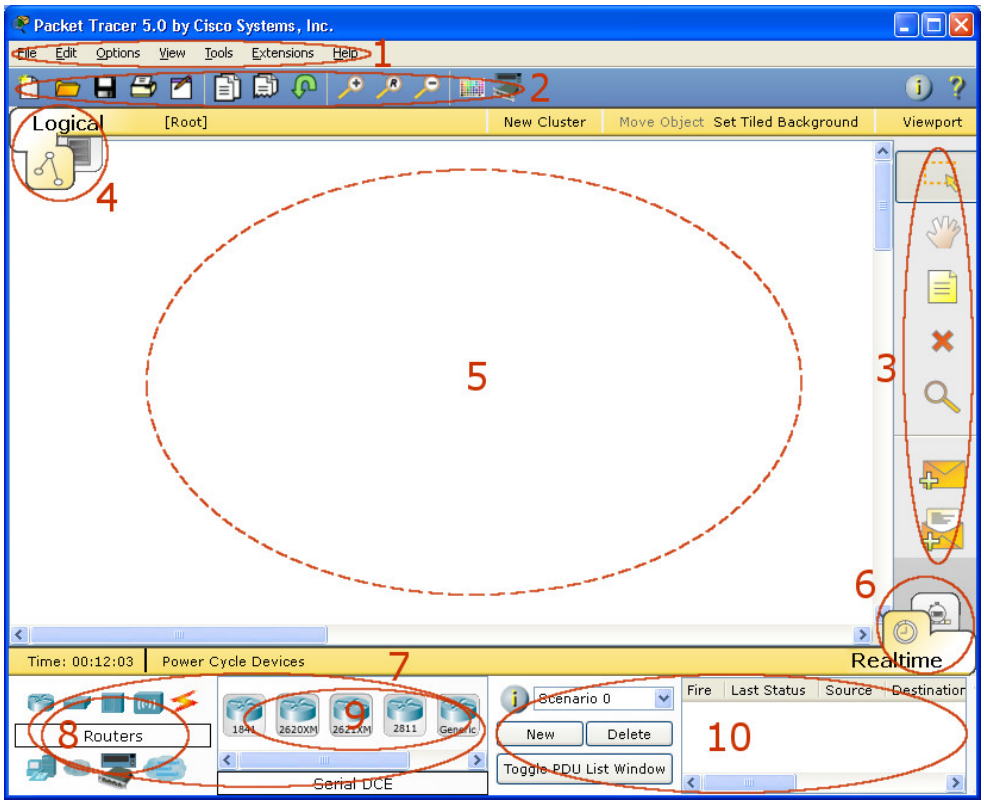

Рисунок 4. – Общий вид программы Packet Tracer

Рабочая область окна программы состоит из следующих элементов:

1. «Menu Bar» – панель, которая содержит меню «File», «Edit», «Options», «View», «Tools», «Extensions», «Help».

2. «Main Tool Bar» содержит графические изображения ярлыков для доступа к командам меню «File», «Edit», «View» и «Tools», а также кнопку «Network Information».

3. «Common Tools Bar» – панель, которая обеспечивает доступ к наиболее используемым инструментам программы: «Select», «Move Layout», «Place Note», «Delete», «Inspect», «Add Simple PDU» и «Add Complex PDU».

4. «Logical/Physical Workspace and Navigation Bar» – панель, которая дает возможность переключать рабочую область (физическую или логическую), а также позволяет перемещаться между уровнями кластера.

5. «Workspace» – область, в которой происходит создание сети, проводятся наблюдения и просматривается разная информация и статистика.

6. «Realtime/Simulation Bar» – с помощью закладок этой панели можно переключаться между режимом *Realtime* и режимом *Simulation*. Она также содержит кнопки, относящиеся к «Power Cycle Devices», кнопки «Play Control» и переключатель «Event List» в режиме *Simulation*.

7. «Network Component Box» – это область, в которой выбираются устройства и связи для размещения их на рабочем пространстве. Она содержит область «Device-Type Selection» и область «Device-Specific Selection».

8. «Device-Type Selection Box» – эта область содержит доступные типы устройств и связей в *Packet Tracer*. Область «Device-Specific Selection» изменяется в зависимости от выбранного устройства.

9. «Device-Specific Selection Box» – эта область используется для выбора конкретных устройств и соединений, необходимых для постройки в рабочем пространстве сети.

10. «User Created Packet Window» – это окно управляет пакетами, которые были созданы в сети во время симуляции сценария.

Для создания топологии необходимо выбрать устройство из панели «Network Component», а затем из панели «Device-Type Selection» выбрать тип выбранного устройства. После этого нужно нажать левую кнопку мыши в поле рабочей области программы («Workspace»). Также можно переместить устройство прямо из области «Device-Type Selection», но при этом будет выбрана модель устройства по умолчанию.

Для быстрого создания нескольких экземпляров одного и того же устройства нужно, удерживая кнопку «Ctrl», нажать на устройство в области «Device-

Specific Selection» и отпустить кнопку «Ctrl». После этого можно несколько раз нажать на рабочей области для добавления копий устройства.

В *Packet Tracer* представлены следующие типы устройств:

– *маршрутизатор* – специализированное устройство, которое пересылает пакеты между различными сегментами сети на основе правил и таблиц маршрутизации. Маршрутизатор может связывать разнородные сети различных архитектур. Для принятия решений о пересылке пакетов используется информация о топологии сети и определенные правила, заданные администратором;

– *коммутатор* – устройство, предназначенное для соединения нескольких узлов компьютерной сети в пределах одного или нескольких сегментов сети. *Сегмент сети* – физически обособленная часть сети;

– *хабы и повторители* – класс устройств для объединения компьютеров в сетях Ethernet с применением кабельной инфраструктуры;

– *конечные устройства* – ПК, серверы, принтеры, IP-телефоны;

– *беспроводные устройства*: точки доступа – базовые станции, предназначенные для обеспечения беспроводного доступа к уже существующей сети (беспроводной или проводной) или создания новой беспроводной сети, и беспроводной маршрутизатор;

Необходимые элементы помещаются в рабочую область программы так, как показано на рисунке 5.

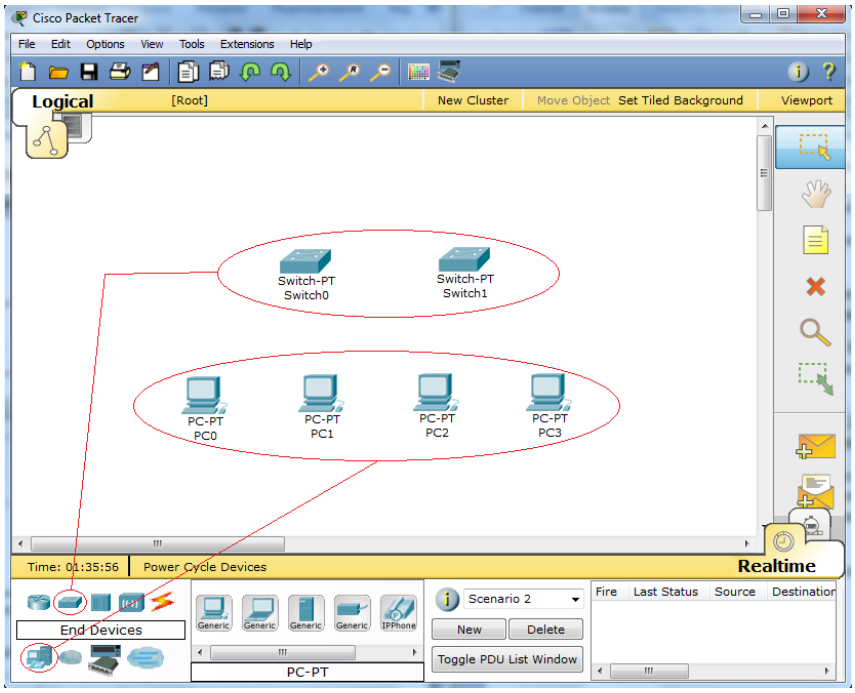

Рисунок 5. – Добавление элементов сети

При добавлении каждого элемента пользователь имеет возможность дать ему имя и установить необходимые параметры. Для этого необходимо нажать на нужный элемент левой кнопкой мыши (ЛКМ) и в диалоговом окне устройства перейти к вкладке «Config».

Диалоговое окно свойств каждого элемента имеет две вкладки:

– «Physical» – содержит графический интерфейс устройства и позволяет симулировать работу с ним на физическом уровне;

– «Config» – содержит все необходимые параметры для настройки устройства и имеет удобный для этого интерфейс.

Также в зависимости от устройства свойства могут иметь дополнительную вкладку для управления работой выбранного элемента: «Desktop» (если выбрано конечное устройство) или «CLI» (если выбран маршрутизатор) и т.д.

Для удаления ненужных устройств с рабочей области программы используется кнопка «Delete» («Del»).

Добавленные элементы связываются с помощью соединительных связей. Для этого необходимо выбрать вкладку «Connections» из панели «Network Component Box». Мы увидим все возможные типы соединений между устройствами. Выбирается подходящий тип кабеля. Указатель мыши изменится на курсор «connection» (имеет вид разъема). На первом устройстве и выбирается соответствующий интерфейс, с которым нужно выполнить соединение, а затем второе устройство, при выполнении той же операции (рисунок 6). Можно также соединить с помощью «Automatically Choose Connection Type» (автоматически соединяет элементы в сети). Между устройствами появится кабельное соединение, а индикаторы на каждом конце покажут статус соединения (для интерфейсов, которые имеют индикатор).

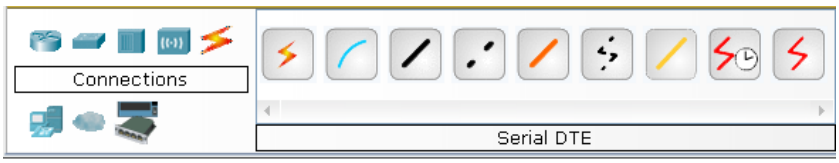

Рисунок 6. – Поддерживаемые в Packet Tracer типы кабелей

Packet Tracer поддерживает широкий диапазон сетевых соединений (таблица 2). Каждый тип кабеля может быть соединен лишь с определенными типами интерфейсов.

После создания макета сети ее нужно сохранить, выбрав пункт меню *File → Save* или иконку «Save» на панели «Main Tool Bar». Файл сохраненной топологии имеет расширение **.pkt*.

Таблица 2. – Типы соединений в Packet Tracer

Тип кабеля	Описание
 Console	Консольное соединение может быть выполнено между ПК и маршрутизаторами или коммутаторами. Должны быть выполнены некоторые требования для работы консольного сеанса с ПК: скорость соединения с обеих сторон должна быть одинаковая, должно быть 7 бит данных (или 8 бит) для обеих сторон, контроль четности должен быть одинаковый, должно быть 1 или 2 стоповых бита (но они не обязательно должны быть одинаковыми), а поток данных может быть чем угодно для обеих сторон
 Copper Straight-through	Этот тип кабеля является стандартной средой передачи Ethernet для соединения устройств, который функционирует на разных уровнях OSI. Он должен быть соединен со следующими типами портов: медный 10 Мбит/с (Ethernet), медный 100 Мбит/с (Fast Ethernet) и медный 1000 Мбит/с (Gigabit Ethernet)
 Copper Crossover	Этот тип кабеля является средой передачи Ethernet для соединения устройств, которые функционируют на одинаковых уровнях OSI. Он может быть соединен со следующими типами портов: медный 10 Мбит/с (Ethernet), медный 100 Мбит/с (Fast Ethernet) и медный 1000 Мбит/с (Gigabit Ethernet)
 Fiber	Оптоволоконная среда используется для соединения между оптическими портами (100 Мбит/с или 1000 Мбит/с)
 Phone	Соединение через телефонную линию может быть осуществлено только между устройствами, имеющими модемные порты. Стандартное представление модемного соединения – это конечное устройство, например ПК, дозванивающееся в сетевое облако
 Coaxial	Коаксиальная среда используется для соединения между коаксиальными портами, такими как кабельный модем, соединенный с облаком Packet Tracer
 Serial DCE and DTE	Соединения через последовательные порты, часто используются для связей WAN. Для настройки таких соединений необходимо установить синхронизацию на стороне DCE-устройства. Синхронизация DTE выполняется по выбору. Сторону DCE можно определить по маленькой иконке «часов» рядом с портом. При выборе типа соединения Serial DCE первое устройство, к которому применяется соединение, становится DCE-устройством, а второе – автоматически станет стороной DTE. Возможно и обратное расположение сторон, если выбран тип соединения Serial DTE

Packet Tracer дает возможность симулировать работу с интерфейсом командной строки (ИКС) операционной системы IOS, установленной на всех коммутаторах и маршрутизаторах компании Cisco.

Подключившись к устройству, можно работать с ним так, как за консолью реального устройства. Симулятор обеспечивает поддержку практически всех команд, доступных на реальных устройствах.

Подключение к ИКС коммутаторов или маршрутизаторов можно произвести, нажав на необходимое устройство и перейдя в окно свойств к вкладке «CLI».

Для симуляции работы командной строки на конечном устройстве (компьютере) необходимо в свойствах выбрать вкладку «Desktop», а затем нажать на ярлык «Command Prompt».

Работа с файлами в симуляторе

Packet Tracer дает возможность пользователю хранить конфигурацию некоторых устройств, таких как маршрутизаторы или коммутаторы, в текстовых файлах. Для этого необходимо перейти к свойствам необходимого устройства и во вкладке «Config» нажать на кнопку «Export…» для экспорта конфигурации *Startup Config* или *Running Config*. Так получается диалоговое окно для сохранения необходимой конфигурации в файл, который будет иметь расширение **.txt*. Текст файла с конфигурацией устройства *running-config.txt* (имя по умолчанию) аналогичен тексту информации полученной при использовании команды *show running-config* в IOS-устройствах.

Необходимо отметить, что конфигурация каждого устройства сохраняется в отдельном текстовом файле. Пользователь также имеет возможность изменять конфигурацию в сохраненном файле вручную с помощью произвольного текстового редактора. Для предоставления устройству сохраненных или отредактированных настроек нужно во вкладке «Config» нажать кнопку «Load...» для загрузки необходимой конфигурации *Startup Config* или кнопку «Merge…» для загрузки конфигурации *Running Config*.

Ход работы

На рабочую область программы добавляются два коммутатора *Switch-PT*, по умолчанию они имеют имена *Switch0* и *Switch1*. Затем – четыре компьютера с именами по умолчанию *PC0*, *PC1*, *PC2*, *PC3*. Соединяются устройства в сеть Ethernet, как показано на рисунке 7. Сохраняется созданная топология нажатием кнопки «Save» (меню *File → Save*).

Откроются свойства устройства *PC0* нажатием на его изображении. Далее по вкладке «Desktop» симулируется работа *run* нажатием «Command Prompt».

Список команд получается вводом «?» и нажатием «Enter». Для конфигурирования компьютера используется команда *ipconfig* из командной строки.

Пример. ipconfig 192.168.1.2 255.255.255.0

IP-адрес и маску сети также можно вводить в удобном графическом интерфейсе устройства (см. рисунки 7, 8). Поле *DEFAULT GATEWAY* – адреса шлюза, не имеет значения, так как создаваемая сеть не требует маршрутизации.

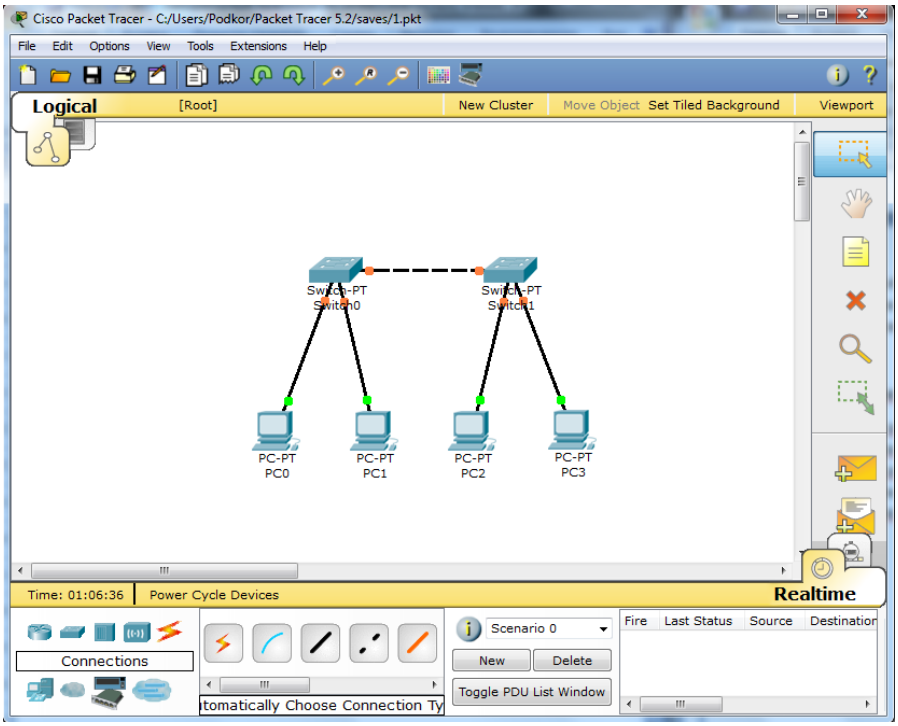

Рисунок 7. – Экспериментальная модель сети

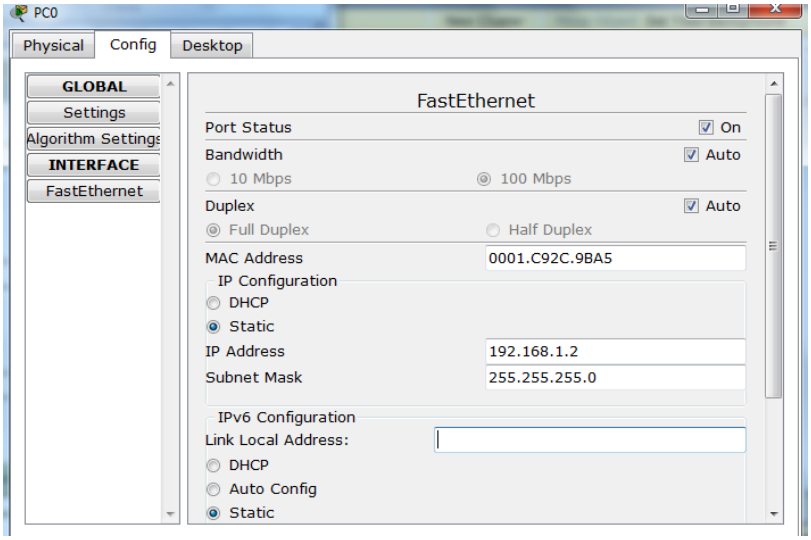

Рисунок 8. – Настройка узла

Таким же путем настраивается каждый компьютер (таблица 3).

Таблица 3. – Адреса узлов ЭВМ

Устройство	IP ADDRESS	SUBNET MASK
PC0	192.168.1.2	255.255.255.0
PC1	192.168.1.3	255.255.255.0
PC2	192.168.1.4	255.255.255.0
PC3	192.168.1.5	255.255.255.0

На каждом компьютере посмотреть назначенные адреса командой *ipconfig* без параметров.

В *Packet Tracer* предусмотрен режим моделирования, в котором подробно описывается и показывается, как работает утилита *Ping*. Поэтому необходимо перейти в данный режим, нажав на одноименный значок в нижнем левом углу рабочей области, или по комбинации клавиш Shift + S. Откроется «Панель моделирования» (рисунок 9), в которой будут отображаться все события, связанные с выполнения ping-процесса.

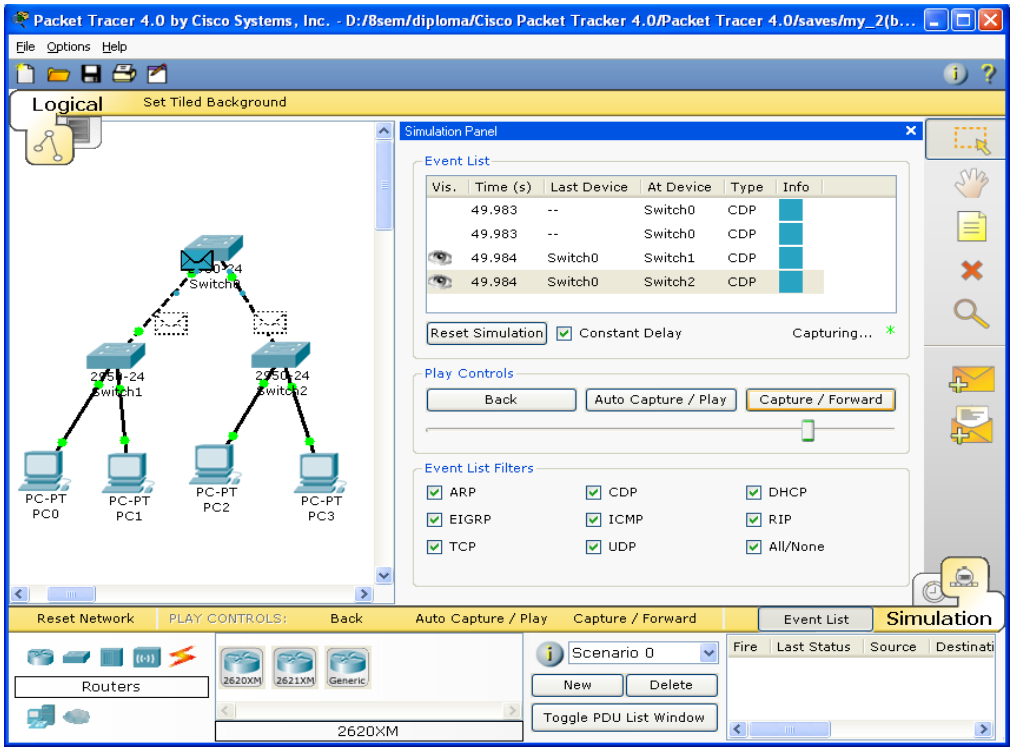

Рисунок 9. – Панель моделирования

Теперь необходимо повторить запуск ping-процесса. После его запуска можно сдвинуть «Панель моделирования», чтобы на схеме спроектированной сети наблюдать за отправкой/приемкой пакетов.

Кнопка «Автоматически» подразумевает моделирование всего ping-процесса в едином процессе, тогда как «Пошагово» позволяет отображать его пошагово.

Чтобы узнать информацию, которую несет в себе пакет, его структуру, достаточно нажать правой кнопкой мыши на цветной квадрат в графе «Информация».

Моделирование прекращается либо при завершении ping-процесса, либо при закрытии окна «Редактирование» соответствующей рабочей станции.

Если все сделано правильно, можно пропинговать любой компьютер. Должен появиться отчет о ping-процессе подобный рисунку 10.

Однако это не все преимущества *Packet Tracer*: в «Режиме симуляции» можно не только отслеживать используемые протоколы, но и видеть, на каком из семи уровней модели OSI данный протокол задействован (рисунок 11).

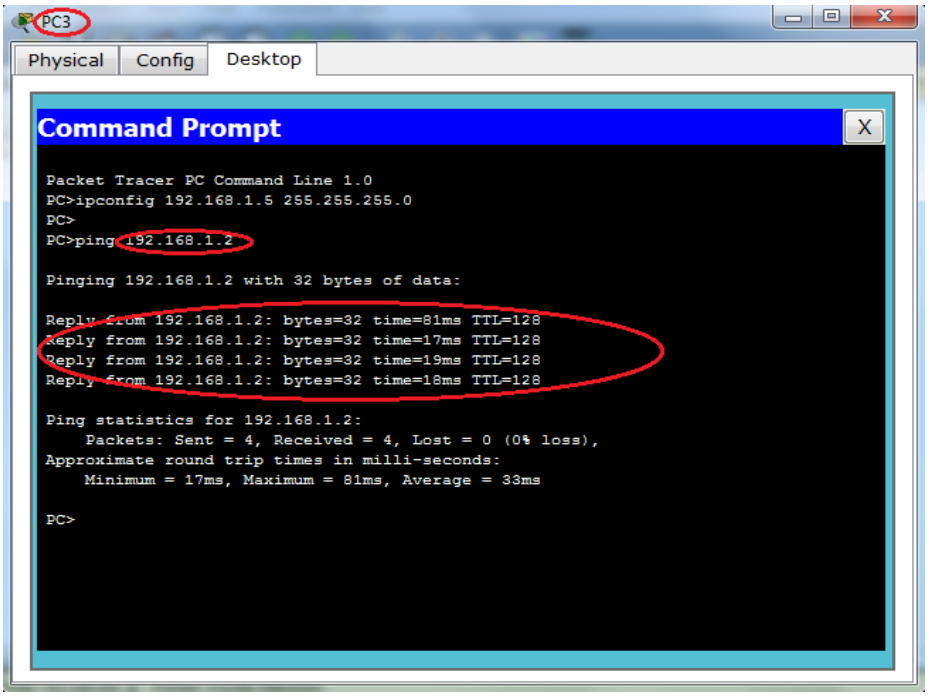

Рисунок 10. – Выполнение команды ping в командной строке

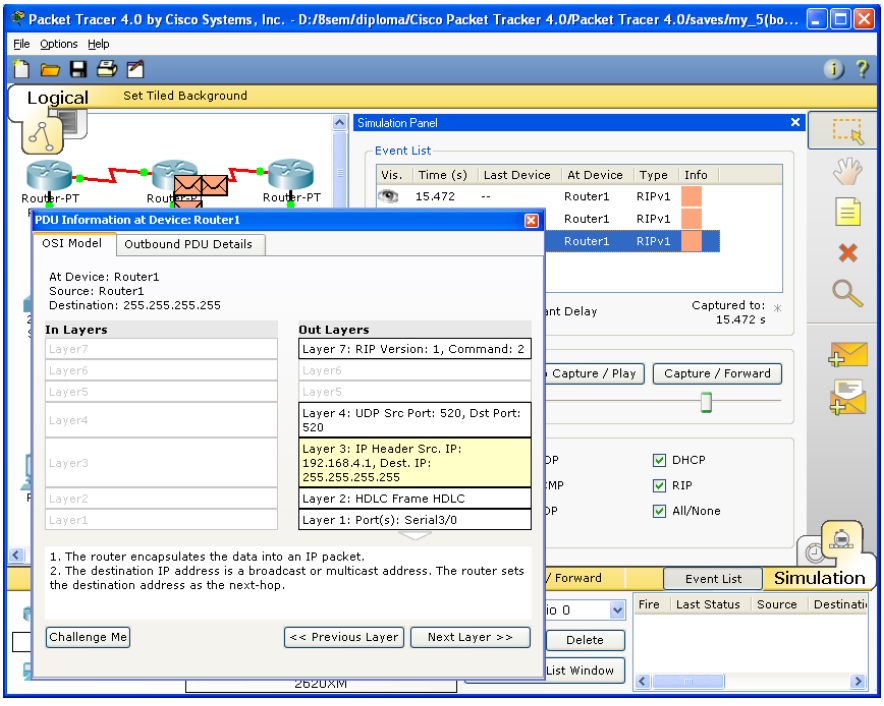

Рисунок 11. – Анализ семиуровневой модели OSI в Cisco Packet Tracer

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Ход работы в виде скриншотов экрана с пояснениями.
5. Выводы.

Контрольные вопросы

1. Какие типы сетевых устройств и соединений можно использовать в *Packet Tracer*?
2. Каким способом можно перейти к интерфейсу командной строки устройства.
3. Как добавить в топологию и настроить новое устройство?
4. Как сохранить конфигурацию устройства в *.txt* файл?
5. Как настроить маршрутизатор?
6. Какие настройки имеются у коммутатора?
7. Какие команды можно вводить в режиме командной строки?

ЛАБОРАТОРНАЯ РАБОТА № 3
Построение простейших сетей в программном обеспечении NetEmul

Цель работы: ознакомиться с основами работы с программным эмулятором ЛВС NetEmul.

Задачи:

– научиться строить простейшие модели ЛВС;

– понять разницу в построении ЛВС на концентраторах и коммутаторах.

Краткие теоретические сведения

Для запуска эмулятора NetEmul необходимо либо воспользоваться соответствующим пунктом главного меню операционной системы, либо выполнить в терминале команду *Netemul*.

Ход работы

С помощью инструмента «Вставить текстовую надпись» добавить на рабочее поле эмулятора надпись, содержащую:

– номер группы;

– ФИО студента, выполняющего работу;

– номер варианта, выданного преподавателем.

Таблица 4. – Варианты заданий к лабораторной работе

Вариант	Адрес сети/маска
1	10.0.1.0/27
2	10.0.1.32/27
3	10.0.3.64/27
4	10.0.4.96/27
5	10.0.5.128/27
6	10.0.6.160/27
7	10.0.7.192/27
8	10.0.8.224/27
9	10.0.9.0/27
10	10.0.0.32/27

Соединение двух ЭВМ напрямую

1. Выбрать исходные данные для выполнения работы согласно своему варианту.

2. Добавить на рабочее поле эмулятора два компьютера (рисунок 12), использовав кнопку «Добавить компьютер» на панели инструментов.

3. Соединить добавленные компьютеры как показано на рисунке 12. Для этого:

а) нажать кнопку «Создать соединение» на панели инструментов;

б) навести указатель на один из компьютеров;

в) зажав ЛКМ, перевести курсор на второй компьютер – за курсором от первого компьютера должна тянуться прямая линия;

г) отпустить ЛКМ – после этого должно появиться окно начальных настроек с выбором соединяемых интерфейсов;

д) подтвердить соединение между интерфейсами *eth0* и *eth0*, нажав «Соединить»;

е) если все сделано правильно, то компьютеры теперь соединены, на каждом конце соединения показан номер используемого интерфейса (в данном случае – 0), а индикатор соединения на иконке компьютера сменил цвет с красного на желтый (соединение есть, но интерфейсы не настроены).

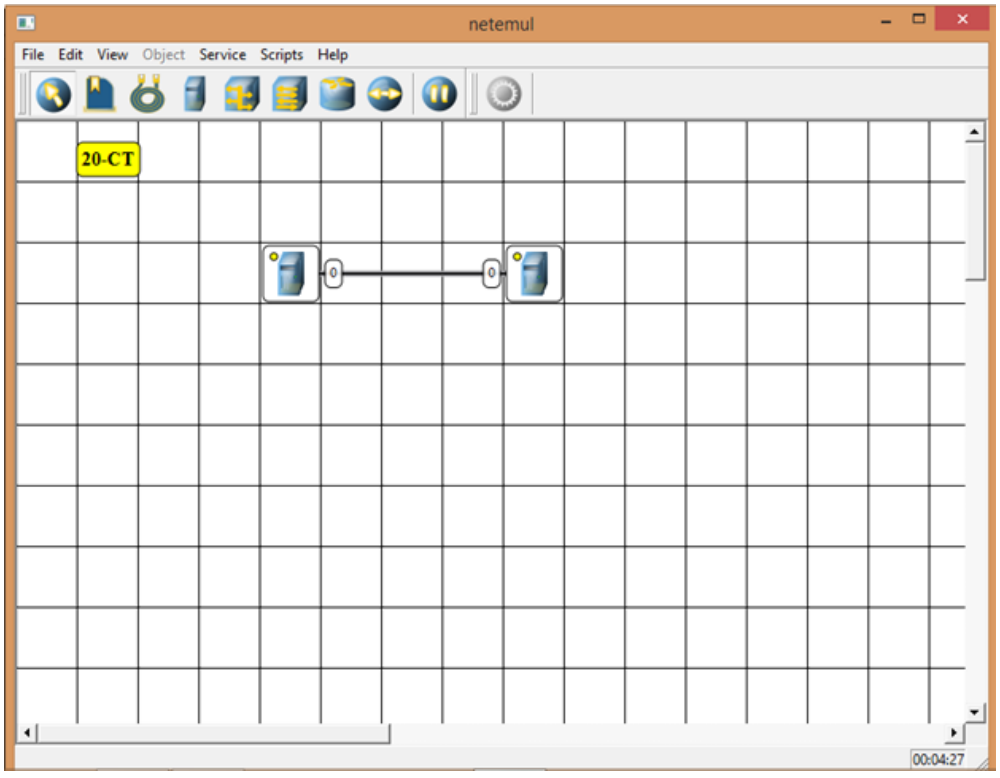

Рисунок 12. – Соединение двух ЭВМ напрямую

4. Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом. Для этого (рисунок 12):

а) выбрать инструмент «Перемещение объектов» на панели инструментов;

б) выделить первый компьютер щелчком ЛКМ;

в) вызвать контекстное меню щелчком ПКМ и выбрать пункт «Интерфейсы»;

г) в появившемся окне указать в соответствующих полях IP-адрес и маску подсети (рисунок 13);

д) подтвердить ввод последовательным нажатием кнопок «Применить» и «ОК»;

е) если все сделано правильно, то индикатор соединения на иконке компьютера должен сменить цвет с желтого на зеленый (соединение есть, и интерфейсы настроены);

ж) добавить возле каждого компьютера надпись с его IP-адресом и маской подсети, как показано на рисунке 14.

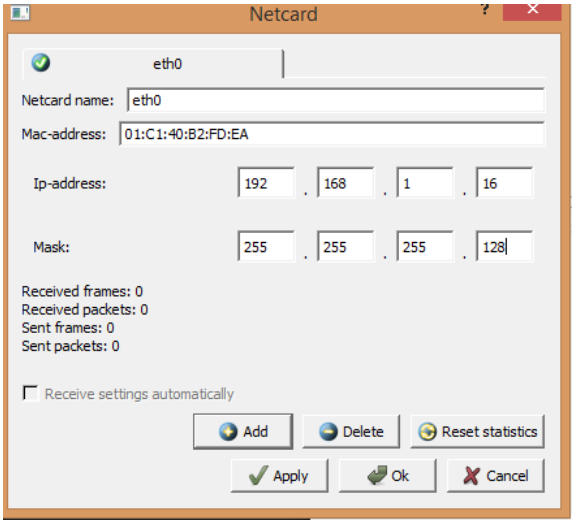

Рисунок 13. – Настойка IP-адреса и маски подсети

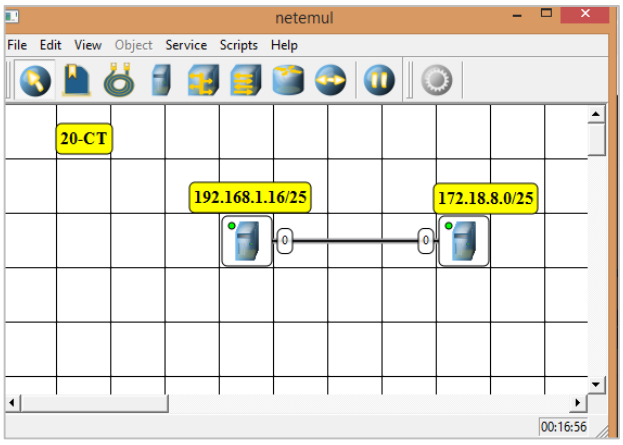

Рисунок 14. – Внешний вид получившегося соединения

5. Проверить работоспособность построенной модели ЛВС, передав пакеты от одного компьютера до другого. Для этого:

а) выбрать инструмент «Отправить данные» на панели инструментов;

б) под курсором (на рабочем поле программы) должен появиться красный круг;

в) навести курсор с красным кругом на передающий компьютер и нажать ЛКМ;

г) в появившемся окне «Отправка» указать: протокол TCP, размер данных 5 КВ;

д) нажать «Далее» – окно пропадет, а кружок под курсором сменит цвет на зеленый;

е) навести курсор с зеленым кругом на принимающий компьютер и нажать ЛКМ;

ж) в появившемся окне подтвердить интерфейс на принимающем компьютере *eth0*, нажав «Отправка»;

з) проследить за перемещением пакетов.

Построение ЛВС на концентраторах

1. Выбрать исходные данные для выполнения работы согласно своему варианту.
2. Добавить на рабочее поле эмулятора шесть компьютеров и три концентратора, как показано на рисунке 15.
3. Соединить устройства, как показано на рисунке 16.
4. Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом.
5. Добавить возле каждого компьютера надпись с его IP-адресом и маской подсети.
6. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 КВ) от одного компьютера до другого. Проследить за перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе концентраторов.

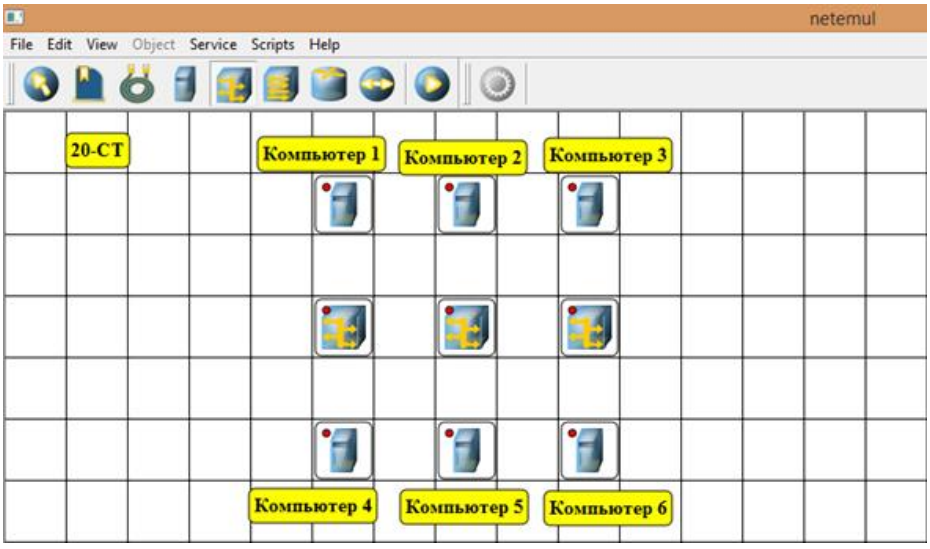

Рисунок 15. – Построение сети на концентраторах

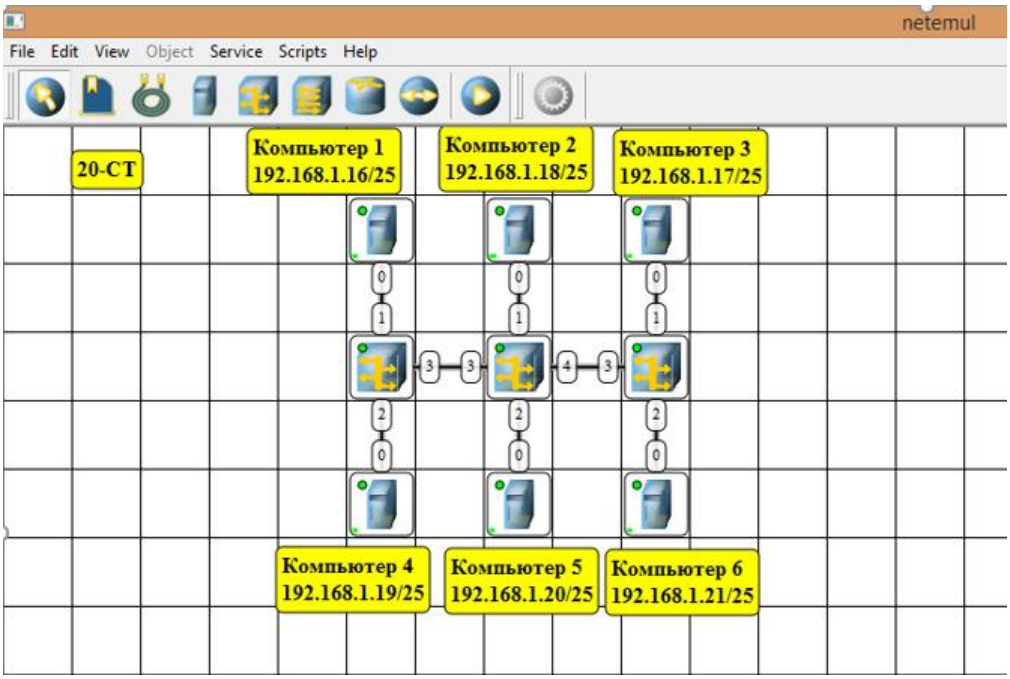

Рисунок 16. – Соединение модели

Построение ЛВС на коммутаторах

1. Выбрать исходные данные для выполнения работы согласно своему варианту.

2. Добавить на рабочее поле эмулятора пять компьютеров и два коммутатора как показано на рисунке 17.

3. Соединить устройства как показано на рисунке 17.

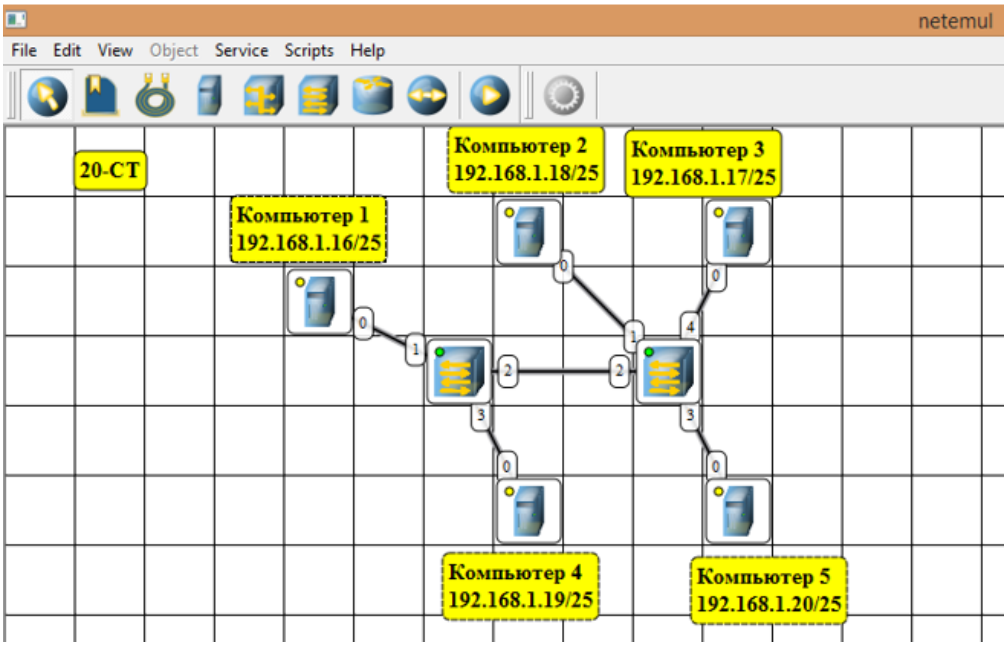

Рисунок 17. – Соединение сети на коммутаторах

4. Настроить компьютеры, задав каждому IP-адрес и маску подсети в соответствии с вариантом.

5. Добавить возле каждого компьютера надпись с его IP-адресом и маской подсети.

6. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от одного компьютера до другого. Проследить за перемещением пакетов и сделать выводы об особенностях работы ЛВС на основе коммутаторов.

После выполнения работы продемонстрировать преподавателю работоспособность построенной модели. Проект сохранить для отчета.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. По каждому пункту лабораторной должна быть приведена схема модели с указанием IP-адресов устройств и номеров интерфейсов.
4. По каждому пункту лабораторной должны быть приведены выводы по работе.
5. Ответы на контрольные вопросы.

Контрольные вопросы

1. Что такое IP-адрес?
2. Что такое маска подсети?
3. Как работает концентратор?
4. Как работает коммутатор?
5. Что представляет собой таблица коммутации?
6. Почему концентратор относится к физическому уровню модели взаимодействия открытых систем (ВОС)?
7. К какому уровню модели ВОС относят коммутаторы?

ЛАБОРАТОРНАЯ РАБОТА № 4
Адресация по протоколу IPv4

Цель работы: изучить адресацию, общую классификацию адресов в стеке TCP/IP, принцип назначения IP-адресов узлам отдельных подсетей.

Задачи:

– научиться работать с классовыми IP-адресами и масками подсетей;
– научиться разбивать сети на подсети с использованием масок.

Краткие теоретические сведения

В стеке TCP/IP используются три типа адресов:

– локальные (называемые также аппаратными);
– IP-адреса;
– символьные доменные имена.

Локальные адреса

Локальный адрес в терминологии TCP/IP – это такой тип адреса, который используется средствами базовой технологии для доставки данных в пределах подсети, которая сама является элементом составной интерсети.

В разных подсетях допустимы разные сетевые технологии, разные стеки протоколов, поэтому при создании стека TCP/IP уже заранее предполагалось наличие разных типов локальных адресов.

Если подсетью интерсети является локальная сеть, то локальный адрес – это MAC-адрес.

MAC-адрес назначается сетевым адаптерам и сетевым интерфейсам маршрутизаторов.

MAC-адреса назначаются производителями оборудования и являются уникальными, так как управляются централизованно.

Для всех существующих технологий локальных сетей MAC-адрес имеет формат 6 байт, например, 11-A0-17-3D-BC-01.

Компьютер в локальной сети может иметь несколько локальных адресов даже при одном сетевом адаптере. И наоборот, некоторые сетевые устройства вообще не имеют локальных адресов. К таким устройствам относятся глобальные порты маршрутизаторов, предназначенные для соединений типа «точка – точка».

IP-адреса – основной тип адресов сетевого уровня

На основании IP-адресов сетевой уровень передает пакеты между сетями.

IP-адреса состоят из 4 байт.

IP-адрес назначается администратором во время конфигурирования компьютеров и маршрутизаторов.

IP-адрес состоит из двух частей: номера сети и номера узла.

Номер сети может быть выбран администратором произвольно либо назначен по рекомендации специального подразделения Internet (Internet Network Information Center, InterNIC), если сеть должна работать как составная часть Internet. Обычно поставщики услуг Internet получают диапазоны адресов у подразделений InterNIC, а затем распределяют их между своими абонентами.

Номер узла в протоколе IP назначается независимо от локального адреса узла.

Маршрутизатор по определению входит сразу в несколько сетей, поэтому каждый порт маршрутизатора имеет собственный IP-адрес.

Конечный узел также может входить в несколько IP-сетей. В этом случае компьютер должен иметь несколько IP-адресов, по числу сетевых связей.

Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а одно сетевое соединение. Напоминаю, что мы поговорим об этом немного позже более подробно.

Символьные имена

Символьные имена имеют символьный вид и в IP-сетях называются доменными.

Доменные имена строятся по иерархическому признаку. Полное символьное имя в IP-сетях состоит из нескольких составляющих, которые разделяются точкой. Они перечисляются в следующем порядке (слева направо):

– сначала простое имя конечного узла;
– затем имя группы узлов (например, имя организации);
– затем имя более крупной группы (поддомена).

И так до имени домена самого высокого уровня (например, домена, объединяющего организации по географическому принципу: UA – Украина, RU – Россия, UK – Великобритания, US – США)

Примером доменного имени может служить имя *base2.sales.zil.ru*. Между доменным именем и IP-адресом узла нет никакого соответствия, поэтому необходимо использовать какие-то дополнительные таблицы или службы, чтобы узел интерсети однозначно мог определяться в сети как по доменному имени, так и по IP-адресу.

IP-адреса. Классы IP-адресов

В первую очередь IP-адреса назначаются не узлам составной сети, а *сетевым интерфейсам* узлов составной сети.

Большинство компьютеров в IP-сети имеют единственный сетевой интерфейс. Но компьютеры и другие устройства могут иметь несколько сетевых интерфейсов, и каждый интерфейс будет иметь свой собственный IP-адрес.

Например, устройство с шестью активными интерфейсами будет иметь шесть IP-адресов: по одному на каждый интерфейс в каждой сети, к которой он подключен.

IP-адрес определяет однозначно сеть и узел, который подключен к данной сети. IP-адрес имеет длину 4 байта (1 байт = 8 бит), это дает в совокупности 32 бита доступной информации.

IP-адрес записывается в виде четырех чисел, разделенных точками.

Пример:

– 128.10.2.30 – десятичная форма представления адреса: 4 (десятичных) числа, разделенных точками (.);

– 10000000 00001010 00000010 00011110 – двоичная форма представления этого же адреса, четыре восьмиразрядных числа (октета).

Так как каждое из четырех чисел – это десятичное представление 8-битного байта, то каждое число может принимать значения от 0 до 255.

Здесь надо отметить, что десятичная форма записи IP-адреса используется в основном в операционных системах как наиболее удобная при настройке.

Кроме двоичной формы, встречается шестнадцатеричная форма записи IP-адреса, например C0.94.1.3.

Использование 32-разрядных двоичных чисел позволяет создавать 4 294 967 296 уникальных IP-адресов – более чем достаточно для любой частной интрасети.

IP-адрес состоит из двух логических частей – номера сети и номера узла в сети.

Для того чтобы более рационально определиться с величиной сети и притом разграничить, какая часть IP-адреса относится к номеру сети, а какая – к номеру узла, используется система классов. Система классов использует значения первых бит адреса, но таким образом, что значения этих первых бит адреса являются признаками того, к какому классу относится тот или иной IP-адрес.

Классы IP-адресов приведены на рисунке 17.

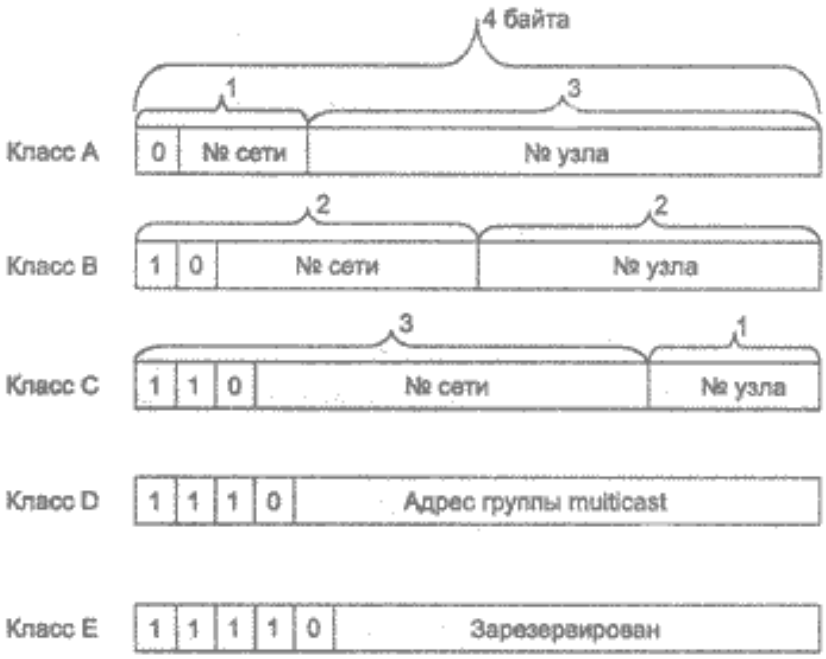

Рисунок 17. – Классы IP-адресов

В таблице 5 приводятся диапазоны номеров сетей и максимальное число узлов, соответствующих каждому классу сетей.

Таблица 5. – Классы сетей

Класс	Первые биты	Наименьший адрес сети	Наибольший адрес сети	Максимальное количество узлов
A	0	1.0.0.0	126.0.0.0	2^{24} (16 777 216 – 2)
B	10	128.0.0.0	191.255.0.0	2^{16} (65536 – 2)
C	110	192.0.1.0	223.255.255.0	2^{8} (256 – 2)
D	1110	224.0.0.0	239.255.255.255	multicast
E	11110	240.0.0.0	247.255.255.255	зарезервирован

Сети класса C являются наиболее распространенными.

Если адрес начинается с последовательности 1110, то он является адресом класса D и обозначает особый, групповой адрес – *multicast*.

Если в пакете в качестве адреса назначения указан адрес класса D, то такой пакет должны получить все узлы, которым присвоен данный адрес.

Если адрес начинается с последовательности 11110, то это значит, что данный адрес относится к классу E. Адреса этого класса зарезервированы для будущих применений.

Таким образом, можно однозначно определить, что большие сети получают адреса класса A, средние – класса B, а маленькие – класса C. В зависимости от того к какому классу (A, B, C) принадлежит адрес, номер сети

может быть представлен первыми 8, 16 или 24 разрядами, а номер хоста – последними 24, 16 или 8 разрядами.

Такова традиционная система классов, но она также недостаточно гибко определяет границы между номером сети и номером узла. С использованием классов границы проходят по границам байтов. Существует другой метод, который может проводить разделение границы между номером сети и номером узла в одном IP-адресе по границам битов.

Особые IP-адреса

Существуют некоторые значения IP-адресов, которые зарезервированы заранее, то есть существуют IP-адреса, которые предназначены для особых целей.

1. Если весь IP-адрес состоит только из двоичных нулей, то он обозначает адрес того узла, который сгенерировал этот пакет:

0 0 0 0 0 0 0 0.

Этот режим используется только в некоторых сообщениях протокола межсетевых управляющих сообщений ICMP.

2. Если в поле номера сети стоят только нули, то по умолчанию считается, что узел назначения принадлежит той же самой сети, что и узел, который отправил пакет:

0 0 0 0 0 Номер узла.

IP-адрес с нулевым номером хоста используется для адресации ко всей сети. Например, в сети класса С с номером 199.60.32 IP-адрес 199.60.32.0 означает сеть в целом.

3. Если все двоичные разряды IP-адреса равны 1, то пакет с таким адресом назначения должен рассылаться всем узлам, находящимся в той же сети, что и источник этого пакета:

1 1 1 1 1 1.

Такая рассылка называется ограниченным широковещательным сообщением (*limited broadcast*).

4. Если в поле номера узла назначения стоят только единицы, то пакет, имеющий такой адрес, рассылается всем узлам сети с заданным номером сети. Например, пакет с адресом 192.190.21.255 доставляется всем узлам сети 192.190.21.0:

Номер сети 1111 11.

Такая рассылка называется широковещательным сообщением (*broadcast*).

Предположим, что один из хостов в сети класса С с сетевым адресом 199.60.32.0 собирается направить сообщение всем остальным хостам, находящимся в той же сети. В этом случае сообщение должно быть передано на адрес 199.60.32.255.

При адресации хостов интерсети администратор должен обязательно учитывать все ограничения, которые вносятся особым назначением некоторых IP-адресов.

Таким образом, каждый администратор должен знать, что ни номер сети, ни номер узла не может состоять только из одних двоичных единиц или только из одних двоичных нулей. Отсюда следует, что максимальное количество узлов, приведенное в таблице для сетей каждого класса, на практике должно быть уменьшено на 2.

Особый смысл имеет IP-адрес, первый октет которого равен 127. Этот адрес зарезервирован для тестирования программ и взаимодействия процессов в пределах одной машины.

Когда программа посылает данные по IP-адресу 127.0.0.1, то образуется петлевое соединение.

Данные не передаются по сети, а возвращаются модулям верхнего уровня как только что принятые.

Соответственно, в IP-сети запрещается присваивать машинам IP-адреса, начинающиеся со 127. Этот адрес имеет название *loopback*.

Можно отнести адрес 127.0.0.0 ко внутренней сети модуля маршрутизации узла, а адрес 127.0.0.1 – к адресу этого модуля на внутренней сети. На самом деле любой адрес сети 127.0.0.0 служит для обозначения своего модуля маршрутизации, а не только 127.0.0.1, например 127.0.0.3.

В протоколе IP нет понятия широковещательности в том смысле, в котором оно используется в протоколах канального уровня локальных сетей, когда данные должны быть доставлены абсолютно всем узлам.

Как ограниченный широковещательный IP-адрес, так и широковещательный IP-адрес имеют свои пределы распространения в интерсети – они ограничены либо сетью, к которой принадлежит узел-источник пакета, либо сетью, номер которой указан в адресе назначения. Поэтому деление сети с помощью маршрутизаторов на части локализует широковещательный шторм пределами одной из составляющих общую сеть частей просто потому, что нет способа адресовать пакет одновременно всем узлам всех сетей составной сети.

Выше упоминалась в таблице форма группового IP-адреса – *multicast*. Так вот именно IP-адрес *multicast* означает, что данный пакет должен быть доставлен сразу нескольким узлам, которые образуют группу с номером, указанным в поле адреса.

Узлы сами идентифицируют себя, то есть определяют, к какой из групп они относятся. Один и тот же узел может входить в несколько групп. Члены какой-либо группы *multicast* не обязательно должны принадлежать одной сети. В общем случае они могут распределяться по совершенно различным сетям, находящимся друг от друга на произвольном количестве промежуточных узлов (хопов).

Групповой адрес не делится на поля номера сети и узла и обрабатывается маршрутизатором особым образом.

Основное назначение *multicast*-адресов – распространение информации по схеме «один-ко-многим».

Она работает следующим образом: хост, который хочет передавать одну и ту же информацию многим абонентам, с помощью специального протокола IGMP (Internet Group Management Protocol) сообщает о создании в сети новой мультивещательной группы с определенным адресом.

Маршрутизаторы, поддерживающие мультивещательность, распространяют информацию о создании новой группы в сетях, подключенных к портам этого маршрутизатора.

Хосты, которые хотят присоединиться к вновь создаваемой мультивещательной группе, сообщают об этом своим локальным маршрутизаторам и те передают эту информацию хосту, инициатору создания новой группы.

Чтобы маршрутизаторы могли автоматически распространять пакеты с адресом *multicast* по составной сети, необходимо использовать в конечных маршрутизаторах специальные модифицированные протоколы обмена маршрутной информацией. В общем, групповая адресация была предназначена для экономичного распространения в Internet или крупной корпоративной сети аудио- или видеопрограмм, предназначенных сразу большой аудитории слушателей или зрителей.

IP-адрес может означать одно из трех представлений:

– адрес IP-сети (группа IP-устройств, имеющих доступ к общей среде передачи: например, все устройства в сегменте Ethernet). Сетевой адрес всегда имеет биты интерфейса (хоста) адресного пространства установленными в 0;

– широковещательный адрес IP-сети (адрес для переговоров со всеми устройствами в IP-сети). Широковещательные адреса для сети всегда

имеют хостовые биты адресного пространства установленными в 1 (если сеть не разбита на подсети);

– адрес интерфейса (например, Ethernet-адаптер или PPP-интерфейс хоста, маршрутизатора, сервера печать и т.д.). Эти адреса могут иметь любые значения хостовых битов, исключая все нули или все единицы.

Для сети класса A (один байт – под адрес сети, три байта – под номер хоста): 10.0.0.0 – сеть класса A, потому что все хостовые биты равны 0; 10.0.1.0 – адрес хоста в этой сети; 10.255.255.255 – широковещательный адрес этой сети, поскольку все сетевые биты установлены в 1.

Для сети класса B (два байта – под адрес сети, два байта – под номер хоста): 172.17.0.0 – сеть класса B; 172.17.0.1 – адрес хоста в этой сети; 172.17.255.255 – сетевой широковещательный адрес.

Для сети класса C (три байта – под адрес сети, один байт – под номер хоста): 192.168.3.0 – адрес сети класса C; 192.168.3.42 – хостовый адрес в этой сети; 192.168.3.255 – сетевой широковещательный адрес.

Маски в IP-адресации

Традиционная схема деления IP-адреса на номер сети и номер узла основана на понятии класса. Класс определяется значениями нескольких первых бит адреса. Теперь можно определить, что поскольку первый байт адреса 185.23.44.206 попадает в диапазон 128–191, то этот адрес относится к классу B, а значит, номером сети являются первые два байта, дополненные двумя нулевыми байтами – 185.23.0.0, а номером узла – 0.0.44.206.

Определение номеров сети по первым байтам адреса также не вполне гибкий механизм для адресации. В качестве такого признака сейчас получили широкое распространение маски.

Маска – это тоже 32-разрядное число, она имеет такой же вид, как и IP-адрес. Маска используется в паре с IP-адресом, но не совпадает с ним.

Принцип отделения номера сети и номера узла сети с использованием маски состоит в следующем: двоичная запись маски содержит единицы в тех разрядах, которые в IP-адресе должны представляться как номер сети, и нули в тех разрядах, которые представляются как номер хоста.

Каждый класс IP-адресов (A, B и C) имеет свою маску, используемую по умолчанию.

Поскольку номер сети является цельной частью адреса, единицы в маске также должны представлять непрерывную последовательность.

Таким образом, для стандартных классов сетей маски имеют следующие значения:

– класс A – 11111111.00000000.00000000.00000000 (255.0.0.0);
– класс B – 11111111.11111111.00000000.00000000 (255.255.0.0);
– класс C – 11111111.11111111.11111111.00000000 (255.255.255.0).

Пример. Если адресу 185.23.44.206 назначить маску 255.255.255.0, получается, что единицы в маске заданы в трех байтах, значит номер сети будет 185.23.44.0, а не 185.23.0.0, как это определено правилами системы классов.

Для записи масок используются и другие форматы, например, удобно интерпретировать значение маски, записанной в шестнадцатеричном коде:

FF.FF.00.00 – маска для адресов класса B.

Часто встречается и такое обозначение: *IP-адрес/префикс сети*.

Пример. 185.23.44.206/16 – эта запись говорит о том, что маска для этого адреса содержит 16 единиц (префикс сети), или что в указанном IP-адресе под номер сети отведено 16 двоичных разрядов.

Нотация с префиксом сети также известна как бесклассовая междоменная маршрутизация (Classless Interdomain Routing – CIDR).

Таким образом, можно, снабжая каждый IP-адрес произвольной маской (не обязательно кратной 8), отказаться от понятий классов адресов и тем самым сделать более гибкой систему IP-адресации.

Пример. Для IP-адреса 129.64.134.5, назначив маску 255.255.128.0, в двоичном виде будет выглядеть так:

IP-адрес	129.64.	134.5 – 10000001.01000000.1	0000110.00000101
Маска	255.255.	128.0 – 11111111.11111111.1	0000000.00000000

В маске 17 последовательных единиц «накладываются» на IP-адрес и определяют номер сети: 10000001. 01000000. 10000000. 00000000 или 129.64.128.0, а номер узла 0000110.00000101 или 0.0.6.5.

Механизм масок очень широко распространен в IP-маршрутизации, причем маски могут использоваться для самых разных целей. С их помощью администратор может структурировать свою сеть, не требуя от поставщика услуг дополнительных номеров сетей.

На основе этого же механизма поставщики услуг могут объединять адресные пространства нескольких сетей путем введения префиксов с целью уменьшения объема таблиц маршрутизации и повышения за счет этого производительности маршрутизаторов.

Маски при записи всегда проходят с соответствующими адресами, *IP-адрес маска подсети* – именно так теперь будет описываться адрес любого хоста сети.

Порядок назначения IP-адресов. Автономные IP-адреса. Автоматизация назначения IP-адресов

Номера сетей могут назначаться либо централизованно, если сеть является частью Internet, либо произвольно, если сеть работает автономно. Номера узлов и в том и в другом случае администратор назначает самостоятельно по своему усмотрению, не выходя из разрешенного для этого класса сети диапазона.

Главную роль в централизованном распределении IP-адресов до некоторого времени играла организация InterNIC (Network Information Center), однако с ростом сети задача распределения адресов стала слишком сложной. InterNIC делегировала часть своих функций другим организациям и крупным поставщикам услуг Internet – *провайдерам*. В частности, распределением IP-адресов для подключения к сети Internet теперь занимаются провайдеры.

Возникает ситуация, когда две сети необходимо соединить глобальной связью. В таких случаях в качестве канала связи используются два маршрутизатора, соединенных по схеме «точка – точка».

Рисунок 18. – Деление сети на подсети

В ситуации, которая приведена в примере, для вырожденной сети, образованной каналом, связывающим порты двух смежных маршрутизаторов, приходится выделять отдельный номер сети, хотя в этой сети имеются всего 2 узла.

В этом случае нужно просто назначить каждому сегменту локальной сети его собственный сетевой номер класса С.

Если все сегменты локальной сети имеют собственные сетевые номера класса С, то в каждом сегменте можно создать по 254 номера хостов.

Однако если есть хотя бы небольшая вероятность того, что когда-либо в будущем сеть может быть подключена к Internet, не следует использовать такие IP-адреса. Они могут привести к конфликту с другими адресами в Internet. Чтобы избежать таких конфликтов, нужно использовать IP-адреса, зарезервированные для частных сетей.

Для этой цели зарезервированы специально несколько блоков IP-адресов, которые называются автономными.

Автономные IP-адреса

Автономные адреса зарезервированы для использования частными сетями. Они обычно используются организациями, которые имеют свою частную большую сеть – intranet (локальные сети с архитектурой и логикой Internet), но и маленькие сети часто находят их полезными.

Эти адреса не обрабатываются маршрутизаторами Internet ни при каких условиях. Эти адреса выбраны из разных классов.

Таблица 8. – Классы сетей

Класс	От IP-адреса	До IP-адреса	Всего узлов адресов в диапазоне
A	10.0.0.0	10.255.255.255	16 777 216-2
B	172.16.0.0	172.31.255.255	65 536-2
C	192.168.0.0	192.168.255.255	256-2

Эти адреса являются зарезервированными для частных сетей. Таким образом, если в будущем необходимо подключить свою сеть к Internet, то даже если трафик с одного из хостов в этой сети и попадет каким-либо образом в Internet, конфликта между адресами произойти не должно. Маршрутизаторы в Internet запрограммированы так, чтобы не транслировать сообщения, направляемые с зарезервированных адресов или на них.

Использование автономных IP-адресов имеет и недостатки, которые состоят в том, что если подключать свою сеть к Internet, то придется заново настроить конфигурацию хостов, соединяемых с Internet.

Подсеть – это метод, состоящий в том, чтобы взять сетевой IP-адрес и локально разбить его так, чтобы этот один сетевой IP-адрес мог в действительности использоваться в нескольких взаимосвязанных локальных сетях.

Один сетевой IP-адрес может использоваться только для одной сети. Разбиение на подсети – это локальная настройка. Разбиение одной большой сети на подсети значительно разгружает общий трафик и позволяет повысить безопасность всей сети в целом.

Алгоритм разбиения сети на подсети

1. Устанавливаются физические соединения (сетевые кабели и сетевые соединители, такие как маршрутизаторы).
2. Принимается решение, насколько большие/маленькие подсети нужны, исходя из количества устройств, которое будет подключено к ним, то есть, сколько IP-адресов требуется использовать в каждом сегменте сети.
3. Вычисляются соответствующие сетевые маски и сетевые адреса.
4. Раздается каждому интерфейсу в каждой сети свой IP-адрес и соответствующая сетевая маска.
5. Настраивается каждый маршрутизатор и все сетевые устройства.
6. Проверяется система, исправляются ошибки.

Пример 1. Пусть нужно разбить сеть на подсети, но имеется только один IP-адрес сети 210.16.15.0.

Решение. IP-адрес 210.16.15.0 – это адрес класса С. Сеть класса С может иметь до 254 интерфейсов (хостов) плюс адрес сети (210.16.15.0) и широковещательный адрес (210.16.15.255).

Прежде всего необходимо определить размер подсети.

Существует зависимость между количеством создаваемых подсетей и потраченными IP-адресами.

Каждая отдельная IP-сеть имеет два адреса, неиспользуемые для интерфейсов (хостов): IP-адрес сети и широковещательный адрес.

При разбивке на подсети каждая подсеть требует свой собственный уникальный IP-адрес сети и широковещательный адрес, и они должны быть корректно выбраны из диапазона адресов IP-сети, которая делится на подсети.

При разбивке IP-сети на подсети, в каждой из которых есть два сетевых адреса и два широковещательных адреса, необходимо помнить, что каждая из них уменьшит количество используемых интерфейсных (хостовых) адресов на два.

Это необходимо всегда учитывать при вычислении сетевых номеров. Следующий шаг – вычисление маски подсети и сетевых номеров.

Сетевая маска – это то, что выполняет все логические манипуляции по разделению IP-сети на подсети.

Для всех трех классов IP-сетей существуют стандартные сетевые маски:

- класс А (8 сетевых битов): 255.0.0.0;
- класс В (16 сетевых битов): 255.255.0.0;
- класс С (24 сетевых бита): 255.255.255.0.

Чтобы создать подсеть, нужно изменить маску подсети для данного класса адресов.

Номер подсети можно задать, позаимствовав нужное для нумерации подсетей количество разрядов в номере хоста. Для этого берутся левые (старшие) разряды из номера хоста, в маске же взятые разряды заполняются единицами, чтобы показать, что эти разряды теперь нумеруют не узел, а подсеть. Значения в остающихся разрядах маски подсети оставляются равными нулю; это означает, что оставшиеся разряды в номере хоста в IP-адресе должны использоваться как новый (меньший) номер хоста.

Например, чтобы разбить сетевой адрес на две подсети, нужно позаимствовать один хостовый бит, установив соответствующий бит в сетевой маске первого хостового бита в 1.

Если нужно четыре подсети, используется два хостовых бита, если восемь подсетей – три бита и т.д. Однозначно, что если нужно пять подсетей, то необходимо использовать три хостовых бита. Соответствующим образом изменяется и маска подсети:

– для адресов класса С при разбиении на 2 подсети это дает маску 11111111.11111111.11111111.10000000 или 255.255.255.128, при разбиении на 4 подсети маска в двоичном виде – 11111111.11111111.11111111.11000000, или в десятичном – 255.255.255.192. и т.д.;

– для адреса сети класса С 210.16.15.0 можно определить следующие способы разбивки на подсети:

Число подсетей	Число хостов	Сетевая маска
2	126	255.255.255.128 (11111111.11111111.11111111.10000000)
4	62	255.255.255.192 (11111111.11111111.11111111.11000000)
8	30	255.255.255.224 (11111111.11111111.11111111.11100000)
16	14	255.255.255.240 (11111111.11111111.11111111.11110000)
32	6	255.255.255.248 (11111111.11111111.11111111.11111000)
64	2	255.255.255.252 (11111111.11111111.11111111.11111100)

Далее нужно решить вопрос об адресах сетей и широковещательных адресах, а также о диапазоне IP-адресов для каждой из этих сетей.

Принимая во внимание только сетевые адреса класса С и показав только последнюю (хостовую) часть адресов, снова получается:

Сетевая маска	Подсети	Сеть	Broadcast	MinIP	MaxIP	Хосты	Всего хостов
128	2	0	127	1	126	126	
		128	255	129	254	126	252
192	4	0	63	1	62	62	
		64	127	65	126	62	
		128	191	129	190	62	
		192	255	193	254	62	248
224	8	0	31	1	30	30	
		32	63	33	62	30	
		64	95	65	94	30	
		96	127	97	126	30	
		128	159	129	158	30	
		160	191	161	190	30	
		192	223	193	222	30	
		224	255	225	254	30	240

Из этой таблицы сразу можно увидеть, что увеличение количества подсетей сокращает общее количество доступных хостовых адресов. Теперь можно назначать хостовые и сетевые IP-адреса и сетевые маски.

Пример 2. Для всех трех классов определить соответственно маски подсети и максимальное количество узлов, возможное в каждой из этих подсетей, если необходимо разбить соответственно сеть класса А, сеть класса В, сеть класса С на отдельные 4 подсети.

Решение.

Для сети класса А максимальное количество узлов – 16 777 216. Для адресации 4-х подсетей необходимо 2 разряда, значит остается 22 разряда для адресации хостов. Таким образом, каждая из четырех подсетей способна обслуживать $2^{22} – 2 = 4\ 194\ 302$ хоста в каждой из подсетей.

Число подсетей	Число хостов	Сетевая маска
4	4 194 302	255.192.0.0 (11111111.11000000.00000000.00000000)

Для сети класса B максимальное количество узлов – 65 536. Для адресации 4-х подсетей в сетевом адресе класса B также нужно использовать 2 разряда, но теперь свободными остается 14 разрядов. Таким образом, каждая из подсетей может обслуживать $2^{14} – 2 = 16\ 382$ хостов.

Число подсетей	Число хостов	Сетевая маска
4	16 382	255.255.192.0 (11111111.11111111.11000000.00000000)

Пример с сетью класса C уже рассмотрен. Теперь самое главное – уметь в двоичном виде читать IP-адреса, а с помощью маски легко можно определить номер сети и номер узла

Назначение IP-адресов узлам сети даже при не очень большом размере сети представляет для администратора очень утомительную процедуру, поэтому вторым шагом в IP-адресации разработчики решили автоматизировать этот процесс.

С этой целью был разработан протокол Dynamic Host Configuration Protocol (DHCP), который освобождает администратора от этих проблем, автоматизируя процесс назначения IP-адресов.

DHCP может поддерживать способ автоматического динамического распределения адресов, а также более простые способы ручного и автоматического статического назначения адресов. Протокол DHCP работает в соответствии с моделью клиент-сервер.

Во время старта системы компьютер, являющийся DHCP-клиентом, посылает в сеть широковещательный запрос на получение IP-адреса. DHCP-сервер откликается и посылает сообщение-ответ, содержащее IP-адрес. Предполагается, что DHCP-клиент и DHCP-сервер находятся в одной IP-сети.

При динамическом распределении адресов DHCP-сервер выдает адрес клиенту на ограниченное время, оно называется временем аренды (lease duration). Это дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру.

Основное преимущество DHCP – автоматизация работы администратора по конфигурированию стека TCP/IP на каждом компьютере. Иногда динамическое разделение адресов позволяет строить IP-сеть, количество узлов которой превышает количество имеющихся в распоряжении администратора IP-адресов.

В ручной процедуре назначения статических адресов активное участие принимает администратор, который предоставляет DHCP-серверу информацию о соответствии IP-адресов физическим адресам или другим идентификаторам клиентов. DHCP-сервер, пользуясь этой информацией, всегда выдает определенному клиенту, назначенный администратором адрес.

При автоматическом статическом способе DHCP-сервер присваивает IP-адрес из пула наличных IP-адресов без вмешательства оператора, а границы пула назначаемых адресов задает администратор при конфигурировании DHCP-сервера.

Адрес дается клиенту из пула в постоянное пользование, то есть с неограниченным сроком аренды. Между идентификатором клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие. Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту. При всех последующих запросах сервер возвращает тот же самый IP-адрес.

DHCP обеспечивает надежный и простой способ конфигурации сети TCP/IP, гарантируя отсутствие дублирования адресов за счет централизованного управления их распределением. Администратору в этом случае остается только управлять процессом назначения адресов с помощью параметра «продолжительность аренды», которая определяет, как долго компьютер может использовать назначенный IP-адрес перед тем как снова запросить его от DHCP-сервера в аренду.

Ход работы

1. IP-адрес 190.235.130.N, сетевая маска 255.255.192.0. Определить, адрес сети и адрес узла.
2. Определить маски подсети для случая разбиения сети с номером N.0.0.0 на 32 подсети.
3. Существует единая корпоративная сеть, количество узлов сети – 50 450. Этой сети выделен адрес для выхода в Internet N.124.0.0. Решено не требовать от провайдера дополнительных адресов и организовать 8 филиалов в этой сети. Необходимо:

   – выяснить, какое максимальное количество узлов может быть в каждом из филиалов;

   – вычислить сетевые маски и возможный диапазон адресов хостов для каждого из филиалов.

4. Корпоративная сеть состоит из 6 подсетей, в каждой подсети – по 25 компьютеров. Необходимо, используя один номер сети класса С 192.168.10.0, определить, правильно ли выбран размер подсети, и назначить маски и возможные IP-адреса хостам сети.

5. Разделить IP-сеть на подсети в соответствии с вариантом из таблицы. Для каждой подсети указать широковещательный адрес.

Таблица 9. – Варианты заданий

Вариант	Сеть	Подсети
1	192.168.16.0/24	5 подсетей с 100, 20, 10, 6 и 40 узлами
2	194.45.27.0/24	5 подсетей с 34, 20, 62,10 и 40 узлами*
3	56.1.1.0/16	4 подсети с 65, 22, 10 и 30 узлами
4	147.168.0.0/16	5 подсетей с 56, 16, 10 и 70 узлами
5	193.68.61.0/24	5 подсетей с 100, 20, 10 и 40 узлами
6	192.100.0.0/24	4 подсети с 80, 20, 12 и 20 узлами
7	195.18.11.0/24	4 подсети с 110, 11, 10 и 40 узлами
8	207.15.0.0/24	4 подсети с 28, 80, 10 и 40 узлами
9	222.11.0.0/24	4 подсети с 110, 20, 10 и 50 узлами
10	200.2.2.0/24	4 подсети с 100, 20, 10 и 40 узлами
11	201.111.32.0/16	5 подсетей с 170, 590, 1500, 800 и 254 узлами*
12	128.200.1.0/16	5 подсетей с 115, 300, 200, 128 и 420 узлами
13	53.11.0.0/16	5 подсетей с 165, 222, 128, 110 и 430 узлами*
14	146.77.0.0/16	5 подсетей с 550, 116, 200, 256 и 170 узлами
15	194.54.45.0/24	4 подсети с 103, 39, 10 и 16 узлами
16	142.51.0.0/16	4 подсети с 180, 120, 12 и 30 узлами
17	43.0.0.0/16	4 подсети с 151, 211, 16 и 70 узлами

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Ход работы в виде скриншотов экрана с пояснениями.
5. Выводы.

Контрольные вопросы

1. Какие бывают классы IP-адресов.
2. Как по первому байту адреса определить его класс?
3. Что такое маска, на что она указывает?
4. Для чего нужны маски переменной длины?

5. Изложите алгоритм деления сетей на подсети с помощью VLM (variable length mask).

6. Что представляют собой бесклассовые сети?

7. Объясните необходимость применения бесклассовых сетей.

ЛАБОРАТОРНАЯ РАБОТА № 5
Исследование статической маршрутизации

Цель работы: изучить статические настройки маршрутизаторов и их объединение в рамках единой ЛВС.

Задачи:

– научиться настраивать статическую маршрутизацию;

– изучить правила построения сетей в эмуляторе Cisco Packet Tracer.

Краткие теоретические сведения

ARP (от англ. Address resolution protocol – протокол преобразования адресов) – протокол в компьютерных сетях, предназначенный для определения MAC-адреса по IP-адресу другого компьютера. Суть функционирования ARP можно рассмотреть на простом примере.

> ***Пример.*** Компьютер А и компьютер Б соединены сетью Ethernet. Компьютеру А необходимо переслать пакет данных на компьютер Б, IP-адрес компьютера Б ему известен.

Назначение протокола: преобразование сетевых адресов в канальные.

Уровень (по модели OSI): канальный.

Семейство: TCP/IP.

Порт/ID: 0x0806/Ethernet.

Название: Address Resolution Protocol.

Основные реализации (клиенты): реализации стека TCP/IP в Microsoft Windows, Linux и BSD.

RARP (от англ. Reverse Address Resolution Protocol – обратный протокол преобразования адресов) – протокол сетевого уровня модели OSI, выполняет обратное отображение адресов, то есть преобразует физический адрес в IP-адрес.

Протокол применяется во время загрузки узла (например, компьютера), когда он посылает групповое сообщение-запрос со своим физическим адресом. Сервер принимает это сообщение и просматривает свои таблицы (либо перенаправляет запрос куда-либо еще) в поисках соответствующего физическому адресу IP-адреса. После обнаружения найденный адрес отсылается обратно на запросивший его узел. Другие станции также могут фиксировать этот диалог и локально сохранить эту информацию в своих ARP-таблицах.

RARP позволяет разделять IP-адреса между нечасто используемыми хост-узлами. После использования каким-либо узлом IP-адреса он может быть освобожден и выдан другому узлу.

RARP является дополнением к ARP.

RARP отличается от обратного ARP (Inverse Address Resolution Protocol, или InARP), который предназначен для получения IP-адреса, соответствующего MAC-адресу другого узла. InARP является дополнением к протоколу разрешения адресов и используется для обратного поиска. RARP является скорее аналогом DHCP/BOOTP.

Маршрутизация

Протоколы маршрутизации – это правила, по которым осуществляется обмен информацией о путях передачи пакетов между маршрутизаторами. Протоколы характеризуются временем сходимости, потерями и масштабируемостью. В настоящее время используется несколько протоколов маршрутизации. Каждый протокол имеет сильные и слабые стороны.

Одна из главных задач маршрутизатора состоит в определении наилучшего пути к заданному адресату. Маршрутизатор определяет пути (маршруты) к адресатам или из статической конфигурации, введенной администратором, или динамически на основании маршрутной информации, полученной от других маршрутизаторов. Маршрутизаторы обмениваются маршрутной информацией с помощью протоколов маршрутизации. Маршрутизатор хранит таблицы маршрутов в оперативной памяти. Таблица маршрутов – это список наилучших известных доступных маршрутов. Маршрутизатор использует эту таблицу для принятия решения куда направлять пакет. Для просмотра таблицы маршрутов следует использовать команду *show ip route*. Даже если на некотором маршрутизаторе X не задавались никакие команды маршрутизации, то он все равно строит таблицу маршрутов для непосредственно подсоединенных к нему сетей. Например:

```
. . .
C       192.168.4.0/24 is directly connected, Ethernet0
        10.0.0.0/16 is subnetted, 3 subnets
C       10.3.0.0 is directly connected, Serial0
C       10.4.0.0 is directly connected, Serial1
C       10.5.0.0 is directly connected, Ethernet1
```

Маршрут на непосредственно подсоединенные сети отображается на интерфейс маршрутизатора, к которому они присоединены. Здесь /24 обозначает маску 255.255. 255.0, а /16 – 255.255.0.0.

Таблица маршрутов отображает сетевые префиксы (адреса сетей) на выходные интерфейсы. Когда X получает пакет, предназначенный для 192.168.4.46, он ищет префикс 192.168.4.0/24 в таблице маршрутов. Согласно таблице пакет будет направлен на интерфейс Ethernet0. Если X получит пакет для 10.3.21.5, он направит его на Serial0.

Эта таблица показывает четыре маршрута для непосредственно подсоединенных сетей. Они имеют метку C. Маршрутизатор X отфильтровывает все пакеты, направляемые к сетям, не указанным в таблице маршрутов. Для направления пакетов к другим адресатам необходимо в таблицу включить дополнительные маршруты.

Новые маршруты могут быть добавлены двумя методами:

– *статическая маршрутизация* – администратор вручную определяет маршруты к сетям назначения;

– *динамическая маршрутизация* – маршрутизаторы следуют правилам, определяемым протоколами маршрутизации для обмена информацией о маршрутах и выбора лучшего пути.

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

Для конфигурации статической маршрутизации в маршрутизаторах Cisco используют две версии команды *ip route*.

Маршрутизация по умолчанию

Совсем необязательно, чтобы каждый маршрутизатор обслуживал маршруты ко всем возможным сетям назначения. Вместо этого маршрутизатор хранит маршрут по умолчанию или шлюз последнего пристанища (last resort). Маршруты по умолчанию используются, когда маршрутизатор не может поставить в соответствие сети назначения строку в таблице маршрутов. Маршрутизатор должен использовать маршрут по умолчанию для отправки пакетов другому маршрутизатору. Следующий маршрутизатор будет иметь маршрут к этой сети назначения или иметь свой маршрут по умолчанию к третьему маршрутизатору и т.д. В конечном счете, пакет будет маршрутизирован на маршрутизатор, имеющий маршрут к сети назначения.

Маршрут по умолчанию может быть статически введен администратором или динамически получен из протокола маршрутизации.

Так как все IP-адреса принадлежат сети 0.0.0.0 с маской 0.0.0.0, то в простейшем случае надо использовать команду

```
ip route 0.0.0.0 0.0.0.0 [адрес следующего хопа | выходной интерфейс]
```

Ручное задание маршрута по умолчанию на каждом маршрутизаторе подходит для простых сетей. В сложных сетях необходимо организовать динамический обмен маршрутами по умолчанию.

Интерфейс «петля»

На сетевых устройствах можно создавать сетевые интерфейсы, не связанные с реальными каналами для передачи данных и назначать на них IP-адреса с масками. Такие интерфейсы называют *петлями* (loopback). Петли полезны при поэтапном проектировании сетей. Если к какому-то реальному сетевому интерфейсу маршрутизатора в дальнейшем будет подсоединена подсеть, то в начале на маршрутизаторе создается loopback, настраивается в плане взаимодействия с остальными участками сети и лишь затем заменяется на реальный интерфейс. Интерфейс «петля» появляется после команды *interface loopback N* или сокращенно *int lN*, где *N* – целое неотрицательное число – номер петли. Например:

```
Router(conf)>int l0 1.1.1.1 255.0.0.0
```

Команда trace

Команда *trace* является идеальным способом для выяснения того, куда отправляются данные в сети. Эта команда использует ту же технологию протокола ICMP, что и команда *ping*, только вместо проверки связи между отправителем и получателем она проверяет каждый шаг на пути. Команда *trace* использует способность маршрутизаторов генерировать сообщения об ошибке при превышении пакетом своего установленного времени жизни (Time To Live, TTL). Эта команда посылает несколько пакетов и выводит на экран данные про время прохождения туда и назад для каждого из них. Преимущество команды *trace* заключается в том, что она показывает очередной достигнутый маршрутизатор на пути к пункту назначения. Это очень мощное средство для локализации отказов на пути от отправителя к получателю.

Таблица 12. – Варианты ответов утилиты trace

Символ	Значение
!H	Зондирующий пакет был принят маршрутизатором, но не переадресован, что обычно бывает из-за списка доступа
P	Протокол недосягаем
N	Сеть недосягаема
U	Порт недосягаем
*	Превышение границы ожидания

Практическая часть

1. Подключиться к маршрутизатору Router1 и посмотреть его ARP-таблицу:

```
Router1# show arp
```

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.1               -   00D0.58B7.80A1  ARPA   FastEthernet0/0
```

Она содержит только одну строку о MAC-адресе своего Ethernet-интерфейса с IP-адресом 10.1.1.1.

2. Подключиться к маршрутизатору Router2 и посмотреть его ARP-таблицу. Она должна содержать только одну строку о MAC-адресе своего Ethernet-интерфейса с IP-адресом 10.1.1.2:

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.2               -   00D0.BA88.005C  ARPA   FastEthernet0/0
```

3. Выполнить *ping* Ethernet-интерфейса маршрутизатора Router1:

```
Router2# ping 10.1.1.1
```

4. Снова посмотреть ARP-таблицу. Она должна содержать уже две строки. Должна появиться запись о MAC-адресе Ethernet-интерфейса Router1 с IP-адресом 10.1.1.1.

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.1               0   00D0.58B7.80A1  ARPA   FastEthernet0/0
Internet  10.1.1.2               -   00D0.BA88.005C  ARPA   FastEthernet0/0
```

5. Подключиться к маршрутизатору Router1 и посмотреть его ARP-таблицу. Она должна содержать уже две строки:

```
Protocol  Address          Age (min)  Hardware Addr   Type   Interface
Internet  10.1.1.1               -   00D0.58B7.80A1  ARPA   FastEthernet0/0
Internet  10.1.1.2               2   00D0.BA88.005C  ARPA   FastEthernet0/0
```

Должна появиться запись о MAC-адресе Ethernet-интерфейса маршрутизатора Router2 с IP-адресом 10.1.1.2. Router1 для ответа на *ping* от Router2 должен был знать о MAC-адресе Ethernet-интерфейса Router2 с IP-адресом 10.1.1.2.

Статические маршруты. Порядок настройки

1. Подключиться к маршрутизатору Router2. Не принимались пакеты с адресов 172.16.10.1 и 172.16.10.2. Посмотреть таблицу маршрутов:

```
Router2# show ip route
```

```
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
```

Видны непосредственно присоединенные сети. Нет маршрута к сети 172.16.10.0/24. Добавить маршрут к сети 172.16.10.0/24 через адрес 10.1.1.1 ближайшего хопа на пути к этой сети:

```
Router2(config)# ip route 172.16.10.0 255.255.255.0 10.1.1.1
```

Здесь и далее 172.16.10.0/24 – это сокращенная запись – определение подсети 172.16.10.0 с маской 255.255.255.0. В маске 255.255.255.0 содержится 24 единицы, что и обозначается /24.

2. Успешно виден через запрос *ping* serial-интерфейс Router1:

```
Router2# ping 172.16.10.1
```

3. Снова посмотреть таблицу маршрутов:

```
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
     172.16.0.0/24 is subnetted, 1 subnets
S       172.16.10.0 [1/0] via 10.1.1.1
```

При этом не удается выполнить команду *ping* для serial-интерфейса Router4:

```
Router2# ping 172.16.10.2
```

Это происходит потому, что ICMP-пакеты не имеют обратного ответа от Router4, так как на Router4 не прописаны маршруты.

4. Подключиться к маршрутизатору router4. Посмотреть таблицу маршрутов:

```
Router4# show ip route
```

```
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

Нет маршрута к сети 10.1.1.0/24. Добавить маршрут к сети 10.1.1.0/24 через адрес 172.16.10.1 ближайшего хопа на пути к этой сети:

```
Router4(config)# ip route 10.1.1.0 255.255.255.0 172.16.10.1
```

5. Снова посмотреть таблицу маршрутов:

```
     10.0.0.0/24 is subnetted, 1 subnets
S       10.1.1.0 [1/0] via 172.16.10.1
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

6. Теперь все сетевые интерфейсы в сети должны быть видны для ICMP-пакетов из каждого сетевого устройства. Проверить это.

Маршрутизация по умолчанию

Сетевые устройства Router2 и Router4 имеют только по одному выходу в сеть: через интерфейсы с адресами 10.1.1.1 и 172.16.10.1 соответственно, поэтому можно не определять, на какие подсети маршрутизируем пакеты и использовать маршрутизацию по умолчанию.

1. Вначале удалить старые маршруты:

```
Router2(config)# no ip route 172.16.10.0 255.255.255.0 10.1.1.1
```

```
Router4(config)# no ip route 10.1.1.0 255.255.255.0 172.16.10.1
```

2. Назначить маршруты по умолчанию:

```
Router2(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.1
```

```
Router4(config)# ip route 0.0.0.0 0.0.0.0 172.16.10.1
```

3. Посмотреть таблицу маршрутов на всех устройствах:

```
Router2# sh ip route
```

```
Gateway of last resort is 10.1.1.1 to network 0.0.0.0

     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 10.1.1.1
```

```
Router4# sh ip route
```

```
Gateway of last resort is 172.16.10.1 to network 0.0.0.0

     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
S*   0.0.0.0/0 [1/0] via 172.16.10.1
```

4. Все сетевые интерфейсы в сети видны для команды *ping* из каждого сетевого устройства. Проверить это.

Loopback

1. Определить интерфейс-петлю на устройстве Router4:

```
Router4(conf)# int loopback 0 1.1.1.1 255.255.255.0
```

2. Прописать на устройстве Router1 маршрут на сеть петли:

```
Router1(conf)# ip route 1.1.1.0 255.255.255.0 172.16.10.2
```

3. Подключиться к устройству Router2 и выполнить команду *ping* для созданной петли:

```
Router2#ping 1.1.1.1
```

Сохранить проект в целом и конфигурацию каждого маршрутизатора в отдельный файл.

Контрольные вопросы

1. Как отправитель узнает MAC-адрес получателя?
2. Как посмотреть ARP-таблицу?
3. Когда в ARP-таблице появляются новые строки?
4. Что такое таблица маршрутов?
5. Если администратор не настраивал никаких маршрутов, то что она будет содержать?
6. Чем статическая маршрутизация отличается от динамической?
7. Какие две формы задания статической маршрутизации вы знаете?
8. Как в команде маршрутизации определяется сеть назначения?
9. Объясните значения полей в командах маршрутизации.
10. Когда используется маршрутизация по умолчанию?
11. Когда используют интерфейс «петля»?
12. Как работает команда трассировки?

Порядок выполнения и сдачи работы

1. Изучить краткие теоретические сведения и практическую часть.
2. Сдать преподавателю теоретический материал путем ответа на контрольные вопросы.
3. Выполнить в Packet Tracer практическую часть.
4. Получить вариант и выполнить в Packet Tracer задание для самостоятельной работы.

5. Предъявить преподавателю результат выполнения пунктов 8 и 9 задания для самостоятельной работы.

6. Оформить отчет. Содержание отчета см. ниже.

7. Защитить выполненную работу.

Задание для самостоятельной работы

1. Построить в Packet Tracer топологию, представленную на рисунке 19. Использовать необходимые маршрутизаторы.

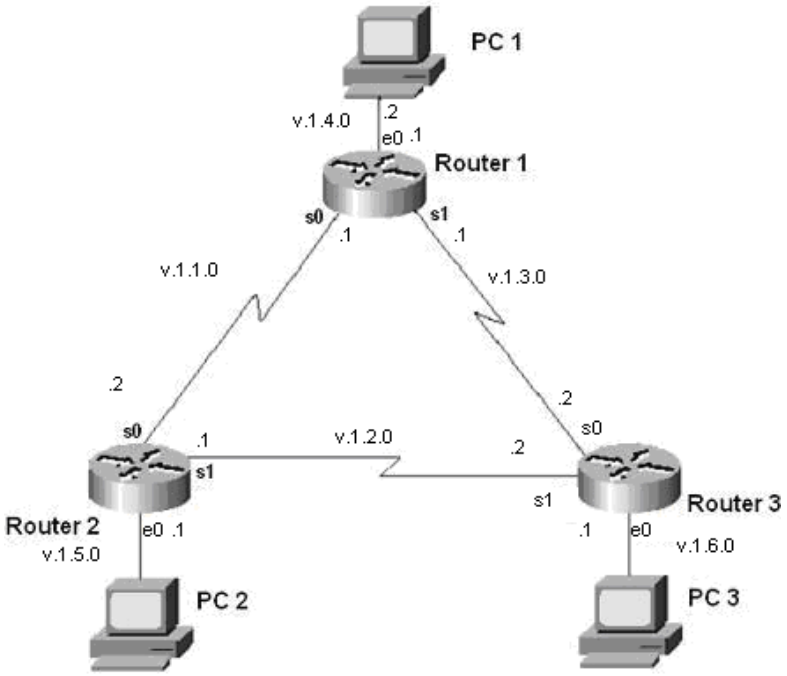

Рисунок 19. – Варианты топологии

В сети шесть подсетей. Каждый маршрутизатор подключен к трем подсетям.

2. На каждом маршрутизаторе настроить используемые интерфейсы и посмотреть соседей командой *show cdp neighbors*. Сделать скриншоты.

3. Назначить интерфейсам сети адреса согласно рисунку 19 и таблице 2 в которых v – это номер варианта. Все маски – 255.255.255.0. Назначить шлюзы по умолчанию для компьютеров согласно таблице 11.

Таблица 13. – Варианты заданий к работе

	v.1.1.0	v.1.2.0	v.1.3.0	v.1.4.0	v.1.5.0	v.1.6.0
1	2	3	4	5	6	7
Router1	S0:v.1.1.1		S1:v.1.3.1	E0:v.1.4.1		
Router2	S0:v.1.1.2	S1:v.1.2.1			E0:v.1.5.1	
Router3		S0:v.1.2.2	S1:v.1.3.2			E0:v.1.6.1

Окончание таблицы 13

1	2	3	4	5	6	7
PC1				E0:v.1.4.2		
PC2					E0:v.1.5.2	
PC3						E0:v.1.6.2

4. Проверить факт назначения адресов путем выполнения на каждом маршрутизаторе команд *show running-config* и *show ip interface brief*. Для компьютеров использовать команду *ipconfig*.

5. Проверить правильность назначения адресов путем выполнения на каждом маршрутизаторе команд *ping* к непосредственным соседним узлам. Например, на маршрутизаторе Router1 выполнить:

```
Router1# ping v.1.1.2
Router1# ping v.1.3.2
Router1# ping v.1.4.2
```

6. Связать между собой компьютеры PC1, PC2 и PC3. Для этого осуществить на маршрутизаторах настройку статической маршрутизации. В каждом маршрутизаторе прописать маршруты на удаленные Ethernet-сети. Для решения поставленной задачи маршрутизировать пакеты на удаленные сети последовательных соединений не надо.

У каждого маршрутизатора есть по два сетевых интерфейса на удаленные Ethernet-сети. Всего надо прописать шесть статических маршрутов.

Чтобы из маршрутизатора Router1 достичь удаленную Ethernet-сеть v.1.5.0/24, пакеты можно направить на IP-адрес 1.1.1.2 ближайшего внешнего интерфейса на пути в эту сеть. Это сделает команда

```
router1(config)# ip route 1.1.5.0 255.255.255.0 1.1.1.2
```

Задать остальные пять команд маршрутизации.

7. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты.

8. На каждом маршрутизаторе сделать скриншоты расширенной команды *ping*:

– на маршрутизаторе Router1 от PC2 к PC3;

– на маршрутизаторе Router2 от PC1 к PC3;

– на маршрутизаторе Router3 от PC1 к PC2.

Пример. Результат расширенного пинга на маршрутизаторе Router1 от PC2 к PC3 для варианта 1 (v = 1):

```
router1#ping
Protocol [ip]:
Target IP address: 1.1.6.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]:y
Source address or interface:1.1.5.2
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.6.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms
```

9. На каждом компьютере сделать скриншоты выполнения команд трассировки *tracert* других компьютеров (всего шесть скриншотов).

Пример. Трассировка из PC1 на PC2 для варианта 1 (v = 1):

```
PC1:#tracert 1.1.5.2


"Type escape sequence to abort."
Tracing the route to 1.1.5.2

 1 1.1.4.1 0 msec 16 msec 0 msec
 2 1.1.1.2 20 msec 16 msec 16 msec
 3 1.1.5.2 20 msec 16 msec *
```

10. Сохранить проект.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.
4. Ход выполнения работы со скриншотами.
5. Настройки роутеров в текстовом файле в приложении.
6. Выводы.

ЛАБОРАТОРНАЯ РАБОТА № 6
Исследование динамической маршрутизации

Цель работы: ознакомиться с работой основных динамических протоколов маршрутизации.

Задачи:

– изучить принципы маршрутизации с использованием протоколов дистанционно-векторного типа;

– изучить принципы маршрутизации с использованием протоколов состояния связи.

Краткие теоретические сведения

Статическая маршрутизация не подходит для больших, сложных сетей потому, что обычно сети включают избыточные связи, многие протоколы и смешанные топологии. Маршрутизаторы в сложных сетях должны быстро адаптироваться к изменениям топологии и выбирать лучший маршрут из многих вариантов.

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

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

Маршрутизаторы используют метрики для оценки или измерения маршрутов. Когда от маршрутизатора к сети назначения существует много маршрутов, и все они используют один протокол маршрутизации, то маршрут с наименьшей метрикой рассматривается как лучший. Если используются разные протоколы маршрутизации, то для выбора маршрута используются

административные расстояния, которые назначаются маршрутам операционной системой маршрутизатора.

RIP использует в качестве метрики количество переходов (хопов). EIGRP использует сложную комбинацию факторов, включающую полосу пропускания канала и его надежность.

Результаты работы маршрутизирующих протоколов заносятся в таблицу маршрутов, которая постоянно изменяется при смене ситуации в сети. Рассмотрим типичную строку в таблице маршрутов, относящуюся к динамической маршрутизации:

R 192.168.14.0/24 [120/3] via 10.3.0.1 00:00:06 Serial0

Здесь R определяет протокол маршрутизации. Так, R означает RIP, а O – OSPF и т.д. Запись [120/3] означает, что этот маршрут имеет административное расстояние 120 и метрику 3. Эти числа маршрутизатор использует для выбора маршрута. Элемент 00:00:06 определяет время, когда обновилась данная строка. Serial0 это локальный интерфейс, через который маршрутизатор будет направлять пакеты к сети 192.168.14.0/24 через адрес 10.3.0.1.

Для того чтобы динамические протоколы маршрутизации обменивались информацией о статических маршрутах, следует осуществлять дополнительное конфигурирование.

Дистанционно-векторная маршрутизация

Эта маршрутизация базируется на алгоритме Белмана – Форда. Через определенные моменты времени маршрутизатор передает соседним маршрутизаторам всю свою таблицу маршрутизации. Такие простые протоколы как RIP и IGRP просто распространяют информацию о таблицах маршрутов через все интерфейсы маршрутизатора в широковещательном режиме без уточнения точного адреса конкретного соседнего маршрутизатора.

Соседний маршрутизатор, получая широковещательный пакет, сравнивает информацию со своей текущей таблицей маршрутов. В нее добавляются маршруты к новым сетям или маршруты к известным сетям с лучшей метрикой. Происходит удаление несуществующих маршрутов. Маршрутизатор добавляет свои собственные значения к метрикам полученных маршрутов. Новая таблица маршрутизации снова распространяется по соседним маршрутизаторам (рисунок 20).

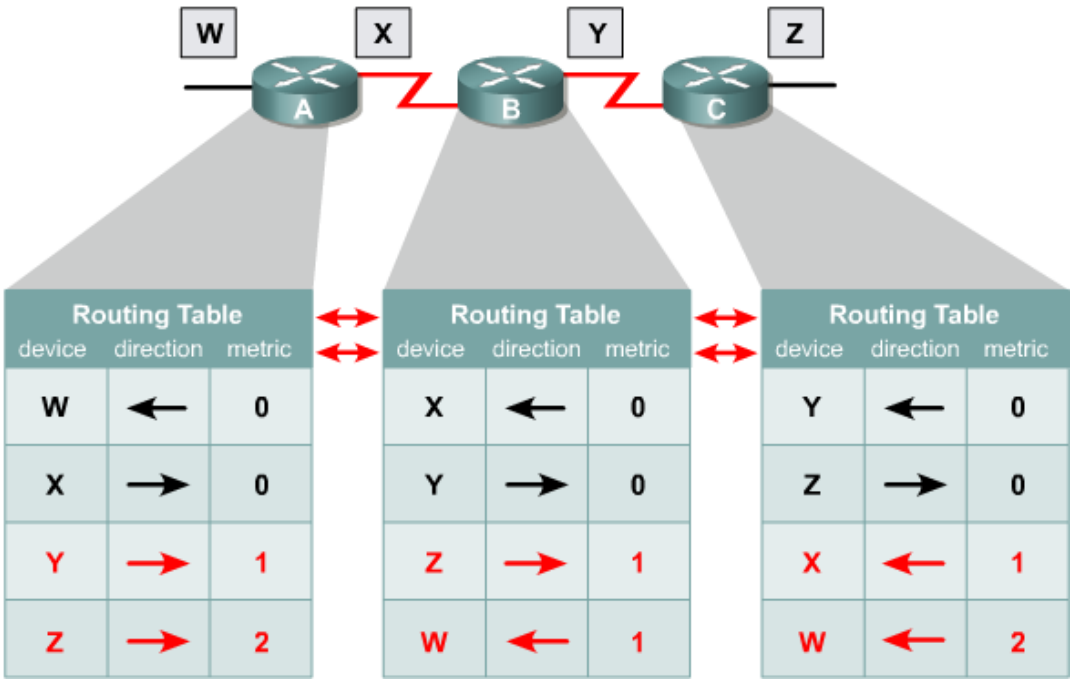

Рисунок 20. – Дистанционно-векторная маршрутизация

Протоколы состояния связи

Эти протоколы предлагают лучшую масштабируемость и сходимость по сравнению с дистанционно-векторными протоколами. Протокол базируется на алгоритме Дейкстры, который часто называют алгоритмом «кратчайший путь – первым» (shortest path first (SPF). Наиболее типичным представителем является протокол OSPF (Open Shortest Path First).

Маршрутизатор принимает во внимание состояние связи интерфейсов других маршрутизаторов в сети. Маршрутизатор строит полную базу данных всех состояний связи в своей области, то есть имеет достаточно информации для создания своего отображения сети. Каждый маршрутизатор затем самостоятельно выполняет SPF-алгоритм на своем собственном отображении сети или базе данных состояний связи для определения лучшего пути, который заносится в таблицу маршрутов. Эти пути к другим сетям формируют дерево с вершиной в виде локального маршрутизатора.

Маршрутизаторы извещают о состоянии своих связей всем маршрутизаторам в области. Такое извещение называют LSA (link-state advertisements).

В отличие от дистанционно-векторных маршрутизаторов, маршрутизаторы состояния связи могут формировать специальные отношения со своими соседями.

Далее обновление маршрутов производится только при смене состояний связи или если состояние не изменилось в течение определенного

интервала времени. Если состояние связи изменилось, то частичное обновление пересылается немедленно. Оно содержит только состояния связей, которые изменились, а не всю таблицу маршрутов.

Протоколы состояния связи имеют более быструю сходимость и лучшее использование полосы пропускания по сравнению с дистанционно-векторными протоколами. Они превосходят дистанционно-векторные протоколы для сетей любых размеров, однако имеют два главных недостатка: повышенные требования к вычислительной мощности маршрутизаторов и сложное администрирование.

Сходимость

Этот процесс одновременно и совместный, и индивидуальный. Маршрутизаторы разделяют между собой информацию, но самостоятельно пересчитывают свои таблицы маршрутизации. Для того чтобы индивидуальные таблицы маршрутизации были точными, все маршрутизаторы должны иметь одинаковое представление о топологии сети. Если маршрутизаторы договорились о топологии сети, то имеет место их сходимость. Быстрая сходимость означает быстрое восстановление после обрыва связей и других изменений в сети. О протоколах маршрутизации и о качестве проектирования сети судят главным образом по сходимости.

Когда маршрутизаторы находятся в процессе сходимости, сеть восприимчива к проблемам маршрутизации. Если некоторые маршрутизаторы определили, что некоторая связь отсутствует, то другие ошибочно считают эту связь присутствующей. Если это происходит, то отдельная таблица маршрутов будет противоречива, что может привести к отбрасыванию пакетов и петлям маршрутизации.

Невозможно, чтобы все маршрутизаторы в сети одновременно обнаружили изменения в топологии. В зависимости от использованного протокола, может пройти много времени пока все процессы маршрутизации в сети сойдутся. На это влияют следующие факторы:

– расстояние в хопах до точки изменения топологии;
– число маршрутизаторов, использующих динамические протоколы;
– полоса пропускания и загрузка каналов связи;
– загрузка маршрутизаторов.

Эффект некоторых факторов может быть уменьшен при тщательном проектировании сети.

Конфигурирование динамической маршрутизации

Для конфигурирования динамической маршрутизации используются две основные команды: *router* и *network*. Команда *router* запускает процесс маршрутизации и имеет следующий вид:

```
Router(config)# router protocol [keyword]
```

Протокол RIP

Ключевые характеристики протокола RIP:

– маршрутизация на основании вектора расстояния;

– метрика при выборе пути в виде количества переходов (хопов);

– максимально допустимое количества хопов – 15;

– по умолчанию пакеты актуализации маршрутной информации посылаются в режиме широковещания каждые 30 секунд.

Выбор протокола RIP как протокола маршрутизации осуществляется командой

```
Router(config)# router rip
```

Процесс маршрутизации связывает интерфейсы с соответствующими адресами и начинает обработку пакетов в заданных сетях.

В показанном на рисунке 21 примере команды *network 1.0.0.0* и *network 2.0.0.0* задают непосредственно подключенные к маршрутизатору Cisco A сети.

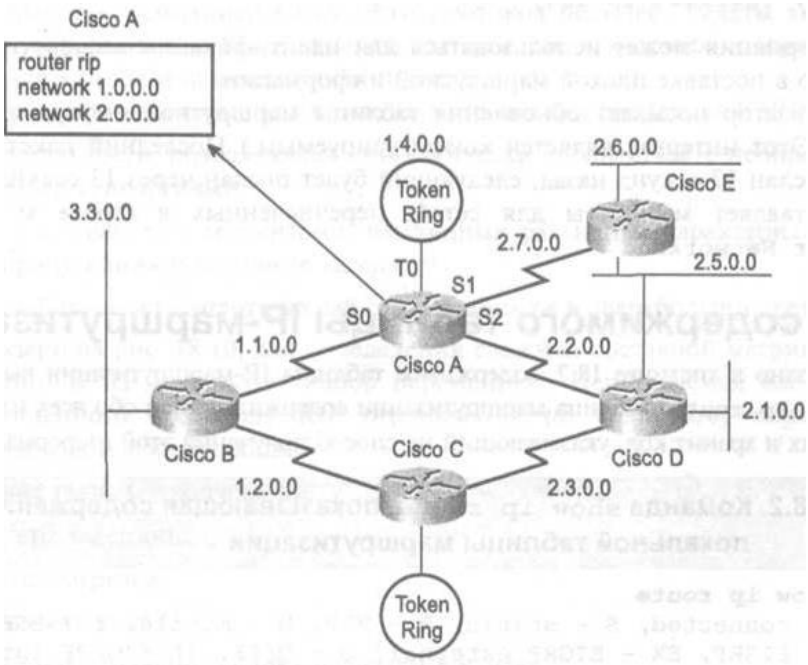

Рисунок 21. – Схема сети

Команда *debug ip rip* выводит содержание пакетов актуализации маршрутной информации протокола RIP в том виде, в котором эти данные посылаются и принимаются.

Протокол IGRP

IGRP представляет собой протокол маршрутизации по вектору расстояния (разработан компанией Cisco). Этот протокол посылает с 90-секундным интервалом пакеты актуализации маршрутной информации, в которых содержатся сведения о сетях для конкретной автономной системы. Этот протокол характеризует универсальность, позволяющую автоматически справляться со сложными топологиями, и гибкость в работе с сегментами, имеющими разные характеристики по полосе пропускания и величины задержки. Используемая им метрика не имеет свойственных протоколу RIP ограничений по количеству переходов. Она включает следующие составляющие: ширина полосы пропускания, величина задержки, уровень загрузки, надежность канала, размер максимального блока передачи в канале.

Выбор протоколу IGRP в качестве протокола маршрутизации осуществляется с помощью команды

Router (config)# router igrp autonomous-system,

где параметр *autonomous-system* называют номером автономной системы и он идентифицирует вычислительный процесс IGRP-маршрутизации. Процессы в маршрутизаторах сети с одинаковым номером *autonomous-system* будут коллективно использовать маршрутную информацию.

В показанном на рисунке 22 примере на маршрутизаторе Cisco A запущен маршрутизирующий процесс, организующий IGRP-маршрутизацию в автономной системе с номером 109. В маршрутизации будут участвовать сети 1.0.0.0 и 2.0.0.0.

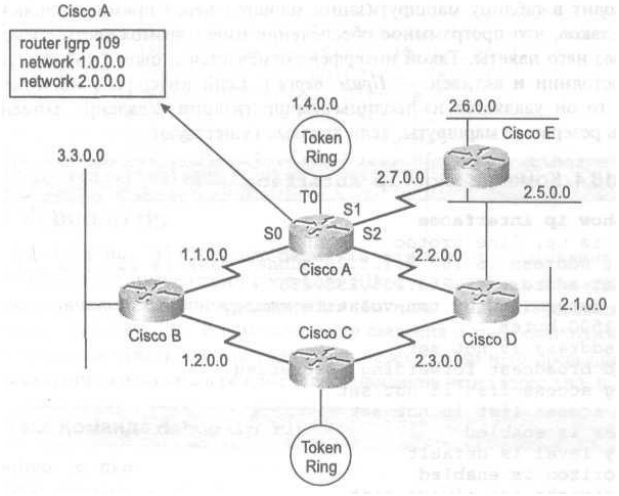

Рисунок 22. – Схема настройки маршрутизатора

Команды *debug ip igrp transactions* и *debug ip igrp events* выводят содержание пакетов актуализации маршрутной информации протокола IGRP в том виде, в котором эти данные посылаются и принимаются.

Протокол OSPF

OSPF – это динамический, иерархический протокол состояния связи, используемый для маршрутизации внутри автономных систем. Он базируется на открытых стандартах и был спроектирован как замена протоколу RIP. Он является развитием ранних версий протокола маршрутизации IS-IS. OSPF – устойчивый протокол, поддерживающий маршрутизацию с наименьшим весом и балансировку загрузки. Кратчайший путь в сети вычисляется по алгоритму Дейкстры. Cisco поддерживает свою версию стандарта OSPF.

Как только маршрутизатор настроен на работу с OSPF, он начинает процесс изучения окружения, проходя несколько фаз инициализации. В начале маршрутизатор использует *Hello* для определения своих соседей и создания отношений для обмена обновлением маршрутной информацией с ними. Затем маршрутизатор начинает фазу *ExStart* начального обмена между базами маршрутов. Следующей является фаза обмена, в которой назначенный маршрутизатор отсылает маршрутную информацию и получает подтверждения от нашего нового маршрутизатора. В течение стадии загрузки, новый маршрутизатор компилирует таблицу маршрутов. По окончании вычислений маршрутизатор переходит в полное состояние, в котором он является активным членом сети.

Для запуска OSPF-маршрутизации служит команда

```
Router(config)# router ospf N
```

Практическая часть

1. Загрузить в симулятор топологию и конфигурацию, использованную в практической части лабораторной работы № 5.
2. Если все сделано правильно, можно пропинговать из любых маршрутизаторов адреса непосредственно соединенных интерфейсов других маршрутизаторов. На каждом устройстве, используя команды *CDP*, *show cdp*, *neighbors detail*, получить IP-адреса соседних устройств и выполнить команду *ping*.
3. Из Router2 была недоступна сеть 172.16.10.0/24, а из Router4 была недоступна сеть 10.1.1.0/24. В этой работе для решения проблемы использовать разные формы динамической маршрутизации.

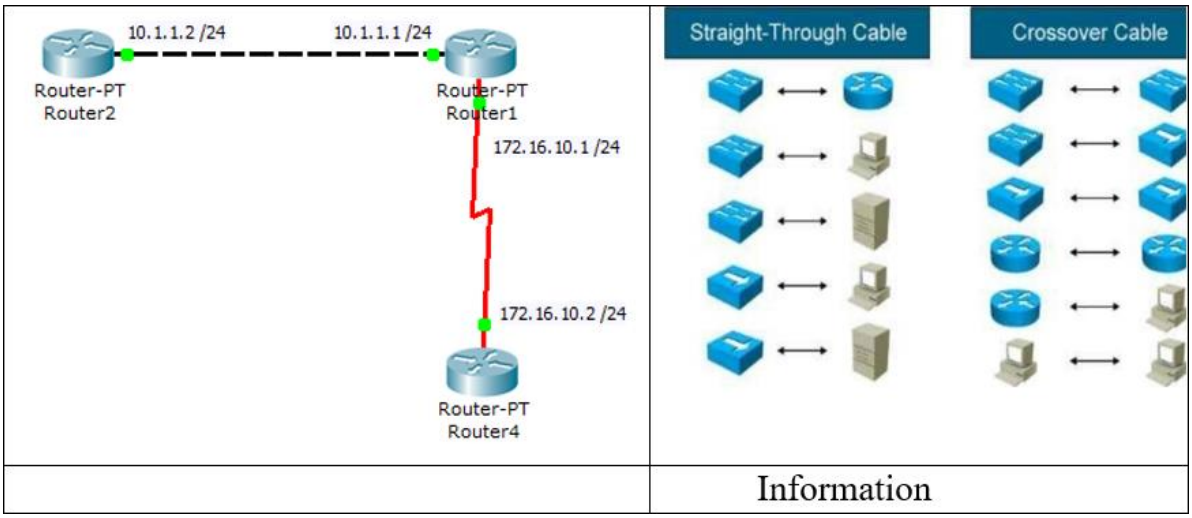

Рисунок 23. – Информация

4. Посмотреть таблицы маршрутов.

```
Router2# sh ip route
```

Нет маршрута на сеть 172.16.10.0/24.

Следовательно, в эту сеть из Router2 не проходит ping.

```
Router4# sh ip route
```

Нет маршрута на сеть 10.1.1.0/24.

Следовательно, в эту сеть из Router4 не проходит ping.

RIP

1. Включить RIP на всех маршрутизаторах:

```
Router1(config)# router rip
Router1(config-router)# network 172.16.10.0
Router1(config-router)# network 10.1.1.0

Router2(config)# router rip
Router2(config-router)# network 10.1.1.0

Router4(config)# router rip
Router4(config-router)# network 172.16.10.0
```

2. На каждом роутере командой *show running-config* посмотреть, как маршрутизаторы ответили на команды. Видим, что сеть 10.1.1.0/24 воспринята как сеть 10.0.0.0/8, а сеть 172.16.10.0/24 воспринята как сеть 172.16.0.0/16. Это связано с классами IP-адресов.

3. Командой *show ip protocols* посмотрим, с какими параметрами работает протокол RIP. Например, для Router1 имеем

```
Routing Protocol is "rip"
Sending updates every 30 seconds, next due in 6 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Redistributing: rip
Default version control: send version 1, receive any version
  Interface             Send  Recv  Triggered RIP  Key-chain
  Serial2/0             1     2 1
  FastEthernet0/0       1     2 1
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
        10.0.0.0
        172.16.0.0
Passive Interface(s):
Routing Information Sources:
        Gateway         Distance      Last Update
Distance: (default is 120)
```

Перевести сообщение.

4. Посмотреть таблицы маршрутов:

Router2# sh ip route

```
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
R    172.16.0.0/16 [120/1] via 10.1.1.1, 00:00:15, FastEthernet0/0
```

Есть маршрут на сеть 172.16.10.0/24 через интерфейс Ethernet на адрес 10.1.1.1.

Команда *ping* в эту сеть из Router2 сработает. Проверить:

Router2# ping 172.16.10.1

Router2# ping 172.16.10.2

5. Перейти на другой маршрутизатор:

Router4# sh ip route

```
R    10.0.0.0/8 [120/1] via 172.16.10.1, 00:00:22, Serial2/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

Есть маршрут на сеть 10.1.1.0/24 через интерфейс Serial на адрес 172.16.10.1.

Команда *ping* в эту сеть из Router4 сработает. Проверить:

Router4# ping 10.1.1.1

Router4# ping 10.1.1.2

6. Командой *debug ip rip* посмотреть, как маршрутизаторы обмениваются маршрутной информацией. Например, для Router1 есть повторяющиеся каждые 30 секунд сообщения:

```
Router1# debug ip rip
RIP: sending update to 255.255.255.255 via Serial0 (172.16.10.1)
 subnet 10.1.1.0, metric 1

RIP: sending update to 255.255.255.255 via Ethernet0 (10.1.1.1)
 subnet 172.16.10.0, metric 1

RIP: received update from 172.16.10.2 on Serial0

RIP: received update from 10.1.1.2 on Ethernet0
```

7. Выключить трассировку:

```
Router1# no debug ip rip
```

8. Сохранить конфигурацию.

EIGRP

Остановить на всех маршрутизаторах RIP командой

```
Router(config)# no router rip
```

1. Включить EIGRP на всех маршрутизаторах, образуя автономную систему с номером 100

```
Router1(config)# router eigrp 100
Router1(config-router)# network 172.16.10.0
Router1(config-router)# network 10.1.1.0

Router2(config)# router eigrp 100
Router2 (config-router)# network 10.1.1.0

Router4(config)# router eigrp 100
Router4 (config-router)# network 172.16.10.0
```

2. На каждом маршрутизаторе командой *show running-config* посмотреть, как маршрутизаторы выполнили команды. Видим, что сеть 10.1.1.0/24 воспринята как сеть 10.0.0.0/8, а сеть 172.16.10.0/24 воспринята как сеть 172.16.0.0/16. Это связано с классами IP-адресов.

3. Командой *show ip protocols* посмотреть, с какими параметрами работает протокол EIGRP. Например, для Router1:

```
Routing Protocol is "eigrp  100 "
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
```

```
    EIGRP maximum hopcount 100
    EIGRP maximum metric variance 1
  Redistributing: eigrp 100
    Automatic network summarization is in effect
    Automatic address summarization:
      172.16.0.0/16 for FastEthernet0/0
        Summarizing with metric 20512000
      10.0.0.0/8 for Serial2/0
        Summarizing with metric 28160
    Maximum path: 4
    Routing for Networks:
       172.16.0.0
       10.0.0.0
    Routing Information Sources:
      Gateway         Distance      Last Update
      172.16.10.2     90            8321604
      10.1.1.2        90            8350639
    Distance: internal 90 external 170
```

Перевести сообщение.

4. Посмотреть таблицы маршрутов:

Router2# sh ip route

```
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
D    172.16.0.0/16 [90/20514560] via 10.1.1.1, 00:08:37, FastEthernet0/0
```

Есть маршрут на сеть 172.16.10.0/24 через интерфейс Ethernet на адрес 10.1.1.1.

Команда *ping* в эту сеть из Router2 не сработает. Проверить:

Router2# ping 172.16.10.1

Router2# ping 172.16.10.2

5. Перейти на другой маршрутизатор

Router4# sh ip route

```
D    10.0.0.0/8 [90/20514560] via 172.16.10.1, 00:12:30, Serial2/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

Есть маршрут на сеть 10.1.1.0/24 через интерфейс Serial на адрес 172.16.10.1.

Команда *ping* в эту сеть из Router4 сработает. Проверить:

Router4# ping 10.1.1.1

Router4# ping 10.1.1.2

6. Командами *debug ip eigrp transactions* и *debug ip eigrp events* посмотреть, как маршрутизаторы обмениваются маршрутной информацией.

Сохранить конфигурацию.

OSPF

Остановить на всех маршрутизаторах EIGRP командой

```
Router(config)# no router eigrp 100
```

1. Включить OSPF на всех маршрутизаторах. Дать процессу OSPF номер 100. Образуем OSPF область 0:

```
Router1(config)# router ospf 100
Router1(config-router)# network 172.16.10.0 0.0.0.255 area 0
Router1(config-router)# network 10.1.1.0 0.0.0.255 area 0

Router2(config)# router ospf 100
Router2(config-router)# network 10.1.1.0 0.0.0.255 area 0

Router4(config)# router ospf 100
Router4(config-router)# network 172.16.10.0 0.0.0.255 area 0
```

2. Команда *show running-config* показывает, что маршрутизаторы поняли команды в том же виде, как их и задавали.

3. Командой *show ip protocols* посмотреть, с какими параметрами работает протокол OSPF. Например, для Router1:

```
Routing Protocol is "ospf 100"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.16.10.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.16.10.0 0.0.0.255 area 0
    10.1.1.0 0.0.0.255 area 0
  Routing Information Sources:
    Gateway         Distance      Last Update
    10.1.1.2             110      00:01:40
    172.16.10.2          110      00:01:45
  Distance: (default is 110)
```

Перевести сообщение.

4. Посмотреть таблицы маршрутов

```
Router2# sh ip route
     10.0.0.0/24 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, FastEthernet0/0
     172.16.0.0/24 is subnetted, 1 subnets
O       172.16.10.0 [110/782] via 10.1.1.1, 00:08:26, FastEthernet0/0
```

Есть маршрут на сеть 172.16.10.0/24 через интерфейс Ethernet на адрес 10.1.1.1.

Команда *ping* в эту сеть из Router2 сработает. Проверить:

```
Router2# ping 172.16.10.1
Router2# ping 172.16.10.2
```

5. Перейти на другой маршрутизатор:

```
Router4# sh ip route
     10.0.0.0/24 is subnetted, 1 subnets
O       10.1.1.0 [110/782] via 172.16.10.1, 00:03:26, Serial2/0
     172.16.0.0/24 is subnetted, 1 subnets
C       172.16.10.0 is directly connected, Serial2/0
```

Есть маршрут на сеть 10.1.1.0/24 через интерфейс Serial на адрес 172.16.10.1.

Команда *ping* в эту сеть из Router4 сработает. Проверить:

```
Router4# ping 10.1.1.1
Router4# ping 10.1.1.2
```

6. Команды *show ip ospf interface*, *show ip ospf database* и *debug ip igrp neighbor* покажут всю информацию о параметрах протокола OSPF.

Сохранить конфигурацию.

Контрольные вопросы

1. Что такое автономная система?
2. Что такое метрика?
3. Какие существуют два класса протоколов динамической маршрутизации?
4. Объясните работу дистанционно-векторных протоколов.
5. Объясните работу протоколов состояния связи.
6. Как узнать, какие протоколы маршрутизации запущены на маршрутизаторе?
7. В чем преимущества и недостатки дистанционно-векторных протоколов и протоколов состояния связи?
8. Что такое сходимость протоколов маршрутизации?
9. Какие параметры влияют на скорость сходимости?
10. Как на маршрутизаторе запустить и настроить протокол маршрутизации RIP?
11. Как на маршрутизаторе запустить и настроить протокол маршрутизации EIGRP?
12. Как на маршрутизаторе запустить и настроить протокол маршрутизации OSPF?
13. Как посмотреть содержание пакетов актуализации маршрутной информации протокола RIP?

14. Как посмотреть содержание пакетов актуализации маршрутной информации протокола EIGRP?

15. Уточните, что в EIGRP понимается под автономной системой?

16. В чем различие в формате команды *router* для EIGRP и OSPF?

17. В чем различие в формате команд *network* для RIP, EIGRP и OSPF?

Задание для самостоятельной работы

1. Использовать топологию своего варианта, представленную на рисунке 24.

В сети шесть подсетей. Видно, что каждый маршрутизатор подключен к трем подсетям.

2. Отредактировать вручную сохраненную конфигурацию предыдущей успешно выполненной лабораторной работы: уберите в rtr-файлах маршрутизаторов команды статической маршрутизации.

3. Загрузить отредактированную конфигурацию в симулятор.

4. На каждом маршрутизаторе проверить правильность загрузки конфигурации командами *show cdp neighbors* и *show ip interface brief*.

Если последовательный интерфейс не сработал, проверить командой *show running-config*, что на интерфейсе DCE стороны последовательной связи задана команда *clock rate*. Если не задана, то задать ее.

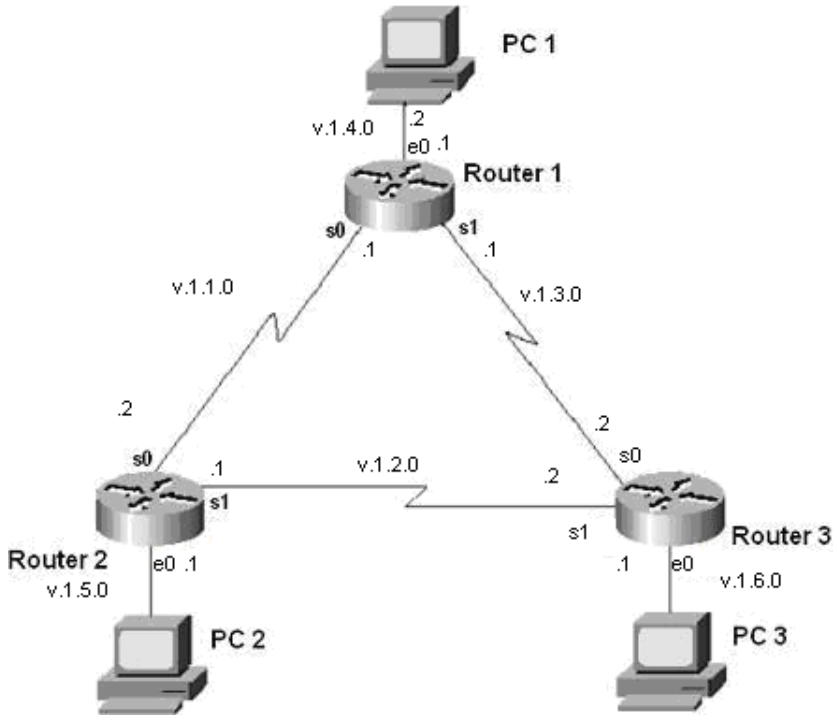

Рисунок 24. – Варианты топологии

5. Настроить на каждом маршрутизаторе динамическую маршрутизацию по протоколу RIP.

6. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора Router1 для варианта 1 (v = 1):

```
C       1.1.4.0/24 is directly connected, Ethernet0
C       1.1.1.0/24 is directly connected, Serial0
C       1.1.3.0/24 is directly connected, Serial1
R       1.1.6.0/24 [120/1] via 1.1.3.2, 00:01:43, Serial1
R       1.1.5.0/24 [120/1] via 1.1.1.2, 00:06:30, Serial0
R       1.1.2.0/24 [120/1] via 1.1.1.2, 00:03:35, Serial0
```

7. На каждом компьютере выполнить команды трассировки *tracert* других компьютеров. Сделать скриншоты. Всего шесть скриншотов. Например, трассировка из PC1 на PC2 для варианта 1 (v = 1) имеет вид:

```
PC1:#tracert 1.1.5.2


"Type escape sequence to abort."
Tracing the route to 1.1.5.2

 1 1.1.4.1 0 msec 16 msec 0 msec
 2 1.1.1.2 20 msec 16 msec 16 msec
 3 1.1.5.2 20 msec 16 msec *
```

Посмотреть, что путь для пакетов из PC1 на PC2 (1.1.5.2) лежит последовательно через Router1 (Ethernet 1.1.4.1) и затем через Router2 (serial0 1.1.1.2).

1. Отключить на маршрутизаторе Router1 последовательный интерфейс serial 0:

```
Router1(config)# interface serial 0
Router1(config-if)# shutdown
```

2. Через пару минут, когда в сети пройдут обновления маршрутной информации, на каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора Router1 для варианта 1 (v = 1):

```
C       1.1.4.0/24 is directly connected, Ethernet0
C       1.1.3.0/24 is directly connected, Serial1
R       1.1.2.0/24 [120/1] via 1.1.3.2, 00:07:26, Serial1
R       1.1.6.0/24 [120/1] via 1.1.3.2, 00:08:41, Serial1
R       1.1.5.0/24 [120/2] via 1.1.3.2, 00:07:44, Serial1
```

Видно, что таблица изменилась: пропала сеть 1.1.1.0/24 и все пакеты теперь маршрутизируются на адрес 1.1.3.2 через интерфейс Serial1.

3. На каждом компьютере выполнить команды трассировки *tracert* других компьютеров. Сделать скриншоты. Всего шесть скриншотов. Например, трассировка из PC1 на PC2 для варианта 1 (v = 1) имеет вид:

```
PC1:#tracert 1.1.5.2


"Type escape sequence to abort."
Tracing the route to 1.1.5.2

 1 1.1.4.1 0 msec 16 msec 0 msec
 2 1.1.3.2 20 msec 16 msec 16 msec
 3 1.1.2.1 20 msec 16 msec 16 msec
 4 1.1.5.2 20 msec 16 msec *
```

Видно, что теперь путь для пакетов из PC1 на PC2 (1.1.5.2) лежит последовательно через Router1 (Ethernet 1.1.4.1), затем через Router3 (serial0 1.1.3.2) и затем через Router2 (serial1 1.1.2.1).

4. Сохранить конфигурацию.

5. Отключить RIP и настроить на каждом маршрутизаторе динамическую маршрутизацию по протоколу IGRP.

6. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора Router1 для варианта 1 (v = 1):

```
C       1.1.4.0/24 is directly connected, Ethernet0
C       1.1.1.0/24 is directly connected, Serial0
C       1.1.3.0/24 is directly connected, Serial1
I       1.1.6.0/24 [100/651] via 1.1.3.2, 00:02:38, Serial1
I       1.1.5.0/24 [100/651] via 1.1.1.2, 00:04:26, Serial0
I       1.1.2.0/24 [100/651] via 1.1.1.2, 00:08:28, Serial0
```

7. Проверить, что вы все сделали правильно. На каждом компьютере выполните команды трассировки *tracert* других компьютеров.

8. Сохранить конфигурацию.

9. Отключить IGRP и настроить на каждом маршрутизаторе динамическую маршрутизацию по протоколу OSPF.

10. На каждом маршрутизаторе посмотреть таблицу маршрутизации командой *show ip route*. Сделать скриншоты. Например, для маршрутизатора Router1 для варианта 1 (v = 1):

```
C       1.1.4.0/24 is directly connected, Ethernet0
C       1.1.1.0/24 is directly connected, Serial0
C       1.1.3.0/24 is directly connected, Serial1
O       1.1.5.0/24 [110/65] via 1.1.1.2, 00:00:39, Serial0
O       1.1.2.0/24 [110/65] via 1.1.3.2, 00:00:22, Serial1
O       1.1.6.0/24 [110/65] via 1.1.3.2, 00:00:26, Serial1
```

11. Проверить, что все сделано правильно. На каждом компьютере выполнить команды трассировки *tracert* других компьютеров.

12. Сохранить конфигурацию.

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения
4. Ход выполнения работы со скриншотами.
5. Настройки роутеров в текстовом файле в приложении.
6. Выводы.

ЛАБОРАТОРНАЯ РАБОТА № 7
Исследование работы протокола ARP

Цель работы: построить сеть на маршрутизаторах с применением протоколов статической и динамической маршрутизации.

Задачи:

– научиться формировать и отправлять пользовательские пакеты;
– ознакомиться с журналом работы сетевого устройства в эмуляторе;
– научиться проводить сетевую атаку вида ARP-спуфинг.

Краткие теоретические сведения

ARP (Address Resolution Protocol – протокол определения адреса) – протокол в компьютерных сетях, предназначенный для определения MAC-адреса сетевого устройства по известному IP-адресу.

Наибольшее распространение ARP получил благодаря повсеместности сетей IP, построенных поверх Ethernet, поскольку в подавляющем большинстве случаев при таком сочетании используется ARP. В семействе протоколов Ipv6 протокола ARP не существует, его функции возложены на ICMPv6. Описание протокола было опубликовано в ноябре 1982 г. в RFC 826.

ARP был спроектирован для случая передачи IP-пакетов через сегмент Ethernet. При этом общий принцип, предложенный для ARP, был использован и для сетей других типов.

Существуют следующие типы сообщений ARP: запрос ARP (ARP-request) и ответ ARP (ARP-reply).

Система-отправитель при помощи запроса ARP запрашивает физический адрес системы-получателя. Ответ (физический адрес узла-получателя) приходит в виде ответа ARP. Принцип работы протокола: узел (хост A), которому нужно выполнить отображение IP-адреса на MAC-адрес, формирует ARP-запрос, вкладывает его в кадр протокола канального уровня, указывая в нем известный IP-адрес (хост B), и рассылает запрос широковещательно (в поле «MAC-адрес назначения заголовка Ethernet» указывается широковещательный MAC-адрес FF:FF:FF:FF:FF:FF).

Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел (хост B) формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес, и отправляет его уже направленно, так как в ARP-запросе отправитель (хост A) указывает свой локальный адрес. Схема работы показана на рисунке 25.

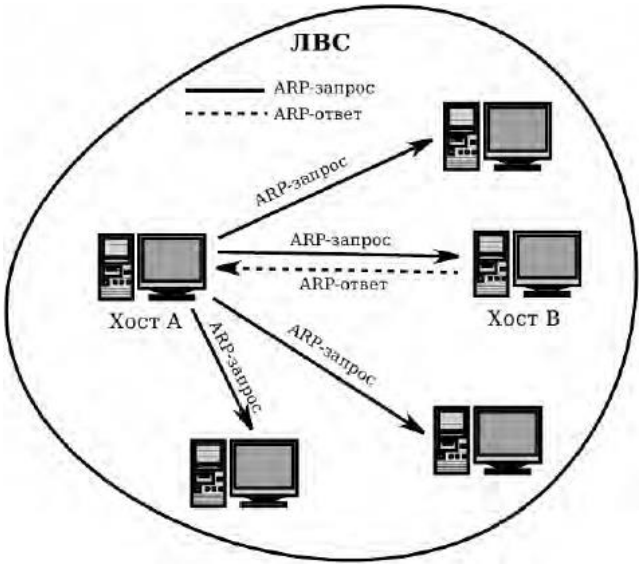

Рисунок 25. – Схема работы протокола ARP

При получении ARP-ответа хост А записывает в кэш ARP запись с соответствием IP-адреса хоста В и MAC-адреса хоста В, полученного из ARP-ответа. Время хранения такой записи ограничено. По истечении времени хранения хост А посылает повторный запрос, теперь уже адресно, на известный MAC-адрес хоста В. В случае, если ответ не получен, снова посылается широковещательный запрос.

Структура кадра ARP с учетом заголовка Ethernet показана на рисунке 26.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Destination MAC						Source MAC						ETH TYPE		HTYPE	
PTYPE		HLEN	PLEN	OP CODE		Sender MAC						Sender IP			
Target MAC						Target IP									

Рисунок 26. – Структура кадра ARP

Самопроизвольный ARP (gratuitous ARP) – такая разновидность ARP, когда ARP-ответ присылается в случае, если с точки зрения получателя в нем нет особой необходимости. Самопроизвольный ARP-ответ – это пакет-ответ ARP, присланный без запроса. Он применяется для определения конфликтов IP-адресов в сети: как только станция получает адрес по DHCP или адрес присваивается вручную, рассылается ARP-ответ gratuitous ARP.

Самопроизвольный ARP может быть полезен в следующих случаях:

– обновление ARP-таблиц, в частности, в кластерных системах;
– информирование коммутаторов;
– извещение о включении сетевого интерфейса.

Несмотря на эффективность самопроизвольного ARP, он является особенно небезопасным, поскольку с его помощью можно уверить удаленный узел в том, что MAC-адрес какой-либо системы, находящейся с ней в одной сети, изменился, и указать, какой адрес используется теперь.

Сетевая атака ARP-спуфинг (ARP-spoofing) основана на использовании самопроизвольного ARP. Чтобы перехватить сетевые пакеты, которые атакуемый хост (A) оправляет на хост B, атакующий хост (C) формирует ARP-ответ, в котором ставит в соответствие IP-адресу хоста B свой MAC-адрес. Далее этот пакет отправляется на хост A. В том случае, если хост A поддерживает самопроизвольный ARP, он модифицирует собственную ARP-таблицу и помещает туда запись, где вместо настоящего MAC-адреса хоста B стоит MAC-адрес атакующего хоста C.

Теперь пакеты, отправляемые хостом A на хост B, будут передаваться хосту C.

Ход работы

Построение сети

1. Построить сеть в соответствии с рисунком 27, используя соответствующие инструменты на панели эмулятора. В свойствах маршрутизатора необходимо указать количество интерфейсов, равное 2.

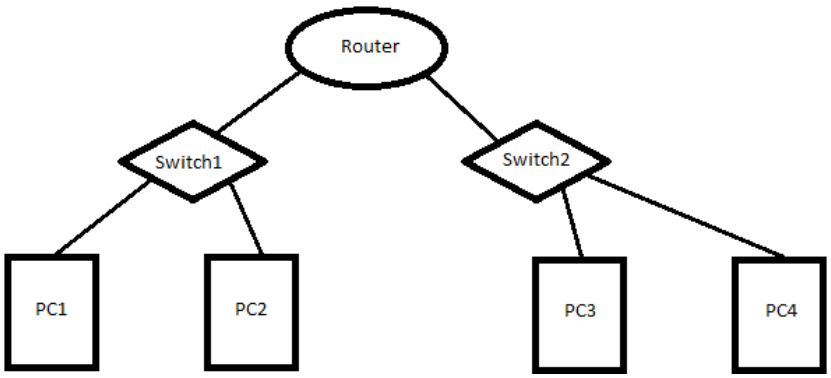

Рисунок 27. – Модель сети

2. Настроить интерфейсы компьютеров и маршрутизаторов, задав каждому IP-адрес и маску подсети (слева – первая подсеть в заданной сети, справа – вторая подсеть). Добавить возле каждого компьютера и интерфейса роутера надписи с их IP-адресом и маской подсети.

3. Настроить на компьютерах маршруты «по умолчанию» (IP-сети – 0.0.0.0, маска подсети – 0.0.0.0). Можно воспользоваться «Таблицей маршрутизации» либо вызвать свойства компьютера двойным щелчком, указать шлюз по умолчанию и включить маршрутизацию.

4. Включить маршрутизацию на маршрутизаторе.

5. Проверить работоспособность построенной модели ЛВС, передав пакеты (TCP, 5 KB) от компьютера в левой подсети до компьютера в правой подсети. Если пакеты не проходят, в меню «Интерфейсы» маршрутизатора нажать кнопку «сбросить статистику» для автоматического формирования ARP-запроса.

6. Задать каждому компьютеру имя-описание, воспользовавшись пунктом контекстного меню «Задать описание».

Определение MAC-адреса с помощью ARP-запроса

1. Запустить для компьютеров 1 и 2 журналы пакетов (пункт меню «Показать журнал»).

2. Очистить ARP-таблицу компьютера 1.

3. Выделить компьютер 1 и с помощью инструмента «Конструктор пакетов» сформировать пакет ARP-запроса для определения MAC-адреса компьютера 2. Помните, что ARP-запрос рассылается широковещательно (MAC-адрес получателя в заголовке Ethernet – FF:FF:FF:FF:FF:FF), а MAC-адрес искомого узла в заголовке ARP приравнивается к нулевому 00:00:00:00:00:00. MAC-адрес компьютера 1 указан в окне «Интерфейсы» для компьютера 1.

4. Запустить ARP-запрос, проследить за ним и за сгенерированным для него ARP-ответом по схеме сети и журналам компьютеров 1 и 2.

5. Открыть ARP-таблицу компьютера 1 и убедиться, что запись добавилась в таблицу.

6. Сохранить скриншот экрана (с открытыми журналами) для отчета.

Реализация атаки ARP-спуфинг

1. Запустить для компьютеров 1 и 2 журналы пакетов (пункт меню «Показать журнал»). При необходимости очистить их.

2. Очистить ARP-таблицу компьютера 1.

3. Выделить компьютер 2 и с помощью инструмента «Конструктор пакетов» сформировать пакет ARP-ответа, в котором будут указаны:

   – MAC отправителя – MAC компьютера 2;
   – IP отправителя – IP интерфейса роутера в левой подсети;
   – MAC получателя – MAC компьютера 1;
   – IP получателя – IP компьютера 1.

4. Запустить ARP-ответ, проследить за ним. Может возникнуть окно о дублировании IP-адресов в сети – это происходит в том случае, если из-за действий коммутатора пакет-атаку получает и роутер. Окно быстро закрыть.

5. Сразу же запустить передачу пакетов (UDP, 5 KB) с компьютера 1 на компьютер 3. Убедиться, что пакеты вначале приходят на компьютер 2 и лишь потом (если на компьютере 2 включена маршрутизация) отправляются на компьютер 3 (через маршрутизатор).

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Разбиение заданной сети /27 на две подсети /28.
4. Схема модели с указанием IP-адресов устройств и номеров интерфейсов.
5. Скриншоты с результатами разрешения адреса и сетевой атаки.
6. По каждому пункту лабораторной должны быть приведены выводы по работе.

Контрольные вопросы

1. Что представляет собой протокол ARP?
2. Каков формат пакета ARP?
3. Что такое самопроизвольный ARP?
4. Как участвует IP-адрес в протоколе ARP?
5. Что такое MAC-адрес?
6. Как работает ARP-спуфинг?

ЛАБОРАТОРНАЯ РАБОТА № 8
Анализ сетевого трафика при помощи Wireshark

Цель работы: научиться анализировать сетевой трафик на примере протоколов ARP, IP и ICMP.

Задачи:

– знать принципы анализа сетевого трафика;

– научиться использовать сетевой анализатор (снифер Wireshark).

Краткие теоретические сведения

Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор трафика, программа или программно-аппаратное устройство, предназначенное для перехвата и последующего анализа либо только анализа сетевого трафика, предназначенного для других узлов.

Перехват трафика может осуществляться:

– обычным «прослушиванием» сетевого интерфейса (метод эффективен при использовании в сегменте концентраторов (хабов) вместо коммутаторов (свичей), в противном случае метод малоэффективен, поскольку на снифер попадают лишь отдельные фреймы);

– подключением снифера в разрыв канала;

– ответвлением (программным или аппаратным) трафика и направлением его копии на снифер;

– через анализ побочных электромагнитных излучений и восстановление таким образом прослушиваемого трафика;

– через атаку на канальном (втором) или сетевом (третьем) уровне, приводящую к перенаправлению трафика жертвы или всего трафика сегмента на снифер с последующим возвращением трафика в надлежащий адрес.

В начале 1990-х гг. широко применялся хакерами для захвата пользовательских логинов и паролей. Широкое распространение хабов позволяло захватывать трафик без больших усилий в больших сегментах сети.

Снифферы применяются как в благих, так и в деструктивных целях. Анализ прошедшего через снифер трафика, позволяет:

– отслеживать сетевую активность приложений;

– отлаживать протоколы сетевых приложений;

– локализовать неисправность или ошибку конфигурации;

– обнаружить паразитный, вирусный и закольцованный трафик, наличие которого увеличивает нагрузку сетевого оборудования и каналов связи;

– выявить в сети вредоносное и несанкционированное ПО, например, сетевые сканеры, троянские программы, клиенты пиринговых сетей и другие;

– перехватить любой незашифрованный (а порой и зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

Постепенно из инструментов, предназначенных только для диагностики, снифферы превратились в средства для исследований и обучения. Например, они постоянно используются для изучения динамики и взаимодействий в сетях. В частности, они позволяют легко и наглядно изучать тонкости сетевых протоколов. Наблюдая за данными, которые посылает протокол, можно понять его функционирование на практике, а заодно зафиксировать, когда некоторая конкретная реализация работает не в соответствии со спецификацией.

На сегодняшний момент существует достаточно большое количество хороших реализаций снифферов. Например:

– Tcpdump (http://www.tcpdump.org/) – консольный вариант сниффера. Портирован почти под все наиболее распространенные ОС;

– Wireshark (http://www.wireshark.org/) – до недавнего момента был известен под названием Ethreal;

– WinDump http://www.winpcap.org/windump.

Сниффер Wireshark

Программа Wireshark является одной из самых удобных реализаций снифферов. Рассчитана на большое количество платформ. Распространяется абсолютно бесплатно.

Базовый принцип работы снифферов. На рисунке 28 схематично изображена структура сетевой подсистемы ОС. Вся базовая инфраструктура реализована в виде драйверов и работает в режиме ядра. Пользовательские процессы и реализации прикладных протоколов, в частности интерфейс сниффера, работают в пользовательском режиме.

На рисунке отображены 2 пользовательских процесса («сетевой процесс 1» и «сетевой процесс 2»).

Основными компонентами сниффера являются: драйвер для захвата пакетов (libpcap-драйвер), интерфейсная библиотека (libpcap) и интерфейс пользователя (Wireshark). Библиотека libpcap (реализация под ОС Windows носит название WinPcap, http://www.winpcap.org) – универсальная сетевая библиотека, самостоятельно реализующая большое количество сетевых протоколов

и работающая непосредственно с NDIS (Network Driver Interface Specification) – драйверами сетевых устройств. На базе данной библиотеки реализовано большое количество сетевых программ, в частности сниффер Wireshark.

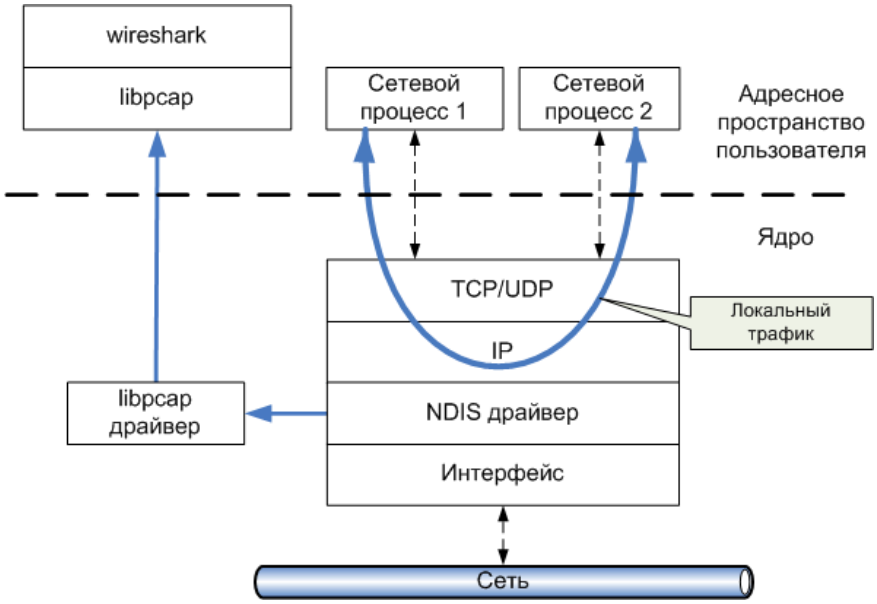

Рисунок 28. – Принцип «захвата» сниффером сетевого трафика

Сниффер использует библиотеку в режиме «захвата» пакетов, т.е. может получать копию ВСЕХ данных проходящих через драйвер сетевого интерфейса. Изменения в сами данные не вносятся.

Основной нюанс использования сниффера заключается в том, что он не позволяет производить анализ локального трафика, т.к. он не проходит через драйвер сетевого устройства (см. рисунок 28). Т.е., если необходимо проанализировать сниффером трафик между двумя сетевыми процессами на локальной машине (например, ftp-сервер и ftp-клиент), то этого не получится. Однако, например, при использовании виртуальных машин, сниффер будет работать без проблем, т.к. виртуальные машины эмулируют реальную среду и сетевые адаптеры, поэтому трафик идет через драйвер, как и в нормальной ситуации при взаимодействии с другими физическими сетевыми машинами.

Также к недостаткам большинства снифферов стоит отнести и тот факт, что, позволяя анализировать трафик, проходящий через сетевой интерфейс, они не могут указать, какое именно приложение генерирует или получает его. Это объясняется тем, что информация об этом хранится на сетевом (например, IP) уровне сетевого стека, а большинство снифферов использует собственную реализацию стека протоколов (например, библиотеку WinPcap), которая (как уже было показано) работает непосредственно с драйверами устройств.

Также снифферы вносят дополнительную нагрузку на процессор, т.к. могут обрабатывать достаточно объемный сетевой трафик, в особенности для высокоскоростных соединений (Fast Ethernet, Gigabit Ethernet и др.).

Использование программы Wireshark. Данный сниффер позволяет в режиме реального времени захватывать пакеты из сети, и анализировать их структуру. Также можно анализировать структуру пакетов из файла, содержащего трафик, полученный, например, программой «tcpdump» (unix/linux).

На рисунке 29 изображено основное окно программы Wireshark. В стандартном режиме окно сниффера делится на 3 панели: список захваченных пакетов, «анализатор» протоколов и исходные данные пакетов. Размер каждой панели можно менять по своему усмотрению.

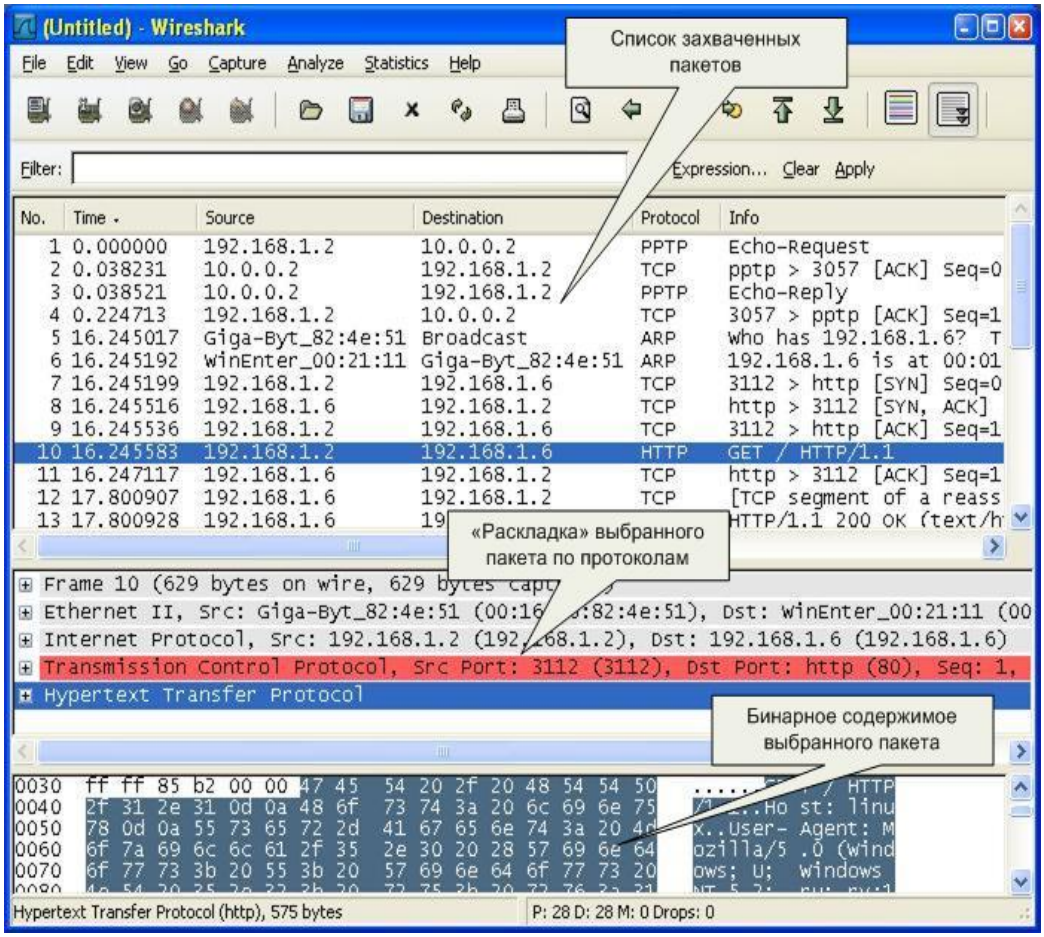

Рисунок 29. – Основное окно сниффера Wireshark

Рассмотрим эти панели подробнее.

Верхняя панель содержит список пакетов, захваченных из сети. Список можно отсортировать по любому полю (в прямом или обратном порядке) – для этого надо нажать на заголовок соответствующего поля.

Каждая строка содержит следующие поля (по умолчанию):

– порядковый номер пакета (No.);
– время поступления пакета (Time);
– источник пакета (Source);
– пункт назначения (Destination);
– протокол (Protocol);
– информационное поле (Info).

Список отображаемых полей настраивается в *Edit → Preferencis → Columns*. Для того, чтобы изменения возымели эффект необходимо перезапустить программу, предварительно нажав кнопку «Save».

При нажатии правой кнопки мыши на том или ином пакете появится контекстное меню. Нажатием на среднюю кнопку мыши можно помечать группу интересующих нас пакетов.

Средняя панель содержит «дерево протоколов» для выбранного в верхнем окне пакета. В этой панели в иерархическом виде для выбранного в верхнем окне захваченного пакета отображается вложенность протоколов в соответствии с моделью взаимодействия открытых систем OSI. По нажатию на правую кнопку мыши вызывается контекстное меню. При «раскрытии» каждого из протоколов нажатием на значок «+» слева – выводятся поля данных соответствующих протоколов.

Нижняя панель содержит шестнадцатеричное представление выбранного пакета. При выборе того или иного поля в средней панели автоматически будет подсвечиваться соответствующий участок шестнадцатеричного представления.

Захват пакетов. Для начала захвата пакетов необходимо задать параметры захвата, в частности, указать сетевой интерфейс, с которого и будет осуществляться захват пакетов. Это действие доступно через меню *Capture → Options* или комбинацию клавиш CTRL + K (см. рисунок 29). Интерфейс, задаваемый в поле «Interface:», можно выбрать из соответствующего поля. В примере на рисунке 29 показано, что доступны три интерфейса: физический сетевой адаптер («Marvel…») и интерфейсы для виртуальных каналов, в частности, установленного VPN-соединения («WAN (PPP/SLIP)…»). В большинстве случае подходит выбор интерфейса сетевого адаптера.

В качестве дополнительных параметров захвата можно указать следующие:

– *Capture Filter* – фильтр захвата (будем рассматривать далее). По нажатию на соответствующую кнопку можно применить тот или иной фильтр

отбора (из ранее сохраненных). Если таковых не имеется, его можно указать явно в строке редактирования;

– *Update list of packets in real time* – обновление списка захваченных пакетов в режиме реального времени;

– *Stop Capture* – набор параметров, позволяющих задать то или иное значение, при достижении которого процесс захвата пакетов прекратится;

– *Name Resolution* – набор параметров разрешения имен позволяет определить, какие из способов разрешения имен должны использоваться.

Для начала мониторинга сетевой активности нужно нажать «Start». После выбора интерфейса, который нас интересует, в дальнейшем можно начинать и останавливать захват пакетов через соответствующие команды в меню «Capture».

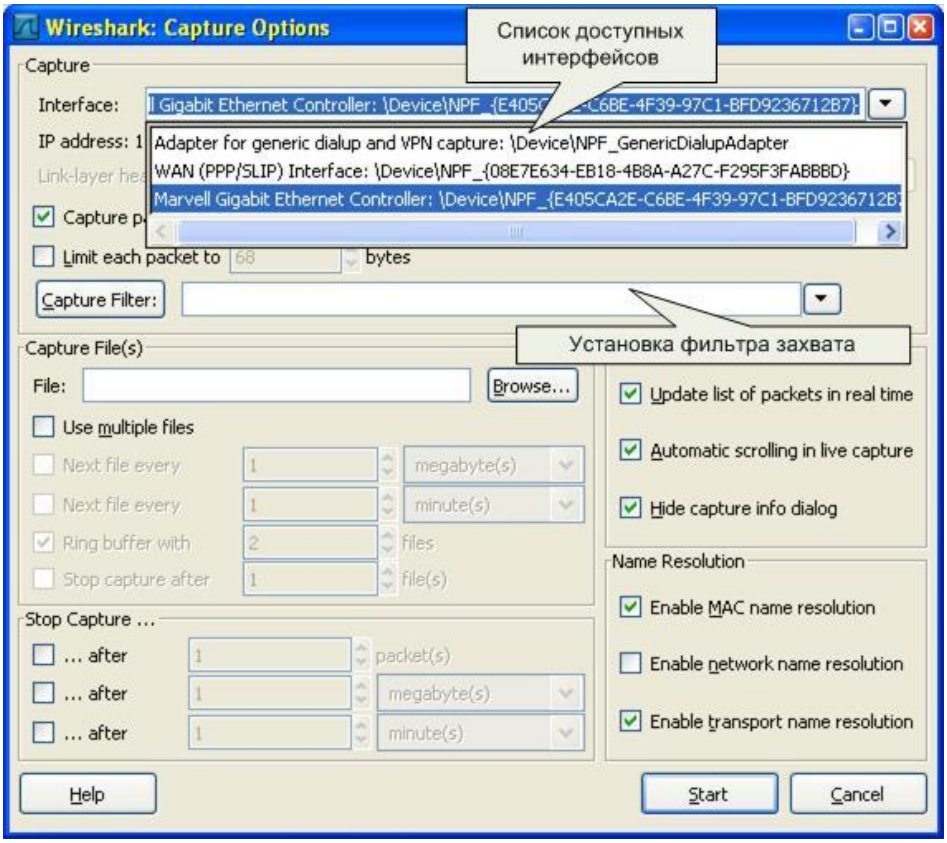

Рисунок 30. – Выбор интерфейса и параметров захвата пакетов

Фильтрация пакетов. Если запустить сниффер без дополнительных настроек, он будет «захватывать» все пакеты, проходящие через сетевые интерфейсы (см. рисунок 28). Для общего ознакомления с процессами, происходящими в сети, необходимо пронаблюдать активность сетевых протоколов в реальных условиях работы системы в сети.

При целенаправленном использовании сниффера очень часто необходимо выборочно отображать или захватывать пакеты по некоторым заданным критериям. Для этих целей служат фильтры отображения и захвата соответственно.

Типы фильтрации трафика. Существует два варианта фильтрации пакетов: на этапе захвата и на этапе отображения пользователю. В первом случае эффективность работы сниффера и потребляемые им системные ресурсы значительно ниже, нежели во втором случае. Это объясняется тем, что при достаточно интенсивном сетевом трафике и продолжительном времени захвата все пакеты должны быть захвачены и сохранены либо в память, либо на дисковое устройство. Самые простые подсчеты могут показать, что даже для 100-мегабитной сети системных ресурсов хватит на непродолжительное время. Фильтрация захвата уже на момент получения пакета гораздо эффективнее, однако в таком случае она должна быть реализована на уровне самих драйверов захвата. Данный факт, естественно, усложняет реализацию сниффера. Wireshark поддерживает оба варианта фильтрации.

Фильтры отображения. Фильтры отображения представляют собой достаточно мощное средство отображения трафика. Фильтры задаются в строке, располагающейся вверху основного экрана («Filter:»). Простейший фильтр отображения, позволяет отобрать пакеты по тому или иному протоколу. Для этого в строке требуется указать название протокола (например, HTTP) и нажать кнопку «Apply». После этого в верхнем окне останутся пакеты, принадлежащие этому протоколу. Кнопкой «Reset» действие фильтра отменяется.

Для работы с фильтрами можно вызвать окно «Analyze/Display Filters». Можно сохранять созданные выражения под определенными именами для последующего использования и т.д.

С помощью логических операций (синтаксис языка Си) можно составлять логические выражения. Логическая истина – 1, ложь – 0.

Список поддерживаемых логических операций:

eq	==	равенство
ne	!=	не равно
gt	>	больше чем
Lt	<	меньше чем
ge	>=	больше равно
Le	<=	меньше равно

Пример

tcp.port == 80 (рисунок 31).

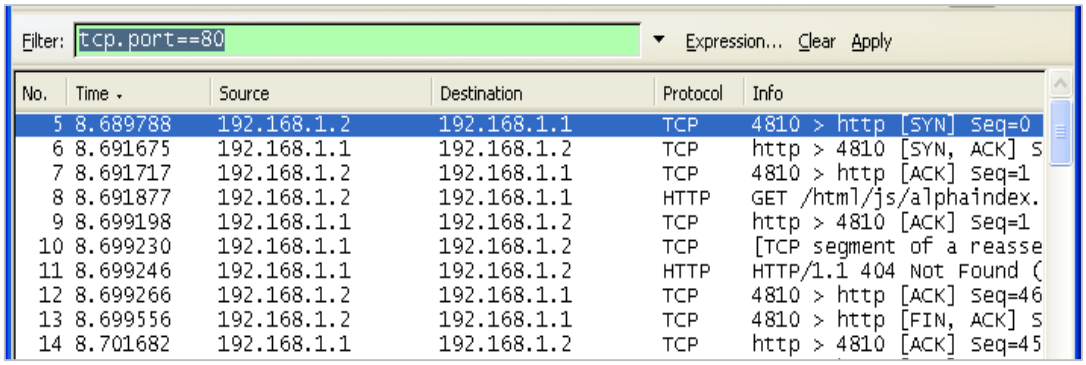

Рисунок 31. – Пример задания простого фильтра отображения

Мастер построения фильтров отображения доступен через кнопку «Expression…» (рисунок 32).

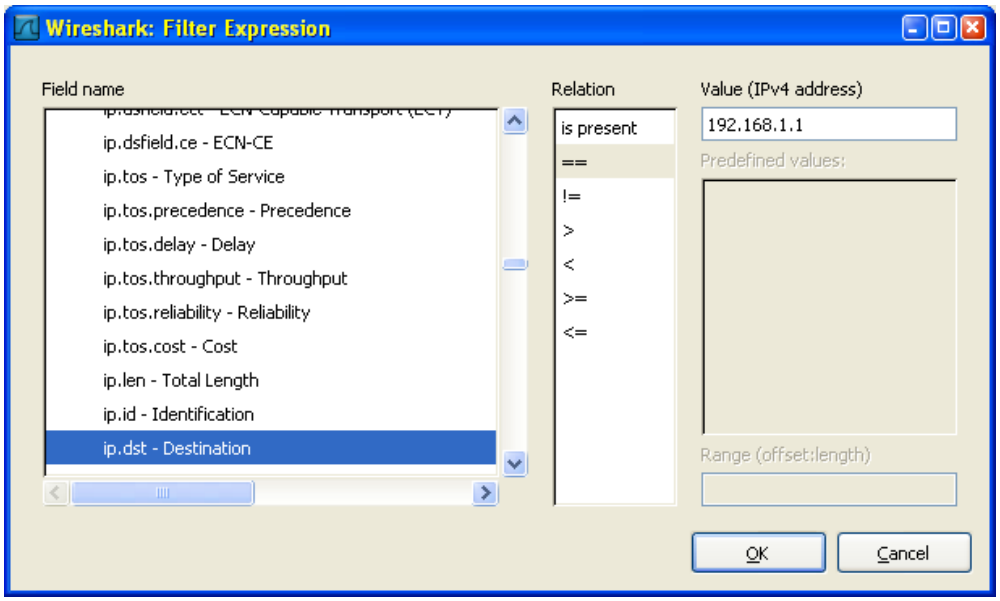

Рисунок 32. – Построение фильтров отображения

Фильтры захвата. С помощью данных фильтров можно захватывать из сети только те пакеты, которые подходят под критерий отбора. Если не задано никакого фильтра (по умолчанию), то будут захватываться все пакеты. В противном случае – только пакеты, для которых указанное выражение будет истинным. Синтаксис фильтров захвата несколько отличается от синтаксиса фильтров отображения. Выражение состоит из одного или более примитивов, разделенных пробельными символами. На рисунке 33 приведен пример фильтра для захвата пакетов, адресованных на 80-й порт (http) узла с IP-адресом 10.197.0.11.

Существует три различных типа примитивов: *type*, *dir*, *proto*.

Спецификатор *type* определяет тип параметра. Возможные параметры – *host*, *net*, *port*.

Пример
host linux
net 192.168.128
port 80

Если не указано никакого типа, предполагается, что это параметр *host*.

Спецификатор *dir* определят направление передачи. Возможные направления – *src*, *dst*, *src or dst*, *src and dst*.

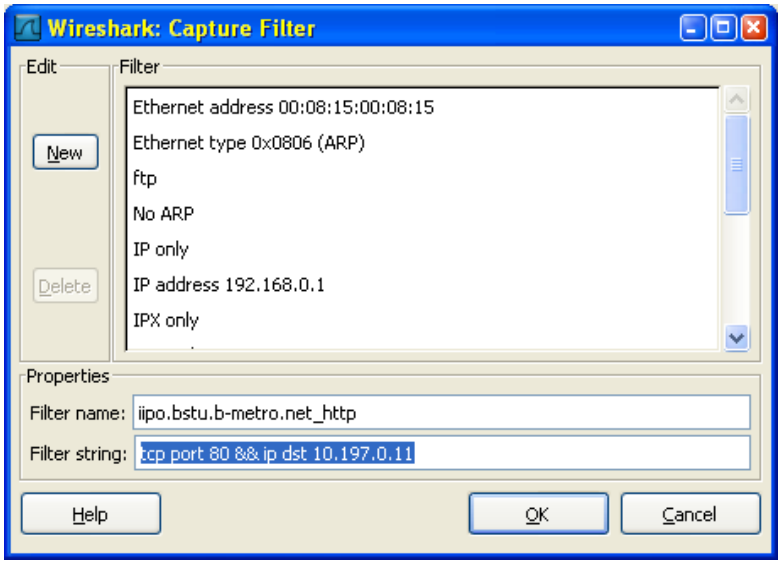

Рисунок 33. – Пример фильтра захвата

Пример
src linux
dst net 192.168.128
src or dst port http

Если не определено направление, то предполагается направление *src or dst*. Для протоколов типа *point-to-point* используются спецификаторы *inbound* и *outbound*.

Спецификатор *proto* определят тип протокола, которому принадлежит пакет.

Возможные протоколы – *ether*, *fddi*, *tr*, *ip/ipv6*, *arp/rarp*, *decent*, *tcp*, *udp*.

Пример
ether src linux
arp net 192.168.128
tcp port 80

Если протокол не определен, то будут захватываться пакеты всех протоколов, т.е. *src linux* означает *(ip or arp or rarp) src linux*; *net ctam* означает *(ip or arp or rarp) net ctam*; *port 53* означает *(tcp or udp) port 53*.

Также существует несколько специальных спецификаторов, которые не попадают в описанные выше случаи:

– *gateway;*
– *broadcast;*
– *less;*
– *greater;*
– *арифметические выражения.*

Сложные фильтры захвата строятся с использованием логических выражений.

Список операций:

not	!	отрицание
and	&&	конкатенация (логическое И)
or	||	альтернатива (логическое ИЛИ)

Примеры фильтров захвата. Ниже рассмотрены некоторые примеры построения фильтров захвата.

Захват всех пакетов на сетевом интерфейсе хоста 192.168.1.2:

```
host 192.168.1.2
```

Захват трафика между хостом host1 И хостами host2 ИЛИ host3:

```
host host1 and (host2 or host3)
```

Захват всех IP-пакетов между хостом host1 и каждым хостом за исключением hostX:

```
ip host host1 and not hostX
```

Захват пакетов ни сгенерированных, ни адресованных локальными хостами:

```
ip and not net localnet
```

Захват IP-пакетов размером больше чем 576 байт, проходящих через шлюз snup:

```
gateway snup and ip[2:2] > 576
```

Захват всех ICMP пакетов, за исключением пакетов ping:

```
icmp[0] != 8 and icmp[0] != 0
```

Статистическая обработка сетевого трафика. Сниффер Wireshark позволяет выполнять различную статистическую обработку полученных данных. Все доступные операции находятся в меню «Statistics».

Общая статистика (количество полученных/переданных пакетов, средняя скорость передачи и т.д.) доступна через пункт *Statistics → Summary*.

Получить информацию по статистике обработанных протоколов в полученных пакетах можно через пункт *Statistics → Protocol Hierarchy*.

Статистику по типу IP-пакетов, их размеру и порту назначения можно получить, выбрав подпункты меню «IP-address…», «Packet length» и «Port type» соответственно.

Одной из наиболее интересных возможностей является генерация диаграммы взаимодействия между узлами, которая доступна из пункта меню «Flow Graph…». В результате можно наблюдать в достаточно наглядной форме процесс взаимодействия на уровне протоколов. Например, на рисунке 34 приведена диаграмма взаимодействия при получении узлом win2013 статической web-странички с сервера http://linux.

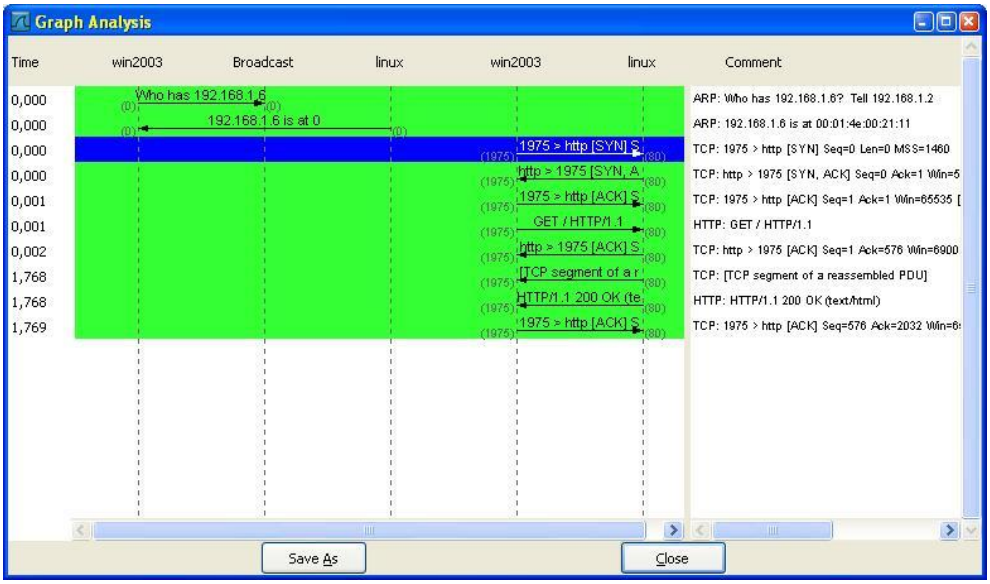

Рисунок 34. – Диаграмма взаимодействия

Протокол ARP

ARP (англ. Address Resolution Protocol – протокол разрешения адресов) – сетевой протокол, предназначенный для преобразования IP-адресов (адресов сетевого уровня) в MAC-адреса (адреса канального уровня) в сетях TCP/IP.

Данный протокол очень распространен и чрезвычайно важен. Каждый узел сети имеет как минимум два адреса – физический адрес и логический

адрес. В сети Ethernet для идентификации источника и получателя информации используются оба адреса. Информация, пересылаемая от одного компьютера другому по сети, содержит в себе физический адрес отправителя, IP-адрес отправителя, физический адрес получателя и IP-адрес получателя. ARP-протокол обеспечивает связь между этими двумя адресами. Существует четыре типа ARP-сообщений: ARP-запрос (ARP request), ARP-ответ (ARP reply), RARP-запрос (RARP-request) и RARP-ответ (RARP-reply). Локальный хост при помощи ARP-запроса запрашивает физический адрес хоста-получателя. Ответ (физический адрес хоста-получателя) приходит в виде ARP-ответа. Хост-получатель вместе с ответом шлет также RARP-запрос, адресованный отправителю, чтобы проверить его IP-адрес. После проверки IP-адреса отправителя начинается передача пакетов данных.

Перед тем, как создать подключение к какому-либо устройству в сети, IP-протокол проверяет свой ARP-кэш, чтобы выяснить, не зарегистрирована ли в нем уже нужная для подключения информация о хосте-получателе. Если такой записи в ARP-каше нет, то выполняется широковещательный ARP-запрос.

RARP (англ. Reverse Address Resolution Protocol – обратный протокол преобразования адресов) – выполняет обратное отображение адресов, то есть преобразует аппаратный адрес в IP-адрес.

Протокол применяется во время загрузки узла (например, компьютера), когда он посылает групповое сообщение-запрос со своим физическим адресом. Сервер принимает это сообщение и просматривает свои таблицы (либо перенаправляет запрос куда-либо еще) в поисках IP-адреса, соответствующего адресу физическому. После обнаружения найденный адрес отсылается обратно на запросивший его узел. Другие станции также могут «слышать» этот диалог и локально сохранить эту информацию в своих ARP-таблицах.

RARP позволяет разделять IP-адреса между нечасто используемыми хост-узлами. После использования каким-либо узлом IP-адреса он может быть освобожден и выдан другому узлу.

Для просмотра ARP-кэша можно использовать одноименную утилиту *arp* с параметром «-a».

Пример

```
D:\>arp –a
Interface: 192.168.1.2 --- 0x10003
Internet Address      Physical Address      Type
192.168.1.1           00-15-e9-b6-67-4f     dynamic
192.168.1.6           00-01-4e-00-21-11     dynamic
```

Из данного результата команды *arp* видно, что в кэше на данный момент находятся две записи и видны соответственно ip-адреса машин и MAC-адреса их сетевых адаптеров. Записи в ARP-кэше могут быть статическими и динамическими. Пример, приведенный выше, описывает динамическую запись кэша. Хост-отправитель автоматически послал запрос получателю, не уведомляя при этом пользователя. Записи в ARP-кэш можно добавлять вручную, создавая статические (static) записи кэша.

Так как на момент начала работы утилиты *ping* в arp-кэше не было информации о MAC-адресе соответствующего узла, то первоначально система должна выполнить определение это самого MAC-адреса, сгенерировав ARP-запрос и отослав его в сеть широковещательным пакетом, после чего она будет ожидать ответа от заданного узла.

После остановки сниффера должны увидеть результат, схожий с тем, что отображен на рисунке 35. В данном случае видны два захваченных пакета: ARP-запрос и ARP-ответ.

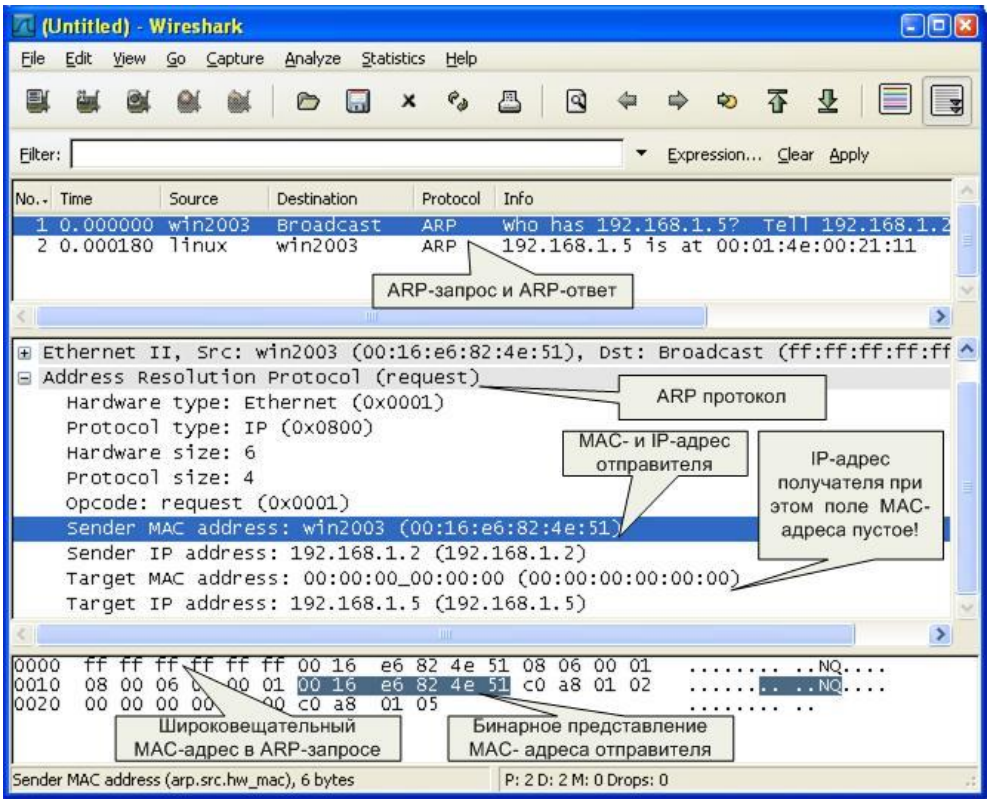

Рисунок 35. – Анализ ARP-запроса

Рассматривается ARP-запрос (пакет № 1). Выделяется пакет курсором, затем получается его раскладка по протоколам (Ethernet + ARP) в среднем окне. Wireshark очень наглядно «раскладывает» заголовок протокола по полям.

Можно видеть, что в пакете указаны MAC- и IP-адреса отправителя («Sender MAC address» и «Sender IP address» соответственно). Это параметры машины, с которой выполняется запрос. В данном случае запрос направлен на получение («Opcode: request» – запрос) MAC-адреса машины, у которой IP-адрес («Protocol type: IP») 192.168.1.5 («Target IP address»). При этом поле «Target MAC address» обнулено. Так как получатель ARP-запроса на момент запроса не известен, Ethernet-пакет отправляется всем машинам в данном локальном сегменте, о чем сигнализирует MAC-адрес Ethernet-пакета «ff:ff:ff:ff:ff:ff».

Примечание. Следует обратить внимание, что пакет представляет собой бинарную последовательность и сниффер выполняет большую работу по преобразованию полей из бинарного представления в удобочитаемый вариант.

Все работающие машины в сети получают пакет с ARP-запросом, анализируют его, а ответ отсылает только та машина, чей IP-адрес соответствует IP-адресу в запросе. Таким образом, второй полученный пакет является ARP-ответом (рисунок 36). Это следует из параметра поля «Opcode: reply». Обратите внимание, что данный пакет был отправлен именно той машиной, чей MAC-адрес нас и интересовал («Sender IP address: 192.168.1.5»). При этом поле «Sender MAC address» заполнено значением «00:01:4E:00:21:11».

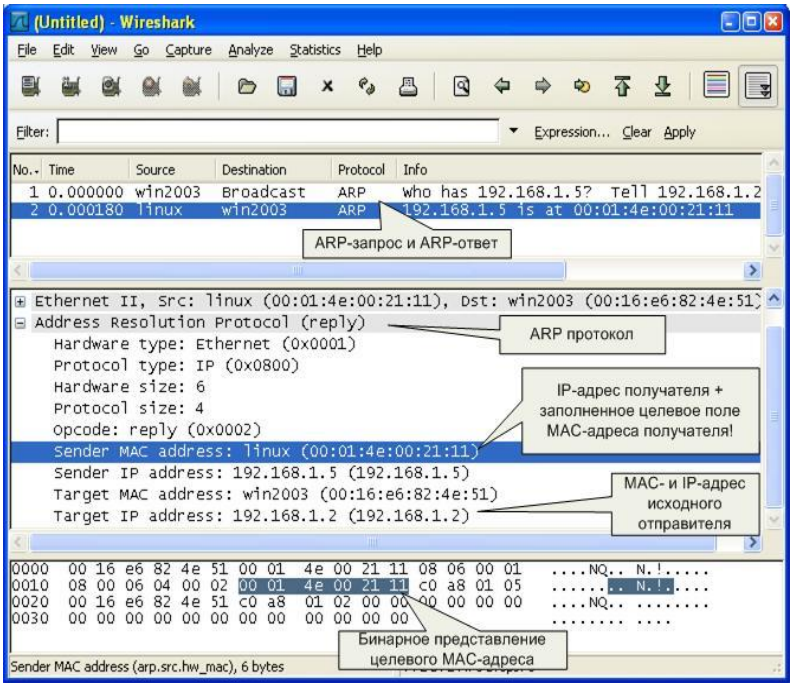

Рисунок 36. – Анализ ARP-ответа

Примечание. Обратить внимание на поле «Info» в списке захваченных пакетов. Сниффер и тут упрощает анализ сетевого трафика, подсказывая назначение пакетов.

Теперь можно повторно просмотреть ARP-кэш и сверить данные в нем с данными, которые получены из анализа пакетов ARP-запрос/ответа:

```
D:\>arp –a
Interface: 192.168.1.2 --- 0x10003
Internet Address      Physical Address      Type
192.168.1.5           00-01-4e-00-21-11     dynamic
```

Стоит также отметить, что в реальных условиях в локальной сети с большим количеством машин arp/rarp трафик бывает гораздо более интенсивным.

Ход работы

1. Изучить интерфейс программы Wireshark.
2. Захватить 100 произвольных пакетов. Определить статистические данные:
   – процентное соотношение трафика разных протоколов в сети;
   – среднюю скорость кадров/сек;
   – среднюю скорость байт/сек;
   – минимальный, максимальный и средний размеры пакета;
   – степень использования полосы пропускания канала (загрузку сети).
3. Зафиксировать 20 IP-пакетов. Определить статистические данные:
   – процентное соотношение трафика разных протоколов стека TCP/IP в сети;
   – средний, минимальный, максимальный размеры пакета.
4. Выполнить анализ ARP-протокола по примеру из методических указаний.
5. На примере любого IP-пакета указать структуры протоколов Ethernet и IP, отметить поля заголовков и описать их.
6. Проанализировать и описать принцип работы утилиты *ping*. При этом описать все протоколы, используемые утилитой. Описать все поля протоколов. Составить диаграмму взаимодействия машин при работе утилиты *ping*.

Примечание. Данная утилита использует протокол ICMP (RFC 792 и RFC 960).

Содержание отчета

1. Титульный лист.
2. Цель работы.
3. Краткие теоретические сведения.

4. Ход выполнения работы со скриншотами.
5. Краткий анализ пакетов.
6. Выводы.

Контрольные вопросы

1. Каковы основные цели мониторинга сетевого трафика?
2. Чем отличается мониторинг трафика от фильтрации?
3. Каково назначении класса программ-снифферов?
4. Какие основные функции выполняют снифферы?
5. Зачем используются фильтры отображения и фильтры захвата сниффера Wireshark? В чем их отличие?
6. Какие базовые функции статистической обработки захваченных пакетов имеет сниффер Wireshark?
7. На решение каких задач рассчитан протокол ARP?

СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ

1. Компьютерные сети : учеб. курс / Пер. с англ. – М. : Издательский отдел «Русская Редакция» ТОО «Channel Trading Ltd.», 2008.

2. Хиллей, В. Секреты Windows Server / В. Хиллей. – Киев : Диалектика, 2017.

3. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 4-е изд. – СПб. : Питер, 2019.

4. Хант, К. Серия «Для специалиста»: персональные компьютеры в сетях TCP/IP / К. Хант. – Киев : BHV, 2007.

5. Кастер, Х. Основы Windows NTFS / Х. Кастер ; пер. с англ. – М. : Издательский отдел «Русская Редакция» ТОО «Channel Trading Ltd.», 2016.