МИНОБРНАУКИ РОССИИ

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

IP-СЕТИ: МАРШРУТИЗАЦИЯ, НАСТРОЙКА, ОРГАНИЗАЦИЯ VLAN

Учебно-методическое пособие

Санкт-Петербург
СПбГЭТУ «ЛЭТИ»
2017

Авторы: **К. А. Борисенко, М. А. Фирсов, В. В. Яновский**

IP-сети: маршрутизация, настройка, организация VLAN. Учебно-методическое пособие. СПбГЭТУ «ЛЭТИ», 2017. 70 с.

Содержит теоретические сведения о структуре IP-сетей. Рассмотрены сетевой уровень организации стека протоколов TCP/IP, виртуальные локальные сети, средства настройки сети ОС MS Windows 10, межсетевой экран ОС Linux. Изложены принципы адресации в IP-сетях, механизмы статической маршрутизации, основы VLAN, различные способы организации изоляции узлов сети друг от друга. Содержит практические задания по моделированию и анализу IP-сетей на сетевом уровне в среде сетевого имитатора javaNetSim; задания по применению сетевых средств ОС; задания по конфигурированию виртуальных сетей.

Предназначено для студентов направлений подготовки 01.03.02 «Прикладная математика и информатика» и 09.03.04 «Программная инженерия», изучающих дисциплину «Сети и телекоммуникации».

Рецензент: зав. каф. информационно-измерительных систем и технологий, д.т.н., проф. В. В. Алексеев

Одобрено
Методической комиссией факультета компьютерных технологий и информатики в качестве учебно-методического пособия

ВВЕДЕНИЕ

Компьютерные сети и коммуникационные технологии являются одной из ключевых областей, знания в которой необходимы специалистам, занимающимся разработкой современных информационных систем и комплексов. Для того чтобы лучше понимать организацию вычислительных сетей, удобно использовать модель глобальной сети, часто называемую TCP/IP моделью. Каждый из уровней этой модели имеет собственные протоколы. Актуальность ее связана прежде всего с распространением и развитием сети Интернет, базирующейся на стеке TCP/IP, а также с большим соответствием процессу межсетевого взаимодействия.

Предлагаемый лабораторный практикум нацелен на изложение основных принципов организации IP-сетей на сетевом и канальном уровнях, а также отдельных сведений о прикладном уровне. Практикум ориентирован на студентов направлений подготовки 01.03.02 «Прикладная математика и информатика» и 09.03.04 «Программная инженерия», изучающих дисциплину «Сети и телекоммуникации».

Лабораторный практикум структурно разделен на три части. Первая часть (гл. 1–5) является основной. Она посвящена отдельным аспектам организации сетей. Во второй части (гл. 6) дается описание имитатора сетевых технологий, разработанного на кафедре МОЭВМ специально для проведения лабораторных работ. Третья часть практикума – справочная – включает в себя глоссарий и список литературы, рекомендуемой для более глубокого освоения предмета.

Каждая глава основной части практикума содержит теоретический материал, необходимый для освоения предмета, задания для самостоятельного выполнения их в лаборатории, а также список вопросов для проверки полученных знаний.

Главами 1 и 3 представлен сетевой уровень модели TCP/IP. В главе 1 рассматриваются принципы адресации, в том числе отображение IP-адресов в адреса физического уровня. В главе 3 освещаются вопросы маршрутизации и управления маршрутами. Лабораторные работы 1 и 3 разработаны усилиями студентов кафедры МО ЭВМ под руководством Алекперова А. И. и Большева А. К.

Глава 2 нацелена на практическое освоение сетевых средств (на примере ОС Microsoft Windows 10) на уровне пользователя. Рассматриваются вопросы настройки сети, брандмауэра, прокси-сервера, устранения неполадок в сети.

В главе 4 дано упрощённое описание межсетевого экрана iptables ОС Linux. В лабораторной работе 4 содержатся задания по использованию iptables.

В главе 5 рассматривается технология VLAN. Теоретический материал включает не только основы и классификацию VLAN, но и вопросы безопасности VLAN, асимметричные VLAN, технологию Q-in-Q. В качестве практических заданий студентам предлагается настроить заданные конфигурации сетей для удовлетворения ими требований по доступности и недоступности узлов.

Для того чтобы наглядно показать связи между уровнями модели TCP/IP, был разработан эмулятор javaNetSim, обеспечивающий иллюстрацию связей уровней между собой. В качестве исходного варианта использована разработка Канберрского технического университета JFirewallSim, распространяемая по лицензии BSD. Функциональность исходного варианта расширяется за счет введения средств иллюстрации уровней приложений, транспортного, сетевого и канального. На сетевом уровне введены функции бесклассовой адресации статической маршрутизации и работы с протоколом ARP.

Необходимость доработок исходного варианта вызвана отсутствием доступных эмуляторов, отвечающих поставленным требованиям, а именно: минимальным временны́м затратам на обучение, комплексным требованиям к аппаратным платформам, инвариантности к операционным системам. Малые объемы требуемой памяти, хорошие временны́е показатели, многоплатформенность, наглядный графический интерфейс, доступность javaNetSim обеспечивают ему преимущества по сравнению с другими эмуляторами.

1. IP-АДРЕСАЦИЯ

IP (Internet Protocol) – протокол сетевого уровня стека TCP/IP; согласно протоколу IP у каждого узла сети есть IP-адрес (адреса), что обеспечивает возможность пересылки пакетов между любыми узлами сети. Пересылка в глобальной сети (между сетями) выполняется маршрутизаторами на сетевом уровне, а в локальном сегменте сети доставка пакета до адресата выполняется на канальном уровне, по физическому адресу.

Сетевой уровень (межсетевой уровень) модели TCP/IP служит для образования единой транспортной системы, объединяющей несколько сетей, причем эти сети могут использовать совершенно различные принципы передачи сообщений между конечными узлами и обладать произвольной топологией.

На этом уровне термин «сеть» означает совокупность компьютеров, соединенных между собой в соответствии с некоторой топологией и использующих для передачи данных физический уровень модели TCP/IP. Единицей данных сетевого уровня является пакет.

На этом уровне определяются два вида протоколов – сетевые протоколы и протоколы маршрутизации. Первые реализуют продвижение пакетов через сеть. Это такие протоколы, как IP, ICMP, ARP и другие. Вторые предоставляют способы обмена информацией о маршрутах. Кроме того, сетевой уровень TCP/IP решает важную задачу идентификации узла-получателя пакета с помощью средств IP-адресации.

Канальный уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками передачи данных. Связь между канальным и сетевым уровнем обеспечивается протоколами преобразования между сетевыми и физическими адресами – ARP и InARP.

1.1. Типы сетевых адресов

Каждый компьютер в сети TCP/IP имеет адреса двух типов: физический (или локальный) и IP-адрес. Физический адрес узла определяется технологией построения отдельной сети, в которую входит данный узел. Для узлов, входящих в локальные сети, – это MAC-адрес сетевого адаптера или порта маршрутизатора. Такие адреса назначаются производителями оборудования и являются уникальными адресами, поскольку управляются централизовано. Для IPv4-сетей MAC-адрес имеет формат 6 байтов: старшие 3 байта – идентификатор фирмы производителя, а младшие 3 байта назначаются уникальным образом самим производителем. Для узлов, входящих в глобальные сети, такие, как X.25 или frame relay, физический адрес назначается администратором глобальной сети или производителем оборудования. Пример физического адреса: 44-BC-89-A2-FE-00.

Адрес IPv4 состоит из 4 байт. Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования компьютеров и маршрутизаторов. IP-Адрес состоит из двух частей: номера сети и номера узла. Номер сети определяет конкретную физическую сеть, а номер узла определяет конкретную рабочую станцию, сервер и пр., включенную в сеть. *Подсеть* – это физический сегмент TCP/IP сети, в котором используются IP-адреса с общим номером сети. Пример IP-адреса: 172.168.10.15.

Номер узла в протоколе IP назначается независимо от физического адреса узла. Деление IP-адреса на поле номера сети и номера узла – гибкое, и граница между этими полями может устанавливаться произвольно. Узел может входить в несколько IP-сетей. В этом случае узел должен иметь несколько IP-адресов (по числу сетевых связей).

Таким образом, IP-адрес характеризует не отдельный компьютер или маршрутизатор, а сетевой интерфейс (физический или виртуальный). IP-адрес имеет длину 4 байта и обычно записывается в виде четырех чисел, представляющих значения каждого байта в десятичной форме и разделенных точками, например:

10.10.1.4 – традиционная десятичная форма представления адреса,

00001010 00001010 00000001 00000100 – двоичная форма представления этого же адреса.

1.2. Структура IP-адреса

Какая часть IP-адреса относится к номеру сети, а какая – к номеру узла, определяется двумя способами: с помощью классов (классовая адресация) или с помощью масок подсети (бесклассовая адресация).

В классовой адресации номер сети и номер узла определяются по принадлежности IP-адреса одному из классов адресов: A, B, C, D или E. Класс определяется значениями первых битов адреса. Например, если первые два бита адреса равны 10, то сеть относится к классу B. В сетях класса B под адрес сети и под адрес узла отводится по 16 бит.

В бесклассовой адресации номер сети, к которой принадлежит узел с заданным IP-адресом, определяется другим способом: вместе с IP-адресом предоставляется и *маска подсети*. В терминологии сетей TCP/IP маской подсети или маской сети называется битовая маска, определяющая, какая часть IP-адреса узла сети относится к адресу сети, а какая – к адресу самого узла в этой сети. Например, узел с IP-адресом 192.168.0.1 и маской подсети 255.255.255.0 находится в сети 192.168.0.0. В настоящее время бесклассовая нумерация сетей преобладает.

Чтобы получить адрес сети, зная IP-адрес и маску подсети, необходимо применить к ним операцию поразрядной конъюнкции. Пример показан в таблице.

Часть адреса	Бинарное представление	Значение
IP-адрес	00001010 00001010 00000001 00000100	10.10.1.4

Маска подсети	11111111 00000000 00000000 00000000	255.0.0.0
Адрес сети	00001010 00000000 00000000 00000000	10.0.0.0

Для стандартных классов сетей маски имеют следующие значения:

– 255.0.0.0 – маска для сети класса A (8 бит на адрес сети);

– 255.255.0.0 – маска для сети класса B (16 бит на адрес сети);

– 255.255.255.0 – маска для сети класса C (24 бита на адрес сети).

1.3. Отображение физических адресов на логические

Отображение физических адресов на IP-адреса происходит с помощью протокола *ARP*. Функционирование ARP происходит различным образом – в зависимости от того, какой протокол канального уровня работает в данной сети. В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом. Узел, которому нужно выполнить отображение IP-адреса на локальный адрес, формирует ARP-запрос, вкладывает его в кадр протокола канального уровня с указанием в нем известного IP-адреса и осуществляет его широковещательную рассылку по сети. Все узлы локальной сети получают ARP-запрос и сравнивают указанный там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свои IP- и локальный адреса.

ARP-запросы и ответы используют один и тот же формат пакета. Так как локальные адреса могут в различных типах сетей иметь различную длину, то формат пакета протокола ARP зависит от типа сети. Для того чтобы не перегружать сеть запросами, ARP использует таблицу отображения (так называемый ARP-кэш). Эта таблица содержит три поля – IP-адрес, соответствующий ему MAC-адрес и тип. Тип может быть статическим или динамическим. Запись в таблице имеет динамический тип, если она внесена в таблицу путем широковещательного запроса ARP. Такие записи имеют время устаревания (обычно 180 или 360 с), после истечения которого они из таблицы удаляются. Запись в таблице будет иметь статический тип, если она добавлена вручную (например, с помощью команды `arp` в ОС Windows или Unix). Статическая запись имеет неограниченное время устаревания.

В IPv6-сетях функции ARP выполняет протокол NDP (Neighbor Discovery Protocol), использующий пакеты ICMPv6 определенных типов.

1.4. Маршрутизация по умолчанию

Для объединения подсетей в единую сеть в простейшем случае используется маршрутизация по умолчанию. Она организуется посредством шлюзов. Шлюзом будем называть узел внутри подсети, который предоставляет доступ в другую подсеть. Чаще всего в виде шлюза выступает маршрутизатор. Схема такой маршрутизации выглядит следующим образом: задан адрес *шлюза по умолчанию*. При попытке отправки пакета в сеть, узел проверяет совпадение подсети назначения пакета с подсетью узла.

Если подсети разные, то пакет отправляется на шлюз. В простейшем случае шлюз сравнивает сеть IP-адреса назначения с номерами сетей на своих интерфейсах и в случае их совпадения направляет пакет в узел назначения через этот сетевой интерфейс. В противном случае он отправляет пакет в узел, указанный в качестве *шлюза по умолчанию* на самом шлюзе. Если такового нет, то пакет теряется.

1.5. Протокол ICMP

Для проверки соединений и корректного функционирования сети обычно используется *протокол ICMP*. ICMP (Internet Control Message Protocol) – протокол управляющих сообщений интернета. ICMP – протокол сетевого уровня и работает поверх протокола IP. Он предназначен для обмена информацией об ошибках между маршрутизаторами (шлюзами) сети и узлом-источником пакета. С помощью специальных пакетов этот протокол сообщает о невозможности доставки пакета, превышении времени жизни, об аномальных значениях параметров, изменении маршрута пересылки, о состоянии системы и т. п.

В простейшем случае, для проверки работоспособности сети используются два сообщения ICMP: Echo-запрос (Echo request) и Echo-ответ (Echo reply). Когда на узел приходит сообщение ICMP типа «Echo-запрос», он отправляет сообщение «Echo-ответ» на тот узел, с которого пришел запрос. Пример реализации такого обмена представлен в утилите ping, входящей в состав почти любой сетевой ОС.

1.6. Лабораторная работа 1. Настройка IP-адресов в сети

Цель: *изучение и практическое освоение основ адресации, разрешения физических адресов и простейшей маршрутизации в IP-сетях.*

1.6.1. Порядок выполнения работы

1. Исправить структуру сети (если это необходимо), обеспечив корректную доставку кадров на физическом уровне.

2. Задать IP-адреса, маски подсети и шлюзы по умолчанию для всех узлов сети, чтобы обеспечить корректную доставку Echo-запроса от К1 к К2 и Echo-ответа обратно. Обосновать свои установки.

3. Выполнить Echo-запрос с К1 на К2. Посмотреть вывод программы.

4. Добавить статическую запись ARP для К3 на К1 (или для ближайшего к К1 маршрутизатора, находящегося между К3 и К1). Подождать устаревания ARP-таблиц и выполнить Echo-запрос с К1 на К3. Объяснить результат.

5. Выполнить Echo-запрос на IP-адрес 200.100.0.1 с К1. Объяснить вывод программы.

6. Выполнить Echo-запросы с К1 и К2 на все узлы сети. Убедиться, что Echo-ответы приходят.

В отчет необходимо включить схему сети, настройки протокола TCP/IP для всех узлов сети и результаты вывода программы, полученные при выполнении Echo-запросов.

1.6.2. Варианты заданий

Вариант 1. Файл со схемой сети: lab1_var1.jfst. Сеть между маршрутизаторами R1, R2 и Boss_R: 117.168.0.0. Компьютер Boss имеет IP-адрес 64.2.0.1. Компьютер Hacker имеет IP-адрес 117.168.0.5. Обозначения в задании: К1 – Boss, К2 – Hacker, К3 – OFFICE2 pc1.

Вариант 2. Файл со схемой сети: lab1_var2.jfst. Сеть между маршрутизаторами OFF_R и R2: 136.15.0.0. Компьютер BIG BOSS имеет IP-адрес 136.15.32.1. Компьютер M_CH_S имеет IP-адрес 10.10.0.2. Сеть между маршрутизаторами R2 и M_CH_S_Router: 192.178.0.0. Обозначения в задании: К1 – BIG BOSS, К2 – M_CH_S, К3 – OFFICE1_pc4.

Вариант 3. Файл со схемой сети: lab1_var3.jfst. Сеть между маршрутизаторами R1, R2 и Boss_R: 172.198.0.0. Компьютер Boss имеет IP-адрес 10.2.0.1. Компьютер Hacker имеет IP-адрес 172.198.99.252. Обозначения в задании: К1 – Boss, К2 – Hacker, К3 – OFFICE2_pc1.

Вариант 4. Файл со схемой сети: lab1_var4.jfst. Сеть между маршрутизаторами OFF_R и R2: 204.188.0.0. Компьютер BIG BOSS имеет IP-адрес 204.188.0.1. Компьютер M_CH_S имеет IP-адрес 10.0.0.2. Сеть между маршрутизаторами R2 и M_CH_S_Router: 192.178.0.0. Обозначения в задании: К1 – BIG BOSS, К2 – M_CH_S, К3 – OFFICE1_pc4.

Вариант 5. Файл со схемой сети: lab1_var5.jfst. Сеть между маршрутизаторами RServers, RManagers и RBosses: 10.0.0.0. Компьютер MegaBoss имеет IP-адрес 172.16.0.5. Компьютер Manager2 имеет IP-адрес 172.16.1.12. Компьютер FileServer имеет IP-адрес 172.16.10.10. Обозначения в задании: K1 – MegaBoss, K2 – Manager2, K3 – File-Server.

Вариант 6. Файл со схемой сети: lab1_var6.jfst. Сеть между маршрутизаторами RServers, RManagers и RBosses: 192.168.0.0. Компьютер MicroBoss имеет IP-адрес 10.0.1.5. Компьютер Manager3 имеет IP-адрес 10.0.2.5. Компьютер PrintServer имеет IP-адрес 10.0.64.1. Обозначения в задании: K1 – Manager3, K2 – PrintServer, K3 – MicroBoss.

Вариант 7. Файл со схемой сети: lab1_var7.jfst. Сеть между маршрутизаторами R1 и ADSL: 172.168.0.0. Компьютер Station1 имеет IP-адрес 172.168.1.2. Компьютер Remote1 имеет IP-адрес 10.0.0.110. Сеть между маршрутизаторами ADSL и ADSL2: 192.168.0.0. Обозначения в задании: K1 – Station1, K2 – Remote1, K3 – Station2.

Вариант 8. Файл со схемой сети: lab1_var8.jfst. Сеть между маршрутизаторами R1 и ADSL: 192.168.0.0. Компьютер Station1 имеет IP-адрес 192.168.1.2. Компьютер Remote1 имеет IP-адрес 99.11.0.11. Сеть между маршрутизаторами ADSL и ADSL2: 172.168.0.0. Обозначения в задании: K1 – Station1, K2 – Remote1, K3 – Station2.

Вариант 9. Файл со схемой сети: lab1_var9.jfst. Сеть между маршрутизаторами R1 и R2: 192.168.100.0. Компьютер PC1 имеет IP-адрес 129.64.128.1. Компьютер PC2 имеет IP-адрес 129.64.127.254. Компьютер PC4 имеет IP-адрес: 10.0.0.2. Длина маски подсети (количество значащих единиц) на PC1, PC2, PC3 должно быть минимально возможным (обеспечивая при этом корректную работу). Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 10. Файл со схемой сети: lab1_var10.jfst. Сеть между маршрутизаторами R1 и R2: 192.168.0.0. Компьютер PC1 имеет IP-адрес 172.168.0.1. Компьютер PC2 имеет IP-адрес 172.168.0.65. Компьютер PC4 имеет IP-адрес: 1.0.0.2. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 11. Файл со схемой сети: lab1_var11.jfst. Сеть между маршрутизаторами R-C-M и R-S-C: 10.1.0.0. Сеть между маршрутизаторами R-C-M и R-M-S: 10.0.32.0. Сеть между маршрутизаторами R-M-S и R-S-C: 10.0.0.128. Компьютер Chief имеет IP-адрес 10.1.0.3. Компьютер Manager1 имеет IP-адрес 10.0.32.11. Компьютер Service имеет IP-адрес: 10.0.0.135. Обозначения в задании: K1 – Chief, K2 – Manager1, K3 – Service.

Вариант 12. Файл со схемой сети: lab1_var12.jfst. Сеть между маршрутизаторами R-C-M и R-S-C: 172.168.128.0. Сеть между маршрутизаторами R-C-M и R-M-S: 172.168.1.0. Сеть между маршрутизаторами R-M-S и R-S-C: 172.168.0.64. Компьютер Chief имеет IP-адрес 172.168.128.5. Компьютер Manager3 имеет IP-адрес 172.168.1.13. Компьютер Service имеет IP-адрес: 172.168.0.76. Обозначения в задании: K1 – Manager3, K2 – Service, K3 – Chief.

Вариант 13. Файл со схемой сети: lab1_var13.jfst. Сеть между маршрутизаторами R120, R230 и R232: 172.31.128.0. Сеть между маршрутизаторами R232 и R233: 10.10.0.0. Компьютер Remote1 имеет IP-адрес 172.31.127.0. Компьютер Remote2 имеет IP-адрес 172.31.200.1. Компьютер Remote3 имеет IP-адрес: 10.0.39.0. Обозначения в задании: K1 – Remote1, K2 – Remote2, K3 – Remote3.

Вариант 14. Файл со схемой сети: lab1_var14.jfst. Сеть между маршрутизаторами R120, R230 и R232: 63.12.95.0. Сеть между маршрутизаторами R232 и R233: 63.12.225.0. Компьютер Remote1 имеет IP-адрес 168.20.88.0. Компьютер Remote2 имеет IP-адрес 63.12.95.1. Компьютер Remote3 имеет IP-адрес: 168.20.120.0. Обозначения в задании: K1 – Remote2, K2 – Remote3, K3 – Remote1.

1.6.3. Пример выполнения лабораторной работы

Рассмотрим конфигурацию сети, приведенную на рис. 1.1. Файл со схемой сети: lab1_sample.jfst. Сеть между маршрутизаторами R1 и R2: 172.168.100.0. Компьютер PC1 имеет IP-адрес 172.168.0.2. Компьютер PC2 имеет IP-адрес 10.0.0.2.

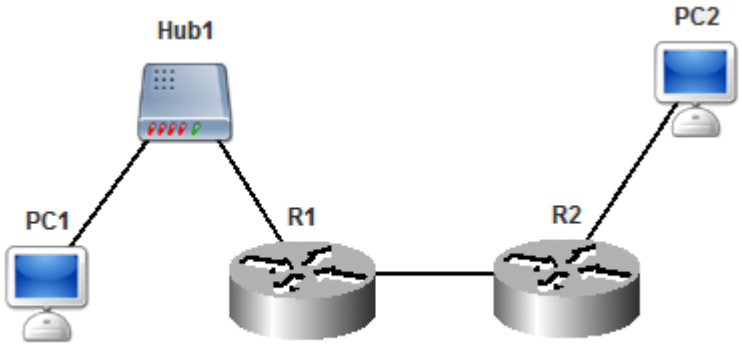

Рис. 1.1

Задание

1. Задать маски подсети и шлюзы по умолчанию для PC1 и PC2, а также IP-адреса из указанного диапазона вместе с масками и шлюзами для R1 и R2

так, чтобы обеспечить корректную доставку Echo-запроса от PC1 к PC2 и Echo-ответа обратно. Обосновать свои установки.

2. Выполнить Echo-запрос с PC1 на PC2. Проанализировать вывод.

3. Выполнить Echo-запрос на IP-адрес 192.168.0.1 с PC1. Объяснить вывод программы.

Выполнение работы

1. Зададим IP-адреса и маски подсети для маршрутизаторов R1 и R2. Сети 172.168.100.0 и 172.168.0.0 (если использовать стандартную маску подсети для класса B) эквивалентны. Будем использовать маску подсети, отличную от стандартной. Зададим для маршрутизатора R1 на интерфейсе eth0 адрес 172.168.0.1 и маску подсети 255.255.255.0. На интерфейсе eth1 установим адрес 172.168.100.1 и маску 255.255.255.0. Теперь необходимо сконфигурировать маршрутизатор R2. На его интерфейсе eth0 зададим IP 172.168.100.2 и маску 255.255.255.0. Установим шлюз по умолчанию в 172.168.100.1. На интерфейсе eth1 для R2 установим любой IP-адрес из диапазона сети PC2, например 10.0.0.1 и соответствующую ему маску: 255.0.0.0. Для корректной маршрутизации осталось задать только шлюз по умолчанию для R1. Он будет адресом маршрутизатора R2.

2. Теперь настроим конечные узлы. На PC1 зададим маску подсети, соответствующую новому адресному пространству: 255.255.255.0. Так как пакеты от узла PC1 в другие сети должны проходить через маршрутизатор R1, зададим шлюз по умолчанию 172.168.0.1 (адрес R1). Аналогичные операции проведем на PC2: установим маску подсети в 255.0.0.0, а шлюз по умолчанию в 10.0.0.1. Стоит заметить, что приведенная конфигурация не является единственно верной.

3. После отправки Echo-запроса с PC1 на PC2 в консоли будет выведен результат прохождения запроса и ответа на него по сети:

```
PC1  Created Echo Request packet to 10.0.0.2
PC1  Created ARP discovery packet to source MAC address.
     for IP 172.168.0.1
PC1  Sending broadcast packet from ProtocolStack.
...
PC1  ProtocolStack received packet from local Interface.
PC1  Confirmed Packet is for this Network Layer Device.
PC1  Echo reply packet received from 10.0.0.2
```

Как видно, PC1 успешно получил Echo-ответ на свой запрос к PC2.

4. Выполним Echo-запрос для несуществующего узла с IP-адресом 192.168.0.1. Для этого выполним на PC1 последовательность действий, аналогичную предыдущему пункту, вместо адреса 10.0.0.2 используя адрес 192.168.0.2:

```
PC1  Created Echo Request packet to 192.168.0.1
PC1  Sending packet from ProtocolStack (to 172.168.0.1).
...
R1   ProtocolStack received packet from local Interface.
R1   Packet Received: Network Layer Device is
     Routable forwarding packet.
R1   Forwarding packet from ProtocolStack
     (to 172.168.100.1).
R2   ProtocolStack received packet from local Interface.
R2   Packet Dropped: Hop count exceeded.
Host 192.168.0.2 Unreachable
```

Как видно, пакет попал в «петлю» между двумя маршрутизаторами и находился там, пока у него не закончилось время жизни (TTL).

1.6.4. Контрольные вопросы

1. Что такое кэш ARP? Какие типы записей могут содержаться в кэше ARP?

2. Какому классу IP-адресов принадлежат адреса 10.11.0.1, 127.1.1.1?

3. Разделите адресное пространство 192.168.1.0 на четыре подсети при помощи масок.

4. Что такое концентратор? Объясните принцип работы концентратора. Чем концентратор отличается от повторителя?

5. Что такое шлюз?

6. Для чего предназначен протокол ICMP?

2. СЕТЕВЫЕ СРЕДСТВА И НАСТРОЙКИ ОС MS WINDOWS 10. ПОДКЛЮЧЕНИЕ К СЕТИ

2.1. Подключение на физическом и канальном уровнях

Подключение компьютера к локальной сети начинается с установки драйвера сетевого адаптера (если он еще не установлен) и обеспечения возможности взаимодействия с сетью на физическом уровне (соединения сетевого адаптера с другим сетевым устройством патч-кордом либо размещения беспроводного адаптера в зоне покрытия сигнала). Установить драйвер сете-

вого адаптера, включить и выключить его можно в диспетчере устройств («Панель управления» – «Система» – «Диспетчер устройств») (рис. 2.1).

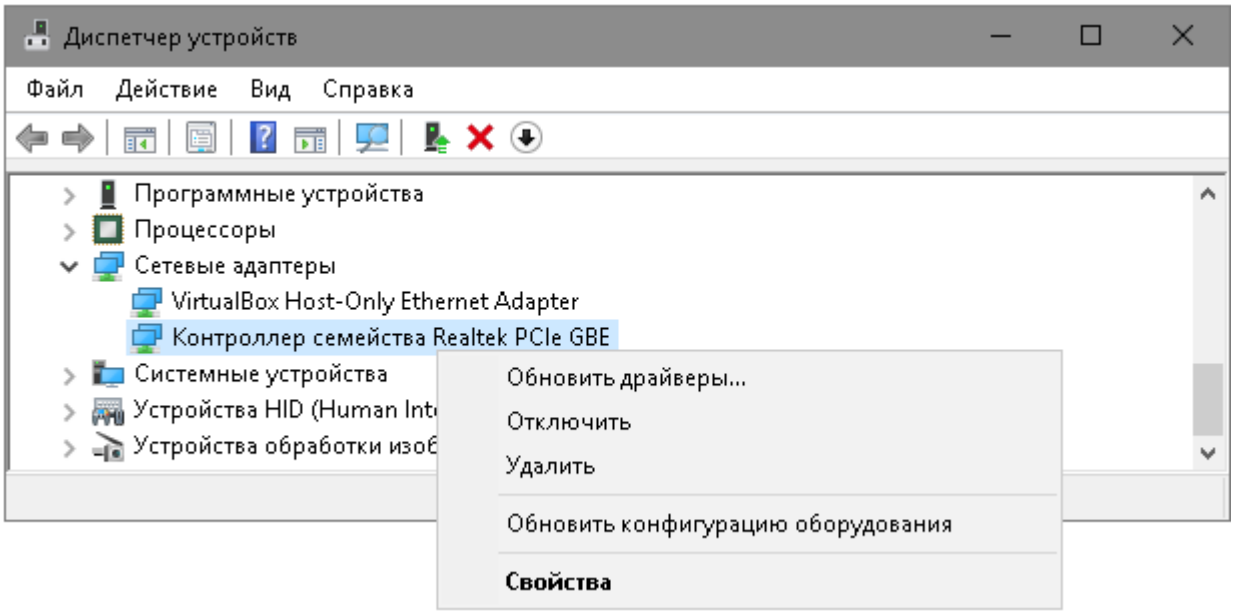

Рис. 2.1

Устройство, драйвер которого не установлен, будет отмечено знаком вопроса.

В окне свойств сетевого адаптера доступны различные его настройки, например: связанные с электропитанием (в отдельной вкладке), с возможностью выхода компьютера из сна по сигналу из сети; управляющие аппаратным ускорением адаптера; включение и отключение использования больших кадров; изменение MAC-адреса; включение и отключение поддержки тегов приоритета и VLAN в кадрах. Список настроек зависит от используемого адаптера и драйвера.

При подключении к беспроводной сети (Wi-Fi) следует выбрать SSID (Service Set Identifier – идентификатор беспроводной сети) точки доступа среди доступных для подключения и ввести пароль. Если используется протокол аутентификации EAP, то вместо ввода пароля может потребоваться установка сертификата-ключа на подключаемое беспроводное устройство. Если точка доступа не транслирует свой SSID, то беспроводная сеть не появится в списке доступных автоматически. В таком случае следует вручную создать беспроводное подключение, введя при этом SSID. На точке доступа возможна настройка таких параметров, как используемый канал, мощность сигнала, способ шифрования и авторизации и др.

При обнаружении новой сети Windows задаст вопрос «Вы хотите разрешить другим компьютерам и устройствам в этой сети обнаруживать ваш ПК?». Ответ определит, какой профиль сети будет использоваться – «Частная сеть» при ответе «Да» или «Общедоступная сеть» при ответе «Нет». От этого зависит, какие ограничения будут налагаться на сетевые взаимодействия. Если сеть будет использоваться только для выхода в Интернет, то лучше ответить «Нет»: большее количество ограничений обеспечивает бóльшую безопасность; если будут использоваться возможности локальной сети, то лучше ответить «Да».

2.2. Базовые настройки сети: IP-адрес, маска подсети, шлюз по умолчанию, DNS-серверы

После обеспечения физического соединения настраиваются автоматически либо вручную базовые параметры подключения: IP-адрес и маска подсети, адрес шлюза по умолчанию, DNS-серверы. Перейти к этим настройкам можно так: «Панель управления» – «Центр управления сетями и общим доступом» – «Изменение параметров адаптера» – открыть свойства подключения – в окне свойств выбрать из списка компонент «IP версии 4 (TCP/IPv4)» и нажать кнопку «Свойства» (рис. 2.2).

При автоматической настройке компьютер получает настройки подключения от DHCP-сервера. Часто DHCP-сервер работает на ближайшем маршрутизаторе в сети. Если DHCP-сервер оказывается недоступен, то на компьютере будет установлен адрес из диапазона APIPA – 169.254.0.0/16.

DNS-сервер используется для преобразования доменных адресов в IP-адреса. Например, чтобы зайти на сайт yandex.ru, компьютер передаст доменное имя yandex.ru DNS-серверу, а в ответ получит информацию об IP-адресе, на котором располагается сайт. Соответствие между некоторыми доменными именами и IP-адресами можно задать для компьютера вручную. Для этого следует добавить соответствующие записи в файл Windows\System32\drivers\etc\hosts.

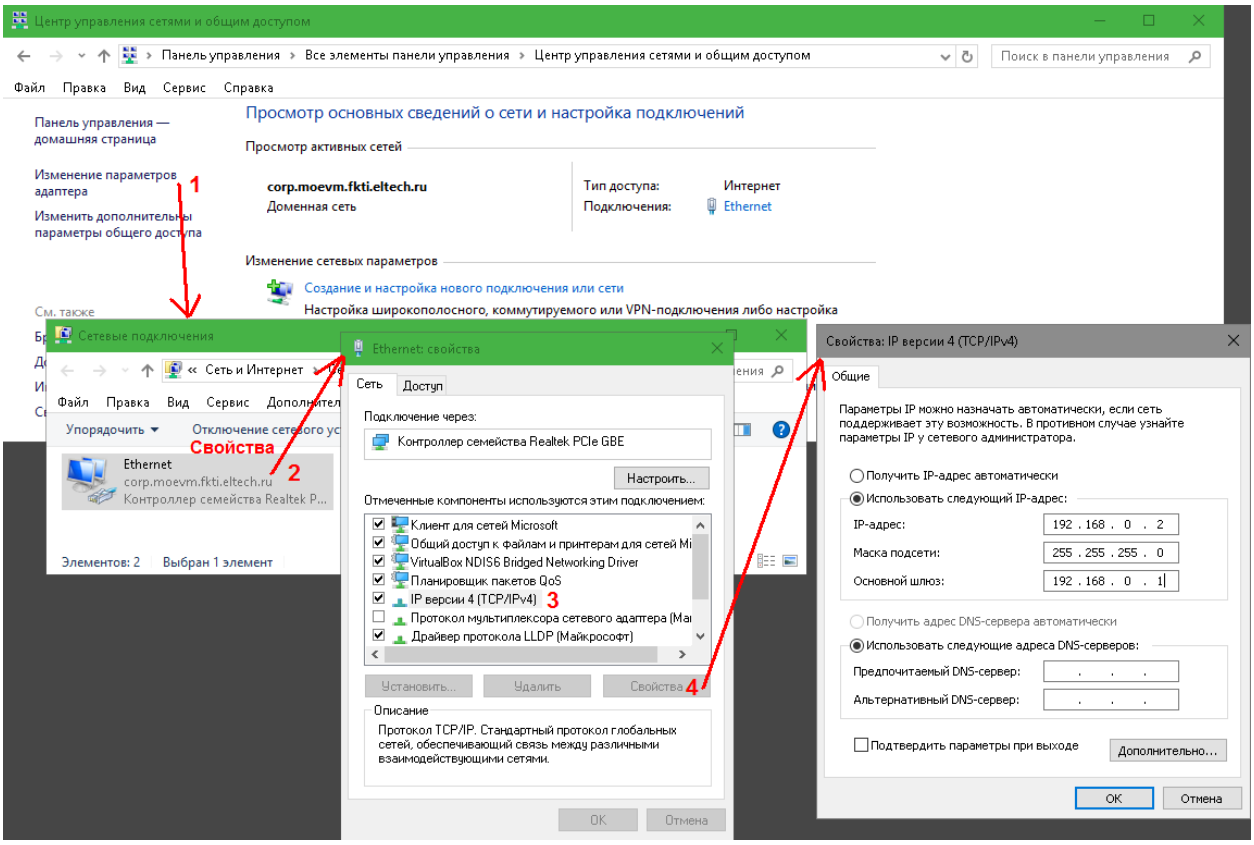

Рис. 2.2

Проверить доступность и работоспособность DNS-сервера можно с помощью команды `nslookup`. Если выполнить в командной строке команду `nslookup <доменное имя>`, то на консоли отобразится ответ, в котором будут IP-адреса, соответствующие доменному имени. `nslookup` обращается только к первому DNS-серверу из известных компьютеру (и по умолчанию запрос нерекурсивный), так что в ответе не будет IP-адреса запрошенного сайта, если первый DNS-сервер его не знает.

В окне свойств подключения (окно «Ethernet: свойства» на рис. 2.2) можно устанавливать и удалять, включать и отключать «компоненты», используемые подключением. Каждый компонент отвечает за те или иные сетевые возможности.

2.3. Изменение профиля сети и настройка брандмауэра

Брандмауэр (межсетевой экран, файрвол) – это программный или программно-аппаратный элемент компьютерной сети, фильтрующий проходящий через него трафик в соответствии с заданными правилами. Правила определяют критерии, по которым трафик должен быть пропущен или заблокирован. Лишние разрешения в брандмауэре могут снижать безопасность ком-

пьютера, а лишние запреты нарушают работу сетевых программ и возможностей.

Набор действующих правил брандмауэра зависит от профиля сети. Так, правила для «общедоступной» сети не будут действовать, если профиль сети – «Частная сеть». Всего есть 3 профиля – «Частная сеть», «Общедоступная сеть» и «Доменная сеть». Профиль сети (тип сети) можно узнать в Центре управления сетями и общим доступом. На рис. 2.2 можно увидеть, что подключение использует профиль «Доменная сеть». Изменить профиль можно разными способами. Один из способов: открыть «Параметры Windows» – «Сеть и интернет» – «Ethernet» (или «Wi-Fi») – выбрать сетевое подключение (рис. 2.3) – изменить положение переключателя «Позвольте другим компьютерам и устройствам в этой сети...» (рис. 2.4).

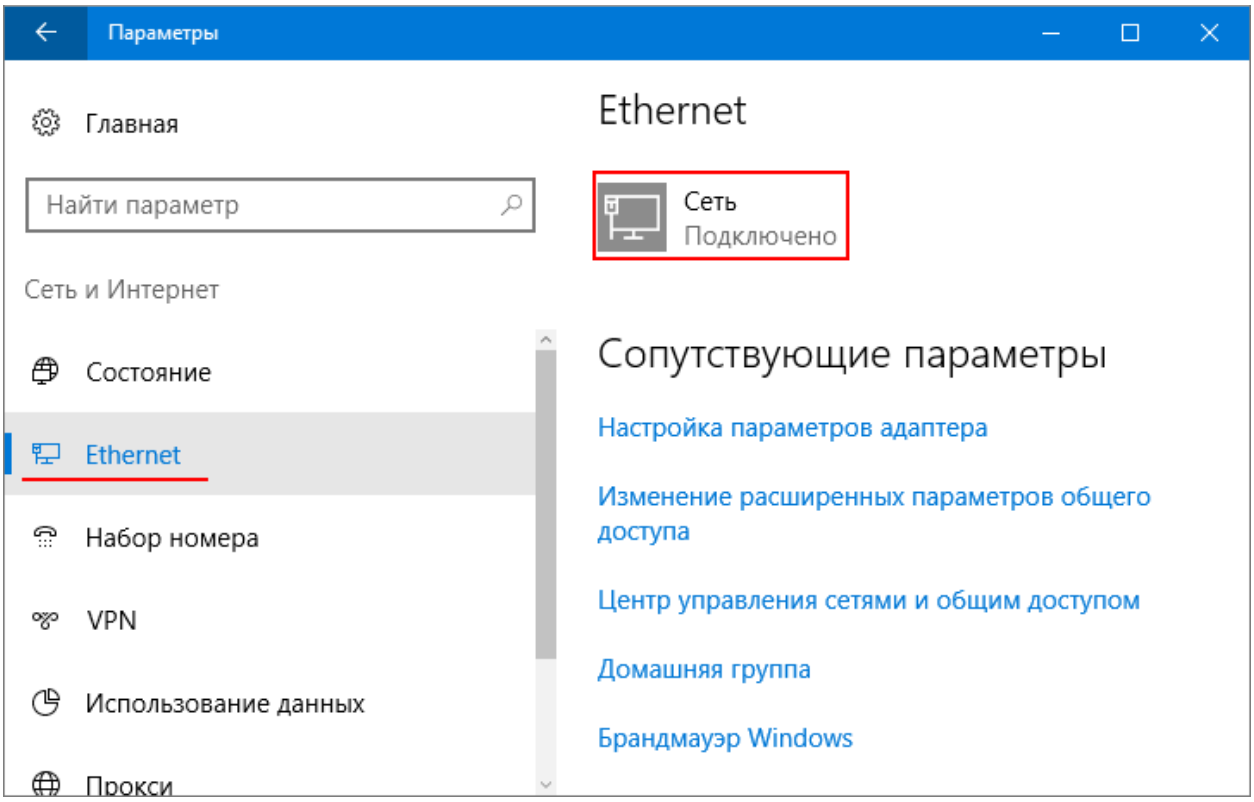

Рис. 2.3

Положение «Вкл.» изменяет профиль сети на «Частная сеть», «Выкл.» – на «Общедоступная сеть».

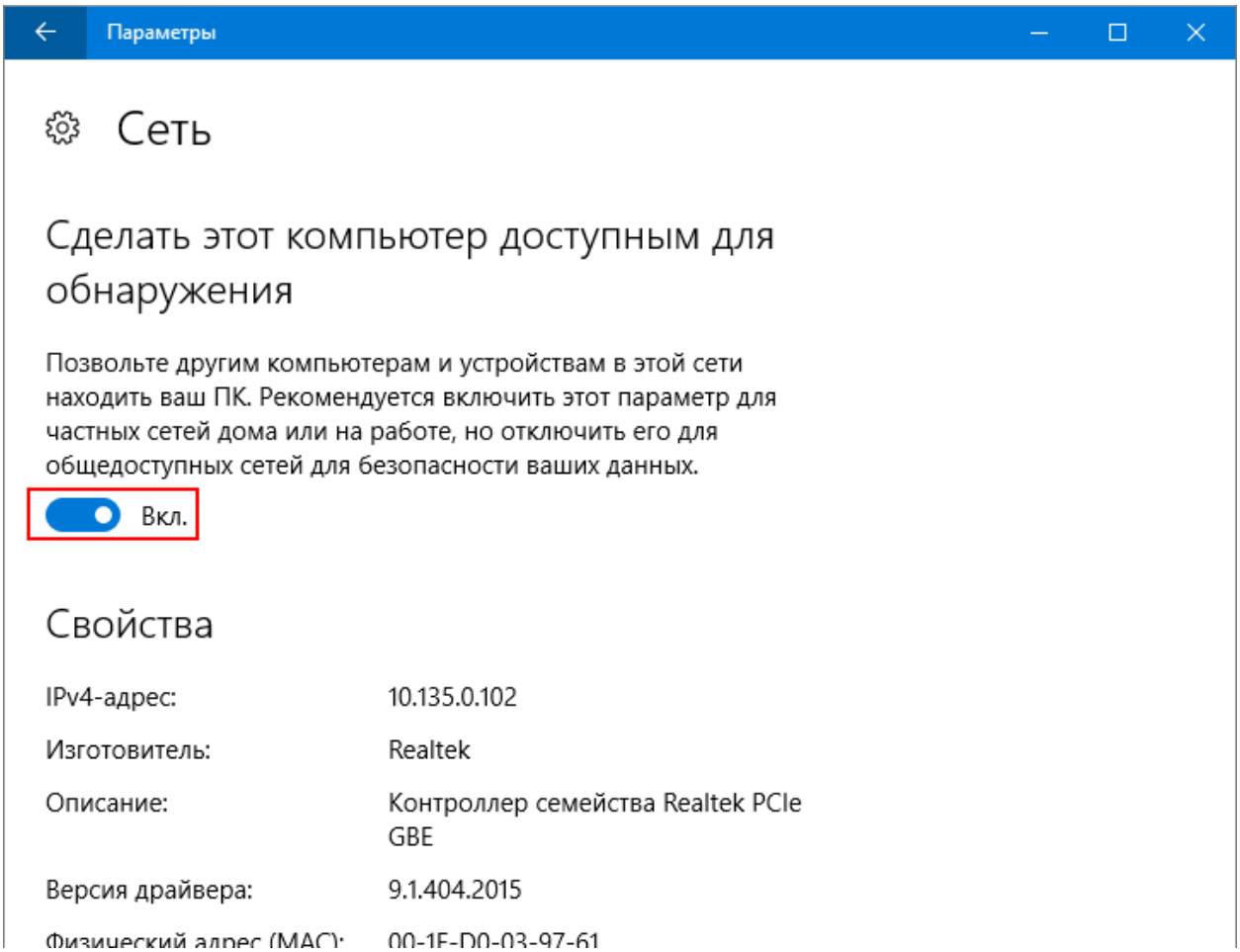

Рис. 2.4

Другой способ изменения профиля сети – изменение значения ключа Category, расположенного в реестре по адресу HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\NetworkList\Profiles\<Ветвь подключения>\ 0 – общедоступная сеть, 1 – частная сеть, 2 – сеть домена.

Перейти к настройке брандмауэра можно из панели управления (рис. 2.5).

Выбрав «Включение и отключение брандмауэра Windows», для каждого из 3 профилей можно включить или выключить брандмауэр; уведомления брандмауэра; блокировку всех входящих соединений, игнорирующую разрешения. Фактическое поведение брандмауэра не изменится, если изменить настройки для неиспользуемого профиля (т.е. такого, в котором нет доступных сетей). Оно изменится, если изменить профиль сети.

Выбрав «Разрешение взаимодействия с приложением или компонентом в брандмауэре Windows», можно создавать и удалять правила, разрешающие входящие подключения для определенных приложений.

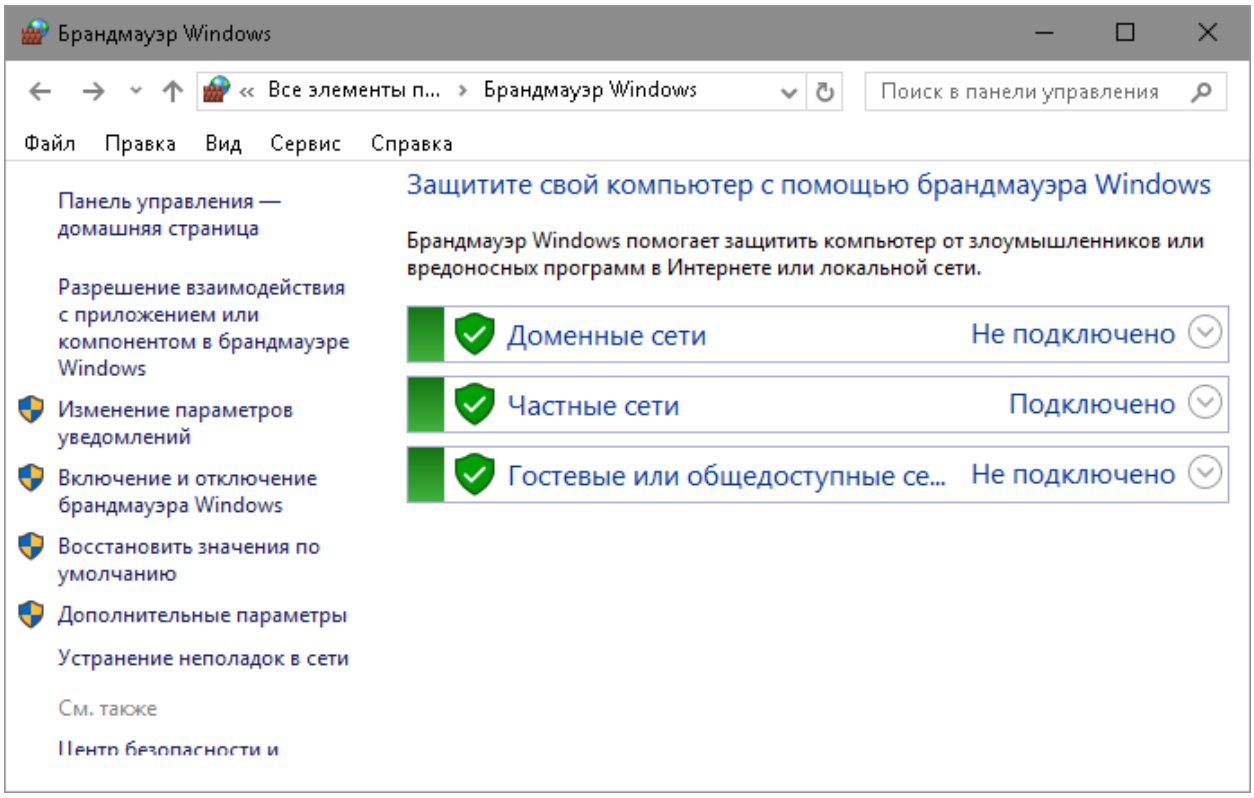

Рис. 2.5

Для тонкой настройки правил брандмауэра следует выбрать «Дополнительные параметры». Откроется окно «Брандмауэр Windows в режиме повышенной безопасности» (рис. 2.6).

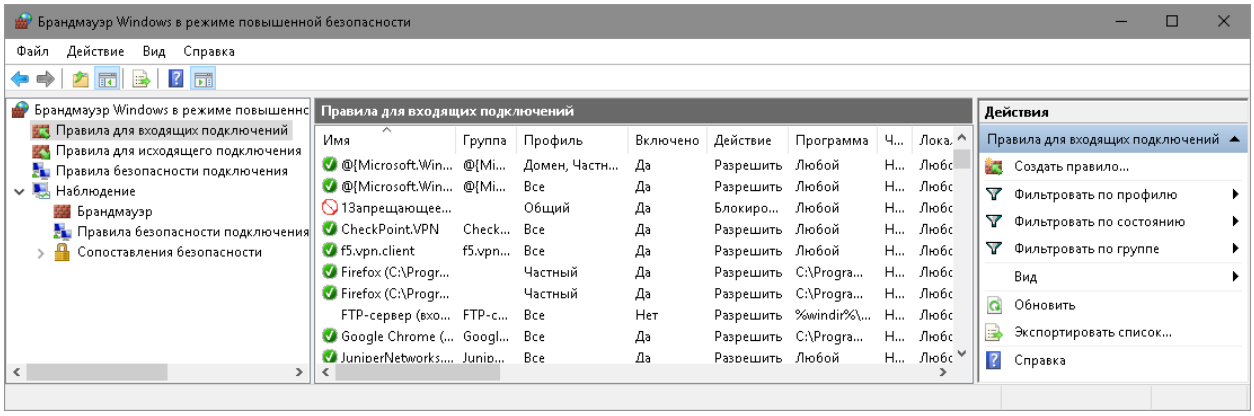

Рис. 2.6

В нем можно настраивать правила для входящих подключений, правила для исходящих подключений и правила безопасности подключения. Для ка-

ждого типа правил можно вывести таблицу со списком правил. В столбцах таблицы выводятся различные характеристики правил. Среди характеристик правил для входящих и для исходящих подключений особо можно отметить следующие:

- «Включено» – определяет, может ли действовать правило;
- «Профиль» – определяет, для каких профилей актуально правило;
- «Действие» – «Разрешить» (разрешающее правило), «Блокировать» (запрещающее правило), «Обеспечить безопасность» (разрешает только безопасные подключения; это действие имеет свои настройки).

Для исходящих соединений действует принцип «разрешено все, что не запрещено», а для входящих – «запрещено все, что не разрешено». Запрещающие правила имеют приоритет над разрешающими.

Среди критериев, которые можно использовать для принятия решения о пропуске трафика, есть используемые протокол и порт; программа, устанавливающая соединение; IP-адрес узла, с которым устанавливается соединение; и др.

Пример правила входящего подключения – предустановленное в Windows правило с именем «Общий доступ к файлам и принтерам (эхо-запрос - входящий трафик ICMPv4)» (правил с таким именем несколько, они различаются списками профилей, к которым применяются). Если оно выключено или удалено, то компьютер не будет отзываться на запросы, выполняемые с другого хоста командой `ping` с использованием IPv4.

Правила безопасности подключений указывают, как и когда выполняется проверка подлинности. Эти правила применяются в тех случаях, когда в правилах для входящих и исходящих подключений установлено действие «Разрешить только безопасное подключение».

В разделе «Наблюдение» можно увидеть список фактически действующих правил. Исключены выключенные правила; правила из неактивных профилей; правила, перекрытые другими правилами.

2.4. Прокси-сервер

Прокси-сервер (от англ. proxy – «представитель, уполномоченный») – сервер, работающий на прикладном уровне в качестве посредника между клиентами и другими узлами. Он позволяет клиентам выполнять косвенные запросы к другим узлам. Клиент подключается к прокси-серверу и запрашивает какой-либо ресурс, расположенный на другом сервере; прокси-сервер

либо подключается к указанному серверу и получает ресурс у него, либо возвращает ресурс из собственного кэша (если прокси имеет свой кэш). В некоторых случаях прокси-сервер может модифицировать запрос клиента или ответ сервера в определенных целях.

Прокси-сервер может поддерживать передачу по отдельным протоколам (HTTP, HTTPS, FTP) или быть универсальным прокси-сервером, поддерживающим практически любые протоколы. В последнем случае используется протокол SOCKS.

На маршрутизаторе может быть настроен прозрачный прокси, в таком случае он используется автоматически без каких-либо настроек ОС или программ на компьютере. Чтобы использовать непрозрачный прокси, соответствующие настройки должны быть сделаны в ОС или в сетевой программе.

Настроить системный прокси можно через браузер Internet Explorer, который его использует: Сервис-кнопка – «Свойства браузера» – вкладка «Подключения» – кнопка «Настройка сети» (рис. 2.7).

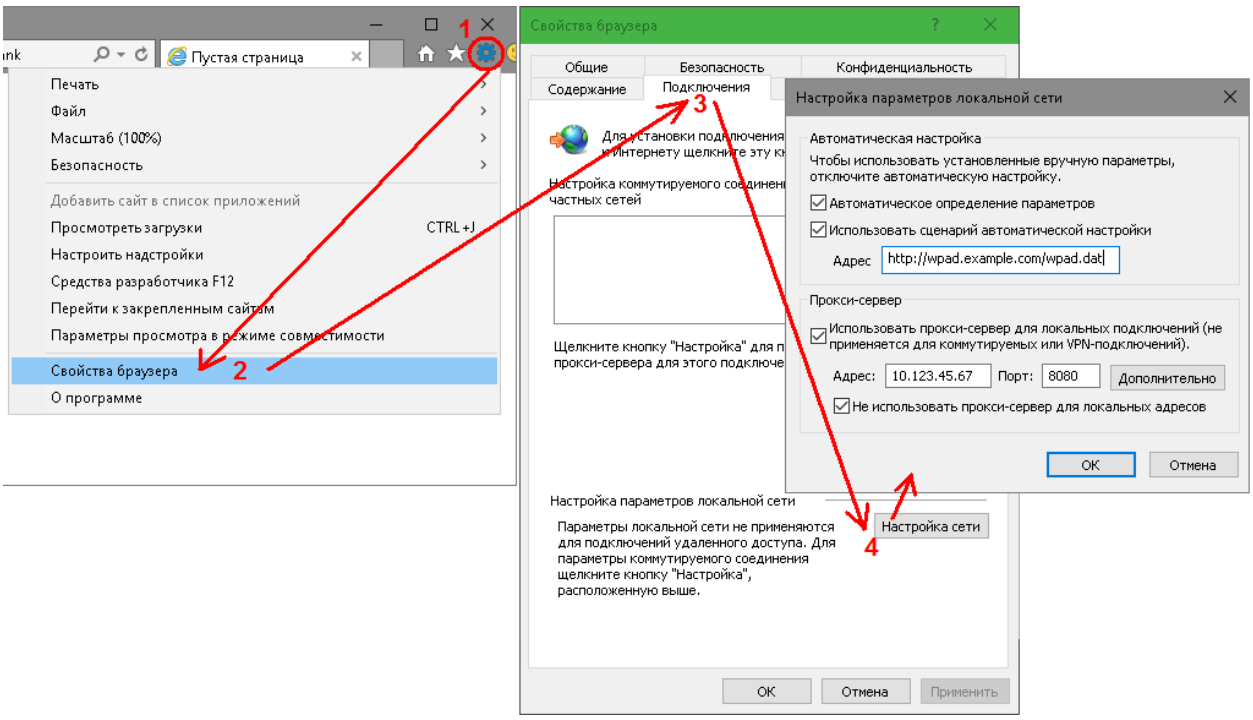

Рис. 2.7

В разделе «автоматическая настройка» можно настроить автоматическое определение прокси из конфигурационного файла. Если выбрана опция «автоматическое определение параметров», то будет выполнена попытка получения конфигурационного файла по протоколу WPAD (Web Proxy Auto-Discovery Protocol). При неудаче, если настроена опция «Использовать сце-

нарий автоматической настройки», будет выполнена попытка получения конфигурационного файла по указанному адресу.

В разделе «Прокси-сервер» прокси может быть настроен вручную. Следует указать, во-первых, IP-адрес или DNS-имя прокси, во-вторых, порт прокси. Если включена опция «Не использовать прокси сервер для локальных адресов», то к локальным адресам компьютер будет обращаться напрямую (не через прокси). Для надежности все адреса, к которым обращаться надо напрямую, лучше ввести в список исключений, который открывается при нажатии на кнопку «Дополнительно». Например, чтобы прокси не использовался для любых адресов, начинающихся на «10.» или «192.168.», и для сайта yandex.ru, в это поле нужно ввести строку «yandex.ru;10.*;192.168.*» (без кавычек). В том же окне можно настроить разные прокси для разных протоколов.

Если прокси настроен в ОС, то это не гарантирует, что все программы будут его использовать. Например, браузер Mozilla Firefox, в зависимости от своих настроек, может использовать системный прокси, другой прокси, прокси из указанного конфигурационного файла, выполнять автоматическую настройку прокси по протоколу WPAD или не использовать прокси.

2.5. Получение параметров и устранение неполадок сети

Узнать параметры подключения (IP-адрес, шлюз по умолчанию и др.) можно, выполнив команду `ipconfig` в консоли, либо `ipconfig /all` для вывода более подробной информации.

Проверить доступность узла по сети можно с помощью команды `ping [необязательные параметры] (<доменное имя> | <IP-адрес>)`. Однако следует иметь в виду, что ответ может не приходить от доступного узла по некоторым причинам, например, из-за запрета брандмауэра или из-за прокси-сервера, не пропускающего ICMP-пакеты.

Если выполнить в консоли команду `tracert [необязательные параметры] (<доменное имя> | <IP-адрес>)`, можно увидеть цепочку узлов, по которой проходит пакет, отправляемый по заданному адресу. Таким образом можно узнавать о маршрутах трафика, действующих в сети, и выяснять, на каком узле связь оказалась нарушена.

Некоторые причины проблем с сетью:

● не подключен патч-корд, неисправность патч-корда, слишком длинный сегмент сети, аппаратная неисправность сетевого адаптера;

● выключен сетевой адаптер или не установлен его драйвер;

● не функционирует один из узлов, через которые осуществляется доступ к сети;

● для беспроводного соединения: недостаточная мощность сигнала точки доступа, неверный пароль или сертификат, неверный канал;

● неверные настройки IP-адреса (в частности, использование уже занятого IP-адреса), маски подсети, основного шлюза (возможно из-за ошибок ручной настройки или недоступности DHCP-сервера);

● неверный адрес DNS-сервера, недоступность DNS-сервера (в таком случае соединение с узлом возможно при указании IP-адреса вместо доменного имени);

● соединение блокируется брандмауэром (в таком случае, вероятно, окажутся заблокированы только некоторые сетевые функции);

● не указан или неправильно указан прокси-сервер или параметры подключения к нему (в таком случае могут быть доступны узлы локальной сети, но не Интернета, или наоборот (если к узлам локальной сети необходимо обращаться напрямую, а не через прокси));

● прокси-сервер не пропускает пакеты некоторых типов (в таком случае некоторые сетевые функции не работают);

● не настроено перенаправление портов на маршрутизаторе, выполняющем трансляцию адресов NAT (в режиме PAT) (это не мешает доступу в Интернет, но не устанавливаются входящие соединения с вашим компьютером, так как он не виден извне локальной сети);

● запрошенный узел доступен только в IPv6-сети, а компьютер имеет выход только в IPv4-сеть;

● проблемы доступа по HTTPS-протоколу: устаревшие списки сертификатов в ОС; устаревшая ОС (устаревшие ОС не поддерживают современные протоколы установления шифрованных соединений, в результате некоторые сайты оказываются недоступны); проблемы совместимости некоторых сайтов со средствами антивирусной проверки шифрованных соединений; неправильное системное время;

● проблемы совместимости некоторых сайтов с некоторыми браузерными расширениями;

● неправильные настройки таблиц маршрутизации;

● неправильные настройки VLAN.

2.6. Лабораторная работа 2. Подключение компьютера с ОС MS Windows 10 к локальной сети с доступом в Интернет

Цель: *получение навыков подключения компьютера к сети и к Интернету, изменения базовых сетевых настроек, определения и исправления простых проблем подключений.*

2.6.1. Порядок выполнения работы

1. Подключить компьютеры патч-кордами к сети. Отключить и включить сетевой адаптер. Просмотреть настройки сетевого адаптера и версию его драйвера. Отключить брандмауэры на обоих компьютерах.

2. Настроить IPv4-адрес и шлюз по умолчанию на компьютерах таким образом, чтобы компьютеры были взаимно доступны по сети; проверить доступность компьютеров друг для друга с помощью команды ping.

3. Проверить настройки подключения с помощью утилиты ipconfig.

4. Включить брандмауэр на одном из компьютеров. Проверить с помощью команды ping доступность по сети компьютера со включенным брандмауэром.

5. Изменить правило брандмауэра, отвечающее за прохождение пакетов команды ping, чтобы доступный компьютер стал недоступным (или наоборот).

6. Настроить компьютер на автоматическое получение IPv4-адреса и шлюза по умолчанию от DHCP-сервера.

7. Проверить наличие доступа в Интернет на компьютере. Задать системный прокси-сервер, указанный преподавателем. Вновь проверить наличие доступа в Интернет.

8. Воспользовавшись интернет-сервисом определения IP, узнать, каким ваш IP видят узлы в Интернете, с которыми вы устанавливаете соединения.

9. Узнать IP-адрес какого-нибудь сайта в Интернете, воспользовавшись командой ping или nslookup.

10. Воспользовавшись командой tracert, попытаться узнать цепочку узлов, которые проходит пакет при обращении к выбранному в предыдущем пункте сайту.

В отчет необходимо включить описание выполненных действий, подтверждающие их выполнение скриншоты, результаты выполнения консольных команд.

2.6.2. Контрольные вопросы

1. Перечислите вещи, которые нужно проверить в первую очередь, если требуется исправить следующую проблему: а) на компьютере, ранее подключенном к Интернету, неожиданно пропал доступ в Интернет; б) при подключении нового компьютера к сети на нем не появляется доступ в Интернет; в) есть доступ к локальной сети, но нет доступа в Интернет.

2. Что такое профиль сети? Чем различаются профили?

3. Что такое брандмауэр? Какие в нем бывают правила?

4. Что такое прокси-сервер? Какие протоколы используются при работе прокси-сервера?

5. Что такое DHCP-сервер и DNS-сервер? Что будет, если они будут недоступны?

3. МАРШРУТИЗАЦИЯ

Маршрутизация – важнейший процесс в IP-сетях. Чтобы некоторая машина могла найти в сети другую, должен иметься механизм описания того, как пакеты должны передаваться от одной машины к другой. Такой механизм и называется маршрутизацией. Более строгое определение звучит так: *Маршрутизация* – процесс определения маршрута следования пакетов данных в компьютерных сетях. Маршрутизация выполняется специальными программными или аппаратными средствами – маршрутизаторами.

Маршрутизатор – сетевое устройство, используемое в компьютерных сетях передачи данных, которое на основании информации о топологии сети (таблицы маршрутизации) и определенных правил принимает решения о пересылке пакетов сетевого уровня их получателю. Маршрутизация, осуществляемая IP, – это процесс поиска в таблице маршрутизации, определение интерфейса, куда будет послан пакет. Существует два типа маршрутизации: статическая и динамическая.

Статическая маршрутизация осуществляется на основе таблиц маршрутизации, задаваемых администратором. Динамическая маршрутизация осуществляется с помощью протоколов маршрутизации. Протокол маршрутизации – это сетевой протокол, используемый маршрутизаторами для определения возможных маршрутов следования данных в составной компьютерной сети. Применение протокола маршрутизации позволяет избежать ручного ввода всех допустимых маршрутов, что, в свою очередь, снижает количество

ошибок, обеспечивает согласованность действий всех маршрутизаторов в сети и облегчает труд администраторов.

3.1. Принцип статической маршрутизации

Далее рассматривается статическая маршрутизация в IP-сетях. Частично эта тема уже затрагивалась при рассмотрении простейшей маршрутизации на основе шлюзов по умолчанию. Таблица маршрутизации содержит информацию, на основе которой маршрутизатор принимает решение о дальнейшей пересылке пакетов. Как только IP-пакет попадает на маршрутизатор (или на любой узел сети, поддерживающий сетевой уровень модели TCP/IP), выполняется последовательность действий:

1. Маршрутизатор определяет, является ли узел-получатель пакета локальным (т.е. находится ли он в той же подсети, что и маршрутизатор). Маршрутизатор имеет несколько интерфейсов и может одновременно находиться в нескольких подсетях. Если получатель находится в той же подсети, то пакет направляется ему напрямую.

2. Если узел-получатель находится в другой подсети, то просматривается таблица маршрутизации на предмет пути к этому узлу. При просмотре сеть узла-получателя сравнивается с записями о сетях в таблице маршрутизации и при обнаружении совпадения пакет направляется маршрутизатору, указанному в соответствующей записи.

3. Если маршрута в таблице маршрутизации не найдено, то пакет отправляется шлюзу по умолчанию.

4. Если запись о шлюзе по умолчанию отсутствует, то пакет уничтожается.

Для указания шлюза по умолчанию в таблице существует специальная запись *default*, которая указывает, на какой узел должен быть направлен пакет, если сеть его назначения не определена. Как правило, в локальной сети присутствует один шлюз, используемый по умолчанию.

3.1.1. Таблицы маршрутизации

Чтобы по адресу сети назначения можно было бы выбрать маршрут дальнейшей пересылки пакета, каждый узел, поддерживающий сетевой уровень, анализирует специальную информационную структуру, называемую *таблицей маршрутизации*.

Простейшая таблица маршрутизации включает в себя информацию об узле (или о сети) назначения, о маске подсети для узла (сети) назначения, об интерфейсе, через который следует направить пакет, и о шлюзе – удаленном узле, которому будет передан пакет.

Рассмотрим на примере, как может выглядеть таблица маршрутизации. Пусть задана конфигурация сети, показанная на рис. 3.1.

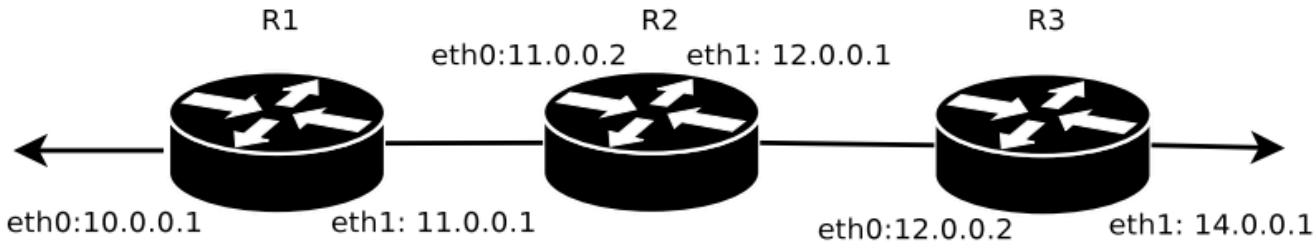

Рис. 3.1

Записи о статических маршрутах маршрутизатора R2 приведены в таблице 3.1.

Таблица 3.1

Destination	Gateway	Genmask	Iface
default	11.0.0.1	255.0.0.0	eth0
14.0.0.0	12.0.0.2	255.0.0.0	eth1

В первой колонке перечисляются номера подсетей, во второй – какому маршрутизатору следует перенаправить пакет для отправки в заданную подсеть. Третья колонка задает маску подсети назначения, в четвертой колонке указывается, через какой интерфейс следует направить пакет.

3.2. Протоколы динамической маршрутизации

Протоколы динамической маршрутизации делятся на протоколы *внутридоменной маршрутизации – IGP* (Interior Gateway Protocol) – и протоколы *междоменной маршрутизации – EGP* (Exterior Gateway Protocol). IGP-протоколы используются для передачи информации о маршрутах в пределах автономной системы (домена маршрутизации); пример автономной системы – сеть одной компании. EGP-протоколы используются для соединения автономных систем между собой.

IGP-протоколы подразделяются на *дистанционно-векторные протоколы* (RIP, EIGRP) и *протоколы состояния каналов связи* (OSPF, IS-IS). Первые основаны на алгоритме DVA (distance vector algorithm), вторые – на ал-

горитме LSA (link state algorithm). Основные различия между этими двумя видами следующие:

1) Тип информации, которой обмениваются роутеры: таблицы маршрутизации в DVA-протоколах и таблицы топологии в LSA-протоколах.

2) Процесс выбора лучшего маршрута.

3) Количество информации о сети, хранящееся в каждом роутере: при DVA роутер знает только своих соседей, при LSA – имеет представление обо всей сети.

Из EGP-протоколов в настоящее время активно используется один – BGP. Предназначенный для оперирования большими объемами данных, он обеспечивает связность всего Интернета, но может применяться и внутри домена.

Протоколы динамической маршрутизации не просто автоматизируют настройку маршрутов в сети, но позволяют выполнять быстрое автоматическое их перестроение при возникновении проблем на участках сети. Еще одна полезная функция – балансировка трафика.

Подробнее об алгоритмах динамической маршрутизации можно прочитать в [10], о протоколах RIP, OSPF и BGP – в [16].

3.3. Лабораторная работа 3. Настройка таблиц маршрутизации

Цель: *изучение методов статической маршрутизации в IP-сетях; овладение управлением таблицами маршрутизации на узлах сетевого уровня.*

3.3.1. Порядок выполнения работы

1. Для всех узлов сети установить IP-адреса, маски подсетей и шлюзы по умолчанию, чтобы добиться успешного выполнения Echo-запроса ближайших соседей (находящихся в одной подсети).

2. Настроить таблицы маршрутизации на маршрутизаторах, чтобы добиться доставки пакетов от узла K1 к узлу K2 и обратно, от узла K2 к K3 и обратно, от узла K3 к K1 и обратно. Пакеты должны доходить до узлов кратчайшим путем.

3. Настроить таблицы маршрутизации на узлах K1, K2 и K3, чтобы обеспечить кратчайшую доставку пакетов между этими узлами, если это невозможно было обеспечить в п. 2.

В отчете привести конфигурацию TCP/IP для каждого из узлов, таблицы маршрутизации, результаты Echo-запросов между узлами K1, K2 и K3, а также обоснование правильности и оптимальности выбранных маршрутов.

3.3.2. Варианты заданий

Вариант 1. Файл со схемой сети: lab2_var1.jfst. Сеть между маршрутизаторами R1, R2 и R3: 192.168.3.0. Сеть между маршрутизаторами R3 и R4: 192.168.4.0. Сеть между маршрутизаторами R5 и R6: 192.168.5.0. Компьютер PC1 имеет IP-адрес 192.168.0.100. Компьютер PC3 имеет IP-адрес 192.168.1.100. Компьютер PC4 имеет IP-адрес: 192.168.2.100. Обозначения в задании: K1 – PC1, K2 – PC3, K3 – PC4.

Вариант 2. Файл со схемой сети: lab2_var2.jfst. Сеть между маршрутизаторами R1, R2 и R3: 172.168.3.0. Сеть между маршрутизаторами R5 и R6: 172.168.4.0. Компьютер PC1 имеет IP- адрес 172.168.0.100. Компьютер PC3 имеет IP-адрес 172.168.1.100. Компьютер PC4 имеет IP-адрес: 172.168.2.100. Обозначения в задании: K1 – PC1, K2 – PC3, K3 – PC4.

Вариант 3. Файл со схемой сети: lab2_var3.jfst. Сеть между маршрутизаторами R1, R2, R3 и R4: 192.168.0.96. Сеть между маршрутизаторами R4 и R5: 172.168.4.0. Маршрутизатор R6 имеет адрес 10.120.0.1 на первом интерфейсе и 10.159.0.1 на втором интерфейсе. Сеть между маршрутизаторами R3 и R8: 11.0.0.0. Компьютер PC1 имеет IP-адрес 192.168.0.4. Компьютер PC3 имеет IP-адрес 192.168.0.34. Компьютер PC4 имеет IP-адрес: 192.168.0.250. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 4. Файл со схемой сети: lab2_var4.jfst. Сеть между маршрутизаторами R1, R2, R3 и R4: 199.0.5.96. Сеть между маршрутизаторами R4 и R5: 172.168.4.0. Маршрутизатор R6 имеет адрес 11.120.0.1 на первом интерфейсе и 11.159.0.1 на втором интерфейсе. Сеть между маршрутизаторами R3 и R8: 12.0.0.0. Компьютер PC1 имеет IP-адрес 199.0.5.2. Компьютер PC3 имеет IP-адрес 199.0.5.52. Компьютер PC4 имеет IP-адрес: 199.0.5.250. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 5. Файл со схемой сети: lab2_var5.jfst. Сеть между узлами PC3 и R3, R4, R6: 204.188.45.128. Сеть между маршрутизаторами R1, R2, R3: 204.188.45.192. Компьютер PC1 имеет IP-адрес 204.188.45.1. Компьютер PC2 имеет IP-адрес 204.188.45.65. Компьютер PC3 имеет IP-адрес 204.188.45.129. Длина маски подсети должна быть минимально возможной. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 6. Файл со схемой сети: lab2_var6.jfst. Сеть между узлами R1, R2, R3: 192.115.120.0. Сеть между узлами R2, R3, R4: 192.115.112.0. Сеть между узлами R4, R5, R7: 192.115.108.0. Сеть между узлами R6 и R7: 192.115.96.0. Компьютер PC1 имеет IP-адрес 192.115.128.1. Компьютер PC2 имеет IP-адрес 192.115.100.1. Компьютер PC3 имеет IP-адрес 192.115.88.2. Длина маски подсети должна быть минимально возможной. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 7. Файл со схемой сети: lab2_var7.jfst. Сеть между узлами PC3 и R3, R4, R6: 204.188.45.128. Сеть между маршрутизаторами R1, R2, R3: 204.188.45.192. Компьютер PC1 имеет IP-адрес 204.188.45.1. Компьютер PC2 имеет IP-адрес 204.188.45.65. Компьютер PC3 имеет IP-адрес 204.188.45.129. Компьютер BOSS имеет IP-адрес 204.188.45.196. Обозначения в задании: K1 – PC1, K2 – BOSS, K3 – PC3.

Вариант 8. Файл со схемой сети: lab2_var8.jfst. Сеть между узлами R1, R2, R3: 192.115.120.0. Сеть между узлами R4 и R7: 192.115.108.0. Сеть между узлами R6 и R7: 192.115.96.0. Компьютер PC1 имеет IP-адрес 192.115.128.1. Компьютер PC2 имеет IP-адрес 192.115.112.4. Компьютер PC3 имеет IP-адрес 192.115.88.2. Длина маски подсети должна быть минимально возможной. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 9. Файл со схемой сети: lab2_var9.jfst. Сеть между узлами R1, R2, R3: 10.0.120.0. Сеть между узлами R3 и R4: 192.168.0.0. Сеть между узлами R4 и R5: 192.168.1.0. Компьютер PC1 имеет IP-адрес 10.0.0.3. Компьютер PC2 имеет IP-адрес 10.0.0.10. Компьютер PC3 имеет IP- адрес 10.0.0.18. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 10. Файл со схемой сети: lab2_var10.jfst. Сеть между узлами R3 и R4: 192.168.0.0. Сеть между узлами R4 и R5: 192.168.1.0. Компьютер PC1 имеет IP-адрес 10.0.0.5. Компьютер PC2 имеет IP-адрес 10.0.0.130. Компьютер PC3 имеет IP-адрес 10.0.0.194. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 11. Файл со схемой сети: lab2_var11.jfst. Все маршрутизаторы и компьютеры имеют адреса из диапазона 192.168.0.1 – 192.168.0.254. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3. Вариант 12. Файл со схемой сети: lab2_var12.jfst. Все маршрутизаторы и компьютеры имеют адреса из диапазона 200.0.1.1 – 200.0.254.254. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 13. Файл со схемой сети: lab2_var13.jfst. Все маршрутизаторы и компьютеры имеют адреса из диапазона 172.1.1.1 – 172.254.254.254. Обозначения в задании: K1 – PC1, K2 – PC2, K3 – PC3.

Вариант 14. Файл со схемой сети: lab2_var14.jfst. Все маршрутизаторы и компьютеры имеют адреса из диапазона 172.0.10.1 – 172.0.88.254. Обозначения в задании: K1 – PC1, K2 – R5, K3 – PC3.

3.3.3. Пример выполнения лабораторной работы

Пусть дана конфигурация сети, показанная на рис. 3.2.

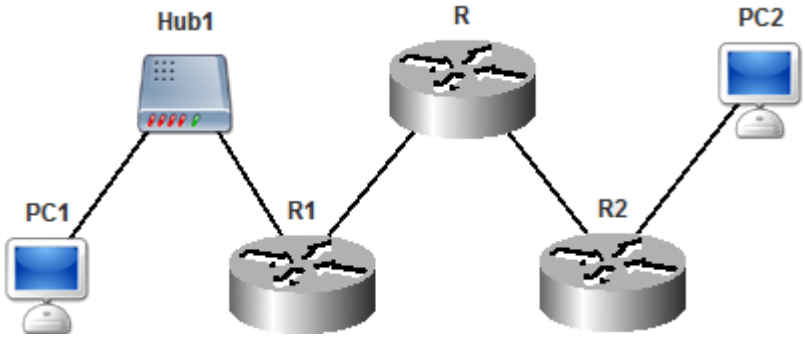

Рис. 3.2

Файл со схемой сети: lab2_sample.jfst. Компьютер PC1 имеет IP-адрес 172.168.0.2. Компьютер PC2 имеет IP адрес 10.0.0.2. Сеть между маршрутизаторами R1 и R: 172.168.100.0. Сеть между маршрутизаторами R2 и R: 192.168.0.0.

Задание

1. Задать маски подсети и шлюзы по умолчанию для PC1 и PC2, а также адреса IP из указанного диапазона вместе с масками и шлюзами для R1, R и R2 так, чтобы обеспечить корректную доставку пакетов от PC1 к PC2.

2. Настроить таблицу маршрутизации на R так, чтобы обеспечить корректную доставку пакетов от PC2 к PC1

3. Выполнить Echo-запрос с PC1 на PC2. Посмотреть вывод программы.

4. Сделать выводы относительно таблиц маршрутизации.

Выполнение работы

1. Установим для всех узлов IP-адреса, маски подсетей и шлюзы по умолчанию так, как показано в таблице 3.2.

Таблица 3.2

Узел	Интерфейс	IP-адрес	Маска подсети	Шлюз по умолчанию
PC1	eth0	172.168.0.2	255.255.255.0	172.168.0.1
PC2	eth0	10.0.0.2	255.0.0.0	10.0.0.1

Окончание табл. 3.2

Узел	Интерфейс	IP-адрес	Маска подсети	Шлюз по умолчанию
R1	eth0	172.168.0.1	255.255.255.0	172.168.100.1
R1	eth1	172.168.100.2	255.255.255.0	172.168.100.1
R	eth0	172.168.100.1	255.255.255.0	192.168.0.1
R	eth1	192.168.0.2	255.255.255.0	192.168.0.1
R2	eth0	192.168.0.1	255.255.255.0	192.168.0.2
R2	eth1	10.0.0.1	255.0.0.0	192.168.0.2

Такая конфигурация обеспечит доставку пакетов от PC1 к PC2, но не обратно. Это объясняется тем, что при попытке отправки пакета с PC2 на PC1 пакет сначала попадет на R2 (поскольку он является шлюзом для PC2), а оттуда – на R. Но с маршрутизатора R он снова будет отправлен на R2, поскольку для R шлюзом является R2. В результате возникнет «петля». Устранить ее можно с помощью настройки таблицы маршрутизации на узле R.

2. Настроим таблицу маршрутизации на R. Она будет выглядеть следующим образом:

```
R# show ip route
Destination    Gateway         Genmask         Type     Iface
default        192.168.0.1     255.255.255.0     0      eth0
```

Для того чтобы пакеты от PC2 прошли через R на R1, необходимо добавить маршрут для сети 172.168.0.0/255.255.255:

```
R# conf t
R(config)# ip route 172.168.0.0 255.255.255.0 172.168.100.2
eth0
R(config)# write mem
```

Теперь любой пакет, попавший на R и имеющий в качестве подсети назначения 172.168.0.0/255.255.255.0, будет направлен на маршрутизатор R1. Чтобы проверить корректность добавления маршрута, необходимо вывести таблицу маршрутизации:

```
R# show ip route
Destination    Gateway         Genmask         Type     Iface
default        192.168.0.1     255.255.255.0     0      eth0
172.168.0.0    172.168.100.2   255.255.255.0     0      eth0
```

3. С PC1 посылаем Echo-запрос PC2:

```
PC1 Created Echo Request packet to 10.0.0.2
PC1 Sending packet from ProtocolStack (to 172.168.0.1).
...
PC2 ProtocolStack received packet from local Interface.
PC2 Confirmed Packet is for this Network Layer Device.
PC2 Created Echo Reply packet to 172.168.0.2
```

```
PC2 Sending packet from ProtocolStack (to 10.0.0.1).
...
PC1 ProtocolStack received packet from local Interface.
PC1 Confirmed Packet is for this Network Layer Device.
PC1 Echo reply packet received from 10.0.0.2
```

4. Как видно из вывода программы, на Echo-запрос пришел ответ, следовательно, таблица маршрутизации настроена верно. Аналогичным образом можно настроить таблицы на остальных устройствах.

3.3.4. Контрольные вопросы

1. Что такое маршрутизация?
2. Для чего предназначен маршрутизатор?
3. Перечислите типы маршрутизации.
4. Что такое таблицы маршрутизации и для чего они нужны?
5. Какие типы записей могут быть в таблице маршрутизации?
6. Объясните механизм статической маршрутизации.
7. Какие есть виды протоколов динамической маршрутизации?

4. ПРИМЕНЕНИЕ МЕЖСЕТЕВЫХ ЭКРАНОВ В ОС LINUX

4.1. Возможности межсетевого экрана ОС Linux

В состав ядра ОС Linux входит подсистема фильтрации пакетов. Данная подсистема в своей работе использует универсальную систему идентификации пакетов iptables. В данной работе рассматривается использование одноименной административной утилиты для настройки межсетевого экрана на базе ОС Linux.

При получении каждого пакета сетевая подсистема сравнивает данные о нем с информацией в таблицах ядра. На основании этого сравнения сетевая подсистема принимает решение о действии, которое необходимо выполнить над данным пакетом.

Есть 4 таблицы: raw, mangle, nat и filter. Каждая таблица обработки пакетов содержит несколько предопределенных цепочек, а также может содержать цепочки, определенные пользователем. Всего есть 5 предопределенных цепочек:

- PREROUTING – для всех пакетов, приходящих извне;
- INPUT – для пакетов, пришедших извне и предназначенных для локальной системы;

● FORWARD – для маршрутизируемых пакетов, т.е. пакетов, пришедших извне и предназначенных для другой системы;

● OUTPUT – для пакетов, созданных в данной системе;

● POSTROUTING – для всех исходящих пакетов.

Каждая цепочка состоит из набора правил, а каждое правило определяет действие над пакетом, которое будет выполняться в случае, если пакет соответствует условиям, заданным в правиле. Такое действие будем называть «цель». «Целью» может быть как переход к любой определенной пользователем цепочке, так и одно из предопределенных действий.

Таблица mangle содержит все пять стандартных цепочек, таблица raw – цепочки PREROUTING и OUTPUT, таблица nat – цепочки PREROUTING, OUTPUT и POSTROUTING, таблица filter – цепочки INPUT, FORWARD, и OUTPUT. Цепочки с одинаковым названием, но в разных таблицах – совершенно независимые объекты. Например, mangle PREROUTING и nat PREROUTING обычно содержат разный набор правил; пакеты сначала проходят через цепочку mangle PREROUTING, а потом через nat PREROUTING.

Обрабатываемые пакеты проходят через множество таблиц и цепочек.

Рассмотрим лишь таблицу фильтрации пакетов (filter). Другие таблицы могут обеспечивать дополнительные возможности, например, модификацию пакетов и преобразование сетевых адресов.

При получении каждого пакета сетевая подсистема выбирает первое правило из таблицы. Если пакет соответствует данному правилу, то выполняется действие («цель»), указанное в правиле, если нет – проверяется следующее в цепочке правило. Если пакет не соответствует условиям ни одного из правил, выбирается «цель» по умолчанию, которая задается для каждой цепочки.

Для каждого правила в таблице фильтрации можно задать в качестве «цели»:

● перевод пакета в пользовательскую цепочку;

● принятие пакета (ACCEPT);

● «выброс» пакета (DROP);

● отказ от получения пакета (REJECT);

● передача пакета служебной программе (QUEUE);

При принятии пакета он передается далее – либо в пользовательский процесс, либо по сети в соответствии с правилами маршрутизации. При выбросе пакета он просто теряется. Отказ от получения пакета вызывает от-

правление сообщения источнику пакета о том, что пакет не может быть принят. Возможен выбор сообщаемой причины непринятия пакета. Передача пакета служебной программе используется для какой-либо специальной обработки пакетов.

По умолчанию для каждой из предопределенных цепочек задана «цель» ACCEPT.

В качестве условий, с которыми сравниваются данные о пакете, могут выступать:

- IP-адрес отправителя;
- IP-адрес получателя;
- используемый протокол;
- сетевой интерфейс, с которого получен пакет;
- порт отправителя и порт получателя;
- другие параметры (размер пакета, физические адреса, флаги протоколов, специальные пометки и т. д.).

4.2. Настройка межсетевого экрана

Настройка межсетевого экрана осуществляется добавлением правил в таблицы ядра. Каждое добавление реализуется выполнением программы iptables с необходимыми параметрами.

Для добавления правила необходимо указать:

- таблицу и имя цепочки, в которую добавляется правило,
- условия выбора пакетов,
- «цель» правила.

Подробную информацию о синтаксисе команды iptables и о возможных параметрах ее вызова можно получить, воспользовавшись электронной документацией. Для этого достаточно в системной консоли набрать команду man iptables. Далее приведены примеры наиболее часто употребляющихся правил фильтрации.

Пример 1:

```
iptables -t filter -A INPUT -s 192.168.1.100 -j DROP
```

Данное правило будет соответствовать всем пакетам, получаемым с адреса 192.168.1.100. Цель правила – выброс, т. е. все пакеты, получаемые с данного адреса, будут игнорироваться так же, как если бы компьютера не было в сети.

Пример 2:

```
iptables -t filter -A INPUT -s 192.168.1.100 -p udp
-dport 139 -j REJECT
```

Правило будет соответствовать всем пакетам протокола UDP, получаемым с адреса 192.168.1.100 для порта 139. Цель правила – отказ. На все пакеты, получаемые с данного адреса, будет отправлен ответ о недоступности данного порта.

Пример 3:

```
iptables -t filter -P <цепочка> <действие>
```

Установка правила по умолчанию для выбранной цепочки.

Пример 4:

```
iptables -t filter -P INPUT DROP
```

Данная команда установит для цепочки INPUT действие по умолчанию «выброс». Таким образом, все пакеты, не попадающие ни под одно разрешающее правило, будут игнорироваться.

Удаление фильтрующих правил. Для того чтобы удалить какое-либо правило, выполняется команда

```
iptables -t filter -D INPUT <номер_правила>
```

Для того чтобы узнать номер правила, можно использовать дополнительный ключ, с которым команда вывода таблиц на экран также печатает для каждого правила его номер. (Значение ключа студенту предлагается выяснить из документации самостоятельно).

Замечание. Вместо номера правила можно также указать весь список условий и цель правила, которое необходимо удалить.

Существует также возможность удаления из цепочки всех правил. Данная возможность реализуется ключом `-F`. Если при использовании этого ключа не указать имя цепочки, то будут удалены все правила из всех цепочек заданной таблицы.

Просмотр текущих настроек фильтра. В любой момент времени можно посмотреть текущее состояние таблиц ядра. Для этого используется команда `iptables -L`. Результатом ее работы будет вывод на экран таблицы. Пример такой таблицы:

```
Chain INPUT (policy ACCEPT)
target  prot opt source           destination
DROP    all  --  10.0.0.0/8        0.0.0.0/0
DROP    all  --  172.16.0.0/12     0.0.0.0/0
DROP    all  --  192.168.0.0/16    0.0.0.0/0
DROP    all  --  127.0.0.0/8       0.0.0.0/0
```

```
DROP    all  --  169.254.0.0/16    0.0.0.0/0
DROP    all  --  224.0.0.0/4       0.0.0.0/0
DROP    all  --  240.0.0.0/5       0.0.0.0/0
DROP    all  --  255.255.255.255   0.0.0.0/0
DROP    all  --  217.174.99.34     0.0.0.0/0
DROP    tcp  -- !127.0.0.0/16      0.0.0.0/0   tcp dpt:3306

Chain FORWARD (policy ACCEPT)
target  prot opt source           destination

Chain OUTPUT (policy ACCEPT)
target  prot opt source           destination
```

В данном примере на входе блокируются некоторые нежелательные источники пакетов, а также пакеты, которые могли попасть по ошибке. Так как цель по умолчанию – ACCEPT, то все пакеты, которые не будут соответствовать ни одному из правил, будут приниматься.

4.3. Настройка трансляции сетевых адресов

Маскарадинг – это тип трансляции сетевых пакетов с подменой IP-адреса на IP-адрес узла, через который данный пакет проходит. Позволяет машинам, не имеющим реальных (внешних) IP-адресов, работать в сети Интернет.

Пусть локальная сеть имеет адрес 192.168.0.0/24. Включается маскарадинг в iptables следующим образом:

1) Разрешить прохождение пакетов между сетевыми интерфейсами из локальной сети:

```
iptables -A FORWARD -s 192.168.0.0/24 -j ACCEPT
```

2) Разрешить прохождение пакетов между сетевыми интерфейсами в локальную сеть:

```
iptables -A FORWARD -d 192.168.0.0/24 -j ACCEPT
```

3) Включить маскарадинг для локальной сети 192.168.0.0/24:

```
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j MASQUERADE
```

SNAT (source NAT) – это тип трансляции сетевых пакетов с подменой IP-адреса на указанный в правиле адрес.

Для его включения первые два шага такие же, как при включении маскарадинга, а на третьем шаге необходимо включить NAT:

```
iptables -t nat -A POSTROUTING -o eth0 -s 192.168.0.0/24 -j SNAT --to-source 111.111.111.111
```

После этой команды включается трансляция адресов сети 192.168.0.0/24 на адрес 111.111.111.111.

4.4. Лабораторная работа 4. Использование межсетевого экрана ОС Linux

Цель: *научиться создавать, удалять и изменять правила межсетевого экрана iptables (настройка блокировки трафика, разрешения принятия трафика, логгирования приходящих пакетов).*

Имеются три виртуальные машины – Ub1, Ub3, UbR. Каждая соединена с двумя другими.

4.4.1. Порядок выполнения работы

0. На всех машинах запустить скрипт toscrath.sh. Проверить, что таблицы ядра пусты.

1. Заблокировать доступ по IP-адресу Ub1 к Ub3.

2. Заблокировать доступ по 21 порту на Ub1.

3. Заблокировать доступ к порту 80 на Ub3 от UbR. Проверить возможность доступа с Ub1.

4. Полностью запретить доступ к Ub3. Разрешить доступ к порту 21.

5. С помощью правила по умолчанию обеспечить блокировку всех входящих и исходящих пакетов узла Ub3, исключая пакеты управления сетью (протокол ICMP). Убедиться, что Ub3 принимает и отвечает на запросы команды ping, но не отвечает на запросы протокола TCP.

6. Запретить подключение к Ub1 по порту 80. Настроить логгирование попыток подключения по 80 порту.

7. Заблокировать доступ по 20 порту к Ub3 с Ub1 по его MAC-адресу.

8. Полностью закрыть доступ к Ub1. Разрешить доступ для Ub3 к Ub1, используя диапазон портов 20-79.

9. Разрешить только одно ssh подключение к UbR.

Каждое задание в отчете оформить отдельным подразделом, указав выполненные команды (их вывод, если есть), представить результаты проверки успешности выполнения задач. После выполнения каждого задания выполнять скрипт delAllTables.sh.

4.4.2. Контрольные вопросы

1. В чем различие между целями DROP и REJECT?

2. Какая очередность применения правил iptables?

3. Какие правила необходимо добавить в таблицу фильтрации для разрешения установления TCP-соединения?

4. Дайте описание цепочек PREROUTING, INPUT, FORWARD, OUTPUT, POSTROUTING. Какие из них есть в таблице фильтрации пакетов?

5. Что делает маскарадинг? Что такое NAT? Каковы отличия между ними?

6. Есть ли способ выполнить п. 5 (см. выше порядок выполнения работы) без использования правил по умолчанию? Если есть, то какой?

5. ВИРТУАЛЬНЫЕ ЛОКАЛЬНЫЕ СЕТИ

VLAN (от англ. Virtual Local Area Network) – группа устройств, имеющих возможность взаимодействовать между собой напрямую на канальном уровне, хотя физически при этом они могут быть подключены к разным сетевым коммутаторам. И наоборот, устройства, находящиеся в разных VLAN'ах, невидимы друг для друга на канальном уровне, даже если они подключены к одному коммутатору, и связь между этими устройствами возможна только на сетевом и более высоких уровнях. Пакет, отправленный из одного VLAN’а в другой VLAN, будет заблокирован на канальном уровне, но может достигнуть цели, если на сетевом уровне организована маршрутизация между VLAN’ами. Таким образом, единый логический сегмент сети разбивается на несколько виртуальных логических сегментов, изолированных друг от друга, – VLAN’ов. Разграничиваются устройства, физически находящиеся в одном сегменте сети; логическая топология сети оказывается не зависящей от ее физической топологии. Это позволяет:

- повысить безопасность в сети (взаимодействие между VLAN можно контролировать на устройствах 3-го уровня); в частности, VLAN используются как средство борьбы с ARP-spoofing'ом;
- повысить управляемость сети (политики можно применять к целым подсетям, а не к отдельному устройству);
- увеличить производительность сети за счет сокращения широковещательного трафика (он будет ограничен одной VLAN).

5.1. Классификация VLAN

Трафик, относящийся к некоторой VLAN, может быть отправлен только по путям, включенным в эту VLAN. В зависимости от способа определения того, к какой VLAN принадлежит трафик, VLAN можно разделить на 2 основных типа:

1) VLAN со статическим назначением портов (port-based VLAN).

2) VLAN с динамическим назначением портов.

При статическом назначении портов принадлежность трафика к той или иной VLAN определяется по физическому порту, через который был получен этот трафик: он считается принадлежащим той VLAN, номер которой указан в настройках порта. Порты коммутатора сопоставляются с VLAN'ами, принадлежность порта VLAN'у задается администратором в процессе настройки коммутатора. Для обеспечения безопасности недоверенным пользователям следует давать доступ только к портам со статическим назначением: напрямую через такой порт будет доступна только та VLAN, которая задана в настройках порта, что бы пользователи ни отправляли в этот порт.

При статическом назначении портов содержимое трафика не используется для определения принадлежности трафика к VLAN, поэтому устройство может принимать через такой порт трафик лишь одной VLAN (весь пришедший трафик считается принадлежащим к той VLAN, к которой приписан порт). Если нужно между устройствами передавать трафик нескольких VLAN, то в случае использования статического назначения потребуется выделять отдельный физический канал для каждой VLAN (рис. 5.1).

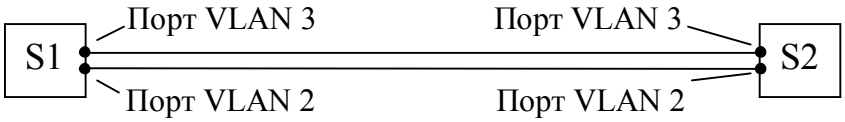

Рис. 5.1. Между коммутаторами S1 и S2, использующими статическое назначение портов, может передаваться трафик VLAN 2 и VLAN 3

Чтобы передавать трафик нескольких VLAN через один кабель, нужно иметь возможность отделять трафик одной VLAN от трафика других VLAN. При динамическом назначении портов содержимое трафика влияет на определение принадлежности трафика к той или иной VLAN, так что через порт можно передавать трафик нескольких VLAN, не меняя настроек коммутатора.

Преимущество динамического назначения в возможности пользователя менять свое местоположение, подключаясь через разные порты к одной VLAN, при этом не требуется заново настраивать конфигурацию VLAN.

В зависимости от того, как содержимое трафика определяет его принадлежность к VLAN, можно выделить следующие типы VLAN с динамическим назначением портов:

● Tag-based VLAN: информация о принадлежности к VLAN передается в явном виде: в каждый фрейм трафика добавлен тег с номером VLAN (фреймы без тега могут считаться относящимися к определенной VLAN, называемой Native VLAN). Формат и место вставки тега определяются стандартом 802.1Q, хотя существуют и другие протоколы.

● MAC-based VLAN: членство в VLAN'е основывается на MAC-адресе подключенной рабочей станции. Коммутатор имеет таблицу MAC-адресов всех устройств вместе с VLAN'ами, к которым они принадлежат. Может использоваться сервер политики членства в VLAN (VMPS – VLAN Membership Policy Server).

● Authentication-based VLAN: устройства могут быть автоматически перемещены в VLAN основываясь на данных аутентификации пользователя или устройства (с использованием протокола 802.1x). По результатам аутентификации на RADIUS-сервере порт коммутатора размещается в той или иной VLAN.

● Protocol-based VLAN: данные 3-4 уровня в заголовке пакета используются для определения членства в VLAN'е. Например, использующие протокол IP машины помещаются в одну VLAN, а использующие протокол IPX – в другую.

● Subnet-based VLAN: членство в VLAN'е основывается на IP-адресе и маске подсети подключенной рабочей станции.

Последние два метода динамического определения членства нарушают независимость уровней: например, переход с IPv4 на IPv6 приведет к нарушению работоспособности сети.

Коммутаторы различаются по возможностям поддержки различных типов VLAN (например, некоторые коммутаторы поддерживают только статическое назначение портов).

Могут использоваться сразу несколько способов определения членства в VLAN (как, например, при классическом построении VLAN на базе портов и тегов). Если коммутаторы позволяют, то администратор может настроить на

них правила распознавания принадлежности трафика к VLAN, учитывающие сразу несколько критериев. Поскольку в таких сетях фреймы постоянно просматриваются на предмет соответствия заданным критериям, принадлежность пользователей к виртуальным сетям может меняться в зависимости от текущей деятельности пользователей.

VLAN'ы могут использоваться для обеспечения безопасности в сетях, а именно, для запрещения взаимодействия между узлами, которые не должны взаимодействовать. В таком случае следует использовать port-based и tag-based VLAN'ы, так как другие типы VLAN'ов сами по себе не обеспечивают надежной защиты. Важно, кроме того, подобрать оборудование, в котором реализация функций VLAN подойдет для обеспечения надежной изоляции VLAN'ов, и правильно настроить его.

В рамках 4-ой и 5-ой лабораторных работ будут исследоваться port-based VLAN и одна из разновидностей VLAN с динамическим назначением портов – tag-based VLAN.

5.2. Port-based и tag-based VLAN

Port-based VLAN'ы позволяют разбить порты коммутатора на группы и запретить обмен трафиком между группами портов. У каждого порта есть ровно один PVID (Port VLAN ID). У каждого порта, кроме того, есть список VID – номеров VLAN'ов, в которые входит порт. В симметричных port-based VLAN'ах у порта есть единственный VID, совпадающий с PVID.

Трафик, входящий в порт коммутатора, получает метку PVID внутри коммутатора. Далее определяется порт назначения; если у порта назначения VID совпадает с меткой PVID трафика, то трафик пропускается и выходит из порта коммутатора. Метка PVID с трафика при этом снимается. Если у порта назначения нет VID, совпадающего с меткой PVID трафика, то трафик отбрасывается.

При использовании tag-based VLAN'ов трафик снабжается метками, указывающими, к какому VLAN принадлежит трафик. Можно считать, что это те же метки PVID, использующиеся внутри коммутаторов в port-based VLAN'ах, но в tag-based VLAN'ах эти метки могут оставаться на исходящем из портов трафике и могут быть на входящем в порты трафике. Благодаря меткам по одному кабелю можно передавать трафик сразу нескольких VLAN'ов, и устройство, принимающее трафик, будет понимать, к какому VLAN'у относится каждый кадр.

5.3. Теги 802.1Q

До появления общепризнанного стандарта по организации виртуальных сетей IEEE 802.1Q каждый производитель сетевого оборудования использовал собственную технологию организации VLAN. Такой подход имел существенный недостаток: технологии одного производителя были несовместимы с технологиями других фирм. Поэтому при построении виртуальных сетей на базе нескольких коммутаторов необходимо было использовать только оборудование от одного производителя.

По стандарту 802.1Q метка (тег) – это 4 байта, добавляемые в каждый кадр трафика (рис. 5.2).

Рис. 5.2. Вставка тега 802.1Q в кадр Ethernet-II

При использовании стандарта Ethernet II, по стандарту 802.1Q тег вставляется после MAC-адреса отправителя и перед полем «E-Type». Так как кадр изменился, пересчитывается контрольная сумма.

Тег состоит из следующих полей:

● Tag Protocol Identifier (TPID, идентификатор протокола тегирования). Размер поля – 16 бит. Указывает, какой протокол используется для тегирования. Для 802.1Q используется значение 0x8100.

● Priority (приоритет). Размер поля – 3 бита. Используется стандартом 802.1Q для задания приоритета передаваемого трафика.

● Canonical Format Indicator (CFI, индикатор канонического формата). Размер поля – 1 бит. Указывает на формат MAC-адреса. 0 – канонический, 1 – не канонический. CFI используется для совместимости между сетями Ethernet и Token Ring.

● VLAN Identifier (VID, идентификатор VLAN). Размер поля – 12 бит. Указывает, какому VLAN принадлежит фрейм. Диапазон возможных значений – от 0 до 4095.

5.4. Тегирующие и нетегирующие порты коммутаторов

Устройства, не поддерживающие тегированный трафик, могут быть подключены к сетям с tag-based VLAN'ами через порты коммутаторов, принимающие и отправляющие нетегированный трафик. Такие порты называются *нетегирующими* или *access-портами*. Порты коммутаторов, принимающие и отправляющие тегированный трафик, называются *тегирующими* или *trunk-портами*. Обычно access-порты используют для подключения конечных узлов, а trunk-порты – для подключения коммутаторов друг к другу.

Access-порт может принадлежать только одному VLAN'у. На пришедший трафик access-порт ставит метку PVID, а с исходящего трафика снимает метку VID=PVID.

Trunk-порт может входить в несколько VLAN'ов (т.е. может иметь несколько VID). Он не изменяет метки трафика на входе и выходе, пропускает трафик любых или некоторых VLAN'ов. Нетегированный трафик считается принадлежащим к VLAN по умолчанию – native VLAN. Native VLAN trunk-порта задается его PVID. В зависимости от настроек trunk-порт может отправлять трафик native VLAN нетегированным или тегированным.

На рис. 5.3 приведен пример сети с 3 виртуальными сетями. Конечные узлы подключены к access-портам соответствующих VLAN'ов и потому отправляют и получают нетегированный трафик. Порт 6 коммутатора 1 и порт 4 коммутатора 2 – trunk-порты, входящие во все три VLAN'а. Между этими портами передается тегированный трафик.

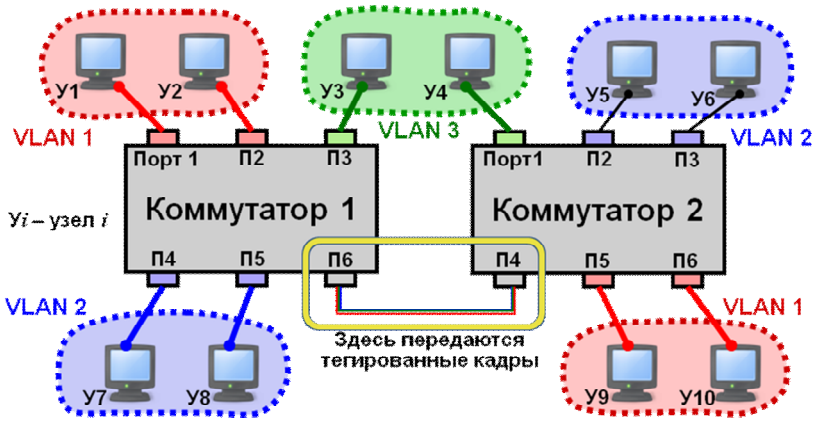

Рис. 5.3

Некоторые коммутаторы поддерживают еще hybrid-порты, способные отправлять трафик нескольких VLAN'ов нетегированным, а нескольких других – тегированным. Для каждого VID из списка VLAN'ов, в которые входит

hybrid-порт, хранится указание, следует ли тегировать исходящий трафик этого VLAN'а. Входящий нетегированный трафик помечается PVID порта.

В некоторых коммутаторах на портах может быть настроена фильтрация входящего трафика: можно разрешить только нетегированный, только тегированный трафик или оба типа, для тегированного трафика – разрешить трафик только с определенными тегами. Правила фильтрации выходящего трафика порта определены списком VID, т.е. VLAN'ами, в которые входит порт.

5.5. Маршрутизация между VLAN-сетями

Каждый VLAN – это отдельный сегмент сети, узлы в этом VLAN'е обычно входят в одну IP-подсеть. Связь между узлами из разных VLAN'ов осуществляется через маршрутизаторы, подключенные к этим VLAN'ам. Если маршрутизатор подключен к access-портам, то он может не поддерживать тегированный трафик; маршрутизатор, поддерживающий тегированный трафик, можно подключать к trunk-портам. Такой маршрутизатор на каждом физическом интерфейсе (порту) может иметь множество виртуальных интерфейсов, каждый из которых отвечает за прием и отправление трафика с определенным тегом. Например, трафик с тегом 2, физически проходящий через физический интерфейс eth0, логически будет проходить через виртуальный интерфейс eth0/2.

5.6. Лабораторная работа 5. Настройка VLAN в ОС Linux

Цель: *настройка подключения компьютера с использованием виртуальных интерфейсов и маршрутизации трафика из одного VLAN в другой.*

Имеются три виртуальные машины – Ub1, Ub3, UbR, на каждой установлен пакет vlan, необходимый для поддержки VLAN.

5.6.1 Порядок выполнения работы

0. На машинах Ub1 и Ub3 запустить скрипт toscrath.sh.

1. Настроить VLAN между ПК Ub1 и Ub3. VLAN ID, IP адреса и маски подсети использовать согласно указанным ниже вариантам. Проверить выполнение ping между ПК, объяснить результат, в случае если выдается ошибка – исправить настройки VLAN.

2. На машинах Ub1 и Ub3 запустить скрипты task2-v*.sh. Исправить ошибку в настройке сетевых адаптеров, после чего продемонстрировать успешный эхо-запрос от одного ПК к другому и обратно.

3. На трех ПК (Ub1, Ub3, UbR) запустить скрипт task3-v*.sh. Организовать подключение Ub1 к Ub3 и обратно через UbR. Настроить UbR таким образом, чтобы эхо-запрос успешно проходил с Ub1 на Ub3.

4. На трех ПК запустить скрипт task4-v*.sh. В данной задаче сеть настроена с ошибками. Необходимо исправить ошибку и показать выполнение эхо-запроса от Ub1 до Ub3.

Схема подключения для задач 3 и 4 изображена на рис. 5.4.

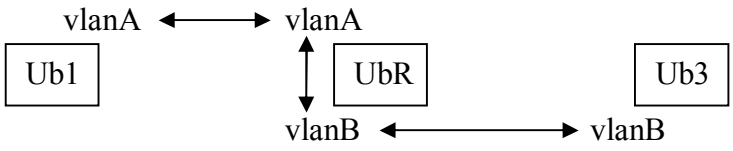

Рис. 5.4

Обосновать в отчете исправления при нахождении ошибок в схеме сети. Не забывать, что для выполнения задачи заново необходимо выполнить скрипты, предназначенные для данной задачи. В имени скрипта вместо «*» подставить номер своего варианта.

В отчет необходимо включить настройки сетевых интерфейсов (имена, VLAN, IP-адреса, маски подсети, шлюз по умолчанию) – в том числе виртуальных – до исправлений и после.

5.6.2. Варианты заданий

Вариант 1. Ub1: vlan id: 100, ip 10.0.0.1, netmask 255.255.255.0; Ub3: vlan id: 100, ip 10.0.0.2 netmask, 255.255.255.0.

Вариант 2. Ub1: vlan id: 101, ip 10.168.16.1, netmask 255.255.240.0; Ub3: vlan id: 101, ip 10.168.30.220 netmask, 255.255.240.0.

Вариант 3. Ub1: vlan id: 102, ip 1.7.0.2, netmask 255.192.0.0; Ub3: vlan id: 102, ip 1.60.60.60 netmask, 255.192.0.0.

Вариант 4. Ub1: vlan id: 103, ip 220.23.12.7, netmask 255.255.248.0; Ub3: vlan id: 103, ip 220.23.8.34 netmask, 255.255.248.0.

Вариант 5. Ub1: vlan id: 104, ip 8.0.0.7, netmask 255.255.224.0; Ub3: vlan id: 104, ip 8.0.31.222 netmask, 255.255.224.0.

Вариант 6. Ub1: vlan id: 105, ip 110.10.12.54, netmask 255.128.0.0; Ub3: vlan id: 105, ip 110.1.13.67 netmask, 255.128.0.0.

Вариант 7. Ub1: vlan id: 106, ip 18.18.18.35, netmask 255.255.255.224; Ub3: vlan id: 106, ip 18.18.18.60 netmask, 255.255.255.224.

Вариант 8. Ub1: vlan id: 107, ip 78.98.178.198, netmask 255.255.255.224; Ub3: vlan id: 107, ip 78.98.179.47, netmask 255.255.254.0.

Вариант 9. Ub1: vlan id: 108, ip 77.97.99.10, netmask 255.255.255.248; Ub3: vlan id: 108, ip 77.97.99.12 netmask, 255.255.255.248.

Вариант 10. Ub1: vlan id: 109, ip 77.97.99.10, netmask 255.255.255.248; Ub3: vlan id: 109, ip 77.97.99.12, netmask 255.255.255.248.

Вариант 11. Ub1: vlan id: 110, ip 154.137.12.8, netmask 255.255.255.224; Ub3: vlan id: 110, ip 154.137.12.27, netmask 255.255.255.224.

Вариант 12. Ub1: vlan id: 111, ip 255.255.192.0, netmask 255.255.192.0; Ub3: vlan id: 111, ip 250.250.190.12, netmask 255.255.192.0.

Вариант 13. Ub1: vlan id: 112, ip 12.13.14.15, netmask 255.255.255.128; Ub3: vlan id: 112, ip 12.13.14.120, netmask 255.255.255.128.

Вариант 14. Ub1: vlan id: 113, ip 222.12.45.76, netmask 255.255.248.0; Ub3: vlan id: 113, ip 222.12.43.12, netmask 255.255.248.0.

5.6.3. Примеры команд, необходимых для выполнения работы

1) Настройка сетевого интерфейса VLAN. В файле /etc/network/interfaces дописать следующее:

```
auto eth0.1001
iface eth0.1001 inet static
 address 1.0.0.1
 netmask 255.255.255.0
 vlan_raw_device eth0
```

Для применения настроек необходимо перезагрузить виртуальную машину. Эти действия создадут виртуальный сетевой интерфейс с VLAN ID 1001 с ip-адресом 1.0.0.1, подключенный к сетевому интерфейсу eth0.

2) настройка маршрутизации пакетов через другой ПК:

```
sudo route add default gw <ip-адрес ПК>
```

3) Проверки получения пакетов на узлах. Для этого следует запустить tcpdump. Это утилита, позволяющая перехватывать и анализировать сетевой трафик, проходящий через компьютер, на котором запущена данная программа. Для перехвата трафика с конкретного интерфейса запустите:

```
sudo tcpdump –i <имя интерфейса>
```

5.6.4. Контрольные вопросы

1. Как настроить в Linux приём и передачу тегированного трафика?

2. Каким образом могут общаться между собой два ПК, принадлежащие разным VLAN?

3. Как выполняется маршрутизация между разными VLAN?

4. Как заставить компьютер выполнять роль коммутатора, поддерживающего VLAN?

5.7. Лабораторная работа 6. Организация и соединение виртуальных сетей на базе портов и тегов

Цель: *освоение принципов построения VLAN на базе портов и тегов.*

Схема сети, используемой в лабораторной работе, приведена на рис. 5.5.

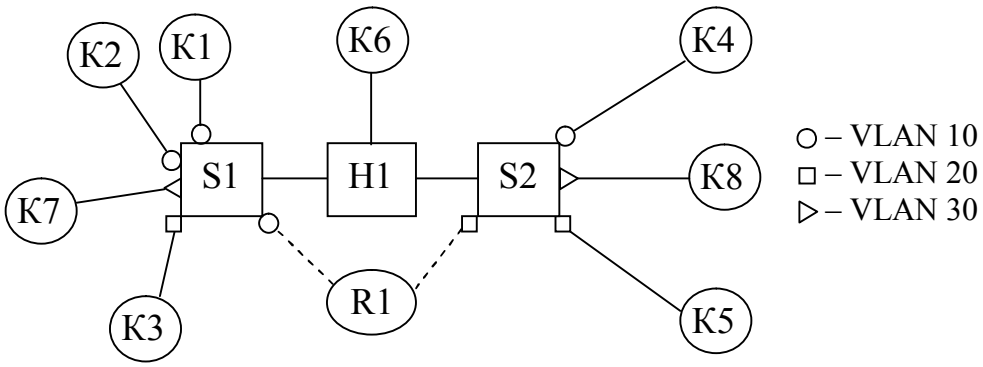

Рис. 5.5. К – компьютеры, H – концентраторы, S – коммутаторы, R – маршрутизаторы.

К1, К2, К4, К7, К8 имеют IP-адреса одной подсети, К3 и К5 – другой подсети.

5.7.1. Порядок выполнения работы

0. Собрать и настроить сеть, описанную выше.

1. Сегментирование сети с использованием VLAN.

1.1. Настроить порты коммутатора S1 таким образом, чтобы К1 и К2 были подключены к VLAN 10, К3 – к VLAN 20, а К7 – к VLAN 30.

1.2. Проверить видимость между К1 и К2, между К1 и К3, между К1 и К4. Объяснить результаты.

1.3. Настроить порты остальных коммутаторов таким образом, чтобы К4 вошел в VLAN 10, К5 – в VLAN 20, а К8 – в VLAN 30.

1.4. Проверить видимость между К1 и К4, между К1 и К5. Объяснить результаты.

1.5. Проанализировать трафик, поступающий к К6, при отправке запросов: с К1 к К4, с К1 к К5, с К1 к К2, с К1 к К3, с К3 к К4, с К3 к К5.

2. Организация маршрутизации между VLAN с использованием обычного маршрутизатора.

2.1. Включить R1. Обеспечить доступность между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 20, через шлюз R1.

2.2. Проверить видимость между компьютерами, принадлежащими разным VLAN. Объяснить результаты.

2.3. Отменить модификации сети, выполненные в п. 2.1.

3. Прямое соединение VLAN.

3.1. Соединить VLAN 10 и VLAN 20 через порты доступа.

3.2. Проверить видимость между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 20. Объяснить результаты.

3.3. Отменить модификации сети, выполненные в п. 3.1.

3.4. Соединить VLAN 10 и VLAN 30 через порты доступа.

3.5. Проверить видимость между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 30. Объяснить результаты.

3.6. Отменить модификации сети, выполненные в п. 3.4.

4. Организация маршрутизации между VLAN с использованием маршрутизатора, подключенного к транковому порту.

4.1. Обеспечить доступность между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 20, через шлюз K6.

4.2. Проверить видимость между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 20. Объяснить результаты.

4.3. Аналогично п. 4.1 попытаться обеспечить доступность между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 30, через шлюз K6. Проверить видимость между компьютерами, принадлежащими VLAN 10, и компьютерами, принадлежащими VLAN 30. Объяснить результаты.

5.7.2. Контрольные вопросы

1. Чем полезны VLAN?

2. Что такое port-based и tag-based VLAN?

3. Что такое access-, trunk- и hybrid-порты? Что такое PVID, VID, native VLAN?

4. Какую информацию содержит тег VLAN?

5.8. Уязвимости VLAN

Под уязвимостью VLAN понимается возможность получить несанкционированный доступ из одного VLAN'а к узлу другого VLAN'а. Уязвимости VLAN могут быть в сети из-за некорректной реализации функций VLAN или некорректных настроек.

Пример уязвимости VLAN, связанной с некорректной реализацией функций VLAN в коммутаторе, – пропускание из одного VLAN'а в другой кадров, содержащих MAC-адрес узла-адресата. В результате узел может получить доступ к узлу из другого VLAN'а, если имеет его MAC-адрес в ARP-кэше. Такое возможно при статическом добавлении MAC-адреса в ARP-кэш или после изменения конфигурации коммутатора, когда узел попадает в другой VLAN, но другие узлы из прежнего VLAN'а еще помнят его MAC-адрес.

Примеры уязвимостей VLAN, связанных с некорректной настройкой VLAN:

1) Атака с использованием двойного тегирования и native VLAN. Позволяет отправлять трафик в другие VLAN'ы, но не получать ответы (рис. 5.6).

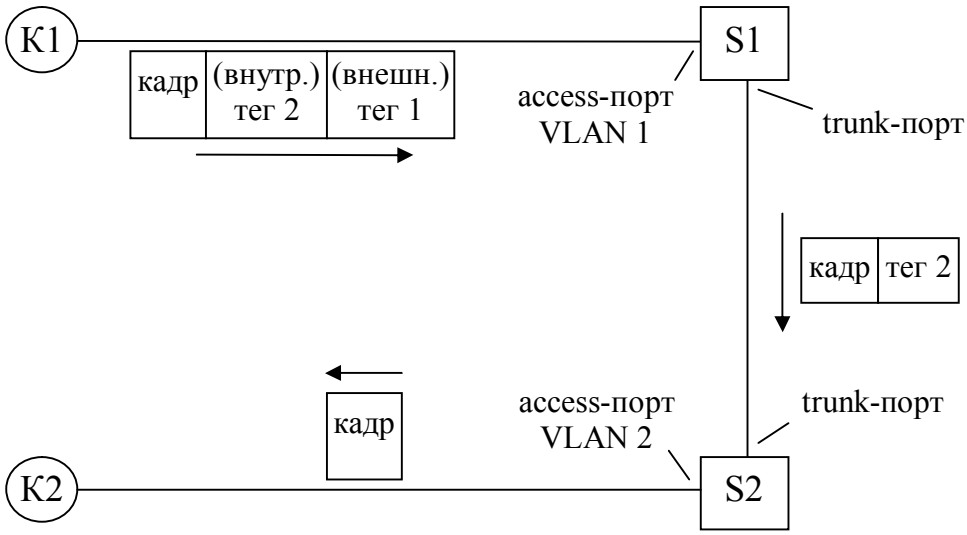

Рис. 5.6. Атака с использованием двойного тегирования и native VLAN. Trunk-порты разрешают VLAN 2 и имеют native VLAN, равный 1.

Узел К1 находится в VLAN 1, узел К2 – в VLAN 2. К1 отправляет к К2 дважды тегированный кадр: во внутреннем теге указан тег 2, во внешнем –

тег 1. Коммутатор S1 видит внешний тег 1 и принимает кадр как принадлежащий к VLAN 1. Далее кадр отправляется к S2. Так как trunk-порт коммутатора S1 имеет native VLAN, равный 1, кадр отправляется к S2 без тега 1 и в результате попадает в S2 с единственным тегом 2. S2 принимает кадр с тегом 2 как принадлежащий к VLAN 2 и отправляет кадр к К2.

Способы устранения уязвимости:

а) Передача трафика native VLAN тегированным на всех trunk-портах.

б) Настройка на всех trunk-портах в качестве native VLAN нигде неиспользуемого VLAN.

в) Фильтрация трафика на access-портах: запрет дважды тегированного трафика или запрет любого тегированного трафика.

2) Отсутствие фильтрации входящего трафика на портах. Может приводить к тому, что подключенный компьютер сможет передавать трафик в чужие VLAN'ы (снабжая пакеты соответствующими тегами), но не получать ответы.

3) Switch spoofing – имитация коммутатора с trunk-портом. Отправляя сообщения протоколов управления VLAN (например, Multiple VLAN Registration Protocol, IEEE 802.1Q, Dynamic Trunking Protocol), используемых коммутаторами, атакующий превращает access-порт коммутатора, к которому подключен, в trunk-порт и получает доступ через него к множеству VLAN'ов. Для защиты следует отключить автоматическую настройку trunk-портов на коммутаторах.

5.9. Вложенные VLAN

В больших сетях могут применяться иерархические VLAN'ы, трафик в которых может содержать более одного тега (функция Q-in-Q). Это может быть полезно, если провайдеру, использующему VLAN'ы, нужно соединить участки сетей клиентов, в которых тоже используются VLAN'ы. Вместо согласования используемых номеров VLAN'ов провайдер может ставить свои теги в качестве внешних на трафик клиентов, теги которых становятся внутренними (рис. 5.7). Внутренние VLAN'ы называются CVLAN (Customer VLAN), внешние – SP-VLAN (Service-provider VLAN). Для внешних VLAN аналогом access-портов будут UNI-порты, а аналогом trunk-портов – NNI порты.

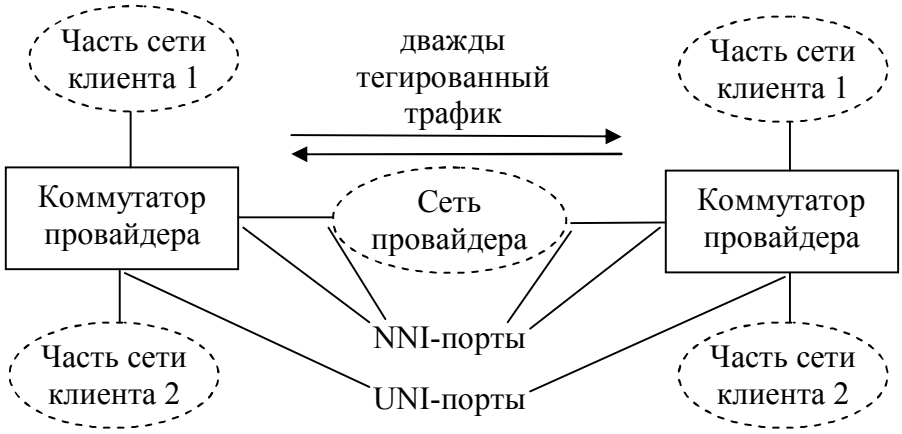

Рис. 5.7. Пример сети с иерархическими VLAN'ами.

Другое применение Q-in-Q – увеличение максимального количества VLAN'ов, которое ограничено 4094 в случае одинарных тегов. Двойные теги позволяют получить 4 094 × 4 094 = 16 760 836 VLAN'ов.

Внутренние и внешние теги обычно имеют разные значения поля TPID.

5.10. Асимметричные VLAN

Если порт принимает трафик некоторого VLAN'а, но не отправляет, или наоборот, то VLAN называется асимметричным. Hybrid-порт, отправляющий нетегированным трафик более чем одного VLAN'а, создает таким образом асимметричные VLAN'ы, так как принимаемый нетегированный трафик может попадать только в один VLAN – с номером PVID.

Применение асимметричных VLAN'ов позволяет без использования тегированного трафика организовать сети с нетранзитивной доступностью узлов (т.е. если могут взаимодействовать А и Б, Б и В, то А и В при этом могут быть изолированы друг от друга). На рис. 5.8 показан пример, когда это может быть полезно.

Узлы К1 и К2 имеют доступ друг к другу и к серверу, но не имеют доступа к узлам К4 и К5; узлы К4 и К5 имеют доступ друг к другу и к серверу, но не имеют доступа к узлам К1 и К2; сервер имеет доступ ко всем узлам.

Не все коммутаторы поддерживают асимметричные VLAN'ы, а если поддержка есть, то на работу коммутатора могут налагаться различные ограничения.

Функция *сегментации трафика* (traffic segmentation) предоставляет более простой способ настройки разрешенных и запрещенных путей трафика, чем асимметричные VLAN. Эта функция, в отличие от асимметричных VLAN, работает только в пределах одного коммутатора (или стека коммута-

торов) и для n-портового коммутатора представляет собой матрицу n×n, каждая ячейка которой показывает, разрешен или запрещен трафик от одного порта к другому.

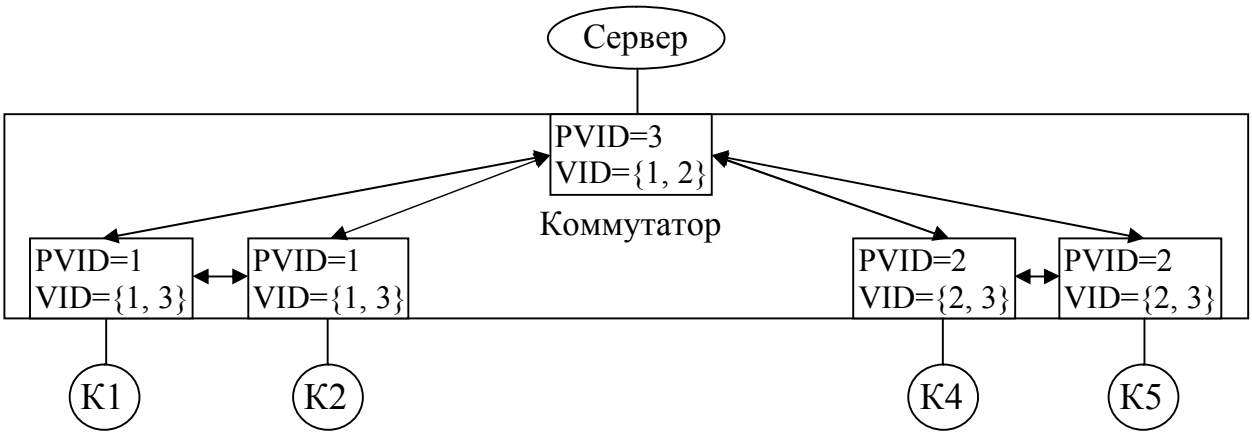

Рис. 5.8. Пример использования асимметричных VLAN. Все порты гибридные нетегирующие. Для каждого порта указаны PVID и VID, стрелками показаны разрешенные пути трафика.

При выборе коммутатора для организации VLAN-сетей может быть важно выяснить:

- Поддерживаемые типы VLAN, возможность функционирования одновременно VLAN разных типов (например, некоторые коммутаторы поддерживают VLAN'ы на основе портов и на основе тегов, но не оба типа одновременно).

- Максимальное количество VLAN, обрабатываемых коммутатором (у многих коммутаторов оно значительно меньше, чем 4096).

- Поддержка асимметричных VLAN, её ограничения.

- Поддержка гибридных портов.

- Поддержка тех или иных режимов фильтрации входящего трафика (ingress filtering, ingress checking).

- Поддержка GVRP и других протоколов автоматизации конфигурирования VLAN.

- Возможности работы с вложенными VLAN.

5.11. Лабораторная работа 7. Применение VLAN для обеспечения безопасности в сетях

Цель: *научиться применять симметричные и асимметричные VLAN, сегментацию трафика, вложенные VLAN для обеспечения безопасности в сетях.*

Схема сети, используемой в лабораторной работе, приведена на рис. 5.9.

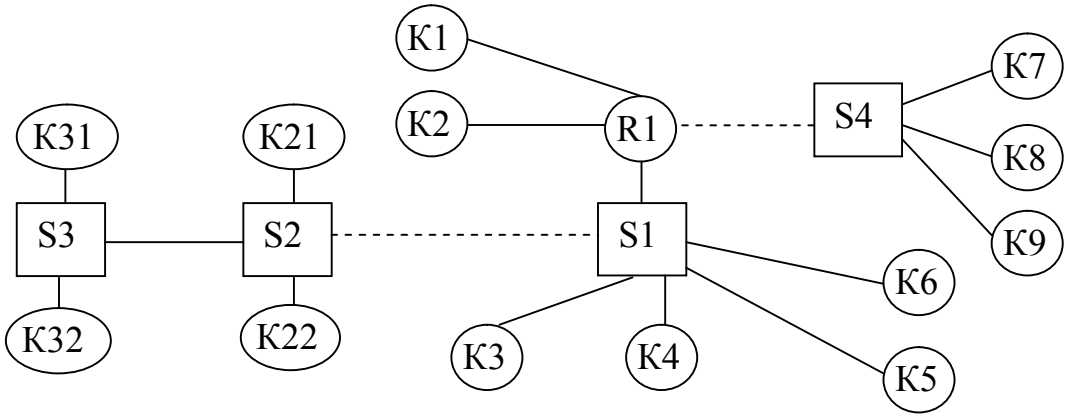

Рис. 5.9

К1 – шлюз в интернет; К2 – сервер организации; К3, К4, К5, К6, К7, К21, К22, К31, К32 – узлы, которым необходимо обеспечить связь с К1 (предоставить доступ в интернет) и связь с К2. К8 и К9 – узлы, которым необходимо предоставить доступ в интернет, но запретить взаимодействие с К2.

Все узлы, которым доступны К1 и К2, должны быть недоступны друг для друга, за исключением следующих взаимодоступных групп узлов: пара К5 и К6; тройка К21, К32 и К3; пара К31 и К22.

5.11.1. Порядок выполнения работы

0. Собрать физически сеть, моделирующую описанную выше.

1. Настроить VLAN 2 и VLAN 3 на коммутаторах S2 и S3 таким образом, чтобы К21 и К32 входили в VLAN 2, а К31 и К22 входили в VLAN 3. Проверить выполнение требуемых условий видимости между компьютерами, подключенными к коммутаторам S2 и S3.

2. Соединить коммутаторы S2 и S1. Настроить на S1 асимметричный VLAN таким образом, чтобы, во-первых, все узлы, подключенные к S1, имели доступ к R1, но не между собой; во-вторых, чтобы имели доступ друг к другу К5 и К6; в-третьих, чтобы были видимы друг для друга К3 и S2. Проверить выполнение требуемых условий видимости.

3. На коммутаторах S2 и S3 настроить native VLAN для обеспечения видимости между К3, К21, К32. Проверить выполнение требуемых условий видимости.

4. Соединить S4 и R1. Настроить ACL на R1 так, чтобы запретить взаимодействие между подсетями S4 и S1-S2-S3. Проверить недоступность подсетей друг для друга.

5. Настроить VLAN на коммутаторе S4 таким образом, чтобы любой трафик от S4 к R1 был с тегом 4, за исключением трафика от К7, который должен быть нетегированным. К8 и К9 оставить видимыми друг для друга. Проверить видимость между узлами, подключенными к S4.

6. Используя функцию сегментации трафика на S4, запретить прямое взаимодействие узлов К8 и К9. Проверить видимость между узлами, подключенными к S4.

7. Настроить ACL на маршрутизаторе R1 таким образом, чтобы запретить доступ от К8 и К9 к К2. Убедиться, что R1 запрещает запросы к К2 от К8 и К9 и пропускает запросы к К1.

8. С К32 отправить такой пакет, который, не выходя за пределы подсегмента S2-S3, достигнет К22. Настроить коммутаторы S2 и S3 таким образом, чтобы такие нарушения стали невозможны. Проверить невозможность обхода изоляции VLAN после изменений настроек коммутаторов.

9. Обеспечить видимость между К3, К21, К32, а затем и К7, чтобы при этом не нарушились условия изоляции других узлов. Какими способами можно этого добиться? Разрешается создание двойных связей между узлами и петель на коммутаторах. Проверить выполнение требуемых условий видимости.

5.11.2. Контрольные вопросы

1. Перечислите способы защиты от атаки с двойным тегированием и native VLAN.

2. Порты каких типов должны быть на коммутаторе для организации асимметричных VLAN?

3. В чем сходства и различия функций асимметричных VLAN и сегментации трафика?

4. Перечислите способы запретить взаимодействие между узлами А и Б в сети, не нарушая взаимодействие между множеством других конечных узлов, если: а) А и Б соединены через коммутаторы; б) А и Б соединены через маршрутизаторы; в) А и Б соединены через цепочку из коммутаторов и маршрутизаторов.

6. ИМИТАТОР JAVANETSIM

Основной задачей имитатора javaNetSim является имитация работы всех уровней стека протоколов TCP/IP. Для этого имитируется работа протоколов

каждого из уровней, чем достигается полная имитация работы сети. Вот почему имитатор javaNetSim удобен для выполнения лабораторных работ. Основные приемы работы с имитатором javaNetSim будут рассмотрены далее.

Имитатор javaNetSim является объектно-ориентированным и написан на языке Java. Программы, написанные на этом языке, являются машинно-независимыми, т.е. имитатор javaNetSim будет работать на любом компьютере, для которого есть виртуальная Java-машина. Использование языка Java не приводит к проблемам с быстродействием имитатора: он разрабатывался для моделирования работы небольших сетей, обработка моделей которых не требует больших вычислительных ресурсов.

Архитектура имитатора javaNetSim выглядит следующим образом. В основе лежит класс Simulation (Имитация), который содержит объекты классов Link (Линия) и Node (Узел). Этот класс предназначен для объединения устройств и линий связи в единую сеть. Класс Link содержит ссылки на объекты класса Node и предназначен для соединения двух узлов между собой. Класс Node содержит ссылки на объекты класса Link и является наиболее общей моделью сетевого устройства. Все реальные сетевые устройства являются производными от объекта класса Node и соответствуют модели стека протоколов TCP/IP:

• Hub (Концентратор) – DataLink Layer Device (Устройство физического уровня) – имеет пять портов, т.е. к нему возможно подключить до пяти линий связи;

• Router (Маршрутизатор) – Network Layer Device (Устройство сетевого уровня) – имеет два порта, а также стек протоколов TCP/IP (ProtocolStack);

• PC (Компьютер) – Applications Layer Device (Устройство уровня приложений) – имеет один порт, стек протоколов TCP/IP, а также возможность выполнять клиентскую или серверную часть какого-либо приложения.

Для взаимодействия с пользователем каждому сетевому устройству нужно графическое соответствие. Его обеспечивают следующие классы:

• GuiHub (Графический пользовательский интерфейс концентратора);

• GuiRouter (Графический пользовательский интерфейс маршрутизатора);

• GuiPC (Графический пользовательский интерфейс компьютера).

Как сами сетевые устройства, так и графический пользовательский интерфейс сетевых устройств должны быть едиными. Этим объединением занимается класс SandBox (Рабочая область). Рабочая область является частью

основного окна программы, представленного на рис. 6.1.

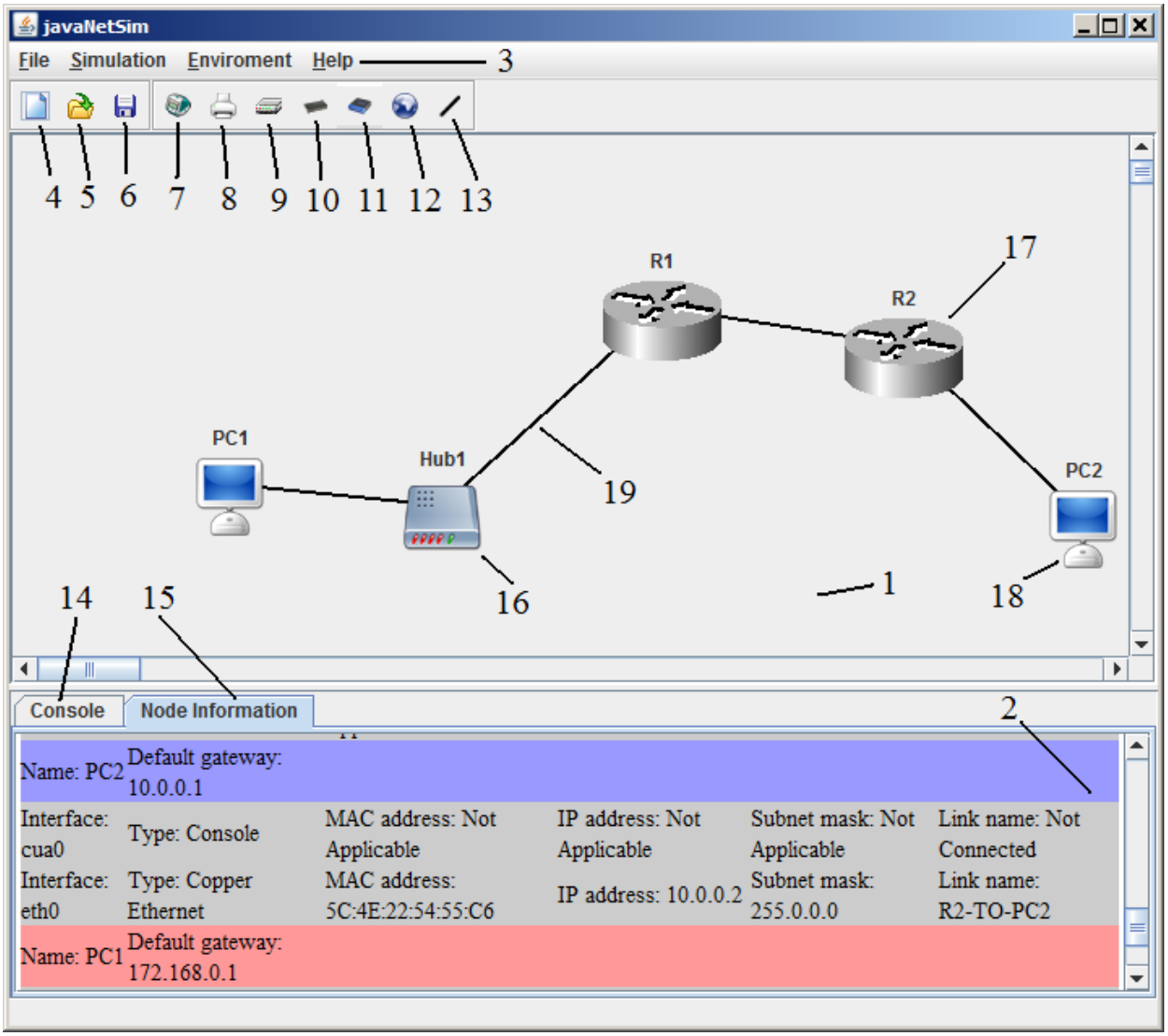

Рис. 6.1

Основное окно программы логически разделено на четыре части:

1. Рабочая область (обозначена на рисунке цифрой 1) содержит сетевые устройства и линии связи между ними:

– концентратор на пять сетевых интерфейсов (16);

– маршрутизатор, соединяющий две подсети (17);

– компьютер или конечный узел сети (18);

– линия связи между двумя сетевыми устройствами (19).

2. Область вывода результатов (обозначена на рисунке цифрой 2) содержит две вкладки:

– вкладку «консоль» (обозначена числом 14), содержащую журнал передачи пакетов по сети;

– вкладку «информация об устройствах» (обозначена числом 15), содержащую для каждого интерфейса всех сетевых устройств IP-адрес, маску подсети и шлюз по умолчанию.

3. Главное меню (обозначено на рисунке цифрой 3) содержит основные действия по управлению имитатором.

4. Линейка инструментов (обозначена на рисунке цифрой 4) содержит следующие кнопки:

– «создать пустую конфигурацию" (4);

– «открыть существующую конфигурацию» – (5);

– «сохранить текущую конфигурацию» – (6);

– «создать компьютер» – (7);

– «создать принтер» – (8);

– «создать маршрутизатор» – (9);

– «создать концентратор» – (10);

– «создать коммутатор» – (11);

– «создать устройство обслуживания канала/данных» – (12);

– «создать соединение» – (13).

Основное окно программы представляет собой инструмент взаимодействия пользователя с имитатором. С помощью этого инструмента пользователь может добавлять, удалять и соединять между собой сетевые устройства, а также работать с сетью на любом из четырех уровней стека протоколов TCP/IP.

6.1. Главное меню программы

Меню File (файл) позволяет создавать, открывать и сохранять конфигурации сетей для их дальнейшего использования. Меню содержит пять пунктов:

● New (Новый) – создать пустую конфигурацию;

● Open... (Открыть...) – открыть существующую конфигурацию;

● Save...(Сохранить...) – сохранить текущую конфигурацию;

● Save As...(Сохранить Как...) – сохранить текущую конфигурацию под новым именем;

● Exit (Выход) – выйти из имитатора javaNetSim.

Режим проектирования сети доступен из меню Simulation (Имитация). Это меню содержит подменю Add (Добавить), позволяющее создавать новые

сетевые устройства (такие, как концентратор, маршрутизатор, коммутатор или компьютер).

Управление параметрами имитатора доступно из меню Environment (Окружение) и позволяет изменять режим отображения информации, а также очищать область вывода результатов. Меню содержит четыре пункта:

● Clear Console (Очистить консоль). Удаляет все записи из вкладки «консоль»;

● Clear Node Information (Очистить информацию об устройствах). Удаляет все записи из вкладки «информация об устройствах»;

● Show simulation messages for: (Показывать сообщения имитатора для:). Позволяет задать режим вывода на вкладку «консоль» сообщений только определенных уровней стека протоколов TCP/IP. Есть возможность выбрать следующие уровни: Link and DataLink Layers (Физический и канальный уровни), Network Layer (Сетевой уровень), Transport Layer (Транспортный уровень), Application Layer(Уровень приложений);

● Show headers (Показывать заголовки). Позволяет задать режим вывода на вкладку «консоль» сообщений с названиями уровней и/или с типами пакетов.

С помощью меню «Environment > Show simulation messages for:» можно отключить сообщение от тех уровней стека протоколов TCP/IP, в которых нет необходимости. Это уменьшит количество информации, выводимой в «консоль», и облегчит поиск нужных данных.

6.2. Контекстное меню

Контекстное меню, вызываемое щелчком правой кнопки мыши, различается для устройств, работающих на разных уровнях стека протоколов TCP/IP.

Основные пункты контекстного меню, общие для всех устройств, перечислены ниже:

● Delete (Удалить). Без подтверждения удаляет выбранное сетевое устройство из текущей конфигурации.

● Properties (Свойства). Вызывает диалог, показывающий сетевые настройки выбранного устройства. Для каждого интерфейса отображается MAC-адрес, IP-адрес, маска подсети, название подключенной линии связи. Также для устройства указаны имя и шлюз по умолчанию.

● Break Link (Разорвать линию связи). Вызывает диалог, в котором можно выбрать интерфейс, линию связи которого требуется разорвать.

● Links Properties (Свойства линий связи). Дает возможность установить свойства линии связи.

При выборе пункта Link Properties (Свойства линий связи) вызывается диалог, который позволяет установить коэффициент пропускания для интерфейса, показывающий, какой процент пакетов линия связи, подключенная к этому интерфейсу, будет пропускать. Коэффициент пропускания задается для интерфейса (eth0, eth1 и т.д.).

В меню концентратора имеются два дополнительных пункта, позволяющих следить за его состоянием и, в случае необходимости, восстанавливать исходное состояние:

● Show state (Показать состояние). Показывает текущее состояние концентратора, может принимать два значения: normal (концентратор работает) и freezed (концентратор был остановлен из-за ошибки).

● Reset (Перезагрузить). Если концентратор находится в состоянии останова, то эта команда вернет его в рабочее состояние.

В меню устройств, работающих на сетевом уровне (маршрутизаторы и компьютеры), в дополнение к основным имеются еще семь пунктов:

● Set TCP/IP Properties (Установка свойств TCP/IP). Вызывает диалог, позволяющий изменить свойства TCP/IP.

● Send Ping... (Послать Echo-запрос). Позволяет послать Echo-запрос адресату.

● ARP-подменю. Позволяет работать с таблицей протокола ARP на выбранном устройстве.

● Counters-подменю содержит два пункта:

– Show Packet Counters (Показать счетчики пакетов). Показывает счетчики для пакетов протоколов ARP, IP, UDP, TCP;

– Reset Packet Counters (Сбросить счетчики пакетов). Устанавливает все счетчики на выбранном устройстве в нуль.

● Console. Вызывает командную строку, позволяющую настраивать таблицы маршрутизации, ARP-таблицы и др.

● Print route table (Показать таблицу маршрутизации). Выводит на вкладку «консоль» таблицу маршрутизации выбранного сетевого устройства.

● Applications (Приложения). Позволяет работать с протоколами SNMP, TELNET и DHCP.

Пункт контекстного меню Send Ping... (Послать Echo-запрос) вызывает диалог, в котором можно настроить параметры Echo-запроса. Во время передвижения пакетов по сети во вкладке «консоль» должны появиться сообщения, аналогичные приведенным далее:

```
PC1 Echo Request Packet Network Created Echo
                                Request packet to 10.0.0.2
...
PC1 Echo Reply Packet   Network Echo reply packet
                                received from 10.0.0.2
```

Меню ARP, позволяющее управлять таблицей протокола ARP на выбранном устройстве, содержит три подпункта:

● Add static entry to ARP table. Вызывает два диалоговых окна: в первом вводится MAC-адрес, а во втором – IP-адрес, после чего в ARP-таблицу заносится статическая запись о связи IP- и MAC- адресов;

● Remove entry from ARP table. Вызывает диалоговое окно, позволяющее ввести IP-адрес, для которого будет удалена запись из ARP-таблицы;

● Print ARP table. Выводит на вкладку «консоль» ARP-таблицу выбранного сетевого устройства.

Подменю Applications контекстного меню компьютеров (устройств, более полно поддерживающих уровень приложений), расширено: добавлены возможности работы с протоколами Echo (UDP, TCP), запуска Telnet-клиента, DHCP-сервера.

6.3. Командная строка

Для запуска командной строки из контекстного меню выберем «Console» – появится окно консоли (рис. 6.2). Окно разделено на 2 части: 1 и 2.

В консоли могут использоваться следующие специальные клавиши:

• Enter – выполнить введенную команду;

• ↑, ↓ – просмотр истории команд;

• ESC – очистить командную строку;

• Ctrl+D – закрыть консоль.

Некоторые команды становятся доступны в командной строке после перехода в режим конфигурирования. Этот переход выполняется командой `conf t`.

Для вывода списка команд, доступных в текущем режиме, введите символ «?».

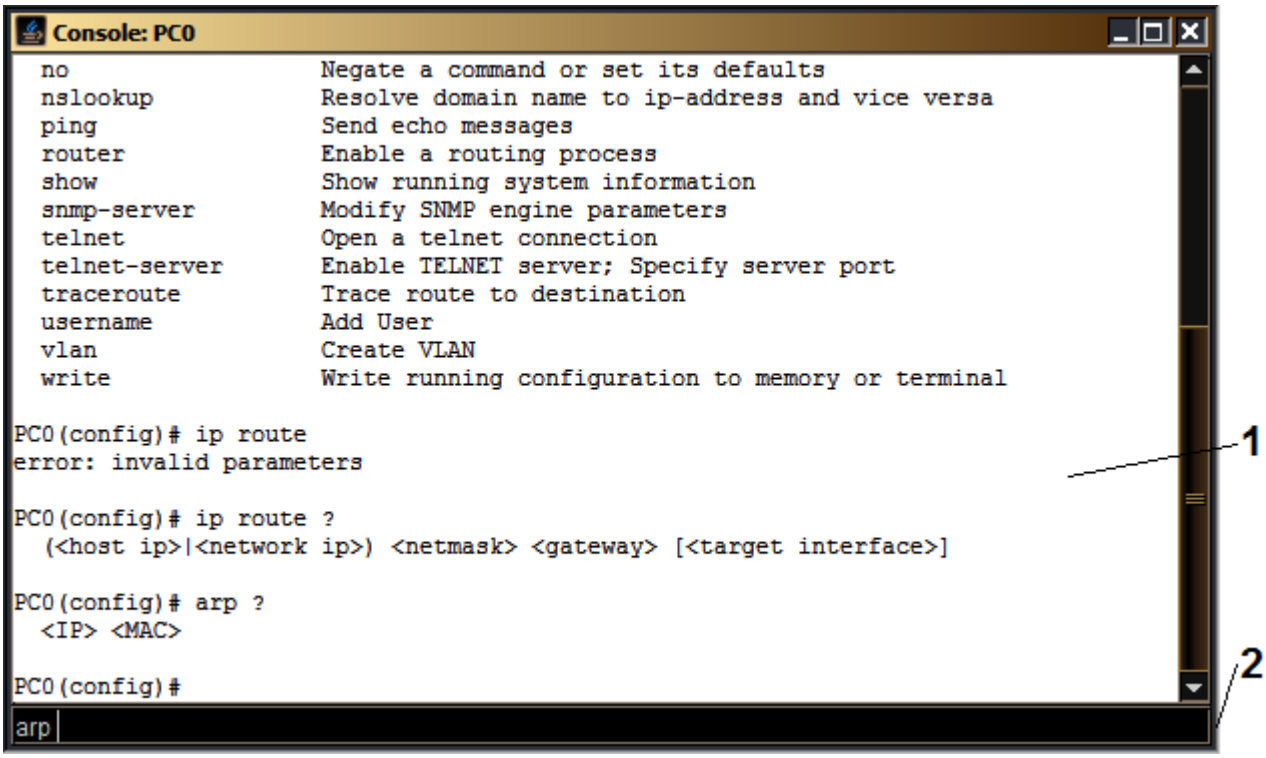

Рис. 6.2. 1 – область для сохранения результата выполнения команд; 2 – командная строка, в которой можно вводить команды на выполнение.

В режиме конфигурирования среди прочих доступны команды настройки таблицы маршрутизации, ARP-таблицы и SNMP-агента.

Синтаксис команд настройки таблицы маршрутизации:

- `ip route (<IP-адрес устройства>|<IP-адрес сети>) <маска подсети> <шлюз> [<интерфейс отправления>]` – добавить новый маршрут для сети или устройства;
- `no ip route (<IP-адрес устройства>|<IP-адрес сети>)` – удалить существующий маршрут для указанного IP адреса устройства или сети;
- `show ip route` – просмотреть список существующих маршрутов.

Для сохранения таблицы маршрутизации в памяти следует ввести команду `write mem`.

Синтаксис команд настройки ARP-таблицы:

- `show arp` – просмотреть ARP таблицу;
- `no arp <IP-адрес>` – удалить из ARP-таблицы запись об IP-адресе;
- `arp <IP-адрес> <MAC-адрес>` – добавить ARP-запись, связывающую IP- и MAC-адреса.

Синтаксис команд управления SNMP-агентом:

- `snmp port <порт SNMP-агента>` – включить SNMP-агент;
- `snmp community <имя группы доступа>` – задать группу доступа для SNMP-агента; если не выполнять эту команду, то используется значение по умолчанию – public;
- `no snmp port` – выключить SNMP-агент.

При вводе строки вида «команда ?» будет выведена краткая информация по использованию этой команды.

Список литературы

1. MCSE: Microsoft TCP/IP: Учебный курс: Сертификационный экзамен 70-059. 3-е изд. / пер. с англ. О. Михальского. М.: Русская редакция, 2001.

2. Rik Farrow. VLAN insecurity. URL: http://rikfarrow.com/Network/net0103.html

3. Scott Hogg. Tagging the Native VLAN. URL: http://www.networkworld.com/article/2234512/cisco-subnet/cisco-subnet-tagging-the-native-vlan.html

4. VLAN. URL: http://xgu.ru/wiki/VLAN

5. VLAN hopping. URL: https://en.wikipedia.org/wiki/VLAN_hopping

6. Андреев В. П., Врублевский В. В. Особенности использования асимметричных VLAN для построения структурированной локальной вычислительной сети // Информационно-измерительные и управляющие системы. 2009. Том7, № 6. С. 109–113.

7. Багиров Р. Технология Q-in-Q. URL: http://ftp.dlink.ru/pub/Trainings/SwitchWhitePapers/Q-in-Q_Port-Based_and_Selective.pdf

8. Герасимов И. В., Калмычков В. А., Чугунов Л. А. Информатика. Применение сетевых компьютерных технологий [Электронный ресурс]: электрон. учеб. пособие / Санкт-Петербургский государственный электротехнический университет им. В.И. Ульянова (Ленина) «ЛЭТИ». 2-е изд., доп. Электрон. текстовые дан. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2013. 1 эл. опт. диск (CD-ROM).

9. Гладцын В. А., Яновский В. В. Сетевые технологии: Учеб. пособие / СПбГЭТУ «ЛЭТИ». — СПб., 1998. 115 с.

10. Гладцын В. А., Яновский В. В., Кринкин К. В. Сети ЭВМ и телекоммуникации: Учеб. пособие / СПбГЭТУ «ЛЭТИ». — СПб., 2010. 96 с.

11. Головин Ю. А., Суконщиков А. А., Яковлев С. А. Информационные сети: учеб. для вузов по направлению подгот. «Информационные системы» 2-е изд., стер. М.: Академия, 2013.

12. Кирх О. Linux для профессионалов: Руководство Администратора Сети. 2-е изд. СПб.: Питер, 2001.

13. Коммутатор агрегации : серия QSW-8400. URL: ftp://ftp.qtech.ru/upload/Switch/Manual/QSW-8400/Command%20Guide%20Edit/QSW-

8400_%D0%BA%D0%BE%D0%BD%D1%84%D0%B8%D0%B3%D1%83%D1%80%D0%B0%D1%86%D0%B8%D1%8F%20VLAN_v1.1.pdf

14. Лекция 2: Начальная настройка коммутатора. URL: http://www.intuit.ru/studies/courses/3591/833/lecture/14253

15. Леинванд А., Пински Б. Конфигурирование маршрутизаторов Cisco. 2-е изд. М.: «Вильямс», 2004.368 с.

16. Олифер В. Г, Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы. Учебник. 5-е изд. 2016.

17. Олифер В. Г., Олифер Н. А. Новые технологии и оборудование IP-сетей. СПб.: БХВ-Петербург, 2001. 512 с.

18. Пахомов С. Возможности современных коммутаторов по организации виртуальных сетей. URL: http://compress.ru/article.aspx?id=10522&iid=430

ГЛОССАРИЙ

802.1Q. Открытый стандарт IEEE, описывающий процедуру тегирования трафика для передачи информации о принадлежности к VLAN.

ACL (Access Control List). Список контроля доступа, который определяет, кто (что) может получать доступ к конкретному объекту и какие именно операции разрешено или запрещено этому субъекту проводить над объектом. В сетях ACL могут быть настроены на маршрутизаторе и служить в качестве межсетевого экрана, управляя входящим и исходящим трафиком.

ARP (Address Resolution Protocol). Протокол разрешения трансляции MAC-адресов на IP-адреса.

FTP. Прикладной протокол передачи файлов.

HTTP. Протокол прикладного уровня передачи данных (изначально – в виде гипертекстовых документов).

HTTPS. Расширение протокола HTTP, в котором данные передаются зашифрованными (с использованием криптографических протоколов SSL или TLS).

ICMP (Internet Control Message Protocol). Протокол передачи команд и сообщений об ошибках.

IP (Internet Protocol). Сетевой протокол стека TCP/IP, функционирующий на уровне межсетевого взаимодействия.

IP-сокет. Номер порта в совокупности с номером сети и номером конечного узла.

LLC (Logical Link Control). Подуровень управления логической связью — верхний подуровень канального уровня модели OSI. Управляет передачей

данных и обеспечивает проверку и правильность передачи информации по соединению.

MAC (Media Access Control). Подуровень управления доступом к среде — нижний подуровень канального уровня модели OSI. Обеспечивает физическую адресацию и механизмы управления доступом к каналам, позволяя нескольким узлам общаться между собой в многоточечной сети.

MAC-таблица. Используемая коммутатором таблица сопоставления номеров портов с MAC-адресами подключенных устройств.

NAT (Network Address Translation). Механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса транзитных пакетов.

PVID (Port VLAN ID). Идентификатор, которым помечается кадр, пришедший в порт коммутатора.

Q-in-Q. Технология передачи дважды тегированного на канальном уровне трафика. Описана в стандарте 802.1ad.

RARP (Reverse Address Resolution Protocol). Протокол обратного разрешения MAC-адресов.

SNMP (Simple Network Management Protocol). Простой протокол, работающий на уровне приложений, предназначенный для наблюдения за сетевыми устройствами и управления ими.

TCP (Transmission Control Protocol). Протокол стека TCP/IP, осуществляющий функции контроля за передачей данных. Функционирует на транспортном уровне.

TPID (Tag Protocol Identifier). Идентификатор протокола тегирования. Указывает, какой протокол используется для тегирования.

UDP (User Datagram Protocol). Протокол передачи пакетов пользователя.

VID (VLAN ID). Идентификатор (номер) VLAN.

VLAN (Virtual Local Area Network). Виртуальная локальная сеть. Единый логический сегмент сети может быть разбит на несколько виртуальных логических сегментов – VLAN, изолированных друг от друга: устройства, подключенные к разным VLAN, не будут видеть друг друга на канальном уровне.

VLAN асимметричная. VLAN, в которой есть порты, пропускающие входящий, но не исходящий трафик этой VLAN, и/или есть порты, пропускающие исходящий, но не входящий трафик этой VLAN.

VLAN симметричная. VLAN, все порты которой пропускают и входящий, и исходящий трафик этой VLAN.

Брандмауэр (файрвол, межсетевой экран). Программа или программно-аппаратная система, фильтрующая проходящий через нее трафик в соответствии с заданными правилами.

Кадр. Блок цифровой информации канального уровня.

Коммутатор (switch). Устройство, предназначенное для соединения нескольких физических сегментов компьютерной сети в логический сегмент.

Концентратор (hub). Устройство для объединения нескольких устройств Ethernet в общий физический сегмент.

Маршрутизация. Процесс определения последовательности промежуточных узлов, которые проходит IP-пакет при движении от отправителя к месту назначения.

Маршрутизация динамическая. Метод маршрутизации, при котором таблицы маршрутизации составляются с помощью протоколов маршрутизации.

Маршрутизация статическая. Метод маршрутизации, при котором таблицы маршрутизации задаются вручную.

Маршрутизатор. Сетевое устройство, используемое в компьютерных сетях передачи данных, которое на основании информации о сети (таблицы маршрутизации) и определенных правил принимает решения о пересылке пакетов сетевого уровня их получателю.

Маскарадинг. Разновидность NAT, при которой адреса подменяются на IP-адрес интерфейса, через который проходит пакет.

Мост. Сетевое устройство канального уровня, предназначенное для объединения сегментов (подсетей) компьютерной сети в единую сеть

Пакет. Цифровой блок данных сетевого уровня, содержащий служебные данные (заголовок пакета, в котором есть адреса отправителя и получателя, версия сетевого протокола, идентификатор протокола следующего уровня, указывающий, данные какого протокола содержит пакет, и др.) и полезные данные (полезная нагрузка (payload)).

Порт. Очередь, организуемая операционной системой, к точке входа прикладного процесса. Используется для определения процесса-получателя пакета в пределах одного хоста.

Прокси-сервер. Сетевой компьютер, исполняющий приложение, которое позволяет ему работать в качестве портала между одной сетью и

другой, например между интрасетью и Интернетом. Может использоваться для защиты сети или выравнивания трафика, причем незаметно для клиентов.

Протокол. Совокупность синтаксических и семантических правил, определяющая взаимодействие одноименных уровней двух узлов вычислительной сети.

Сервер. Аппаратный или программный компонент системы, выполняющий функции управления ресурсами в вычислительной сети.

Сообщение (message). Единица данных при взаимодействии клиента и сервера посредством протокола прикладного уровня.

Таблица маршрутизации. Специальная информационная структура, используемая для определения маршрута следования пакета по адресу его сети назначения.

Шлюз (gateway). 1) Устройство или программа, служащая для передачи трафика в другую сеть. 2) Устройство для объединения сетей разных архитектур, работающих под разными протоколами.

Оглавление

Учебно-методическое пособие

Борисенко Константин Александрович,
Фирсов Михаил Александрович,
Яновский Владислав Васильевич

IP-сети: маршрутизация, настройка, организация VLAN.

Издание публикуется в авторской редакции

СПбГЭТУ «ЛЭТИ»
197376, Санкт-Петербург, ул. Проф. Попова, 5