

Министерство науки и высшего образования РФ  
Федеральное государственное автономное образовательное  
учреждение высшего профессионального образования  
«Севастопольский государственный университет»

# **ИССЛЕДОВАНИЕ СПОСОБОВ ПОСТРОЕНИЯ И КОНФИГУРАЦИИ ОБОРУДОВАНИЯ КОМПЬЮТЕРНЫХ СЕТЕЙ**

**Методические указания**  
к выполнению лабораторных работ по дисциплине  
**«Инфокоммуникационные системы и сети»**  
Для студентов, обучающихся по направлению 09.03.02  
"Информационные системы и технологии"  
и 09.03.03 «Прикладная информатика»  
по учебному плану подготовки бакалавров  
дневной и заочной форм обучения

**Севастополь  
2020**

УДК 004.732

**Исследование способов построения и конфигурации оборудования компьютерных сетей.** Методические указания к лабораторным занятиям по дисциплине «Инфокоммуникационные системы и сети» / Сост., В.С. Чернега, А.В. Волкова – Севастополь: Изд-во СевГУ, 2020 – 67 с.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Инфокоммуникационные системы и сети». Целью методических указаний является помощь студентам в исследовании принципов построения и исследования локальных компьютерных сетей и конфигурации телекоммуникационного оборудования. Излагаются теоретические и практические сведения, необходимые для выполнения лабораторных работ, требования к содержанию отчета.

Методические указания рассмотрены и утверждены на методическом семинаре и заседании кафедры «Информационные системы» (протокол № 1 от « 31 » августа 2020 г.)

Рецензент: Брюховецкий А.А., к.т.н., доцент

## Содержание

1. Лабораторная работа №1. Исследование способов построения виртуальных локальных компьютерных сетей.....	4
2. Лабораторная работа №2. Исследование способов динамической маршрутизации пакетов в компьютерных сетях.....	29
3. Лабораторная работа №3. Исследование способов назначения списков контроля доступа в локальных компьютерных сетях.....	43
4. Лабораторная работа №4. Исследование способов конфигурации сетевых серверных служб стека протоколов TCP/IP.....	59

## Лабораторная работа №1

**Исследование способов построения виртуальных локальных компьютерных сетей****1 Цель работы**

Исследование принципов работы коммутаторов и виртуальных локальных сетей, способов конфигурации коммутаторов для построения виртуальных локальных сетей, приобретение практических навыков конфигурации коммутаторов и исследования функционирования виртуальных сетей.

**2 Основные теоретические положения****2.1 Локальные и виртуальные локальные компьютерные сети**

Локальные компьютерные сети (ЛКС) представляет собой такую разновидность сетей, в которой все ее компоненты, включая ЭВМ различных классов, расположены на ограниченной территории одного предприятия или учреждения и соединены через единую физическую среду. Расстояния между компьютерами локальной сети составляют от сотен метров до десятков (10...20) км. В локальных сетях сетевые компьютеры называют **рабочими станциями**. Ограниченность территории создает предпосылки для использования специфических способов передачи данных, отличных от традиционных, применяемых в глобальных сетях. Благодаря этому в ЛКС удастся реализовать значительно более высокую скорость передачи (до тысяч Мбит/с) и на несколько порядков более низкую вероятность ошибок при существенно меньших затратах. Расположение локальной сети на ограниченной территории влияет также на способы административного сетевого управления, а технические характеристики ЛКС приводят к необходимости введения новых протоколов.

В настоящее время наиболее распространенным типом локальных компьютерных сетей являются сети Fast Ethernet со скоростью передачи 100 Мбит/с, построенная по древовидной (иерархической) топологии (рисунок 2.1).

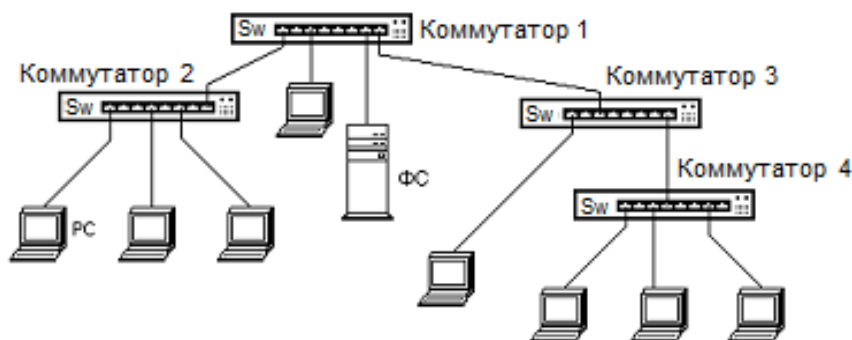


Рисунок 2.1 – Структура типовой локальной компьютерной сети Fast Ethernet

Локальная сеть строится на основе коммутаторов 2-го уровня Sw и линий связи типа витая пара. **Коммутатор (Switch)** представляет собой мультипроцес-

сорный мост, способный независимо транслировать кадры между всеми парами своих портов. Благодаря этому коммутаторы, разделяя локальную сеть на подсети, делят единый коллизийный домен на отдельные поддомены, свободные от коллизий. Коммутатор создает соединение между своими портами по принципу "точка-точка". Поэтому компьютеры, подключенные к этим портам, имеют в своем распоряжении пропускную способность (10 или 100 Мбит/с), которую способны обеспечить соответствующие порты коммутатора.

В такой сети, если не предусмотрено никаких ограничений, каждая рабочая станция РС может осуществлять обмен информацией с любой другой РС сети или получать доступ к файл-серверу. Недостаток такой ЛКС состоит в том, что пользователи одних рабочих групп могут получить доступ к рабочим станциям пользователей других групп. Это снижает уровень безопасности сети, а также скорость доступа к общим ресурсам.

Для устранения указанных недостатков разработана технология виртуальных локальных сетей *VLAN (Virtual LAN)*. Виртуальной локальной сетью называется совокупность узлов (рабочих станций и серверов) некоторой компьютерной сети, трафик которой, в том числе широковещательный, на канальном уровне полностью изолирован от трафика других узлов этой сети. Основное назначение *VLAN* – недопущение трафика из одной сети в другую. Это делается либо с целью увеличения реальной пропускной способности сегментов сети, или с целью защиты от несанкционированного доступа. Технология *VLAN* позволяет осуществить взаимодействие двух и более сетевых устройств на канальном уровне, хотя физически данные устройства, могут быть подключены к разным коммутаторам. *VLAN* ведут себя так же, как и физически разделённые локальные сети. То есть после разбивки сети на *VLAN* образуется несколько локальных сетей, которые далее возможно объединить в единое целое с помощью маршрутизации на третьем, сетевом уровне модели OSI.

Виртуальные сети возможно создавать на основе коммутаторов из групп пользователей, основываясь на их задачах, а не по физическому расположению в сети. *VLAN* могут быть построены на базе одного или нескольких коммутаторов.

## **2.2 Разновидности и возможности коммутаторов**

Коммутаторы по способу управления подразделяются на управляемые и неуправляемые. Неуправляемый коммутатор автоматически распределяет скорость и трафик между всеми клиентами сети. Неуправляемые коммутаторы широко используются в малых сетях с небольшим количеством (5-12) подключенных пользователей. Достоинством является простота в управлении и подключении.

Управляемые (программируемые) коммутаторы позволяют изменять режимы и способы коммутации путем загрузки в них управляющих программ. Управление коммутатором выполняет собственная операционная система, например Cisco IOS (*Internetwork Operating System*). Она хранится обычно в ПЗУ или флэш-памяти коммутатора. Многие управляемые коммутаторы позволяют настраивать такие функции как создание *VLAN*, задание качества обслуживания QoS, агрегирование и зеркалирование портов и др. Управляемые ком-

мутаторы позволяют управлять коммутацией на канальном (втором) или сетевом (третьем) уровнях модели OSI. Обычно их именуют соответственно «Layer 2 Switch» или «Layer 3 Switch» сокращенно «L2 и L3 Switch». Управление коммутатором может осуществляться посредством Web-интерфейса, интерфейса командной строки (CLI), протокола SNMP и т.п. В настоящее время существуют коммутаторы и программные средства, которые позволяют создавать VLAN и на базе **протоколов**, и на базе **правил**.

Все программируемые коммутаторы имеют **консольный порт**, функции которого выполняет асинхронный интерфейс RS-232. Такой порт позволяет управлять коммутатором с персонального компьютера, который с помощью консольного кабеля соединяется с COM-портом ПЭВМ. В новых типах коммутаторов консольный порт имеет разъем RJ-45. Этот разъем можно соединить посредством специального консольного кабеля и переходника с COM-портом компьютера.

В коммутаторах имеется две разновидности портов: порты доступа и магистральные (транковые) порты.

### 2.3 Способы создания VLAN

Виртуальные сети могут создаваться на основе способа *группирования портов* коммутатора или на основе группирования MAC-адресов сетевых устройств. При использовании способа группирования портов каждый порт программным образом назначается одной из виртуальных сетей. Обмен данными в таком случае будет осуществляться только между указанными портами. Порт можно приписать нескольким виртуальным сетям, однако, в случае требований повышенной безопасности это действие не допускается. В виртуальных сетях на основе группирования MAC-адресов каждый физический адрес приписывается той или иной виртуальной сети.

Достоинством VLAN на базе портов является высокий уровень управляемости и безопасности. К недостаткам такого вида сетей следует отнести необходимость физического переключения устройств при изменении конфигурации отдельных сетей.

Для уменьшения количества связей между коммутаторами, на которых сконфигурированы несколько виртуальных сетей, используется одна магистральная линия. По терминологии Cisco такое соединение называется транковым (*Trunk*). В магистральной линии мультиплексируются кадры, принадлежащие различным VLAN.

Разделение (демультиплексирование) входящих кадров производится на основании идентификаторов виртуальных сетей, которые включаются (инкапсулируются) в кадры Ethernet. Способ маркировки виртуальных сетей и формат Ethernet-кадров регламентируется международным стандартом **IEEE 802.1Q**. Корпорация Cisco разработала собственный протокол маркирования VLAN, который получил название «межкоммутаторный канал» ISL (*Inter Switch Link*). Коммутаторы Cisco поддерживают оба протокола. В соответствии со стандартом IEEE 802.1Q к кадру Ethernet добавлен специальный маркер (тег) виртуальной сети (*Tag*) размером в четыре байта. Эти 32 битовых бита содержат инфор-

мацию о принадлежности кадра Ethernet к конкретной VLAN и о его приоритете. Процедура добавления информации о принадлежности к 802.1Q VLAN в заголовок кадра называют маркированием кадра (*Tagging*), а извлечение маркера — *Untagging*.

Изменение структуры кадра Ethernet привело к нарушению совместимости со всеми традиционными устройствами Ethernet, ориентированными на старый формат кадра. Это связано с тем, что данные 802.1q размещаются перед полем с информацией о длине полезной нагрузки (или типе протокола). Традиционное сетевое устройство в процессе анализа заголовка не обнаружит эту информацию на обычном месте. На его месте располагается "маркер" виртуальной сети (рисунок 2.2). Новое поле состоит из тэга (маркера) протокольного идентификатора **TPID** (*Tag Protocol Identifier*) и тега управляющей информации **TCI** (*Tag Control Information*). Поле TPID имеет длину два байта и содержит фиксированный код 0x8100, который информирует, что кадр содержит тег протокола 802.1Q/802.1P. Поскольку это число больше максимальной длины кадра *Ethernet* (1500), то сетевые карты *Ethernet* будут интерпретировать его как тип, а не как длину кадра. Структура полей TCI изображена в нижней части рисунка 2.2



Рисунок 2.2 - Формат кадра Ethernet с меткой виртуальной сети

Трехбитовое поле "**Приоритет**" позволяет задавать 8 уровней приоритета передаваемых кадров и тем самым выделять *трафик реального времени*, *трафик со средними требованиями* и трафик, для которого *время доставки не критично*. Это открывает возможность использования сети Ethernet для задач управления и обеспечения качества обслуживания (QoS) при транспортировке мультимедийных данных. Наивысший уровень приоритета имеют кадры управления сетью, следующий приоритет задается кадрам передачи голосового трафика, а следующий, более низкий уровень, установлен для видеоданных. Остальные уровни предназначены для маркировки данных с разными требованиями по задержке доставки пакетов.

Однобитовое поле **CFI** (*Canonical Format Indicator*) зарезервировано для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet. Значение CFI=1 является указанием того, что в поле данных содержится кадр сети *Token Ring* (Стандарт IEEE 802.5).

Поле "**Идентификатор VLAN**" VID (*VLAN Identifier*) длиной 12 бит определяет, какой виртуальной сети принадлежит кадр. 12-битовое поле позволяет коммутаторам разных производителей создавать до 4096 общих виртуаль-

ных сетей. Обычно виртуальные сети с номерами VID0 и VID4095 резервируются.

Управление виртуальными локальными сетями по умолчанию осуществляется через VLAN1 (*Default VLAN*). Поэтому при конфигурировании коммутатора, как минимум, один порт должен относиться к VLAN1, чтобы можно было управлять коммутатором. Все остальные порты коммутатора могут быть назначены другим виртуальным сетям.

Передача пакетов между виртуальными сетями может быть осуществлена только через маршрутизатор. Поэтому, чтобы виртуальные сети могли обмениваться между собой пакетами каждой VLAN при конфигурировании должен быть назначен IP-адрес с соответствующей маской.

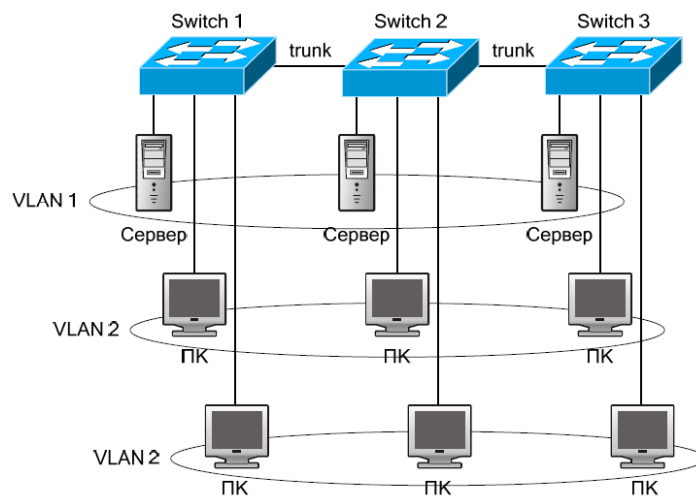


Рисунок 1.1 – Сеть VLAN, определенная логически

Сети VLAN обладают теми же свойствами, что и физические локальные сети, за исключением того, что VLAN являются логическими, а не физическими сетями. Поэтому конфигурирование сетей VLAN может выполняться безотносительно к физическому расположению устройств. Широковещательный, многоадресный и одноадресный трафики отдельно взятой VLAN отделены от трафика других VLAN.

Концепция VLAN, помимо решения проблемы с широковещательным трафиком даёт также ряд дополнительных преимуществ: формирование локальных сетей не по месту расположения ближайшего коммутатора, а по принадлежности компьютеров к решению той или иной производственной задачи; создание сети по типу потребляемого вычислительного ресурса и требуемой серверной услуги (файл-сервер, сервер баз данных). VLAN позволяют вести различную политику безопасности для разных виртуальных сетей; переводить компьютер из одной сети в другую без осуществления физического перемещения или переподключения.

Таким образом, технология VLAN обеспечивает следующие преимущества:

- улучшается производительность сети;
- экономятся сетевые ресурсы;

- упрощается управление сетью;
- снижается стоимость сети;
- улучшается безопасность сети.

В коммутаторах VLAN реализован в соответствии со стандартом 802.1Q.

### 2.2.1 Членство в сети VLAN

Сеть VLAN обычно создается администратором, который присваивает ей порты переключателя. Такой способ называется статической виртуальной локальной сетью (static VLAN). Если администратор немного постарается и присвоит через базу данных аппаратные адреса всех хостов, переключатель можно настроить на динамическое создание сети VLAN.

*Статические сети VLAN* являются типичным способом формирования таких сетей и отличаются высокой безопасностью. Присвоенные сети VLAN порты переключателей всегда сохраняют свое действие, пока администратор не выполнит новое присваивание портов. Этот тип VLAN легко конфигурировать и отслеживать, причем статические VLAN хорошо подходят для сетей, где контролируется перемещение пользователей. Программы сетевого управления помогут выполнить присваивание портов. Однако подобные программы использовать не обязательно.

*Динамические сети VLAN* автоматически отслеживают присваивание узлов. Использование интеллектуального программного обеспечения сетевого управления допускает формирование динамических VLAN на основе аппаратных адресов (MAC), протоколов и даже приложений. Предположим, MAC-адрес был введен в приложение централизованного управления VLAN. Если порт будет затем подключен к неприсвоенному порту переключателя, база данных управления VLAN найдет аппаратный адрес, присвоит его и сконфигурирует порт переключателя для нужной сети VLAN. Это упрощает административные задачи по управлению и настройке. Если пользователь перемещается в другое место сети, порт переключателя будет автоматически присвоен снова в нужную сеть VLAN. Однако для первоначального наполнения базы данных администратору придется поработать.

### 2.2.2 Коммутатор и VLAN

Компьютер при отправке трафика в сеть даже не догадывается, в каком VLAN'е он размещен. Об этом думает коммутатор. Коммутатор знает, что компьютер, который подключен к определенному порту, находится в соответствующем VLAN'е. Трафик, приходящий на порт определенного VLAN'а, ничем особенным не отличается от трафика другого VLAN'а. Другими словами, никакой информации о принадлежности трафика определенному VLAN'у в нём нет.

Однако, если через порт может прийти трафик разных VLAN'ов, коммутатор должен его как-то различать. Для этого каждый кадр трафика должен быть помечен каким-то особым образом. Пометка должна говорить о том, какому VLAN'у трафик принадлежит. Наиболее распространённый сейчас способ ставить такую пометку описан в открытом стандарте IEEE 802.1q.

## 2.3 Принцип коммутации

Внутри фрейма после Source MAC-адреса добавляется ещё одно поле, содержащее номер VLAN'а. Длина, выделенная для номера VLAN'а равна 12 битам, это означает, что максимальное число VLAN'ов 4096.

Кадры первого VLAN'а обычно не тегируются – он является родным VLAN'ом (native vlan). Каждый коммутатор принимает теперь решение на основе этой метки-тега (или его отсутствия).

Коммутация пакетов осуществляется с помощью таблицы коммутации, которая динамически составляется по мере работы коммутатора. Она представляет собой таблицу, содержащую записи о порте, соответствующем MAC-адресе устройства, а также номера VLAN, по-умолчанию «1» (см. таблицу 1.1). При поиске пары MAC-адрес/порт теперь будет сравниваться тег кадра с номером VLAN'а в таблице.

Таблица 1.1 – Таблица коммутации

Порт коммутатора	VLAN	MAC-адрес хоста
1	2	A
2	2	B
3	10	C
4	10	D

Каждая новая VLAN фактически создает новую таблицу коммутации. Тем не менее, все базовые механизмы коммутатора остаются точно такими же, как и до разделения на VLAN, но они используются только в пределах соответствующего VLAN.

### 2.3.1 Принадлежность VLAN

Порты коммутатора, поддерживающие VLAN'ы, (с некоторыми допущениями) можно разделить на два множества:

- нетегированные порты (access-порты, связи доступа) – к ним подключаются, как правило, конечные узлы. За каждым access-портом закреплён определённый VLAN, иногда этот параметр называют PVID. Весь трафик, входящий на этот порт от конечного устройства, получает метку этого VLAN'а, а исходящий уходит без метки. Трафик этого VLAN передается без тега. На Cisco нетегированным порт может быть только в одном VLAN, на некоторых других коммутаторах данного ограничения нет;

- тегированные порты (trunk-порты, магистральные связи) – линия между двумя коммутаторами или от коммутатора к маршрутизатору. Внутри такой линии (транка) передаётся трафик нескольких VLAN'ов. Тут трафик уже идёт с тегами, чтобы принимающая сторона могла отличить кадр, который идёт в бухгалтерию, от кадра, предназначенного для ИТ-отдела. За транковым портом закрепляется целый диапазон VLAN'ов. Без тега коммутатор не сможет различить трафик различных VLAN'ов.

Существует native vlan. Трафик этого VLAN'а не тегируется даже в транке, по умолчанию это 1-й VLAN и по умолчанию он разрешён. Можно пе-

реопределить эти параметры. Нужен он для совместимости с устройствами, незнакомыми с инкапсуляцией 802.1q. Например, через Wi-Fi мост нужно передать 3 VLAN'а, и один из них является VLAN'ом управления. Если Wi-Fi модули не понимают стандарт 802.1q, то управлять ими можно, только если этот VLAN настроить, как native vlan с обеих сторон.

Если порт тегирован для нескольких VLAN'ов, то в этом случае весь нетегированный трафик будет приниматься специальным родным VLAN'ом (native VLAN). Если порт принадлежит только одному VLAN как нетегированный, то тегированный трафик, приходящий через такой порт, должен удаляться. На практике это поведение обычно настраивается.

Обычно, по умолчанию все порты коммутатора считаются нетегированными членами VLAN 1. В процессе настройки или работы коммутатора они могут перемещаться в другие VLAN'ы.

### 2.3.2 Использование VLAN

VLAN позволяет разделять устройства на логические группы. Как правило, одному VLAN соответствует одна подсеть.

Устройства, находящиеся в разных VLAN, будут находиться в разных подсетях. Но в то же время VLAN не привязан к местоположению устройств и поэтому устройства, находящиеся на расстоянии друг от друга, все равно могут быть в одном VLAN независимо от местоположения. Суть вышесказанного показана на рисунке 2.2.

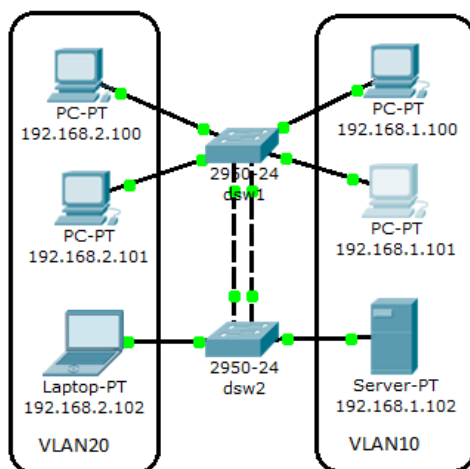


Рисунок 2.2 – Реализация нескольких VLAN

К первому коммутатору dsw1 подключены хосты из подсети 192.168.1.0/24, а также 192.168.2.0/24. Также к dsw1 подключен коммутатор dsw2, к которому в свою очередь подключен хост из подсети 192.168.2.0/24. Портam коммутатора, к которым подключены хосты подсети 192.168.2.0/24 назначен VLAN20, а подсети 192.168.1.0/24 – VLAN10.

Для того чтобы хосты 1.101 и 1.102 в VLAN'е 10 на коммутаторе dsw1, могли обмениваться информацией с хостами VLAN'а 10 добавлен линк (физическое соединение) между коммутаторами, который представляет собой соеди-

нение двух нетегированных портов, находящихся в области видимости каждого VLAN'a соответственно.

Однако, когда количество VLAN возрастает, то схема явно становится очень неудобной, так как для каждого VLAN надо будет добавлять линк между коммутаторами для того, чтобы объединить hosts в один широковещательный сегмент. Для решения этой проблемы используются тегированные порты (см. рисунок 2.3).

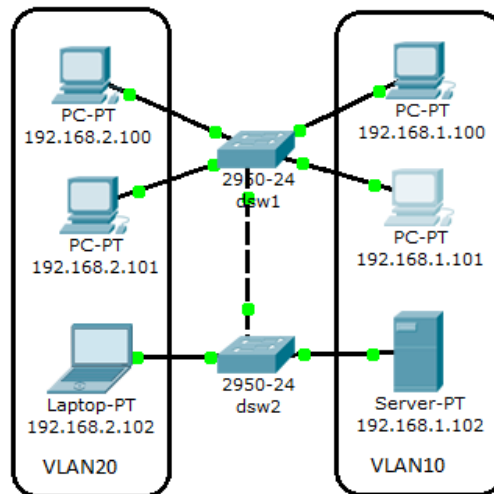


Рисунок 2.3 – Тегированный порт между коммутаторами

## 2.4 Принцип работы VLAN

1. От компьютера отправляется пакет другому компьютеру этой же сети. Этот пакет инкапсулируется в кадр, и пока никто ничего не знает о VLAN'ах, поэтому кадр уходит, как есть, на ближайший коммутатор.

2. У каждого VLAN'a есть номер. Существуют два типа VLAN:

- стандартный диапазон VLAN – от 1 до 1000;
- расширенный диапазон VLAN – от 1025 до 4096.

На каждом коммутаторе существует VLAN 1, все интерфейсы по умолчанию относятся к нему. Процесс настройки практически идентичен для всех коммутаторов Catalyst.

Сначала необходимо создать VLAN и задать ему имя:

```
switch(config)# vlan 2
switch(config-vlan)# name test
```

Просмотр информации о VLAN'ах:

```
switch# show vlan brief
```

3. На коммутаторе необходимый порт отметим как член 2-й VLAN командой:

```
Switch0(config)#interface fa0/1
Switch0(config-if)#description "I am using simple frames"
Switch0(config-if)#switchport mode access
Switch0(config-if)#switchport access vlan 2
```

Это означает, что любой кадр, пришедший на этот интерфейс, автоматически тегруется: на него вешается метка с номером VLAN'а. В данном случае с номером 2.

Далее коммутатор ищет в своей таблице MAC-адресов среди портов, принадлежащих 2-му VLAN'у, порт, к которому подключено устройство с MAC-адресом получателя.

3. Если получатель подключен к такому же access-порту, то метка с кадра исключается, и кадр отправляется в этот самый порт таким, каким он был изначально. То есть получателю также нет необходимости знать о существовании VLAN'ов.

4. Если же искомый порт, является транковым, то метка на нём остаётся.

```
Switch(config)#interface fa0/2
Switch(config-if)#description "I am using tagged frames"
Switch(config-if)#switchport mode trunk
```

Если тегированный кадр прилетит на access-порт, то он будет отброшен.

Если нетегированный кадр поступит на trunk-порт, то он будет помещён в native VLAN. По умолчанию им является 1-й VLAN. Но можно поменять его командой `switchport trunk native vlan 2`. В этом случае все кадры, помеченные 2-м VLAN'ом будут уходить в этот порт нетегированными, а нетегированные кадры, приходящий на этот интерфейс, помечаться 2-м VLAN'ом. Кадры с тегами других VLAN'ов останутся неизменными, проходя, через такой порт.

Конечным узлам (компьютерам, ноутбукам, планшетами, телефонами) можно отправлять тегированные кадры и соответственно подключать их к транковым портам только если сетевая карта и программное обеспечение поддерживает стандарт 802.1q, то узел может работать с тегированными кадрами.

Если тегированные кадры попадут на обычный неуправляемый коммутатор или другое устройство, не понимающее стандарт 802.1q, то скорее всего, коммутатор его отбросит из-за увеличенного размера кадра. Реакция коммутатора зависит от разных факторов: производитель, софт (прошивка), тип форвардинга (cut-through, store-and-forward).

По умолчанию в транке разрешены все VLAN. Можно ограничить перечень VLAN, которые могут передаваться через конкретный транк.

Указать перечень разрешенных VLAN для транкового порта fa0/0 можно командой:

```
switch(config)# interface fa0/0
switch(config-if)# switchport trunk allowed vlan 1-2,10,15
```

Добавление ещё одного разрешенного VLAN:

```
switch(config)# interface fa0/0
switch(config-if)# switchport trunk allowed vlan add 160
```

Удаление VLAN из списка разрешенных:

```
switch(config)# interface fa0/0
```

```
switch(config-if)# switchport trunk allowed vlan remove 160
```

## 2.5 Сеть управления

Настроим IP-адрес для управления. В реальной жизни это жизненно необходимо. Для этого мы создаём виртуальный интерфейс и указываем номер интересующего нас VLAN'а. А далее работаем с ним, как с самым обычным физическим интерфейсом.

```
switch(config)#interface vlan 2
switch(config-if)#description Management
switch(config-if)#ip address 172.16.1.2 255.255.255.0
switch(config-if)#no shutdown
```

## 2.6 InterVlan routing

VLAN'ы предназначены для разделения доменов широковещательной рассылки в локальной сети. Всякий раз, когда хостам в одном VLAN нужен доступ к хостам в другом VLAN, трафик должен маршрутизироваться между VLAN'ми. Данное перемещение трафика между VLAN называется inter-VLAN routing (внутренний роутинг между VLAN). На коммутаторах Cisco Catalyst он создаётся на интерфейсах третьего уровня (мы говорим о модели OSI) и называются данные интерфейсы как Switch virtualinterfaces (SVI) – виртуальные интерфейсы коммутатора. Рассмотрим, как настраивать данные интерфейсы и как решать связанные с данными интерфейсами проблемы.

### 2.6.1 Настройка маршрутизации между VLAN

1. На коммутаторе нужно настроить транковый порт в сторону маршрутизатора с помощью команд:

```
switch(config)#interface FastEthernet0/24
switch(config-if)# description InterVlan-switch
switch(config-if)# switchport trunk allowed vlan имена_vlan
switch(config-if)# switchport mode trunk
```

2. Лучше сразу же настроить время на маршрутизаторе. Это поможет корректно идентифицировать записи в логах.

```
Router#clock set 12:34:56 7 august 2012
```

3. Перейти в режим настройки интерфейса, обращённого в локальную сеть и включить его, так как по умолчанию он находится в состоянии Administratively down.

4. Создадим виртуальный интерфейс или иначе его называют подинтерфейс или ещё сабинтерфейс (sub-interface).

```
Router(config)#interface fa0/0.2
Router(config-if)#description Management
```

Таким образом, сначала указываем обычным образом физический интерфейс, к которому подключена нужная сеть, а после точки ставим некий уникальный идентификатор этого виртуального интерфейса. Для удобства, обычно номер сабинтерфейса делают аналогичным VLAN'у, который он терминирует.

5. Следующей командой обозначим, что кадры, исходящие из этого виртуального интерфейса, будут помечены тегом 2-го VLAN'а. А кадры, входящие на физический интерфейс FastEthernet0/0 с тегом этого VLAN'а будут приняты виртуальным интерфейсом FastEthernet0/0.2.

```
Router(config-if)#encapsulation dot1q 2
```

6. Определим IP-адрес на интерфейсе, который будет шлюзом по умолчанию (default gateway) для всех оконечных устройств в этом VLAN, который дополнительно, помимо IP-адреса и маски, необходимо прописать в настройках оконечного оборудования.

```
Router(config-if)#ip address 172.16.1.1 255.255.255.0
```

Аналогичным образом необходимо настроить все VLAN. Теперь хосты из разных VLAN'ов смогут видеть друг друга.

## 2.6.2 Пример настройки с использованием маршрутизатора

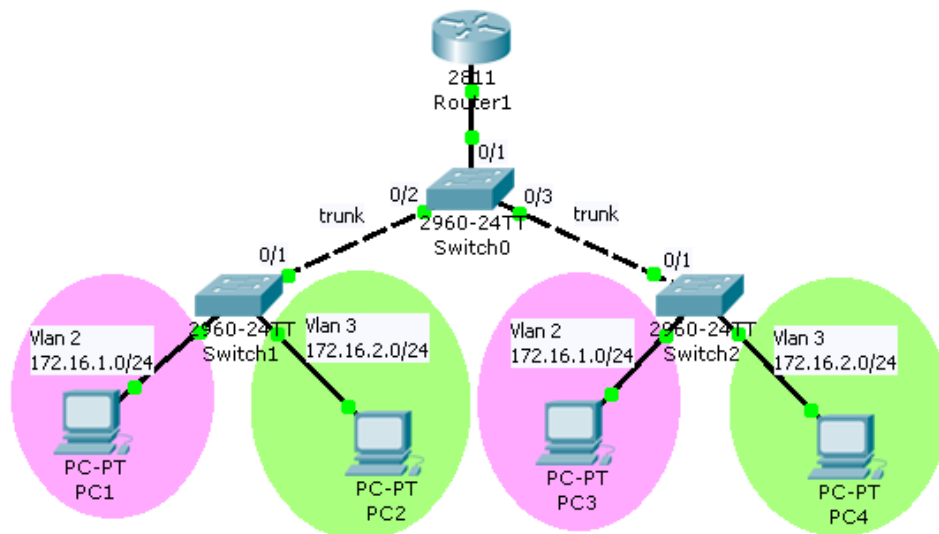


Рисунок 1.4 – InterVlan routing на маршрутизаторе

```
Router1(config)# interface fastethernet 0/0
Router1(config-if)# no shutdown
Router1(config)# interface fastethernet 0/0.2
Router1(config-if)# encapsulation dot1q 2
Router1(config-if)# ip address 172.16.1.1 255.255.255.0
Router1(config)# interface fastethernet 0/0.3
Router1(config-if)# encapsulation dot1q 3
Router1(config-if)# ip address 172.16.2.1 255.255.255.0
Router1# copy running-config startup-config
```

```
Switch0(config)# interface range fastethernet 0/1-3
Switch0(config-if)# switchport mode trunk
Switch0(config-if)# switchport trunk allowed vlan 2,3
```

```
S0# copy running-config startup-config
```

### 2.6.3 Пример настройки с использованием коммутатора 3 уровня

По умолчанию все порты коммутатора 3-го уровня (L3 коммутатор) работают в режиме L2, то есть это обычные порты коммутатора, на которых мы можем настроить VLANы. Но любой из них мы можем перевести в L3-режим, сделав портом маршрутизатора. Тогда на нём можно настроить IP-адрес:

```
MultiplayerSwitch0(config)#interface fa0/0
MultiplayerSwitch0(config-if)#no switchport
MultiplayerSwitch0(config-if)#ip address 192.168.1.1
255.255.255.252
```

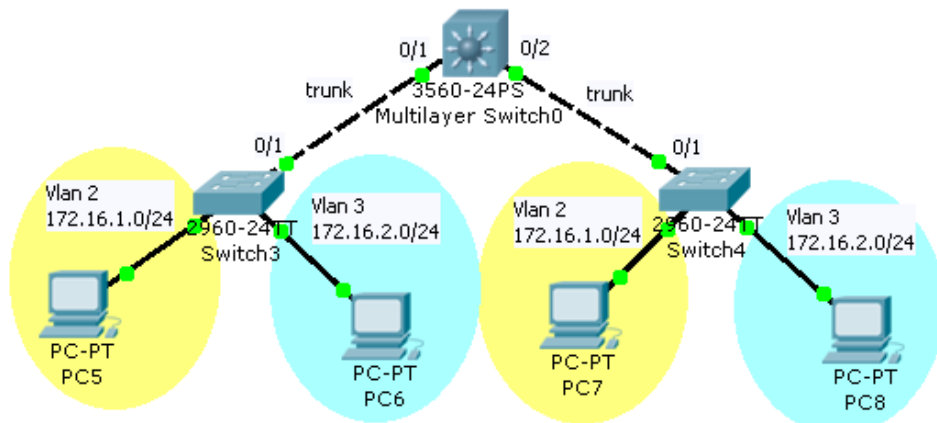


Рисунок 1.5 – InterVlan routing на коммутаторе L3-уровня

Чтобы коммутатор превратился в почти полноценный маршрутизатор, надо использовать команду: `ip routing`.

```
MultiplayerSwitch0(config)# interface range fastethernet 0/1-2
MultiplayerSwitch0(config-if)# switchport trunk encapsulation dot1q
MultiplayerSwitch0(config-if)# switchport mode trunk
```

```
MultiplayerSwitch0(config)# interface vlan 2
MultiplayerSwitch0(config-if)# ip address 172.16.1.1 255.255.255.0
```

```
MultiplayerSwitch0(config)# interface vlan 3
MultiplayerSwitch0(config-if)# ip address 172.16.2.1 255.255.255.0
```

```
MultiplayerSwitch0# copy running-config startup-config
```

## 2.7 Особенности работы протокола VTP

Зачастую, во многих организациях существует необходимость разделения ресурсов на различные подсети (сети) с контролем доступа в них. Для таких целей есть много решений. Один из самых простых способов – создание и

настройка VLAN на коммутаторах Cisco. Если надо добавить один-два VLAN, то никаких проблем это не составляет. Но что, если необходимо добавить не одну виртуальную сеть, а с десятков, и это требуется сделать не на одном коммутаторе, а на 10 или даже на 100? Вводить сотни строк кода каждый раз, когда необходимо изменить присутствующие на коммутаторах VLAN? Избавиться от такой нудной и кропотливой работы поможет протокол VTP, который используется для централизованного управления VLAN на коммутаторах.

Протокол VTP является одним из самых известных протоколов, занимающихся L2-задачами. История протокола VTP достаточно продолжительна – первый раз он появляется в CatOS 2.1 (это Cisco Catalyst 2900 и 5000), после чего слегка модернизируется до второй версии и живёт очень долго. Третья версия, в которой реально много полезных изменений, вышла несколько лет назад.

### 2.7.1 Назначение протокола VTP

Протокол VTP (англ. VLAN Trunking Protocol) – это протокол, который используется для обмена информацией о VLAN (виртуальных сетях), протокол разработан в Cisco. Ключевое назначение протокола VTP – это обмен коммутаторов специфической базой данных с информацией о VLAN'ах. В данном процессе могут участвовать и маршрутизаторы в случае, если у них установлена специальная карта, являющаяся мини-коммутатором (т.н. switchboard). VTP помогает упрощать операции с VLAN'ами в организации – добавление, удаление, изменение параметров, а также оптимизирует трафик, благодаря наличию функции `ntp pruning`. VTP 3-й версии ещё и помогает жить протоколу MST (стандарт 802.1s), плюс исправляет проблемы VTP версий 1 и 2.

Протокол VTP объединяет физически подключённые друг к другу коммутаторы (т.е. не через роутер или даже через другой свитч, но не поддерживающий VTP) в именованные области, называемые доменами VTP. В одной организации таких доменов может быть несколько, главное – чтобы они непосредственно не общались друг с другом.

### 2.7.2 Техническая реализация протокола VTP

Технологически протокол VTP реализован как SNAP-вложение в кадры ISL или 802.1Q. Работать может на 802.3 (Ethernet) и 802.5 (Token Ring).

Служебные данные VTP вкладываются не сразу в кадр 802.3, а после транкового заголовка. Выглядит это так:

- обычный заголовок 802.3 (Destination MAC, Source MAC, тип вложения – например, в случае 802.1Q это будет 0x8100);
- субзаголовок LLC-уровня, содержащий код 0xAA 0xAA, обозначающий, что далее идёт SNAP-вложение;
- субзаголовок SNAP – Subnetwork Access Protocol, показывающий, что будет разделение на субпротоколы канального уровня, а не сразу обработка уйдёт на сетевой уровень – несёт уже конкретную информацию, что вложен будет протокол, идентифицируемый как Cisco'вский протокол VTP.

А далее – уже данные самого протокола VTP, относящиеся к одному из типов сообщений – VTP summary advertisement, VTP subset advertisement, VTP advertisement request, VTP join message.

Весь трафик VTP идёт на специальный мультикастовый MAC-адрес вида 01-00-0c-cc-cc-cc. Так как на этот же адрес, допустим, идёт и трафик протокола CDP, тоже цисковского, то используется схема разделения по SNAP-типу – для CDP он выбран 0x2000, для VTP – 0x2003. Это мультиплексирование канального уровня.

### 2.7.3 Логическая реализация протокола VTP

#### 1. **Server** (режим по умолчанию):

- можно создавать, изменять и удалять VLAN из командной строки коммутатора;
- генерирует объявления VTP и передает объявления от других коммутаторов;
- может обновлять свою базу данных VLAN при получении информации не только от других VTP-серверов, но и от других VTP-клиентов в одном домене, с более высоким номером ревизии;
- сохраняет информацию о настройках VLAN в файле vlan.dat во flash;
- поддержка Private VLAN (VTPv3);
- возможность анонсирования VLAN из расширенного диапазона (VTPv3).

#### 2. **Client** (устройства с Read only доступом):

- нельзя создавать, изменять и удалять VLAN из командной строки коммутатора;
- передает объявления от других коммутаторов;
- синхронизирует свою базу данных VLAN при получении информации VTP;
- сохраняет информацию о настройках VLAN в файле vlan.dat во flash;
- настройки VLAN сохраняются в NVRAM и в режиме клиента (VTPv3);
- поддержка Private VLAN (VTPv3);
- возможность анонсирования VLAN из расширенного диапазона (VTPv3).

#### 3. **Transparent** (прозрачный):

- можно создавать, изменять и удалять VLAN из командной строки коммутатора, но только для локального коммутатора;
- не генерирует объявления VTP;
- передает объявления от других коммутаторов;
- не обновляет свою базу данных VLAN при получении информации по VTP;
- сохраняет информацию о настройках VLAN в NVRAM;
- всегда использует configuration revision number 0.

#### 4. **Off** (отключенный, новый режим работы VTP, добавился в 3 версии):

- не передает на другие порты полученные по транкам объявления VTP;
- в остальном аналогичен режиму Transparent.

В сети желательно иметь хотя бы один VTP-сервер, а если коммутатор один, то имеет смысл включить его сразу в режим VTP transparent. Этот режим

удобен тем, что коммутатор, работающий в нём в случае, если принимает кадр протокола VTP на любом порту, сразу передаёт этот кадр на все остальные транковые порты – т.е. просто ретранслирует этот кадр, не обрабатывая его. Этим (переключением в vtp transparent mode) заранее ликвидируется потенциальная возможность, что какой-то другой коммутатор повлияет на конфигурацию данного.

**Примечание:** Это всё – только на портах в режиме транка.

В разных режимах VTP хранение информации о VLAN'ах реализовано различными способами:

- коммутатор с ролью VTP Server будет хранить настройки в файле vlan.dat на указанном устройстве хранения (один из flash'ей устройства);
- коммутатор с ролью VTP Transparent или VTP Off будет хранить настройки в конфигурации (это config.text, или, говоря проще, NVRAM);
- коммутатор с ролью VTP Client будет хранить настройки в оперативной памяти (их не будет видно в конфигурации или на flash).

#### 2.7.4 Домены VTP

Домены VTP – это подмножества непосредственно подключенных друг к другу коммутаторов.

Для идентификации принадлежности устройства к домену VTP используется сравнение названия домена и хэша пароля (безусловно, не друг с другом, а между устройствами – т.е. оба этих параметра должны совпадать, чтобы VTP-устройства могли корректно общаться). Название домена VTP – это текстовая строка длиной до 32 байт (в случае, если название короче, оно добивается нулями – zero-padding), пароль – тоже текстовая строка, в чистом виде в сети не передающаяся, хранящаяся в случае работы устройства в роли VTP Server в файле vlan.dat, а в случае работы в режиме VTP Transparent – в config.text

Обратите внимание на следующие два важных момента. Первый – то, что название домена является case-sensitive, потому что строки сравниваются побайтово. Поэтому коммутаторы из домена Domain и из домена DOMAIN работать друг с другом не будут. Второй – работа с паролем. В кадрах передаётся хэш пароля, а не хэш кадра. Поэтому этот пароль нужен только для идентификации принадлежности к домену и никак не подтверждает целостность сообщения VTP.

Примечание: для успешной связи двух коммутаторов необходимо, чтобы у них были одинаковые версии протокола VTP.

Примечание: даже если Вы не используете протокол VTP как таковой по его основной задаче – синхронизации базы VLAN'ов, то Вам желательно, чтобы коммутаторы обладали одинаковыми настройками VTP в части имени домена. Причина – протокол динамического согласования транков – DTP – отправляет в ходе процедуры анонса имя VTP-домена, и в случае, если имя не совпадает, согласование не происходит. Т.е. два коммутатора, «смотрящие» друг на друга портами в режиме dynamic auto, в случае разных VTP-доменов просто не смогут согласовать транк. Решений будет несколько – отключить автосогласование

(`switchport mode trunk`), отключить DTP (`switchport nonegotiate`), или выставить одинаковые имена доменов.

### 2.7.5 Принцип работы протокола VTP

Протокол VTP в основном занимается передачей базы данных VLAN между устройствами. Делает он это в следующих случаях:

- если Вы изменили базу данных VLAN'ов на устройстве с ролью VTP Server (т.е. провели успешную запись – не важно, какую – добавили VLAN, удалили VLAN, переименовали VLAN), то изменение будет передано немедленно после проведения записи;

- если после последней успешной записи (см. выше) прошло 300 секунд.

Примечание: в случае, если у Вас коммутатор, записи в VLAN database идут сразу же – т.е. создали Вы VLAN, допустим – сразу разошлось уведомление. Если маршрутизатор со свичкартой – то Вы вначале вносите изменения «пачкой», а после выполняете команду `APPLY`, и только когда она успешно применится, будет разослан анонс.

VTP версий 1 и 2 передаёт только данные по VLAN'ам с номерами от 1 до 1005 (десятибитовые номера VLAN'ов). Зато в VTPv3 это успешно решено и обмен данными про VLAN'ы охватывает весь диапазон – от 1 до 4094.

Несмотря на то, что стартовым инициатором рассылки VTP может быть только устройство с ролью VTP Server, пересылать обновление могут любые коммутаторы – т.е. устройству с ролью VTP Client совсем необязательно быть непосредственно подключённому к VTP Server. Клиент, получив от сервера рассылку с новой версией базы данных VLAN'ов, передаст её на все другие транковые порты, за которыми опять же могут быть другие VTP Client'ы и VTP Transparent, и так – до упора. Ограничения стандарта 802.1D на максимальный диаметр «поля» коммутаторов здесь не действуют.

Итак, обновление базы данных – штука достаточно несложная. Вы – VTP Server, на Вас обновлена БД VLAN'ов, Вы разослали во все транковые порты анонс, все коммутаторы, поддерживающие VTP, его прочитали, и, если они VTP Client или VTP Server, то применили к себе и отправили дальше, а если VTP Transparent – то не применили к себе, но отправили дальше. Какая база данных актуальнее определяется при помощи системы версий.

У каждой базы VLAN'ов будет своя версия. В случае, если устройством получено обновление, которое старше по номеру, чем имеющаяся БД, то имеющаяся БД заменяется новой. Целиком, без всяких join/merge.

### 2.7.6 VTP Pruning

Задача функции pruning – каждый коммутатор будет «считать» фактически используемые VLAN'ы, и в случае, когда по VTP приходит неиспользуемый VLAN, уведомлять соседа, что этот трафик не имеет смысла присылать. Под этот механизм будут подпадать только первые 1000 VLAN'ов, исключая самый первый (т.е. pruning работает только для VLAN'ов с номерами от 2 до 1001). Более того, под pruning будет подпадать только уникастовый и неизвестный мультикастовый трафик, поэтому, к примеру, BPDU протоколов семейства STP фильтроваться не будут.

Т.е. допустим, у нас есть два коммутатора – А и В. Коммутатор А имеет роль VTP Server, а В – VTP Client. Между ними – транковый канал, 802.1Q. На коммутаторе В включен vtp pruning. Допустим, на коммутаторе А в базу VLAN добавлены VLAN 10 и VLAN 20. Соответственно, коммутатор А уведомит по протоколу VTP своего соседа – В – о новой ревизии базы VLAN'ов. Сосед В добавит эти VLAN'ы в базу и теперь, когда подключенный к коммутатору А клиент, например, передаст бродкаст в VLAN'e 10, этот бродкаст дойдёт и до коммутатора В. Невзирая на то, что у коммутатора В может вообще не быть ни одного порта и интерфейса в VLAN 10, а также не быть других транков (т.е. трафик 10-го VLAN'a коммутатору В совсем не нужен). Вот в данном случае механизм pruning сможет сэкономить полосу пропускания канала между коммутаторами А и В просто не отправляя трафик неиспользуемого VLAN'a коммутатору В.

### 2.7.7 Базовая настройка протокола VTP

Первым делом необходимо нарисовать топологию сети, в которой собираетесь применять протокол VTP. Посмотрите, какие версии протокола поддерживаются устройствами (обычно везде есть VTPv2). Выберите устройство, которое будет сервером (ему не надо быть каким-то особо быстрым, специфической нагрузки на VTP Server нет, ему лишь желательно обладать максимальным uptime – временем бесперебойной работы). Если Вы не хотите использовать VTP (например, из соображений безопасности) – тогда просто переведите все устройства в режим VTP Transparent (либо off, если поддерживается оборудованием и ОС).

Настройка имени домена VTP:

```
host(config)#vtp domain имя_домена
```

Стереть имя домена штатно нельзя, только сменить.

Настройка пароля VTP:

```
host(config)#vtp password пароль
```

Пароль можно сбросить на пустой, если ввести команду `no vtp password`. Пароль VTP хранится небезопасно (у VTP Server – в файле `vlan.dat`, у VTP Transparent – в NVRAM), поэтому если пользуетесь VTP, делайте такой пароль, который более нигде не дублируется, т.к. получить пароль VTP – относительно несложно. Всё, от чего защищает этот пароль – это, например, случайное добавление в сеть неправильно настроенного коммутатора и последующие проблемы. Пароль VTP не защищает передаваемую между коммутаторами информацию.

Настройка версии VTP:

```
host(config)#vtp version версия
```

Настройка VTP pruning:

Включение выполняется командой:

```
host(config)#vtp pruning
```

а выключение – `host(config)#no vtp pruning`

## Настройка режима VTP:

```
host(config)#vtp mode режим
```

где режим – это server, client, transparent или off. Режим off получится поставить только на устройствах, поддерживающих VTPv3; на коммутаторах, которые поддерживают только VTPv1 и VTPv2 отключить протокол нельзя.

### 2.7.8 Расширенная настройка протокола VTP

Коммутатор, находящийся в режиме VTP Server, хранит информацию в своей флэш-памяти. Если потенциальных мест хранения несколько, то нужное можно указать в явном виде, командой

```
host(config)#vtp file имя_файловой_системы
```

Для VTP Client и VTP Transparent это не имеет особого смысла. Также имеется возможность упростить выявление и устранение причин неисправностей, указав в явном виде интерфейс, с которого будет браться IP-адрес, пишущийся в результатах вывода команды show vtp status. То есть это влияет на выбор того адреса, который будет указываться у клиентов в строчке вида

```
Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
```

Примечание: Если в выводе команды show vtp status ниже этой строчки есть что-то вида Local updater ID is 0.0.0.0 (no valid interface found), то на устройстве нет рабочих IPv4-интерфейсов, и данная команда не имеет смысла – надо вначале сделать хотя бы один интерфейс, с которого можно будет забрать IP-адрес для идентификации VTP-устройства.

Делаться это будет такой командой:

```
host(config)#vtp interface интерфейс
```

где *интерфейс* – это, например, loopback 0.

### 2.7.9 Устранение неисправностей (troubleshooting) протокола VTP

Неисправностей в VTP может быть очень много. Рассмотрим основные из них.

Проверка каналов между коммутаторами:

- проверьте физическую доступность интерфейсов;
- проверьте корректность режима дуплекса и скорости;
- проверьте, что корректно согласовался транк;
- проверьте, совпадают ли native vlan'ы.

Проверка настройка VTP:

- коммутаторы должны быть непосредственно подключены друг к другу;
- должен быть хотя бы один VTP Server;
- версии VTP, а также имя домена и пароль должны быть идентичны у всех устройств.

Проблема добавления нового коммутатора: заключается в следующем: новый коммутатор при добавлении делает следующее – он слушает трафик VTP и при получении первого же advertisement берёт из него настройки (имя домена и пароль). Настройка коммутатора по умолчанию – это режим VTP Server, Вы сразу можете на нём создавать VLAN'ы, в случае VTP Client это было бы невозможно).

### 2.7.10 Протокол VTPv3

Третья версия протокола принесла множество изменений и нововведений.

Протоколы VTP первой и второй версии поддерживали только первую тысячу VLAN'ов. VTPv3 работает со всем диапазоном, от 1 до 4094го.

Если Вы использовали private VLAN (т.е. назначали порты как promiscuous / isolated / community и определяли их в отдельные группы), то VTP не работал с этим механизмом. Третья версия работает.

Теперь через VTP можно обмениваться данными не только БД VLAN'ов, но и базу маппингов MST, что значительно упрощает конфигурирование 802.1s

Первое, что нужно сделать, чтобы включить поддержку VTPv3 – задать имя VTP-домена в явном виде. В предыдущих версиях протокола это было не нужно – коммутатор стартово обладал именем VTP-домена «NULL» и менял его на другое, получив первое VTP-сообщение. Теперь же до включения протокола VTPv3 Вы должны задать не-дефолтное имя VTP домена в явном виде. Команда, та же

```
host(config)#vtp domain имя_vtp_домена
```

Второе – надо включить поддержку 802.1t, называемого чаще extended system-id. Соответствие данному стандарту будет подразумевать, что в параметре BID во всех BPDU будет не два поля – приоритет и Base MAC, а три – приоритет, VLAN ID и Base MAC. То есть, если без 802.1t на приоритет отдавалось 16 бит, то после его включения формат bridge ID (BID) будет выглядеть так:

- 4 бита на приоритет;
- 12 бит на VLAN ID;
- 48 бит на Base MAC.

Включается поддержка 802.1t командой `host(config)#spanning-tree extend system-id`.

После этого уже можно переключаться – `host(config)#vtp version 3`. Убедиться, что переключение произошло, можно сделав `host#show vtp status` и увидев в первых строчках такое:

```
VTP Version capable : 1 to 3
VTP version running : 3
```

### 3 Задания для самостоятельного выполнения

#### Задание №1. Изучение работы протокола VTP

По умолчанию коммутаторы являются серверами. Если коммутатор в серверном режиме отправляет обновление с номером версии, превышающим текущий номер версии, все коммутаторы изменяют свои базы данных в соответствии с новым коммутатором.

Настройте протокол VTP между коммутатором **Switch-Server** и **Switch-Client1**. Затем добавьте коммутаторы **Switch-Transparent** и **Switch-Client2** и настройте их соответствующим образом (см. рисунок 3.1). При каждом добавлении нового коммутатора или новой команды необходимо просматривать текущую конфигурацию VLAN коммутаторов с помощью команды `show vlan`.

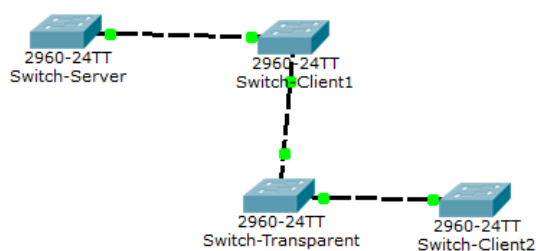


Рисунок 3.1 – Изучение работы протокола VTP

При добавлении нового коммутатора в существующий домен VTP выполните следующие действия:

1. Настройте протокол VTP в автономном режиме.
2. Проверьте конфигурацию VTP.
3. Перезагрузите коммутатор.

#### Задание №2. Настройка VLAN

Составить в рабочем окне Packet Tracer схему, изображенную на рис. 3.2.

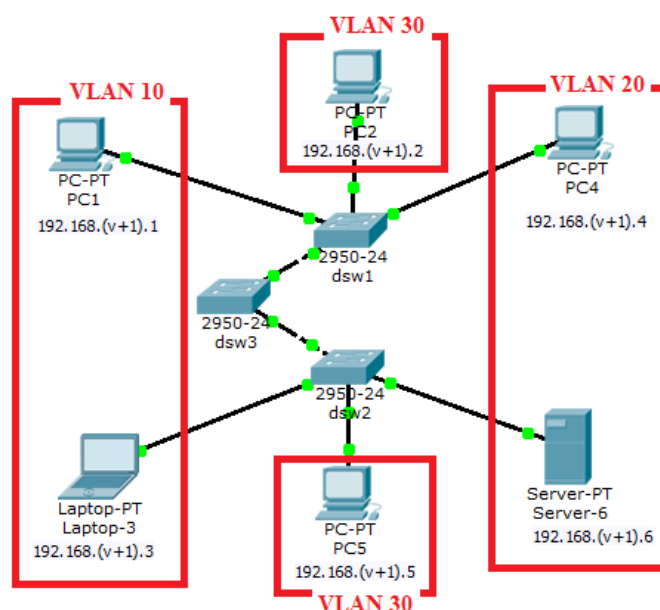


Рисунок 3.2 – Схема для изучения работы протокола VTP

Настроить VLAN на коммутаторах в соответствии с вариантом (v – номер по списку в журнале) и используя протокол VTP (как Вы считаете, какой коммутатор должен остаться в режиме сервера?). Условием проверки является отсутствие связи между хостами, принадлежащими разным VLAN.

После настройки VLAN посмотрите текущую конфигурацию сети командами: `show running-config`, `show vlan`, `show vlan brief`, `show mac address-table`. Результат приведите в отчет.

### Задание №3. Настройка interVLAN routing с помощью маршрутизатора

Возьмите за основу топологию сети из задания №2. Измените ip-адреса хостов в соответствии с вариантом (v – номер по списку в журнале) и добавьте роутер, как показано на рис. 3.3. Хосты подсети  $192.168.(v+1).0$  принадлежат VLAN 10, хосты подсети  $192.168.(v+2).0$  – VLAN 20, а хосты подсети  $192.168.(v+3).0$  – VLAN 30. Настройте маршрутизацию между VLAN 10, 20 и 30. Условием проверки является наличие связи между хостами, принадлежащими разным VLAN.

Просмотрите текущую конфигурацию сети командами: `show vlan`, `show vlan brief`, `show mac address-table`, `show ip route`. Результат приведите в отчет.

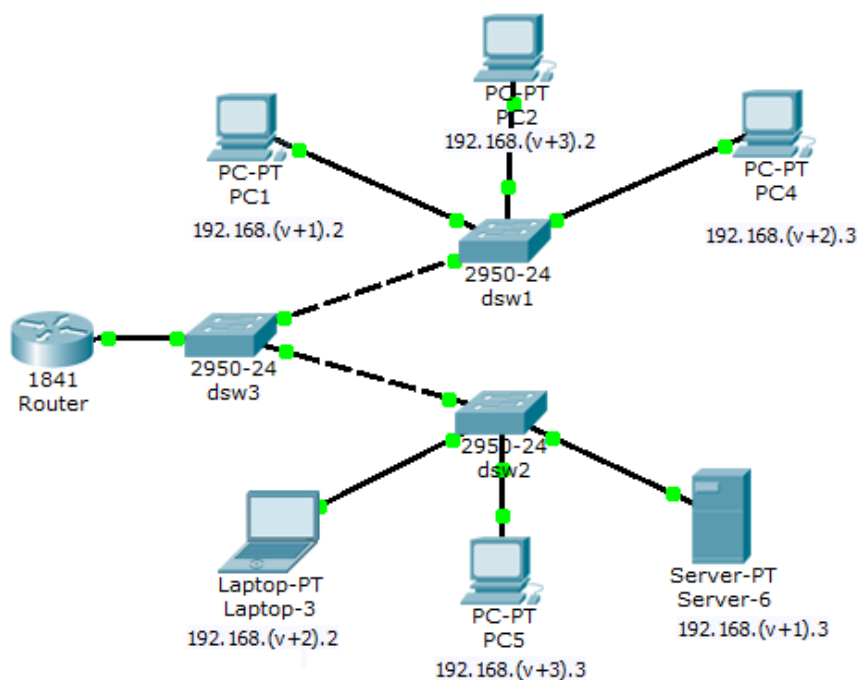


Рисунок 3.3 – InterVlan routing на маршрутизаторе

### Задание №4. Настройка interVLAN routing с помощью L3-коммутатора

Возьмите за основу топологию сети и задания №3, только вместо маршрутизатора и L2-коммутатора dsw3 поставьте L3-коммутатор. Условием проверки является наличие связи между хостами, принадлежащими разным VLAN.

Просмотрите текущую конфигурацию сети командами: `show vlan`, `show vlan brief`, `show mac address-table`, `show ip route`. Результат приведите в отчет.

### **3 Описание лабораторной установки**

В качестве лабораторного стенда используется персональный компьютер с установленной программой Cisco Packet Tracer. Работа с этим пакетом детально описана в методических указаниях (часть 1) лабораторная работа №1.

### **4 Программа выполнения работы**

4.1 Изучить теоретический материал, относящийся к разделу «Локальные компьютерные сети». Особое внимание следует уделить подразделу «Виртуальные локальные сети» и устройству и конфигурации коммутаторов. (Выполняется в процессе домашней подготовки).

4.2 Построить в окне эмулятора Packet Tracer локальную сеть на основе одного коммутатора. Задать узлам сети IP-адреса. Количество серверов и рабочих станций определяется вариантом задания (Приложение А.1).

4.3 Исследовать достижимость сетевых узлов путем их пингования. Результаты пингования сохранить для отчета.

4.4 Разделить сеть, построенную на этапе 4.2, на виртуальные сети способом группирования портов. Количество коммутаторов, виртуальных сетей и рабочих станций в виртуальных сетях определяется вариантом задания (Приложение А.2).

4.5 Исследовать пингованием достижимость сетевых узлов внутри каждой из виртуальных сетей и между виртуальными сетями. После настройки VLAN посмотреть текущую конфигурацию сети командами: `show running-config`, `show vlan`, `show vlan brief`, `show mac address-table`. Результаты пингования и просмотра конфигурации включить в отчет.

4.6 Повторить п.4.4 и 4.5 при условии, что в сети существует два коммутатора. Виртуальные сети включают компьютеры, соединенные как с первым и так и со вторым коммутаторами. Количество линий связи между коммутаторами равно количеству виртуальных сетей.

4.7 Повторить п.4.6 при использовании транковых соединений между коммутаторами.

4.8 Построить сеть, изображенную на рисунке 2.8 и сконфигурировать ее так, чтобы обеспечить обмен пакетами между виртуальными сетями и исследовать корректность функционирования сети.

### **4 Содержание отчета**

1. Титульный лист.
2. Исходные данные в соответствии с индивидуальным вариантом.
3. Описание всех использованных команд.
4. Скриншоты полученных топологий и осуществленных настроек.
5. Выводы.

## 5 Контрольные вопросы

1. Что такое виртуальные локальные сети и зачем они применяются?
2. Зачем применяется разбиение сети на VLAN-ы?
3. Расскажите, как функционирует неуправляемый коммутатор после включения питания.
4. Что означают понятия «агрегирование» и «зеркалирование» портов?
5. В чем состоит отличие портов доступа от магистральных (транковых) портов?
6. С какой целью в коммутационном оборудовании установлен консольный порт?
7. Как обеспечивает передача пакетов между изолированными виртуальными сетями?
8. В чем состоит суть процедуры interVLAN routing?
9. Сравните способы организации interVLAN routing?
10. С какой целью используется протокол VTP?
11. Поясните принцип работы протокола VTP?

## Приложение А.1 - Таблица вариантов

Вариант	Количество РС	Количество серверов	Количество ноутбуков
1	2	2	1
2	4	1	2
3	6	1	1
4	4	2	2
5	5	1	1
6	3	3	4
7	6	2	3
8	4	3	3
9	3	3	3
10	8	2	2

## Приложение А.2 - Таблица вариантов

Вариант	Количество				
	РС	серверов	ноутбуков	коммутаторов	VLAN
1	2	2	1	2	2
2	4	3	2	3	3
3	6	3	3	3	3
4	4	2	2	2	2
5	5	3	4	3	3
6	5	3	4	3	4
7	6	2	3	2	3
8	4	3	3	3	3
9	3	3	3	2	2
10	8	2	2	3	4

## Лабораторная работа №2

### Исследование способов динамической маршрутизации пакетов в компьютерных сетях

#### 1 Цель работы

Углубление теоретических знаний в области архитектуры компьютерных сетей, исследование способов статической и динамической маршрутизации, приобретение навыков составления сценариев конфигурации телекоммуникационного оборудования, а также моделирования локальных сетей в среде симулятора Cisco Packet Tracer.

#### 2 Краткие теоретические сведения

##### 2.1 Виды маршрутизации в компьютерных сетях

**Маршрутизация (Routing)** — процесс определения наиболее эффективного маршрута (последовательности узлов) прохождения пакетов по сети. В компьютерных сетях маршрутизация осуществляется устройствами третьего уровня — маршрутизаторами называемыми также сетевыми шлюзами. Маршрутизатор имеет несколько (обычно 2-4) интерфейсов (портов), каждый из которых имеет свой MAC- и IP адрес. В принципе, маршрутизация может выполняться и компьютерами общего назначения при наличии двухпортовой сетевой карты и установке соответствующих программ.

Несмотря на то, что маршрутизатор функционирует на 3-м уровне модели OSI, т.е. анализирует заголовки IP-пакетов, он работает также на физическом и канальном уровнях. На физическом уровне интерфейс маршрутизатора усиливает и ограничивает принимаемые сигналы, стробирует их и передает модулю канального уровня. На канальном уровне из потока битов составляется кадр данных, выполняется проверка на отсутствие ошибок и сравнивается MAC-адрес устройства назначения с аппаратным адресом интерфейса. При совпадении этих адресов инкапсулированный в кадр IP-пакет передается модулю сетевого уровня. Кроме этого, в маршрутизаторе также имеется модуль разрешения адресов, формирующий ARP-таблицу, в которую он записывает соответствие MAC- и IP-адресов и через какой интерфейс нужно передавать. ARP-таблица у каждого сетевого интерфейса своя.

Маршрутизация осуществляется в соответствии с протоколами маршрутизации, которые регламентируют процесс обмена служебной информацией между маршрутизаторами для формирования и поддержки таблиц маршрутизации, а также обновления записей в таблицах при возникновении изменений в сети. После подачи питания на маршрутизатор он сразу же начинает формировать таблицу маршрутизации. Но запись о возможных путях достижения существующих сетей маршрутизатор вначале может внести только о сетях, с которыми он связан напрямую (состояние *connected*).

При задании пути прохождения пакетов по инфокоммуникационной сети используются два вида маршрутизации: статическая и динамическая. При **статической маршрутизации** маршруты указываются администратором сети в процессе ручной конфигурации маршрутизаторов. Путь прохождения пакетов в процессе всего периода функционирования сети остается неизменным. Протоколы маршрутизации при этом не используются. Статическая маршрутизация применяется обычно на небольших сетях, а также в целях дополнительного обеспечения безопасности. К достоинствам статической маршрутизации следует также отнести ее стабильность при наличии внешних угроз и минимизация использования аппаратных ресурсов маршрутизатора для обслуживания таблицы маршрутизации.

При **динамической маршрутизации** путь прохождения пакетов может изменяться, в зависимости от состояния сети. При этом маршрутизатор выбирает оптимальный путь из нескольких доступных путей. В процессе реализации динамической маршрутизации периодически осуществляется обмен маршрутной информацией между соседними маршрутизаторами, в ходе которого они сообщают друг другу, какие сети в данный момент доступны через них. Полученная информация обрабатывается маршрутизатором и помещается в таблицу маршрутизации. Динамическая маршрутизация осуществляется по стандартным правилам, определяемым протоколами маршрутизации.

Глобальная компьютерная сеть представляет собой объединение отдельных сетей, называемых автономными системами AS (*Autonomous System*), к которым относятся сети, управляемые одним или несколькими операторами, использующими единую политику маршрутизации. При этом отдельно регламентируется маршрутизация как внутри автономных систем, так и маршрутизация между автономными системами.

Протоколы для работы внутри автономных систем называют внутренними (внутридоменными) протоколами шлюзов IGP (*Interior Gateway Protocols*), а протоколы для работы между автономными системами — внешними (междоменными) протоколами шлюзов EGP (*Exterior Gateway Protocols*). К внутренним протоколам относятся RIP, RIP v2, IGRP, EIGRP, OSPF и IS-IS, а к внешним — протоколы EGP3 и BGP4.

Маршрутизатор выбирает оптимальный маршрут на основе некоторой метрики. В качестве метрики в протоколах маршрутизации наиболее часто используются **пропускная способность** (*Bandwidth*), **задержка** (*Delay*) — время прохождения пакета от источника до получателя, **количество переходов** (*Hop* от маршрутизатора к маршрутизатору), через которые пакет должен пройти на пути к адресату назначения, **стоимость** (*Cost*) — обобщенный параметр затрат на передачу пакета к адресату назначения (часто, с целью упрощения, стоимость задается в виде величины, обратной пропускной способности).

Если от маршрутизатора к сети назначения существует много маршрутов, и все они используют один протокол маршрутизации, лучшим считается маршрут с минимальной метрикой. В случае использования в сети нескольких различных протоколов маршрутизации для выбора маршрута применяются адми-

нистративные расстояния, которые назначаются маршрутам операционной системой маршрутизатора.

Одним из важнейших качественных показателей компьютерной сети является **сходимость** (конвергенция). Под сходимостью сети понимают состояние сети, когда все маршрутизаторы будут иметь согласованную информацию о сетевых соединениях. Параметром сходимости является время сходимости (конвергенции), оцениваемое временем, которое требуется маршрутизаторам, чтобы осуществить обмен маршрутной информацией, вычислить лучшие пути и обновить свои таблицы маршрутизации после обрыва линий или других изменений в сети.

На время сходимости влияют ряд факторов:

- расстояние до точки изменения в сети;
- число маршрутизаторов, использующих динамические протоколы;
- пропускная способность и загрузка каналов связи;
- загрузка маршрутизаторов.

При внутренней динамической маршрутизации используются два вида маршрутизации: дистанционно-векторная и маршрутизация на основе учета состояния линий связи.

### **Дистанционно-векторная маршрутизация**

Дистанционно-векторная маршрутизация базируется на алгоритме Беллмана-Форда. В качестве метрики сети в самом простом случае используется количество переприемов (хопов) на пути от источника до получателя. В соответствии с этим алгоритмом каждый маршрутизатор через фиксированные промежутки времени передает широковещательно соседним маршрутизаторам всю свою таблицу маршрутизации.

Соседний маршрутизатор, получая широковещательное сообщение, сравнивает маршрутизационную информацию со своей текущей таблицей маршрутов. В нее добавляются маршруты к новым сетям или маршруты к известным сетям с лучшей метрикой. Происходит удаление несуществующих маршрутов. Маршрутизатор добавляет свои собственные значения к метрикам полученных маршрутов. Новая таблица маршрутизации снова распространяется по соседним маршрутизаторам. Этот процесс схематично изображен на рисунке 2.1.

В каждой строке таблицы содержится IP-адрес сети, интерфейс, через который достижима эта сеть и метрика пути. При непосредственном подключении к сети метрика равно нулю.

К недостатку дистанционно-векторных алгоритмов относятся:

- относительно большое время конвергенции;
- алгоритмы хорошо работают только в относительно небольших компьютерных сетях в связи с тем, что максимальное количество переприемов ограничивается 15-ю;
- перегрузка сети широковещательным трафиком по причине регулярного (через каждые 30 с) обмена между маршрутизаторами векторами расстояний.

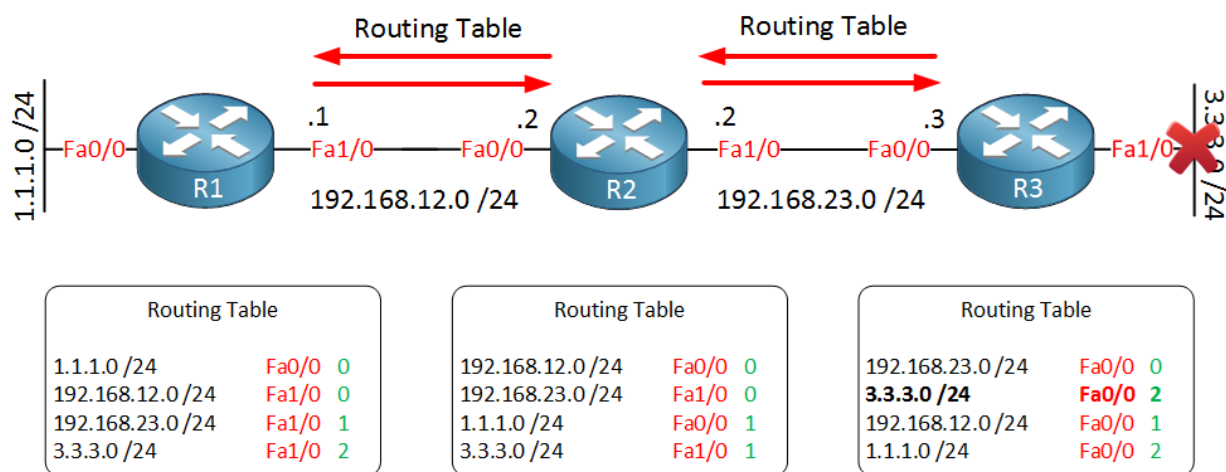


Рисунок 2.1 – Дистанционно-векторная маршрутизация

Самым распространенным представителем дистанционно-векторного алгоритма является открытый протокол маршрутной информации **RIP** (*Routing Information Protocol*) и проприетарный (фирменный) протокол консорциума Cisco **EIGRP** (*Enhanced Interior Gateway Routing Protocol*).

## 2.3 Протоколы маршрутизации на основе учета состояния линий

Коренное отличие протоколов маршрутизации с учетом состояния линий (каналов) от дистанционно-векторных протоколов состоит в следующем:

- 1) в типе информации, которой обмениваются маршрутизаторы: таблицы маршрутизации Distance-Vector и таблицы топологии Link State;
- 2) в процессе выбора лучшего маршрута;
- 3) в количестве информации о сети, которое хранит в памяти каждый маршрутизатор: Distance-Vector знает только своих соседей, Link State имеет информацию обо всей сети;
- 4) рассылка обновления осуществляется только в случае появления изменений, а рассылка полного обновления таблицы состояния выполняется значительно реже (примерно один раз каждые 30 минут).

Протоколы учета состояния линий связи обеспечивают лучшую масштабируемость и сходимость по сравнению с дистанционно-векторными протоколами. Протокол базируется на алгоритме Дейкстры, который часто называют алгоритмом «кратчайший путь – первым» (*Shortest Path First – SPF*). Наиболее типичным представителем является протокол OSPF (*Open Shortest Path First*). Алгоритм работы протокола динамической маршрутизации OSPF основан на использовании всеми маршрутизаторами единой *базы данных*, описывающей, с какими сетями связан каждый маршрутизатор и какова метрика каждой связи.

Маршрутизатор с целью уменьшения размеров таблицы маршрутизации и соответственно времени сходимости (конвергенции), а также снижения нагрузки на центральный процессор строит полную базу данных состояний линий связи обычно не для всей сети, а для некоторой ограниченной области (зоны, англ. *area*). Каждый маршрутизатор затем самостоятельно реализует SPF-

алгоритм с учетом базы данных состояний связи для определения лучшего пути, который заносится в таблицу маршрутов. Эти пути к другим сетям образуют дерево с вершиной в точке данного локального маршрутизатора. Каждый маршрутизатор имеет собственное представление топологии, но при этом все маршрутизаторы используют одну базу данных состояний канала для вычисления кратчайшего пути. Маршрутизаторы извещают о состоянии своих связей всех маршрутизаторов только в своей зоне. Такое сообщение называется извещением о состоянии связи (*Link-State Advertisements, LSA*).

На начальном этапе поступающие LSA пакеты служат для построения базы данных состояний связи. После этого обновление маршрутов производится только при смене состояний связи или, если состояние не изменилось в течение определенного интервала времени. Если состояние связи изменилось, то в этот же момент отправляются пакеты обновления и выполняется частичное обновление таблиц маршрутизации. Сообщения LSA рассылаются всем соседним маршрутизаторам, а каждый маршрутизатор, получивший LSA, производит обновление своей базы данных топологии сети и производит дальнейшую рассылку LSA всем своим соседям. Пакеты обновления содержит не всю таблицу маршрутов, а только сведения о состоянии изменившихся связей. Сообщения LSA имеют порядковые номера, чтобы каждый маршрутизатор мог сравнить порядковый номер, поступившего LSA, с уже имеющимся в его базе данных, и при необходимости обновить ее. В протоколах маршрутизации с учетом состояния канала должно проводиться периодическое обновление записей таблицы топологии для актуализации имеющейся в ней информации. В протоколе OSPF по умолчанию интервал обновления информации таблицы топологии составляет 30 минут.

Протоколы учета состояния связей характеризуются более быстрой сходимостью и лучшим использованием полосы пропускания по сравнению с дистанционно-векторными протоколами, возможность балансировки загрузки. К основным недостаткам протоколов следует отнести повышенные требования к вычислительной производительности маршрутизаторов и сравнительно сложное администрирование.

**Алгоритм динамической маршрутизации SPF реализован в протоколе OSPF.** Это динамический, иерархический протокол состояния связи, используемый для маршрутизации внутри автономных систем. Он базируется на открытых стандартах и был разработан для замены протокола RIP. Кратчайший путь в сети вычисляется по алгоритму Дейкстры. Протокол OSPF может быть настроен на всех типах маршрутизаторов, а также на всех коммутаторах 3-го уровня.

Для уменьшения служебного трафика при рассылке LSA-пакетов выделяется так называемый назначенный маршрутизатор (*Designated Router, DR*). Каждый маршрутизатор сети устанавливает отношения соседства с DR. При обнаружении одним из маршрутизаторов изменения в сети он отправляет сообщение об этом событии только выделенному маршрутизатору по адресу 224.0.0.6, а DR рассылает эту информацию по групповому адресу 224.0.0.5 всем остальным маршрутизаторам сети. Маршрутизатор, на котором активизирован

протокол OSPF, автоматически становится членом группы многоадресной рассылки с адресом 224.0.0.5 и начинает рассылать и получать групповые сообщения OSPF. С целью повышения надежности сети выделяется также резервный назначенный маршрутизатор (*Backup Designated Router*, **BDR**), адрес которого совпадает с адресом DR, т.е. 224.0.0.6. DR и BDR должны иметь полноценное физическое подключение ко всем маршрутизаторам зоны.

Протоколом OSPF предписано в каждом маршрутизаторе создавать 3 таблицы:

**Таблица смежности** или **таблица соседей** (*Adjacency table*) — содержит список соседей и информацию о состоянии всех непосредственно подключенных соседних OSPF маршрутизаторов.

**Топологическая таблица** (*Link State Data Base*, LSDB) — хранит сведения о состоянии всех сетей, маршрутизаторов и их активных интерфейсах в пределах зоны OSPF. Причем все маршрутизаторы одной зоны должны иметь одинаковую таблицу.

**Таблица маршрутизации** (*Route table*) — создается по алгоритму SPF на основе информации из топологической таблицы.

После включения маршрутизатора, настроенного на работу с OSPF, он начинает процесс изучения окружения, проходя несколько фаз инициализации. В начале маршрутизатор отправляет через каждые 10 с Hello-сообщения для определения своих соседей и создания отношений для обмена обновлением маршрутной информацией с ними. Каждый маршрутизатор в результате обмена приветственными сообщениями создает локальную **таблицу соседей**. После завершения установки соседских отношений между смежными OSPF маршрутизаторами начинается обмен топологической информацией. Результатом обмена информацией об элементах топологии является **таблица топологии**. Затем маршрутизаторы запускают процедуру расчета кратчайших путей по алгоритму SPF и формируют **таблицы маршрутизации**.

Для транспортировки пакетов OSPF по сети они инкапсулируются непосредственно в IP-пакеты с указанием номера (89) на протокол последующей обработки.

## 2.4 Особенности конфигурации телекоммуникационного оборудования при использовании протокола OSPF

Для задания динамической маршрутизации используются две основные команды: **router** и **network**. Команда **router** активирует процесс маршрутизации и имеет следующий формат:

```
Router(config)# router protocol PROCESS_NUMBER
```

где **protocol** — любой из протоколов маршрутизации: RIP, IGRP, OSPF и т.п., **PROCESS\_NUMBER** — номер процесса (может быть любой).

При реализации динамической маршрутизации с учетом состояния линий нужно на каждом маршрутизаторе запустить протокол OSPF. Этот процесс осуществляется по команде **router ospf номер-процесса**. Параметр *номер-*

процесса должен быть одинаков на всех маршрутизаторах домена маршрутизации. Чаще всего этот номер устанавливают равным 1.

Для указания сетей, непосредственно подключенных к интерфейсам маршрутизатора, используется команда **network area**. Синтаксис команды **network area** для протокола OSPF имеет вид:

```
(config-router)# network network-address [wildcard-mask] area area-id
```

```
(config-router)# no network network-addressr [wildcard-mask] area area-id
```

Здесь *network-address* – IP-адрес сети, подключенной к интерфейсу маршрутизатора и участвующей в процессе маршрутизации OSPF; *wildcard-mask* – обратная маска, которая указывает с помощью 0 какая часть из указанной сети должна совпадать, а с помощью 1 какая часть сети может быть произвольной; *area-id* – номер зоны OSPF, в которой будет работать интерфейс маршрутизатора. Обычно номер зоны для малых сетей принимается равным 0.

Информация об указанной в команде сети будет передаваться другим маршрутизаторам (при условии, что на маршрутизаторе есть рабочий интерфейс в данной сети). Через интерфейс, находящийся в этой сети маршрутизатор начинает общаться с соседями. Таким образом, необходимо описать на каждом маршрутизаторе все сети, непосредственно подключенные к его интерфейсам.

Для просмотра информации о OSPF маршрутизации применяется команда **show ip ospf interface**, в результате которой для каждого интерфейса выводится вся OSPF информация: IP адрес, область, номер процесса, идентификатор маршрутизатора, стоимость, приоритет, тип сети, интервалы таймера.

По команде **show ip ospf neighbor** выводится важная информация, касающаяся состояния соседей. Вид таблицы изображен на рисунке 2.

R2#show ip ospf neighbor					
Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.0.1	1	FULL/BDR	00:00:37	192.168.0.1	FastEthernet0/0
192.168.3.1	1	FULL/BDR	00:00:37	192.168.1.3	FastEthernet0/1

Рисунок 2.2 – Вид таблицы состояния соседей

**Neighbor ID** – идентификатор соседей (обычно это наибольший из адресов их loopback интерфейсов). Интерфейс loopback — это логический интерфейс внутри маршрутизатора. Он не назначается физическому порту, поэтому его нельзя подключить к другому устройству. Он считается программным интерфейсом, который автоматически переводится в состояние up (активен) во время работы маршрутизатора. На маршрутизаторе можно активировать несколько интерфейсов loopback. IPv4-адрес для каждого интерфейса loopback должен быть уникальным и не должен быть задействован другим интерфейсом.

Поле **Pri** указывает приоритет соседнего маршрутизатора. Маршрутизатор с наивысшим приоритетом становится назначенным маршрутизатором DR (Designated Router). Если приоритеты одинаковы, то маршрутизатор с самым

высоким идентификатором становится назначенным. По умолчанию приоритеты устанавливаются в 1. Состояние **FULL/BDR** показывает, что установлена полная смежность с резервным назначенным маршрутизатором, т.е., когда маршрутизатор имеет в своей базе данных состояний соединений синхронизированные данные; если указано состояние **2WAY/DROTHER**, то это индицирует состояние между обычными соседями. **Dead Time** (мертвое время) — интервал времени, по прохождению которого, сосед считается недоступным, если не было Hello. Если маршрутизатор не получает ни одного пакета в течении Dead-интервала, то считается, что сосед пропал и отношения разрываются, что влечёт за собой потерю связи, отправку LSU, пересчёт топологии и т.д. **Address** — адрес интерфейса удалённой стороны, через который установлено соседство.

Командой **show ip protocols** можно посмотреть с какими параметрами работает протокол OSPF. При необходимости внесения некоторых изменений в конфигурацию процесса маршрутизации OSPF, требуется производить перезапуск процесса маршрутизации. Для этого используется команда **clear ip ospf**.

### 3 Описание лабораторной установки

В качестве лабораторной установки используется персональный компьютер с установленной программой Packet Tracer, позволяющей осуществлять моделирование компьютерных сетей, построенных на оборудовании корпорации Cisco.

## 4. Программа и методические рекомендации выполнения работы

4.1. Повторить теоретический материал по темам: «Маршрутизация пакетов в компьютерных сетях».

4.2. В программе Cisco Packet Tracer построить сеть, изображённую на рисунке 4.1. Выполнить статическую маршрутизацию и проверить взаимным пингованием достижимость PC0 и PC1. Сетевые адреса телекоммуникационного оборудования приведены на рисунке. Статическую адресацию можно задать путем использования графического интерфейса или с использованием интерфейса командной строки (рекомендуется).

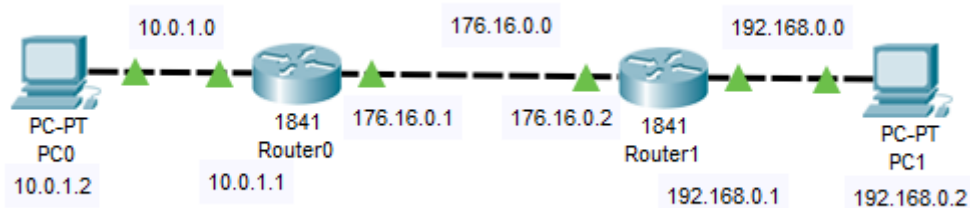


Рисунок 4.1 – Схема фрагмента сети с двумя маршрутизаторами

Задания статической адресации маршрутизатору Router1 с использованием интерфейса командной строки может быть выполнено следующим образом:

```
Router>en
Router#configure terminal
Router(config)#hostname Router1
Router1(config)#ip route 10.0.0.0 255.0.0.0 176.16.0.1
Router1(config)#exit
Router1#
```

Аналогично осуществляется конфигурация Router0 с указанием сети назначения 192.168.0.0 через интерфейс 172.16.0.2. После этого нужно путем поочередного пингования убедиться в доступности удаленных компьютеров.

4.3. В эмуляторе Cisco Packet Tracer построить сеть, изображенную на рисунке 4.2 (аналогичную предыдущей схеме), настроить динамическую маршрутизацию с помощью протокола OSPF и обеспечить возможность взаимодействия конечных устройств, входящих в подсети PC0-PC1.

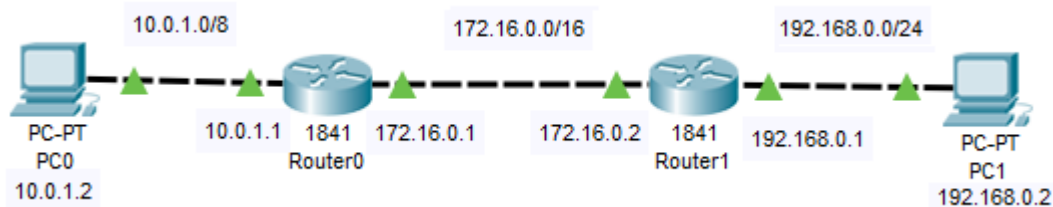


Рисунок 4.2 – Схема фрагмента сети с динамической маршрутизацией

После задания сетевых адресов и сетевых масок всем интерфейсам телекоммуникационных устройств, а также адреса шлюза (Gateway) следует осуществить настройку маршрутизаторов для выполнения ими динамической маршрутизации по протоколу OSPF. Конфигурация маршрутизатора Router0 выполняется следующим образом:

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname Router0
Router0(config)#router ospf 1
Router0(config-router)#network 10.0.0.0 0.255.255.255 area 0
Router0(config-router)#network 172.16.0.0 0.0.255.255 area 0
Router0(config-router)#exit
Router0(config)#
```

4.4. В эмуляторе Cisco Packet Tracer построить сеть, изображенную на рисунке 5.3, настроить динамическую маршрутизацию с помощью протокола OSPF и обеспечить возможность взаимодействия конечных устройств, входящих в подсети PC0-PC1, PC2-PC3, PC4-PC5 и PC6-PC7, между собой.

Планирование адресного пространства необходимо выполнить самостоятельно. Результат следует занести в таблицу, представленную ниже.

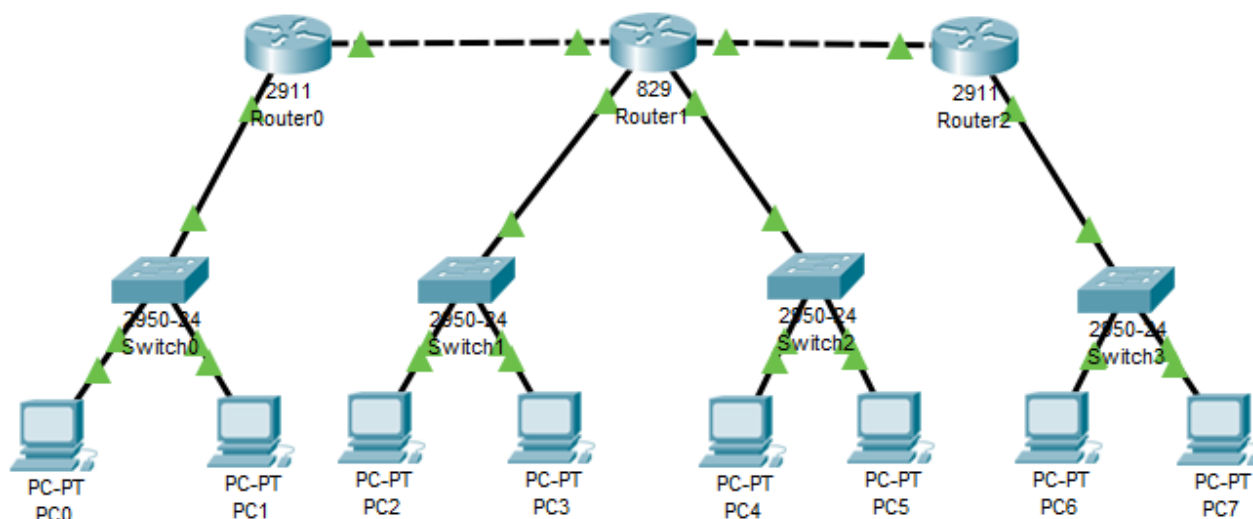


Рисунок 4.3 – Объединение локальных подсетей с помощью маршрутизаторов

Таблица сетевых адресов

Устройство	Интерфейс	IP-адрес	Маска	Шлюз
R0				
R1				
R2				
PC0				
PC1				
PC2				
PC3				
PC7				

4.5. Исследовать процессы обмена пакетами в сети в реальном режиме и режиме симуляции.

4.6. Исследовать настройки динамической маршрутизации:

просмотреть содержимое таблицы IP маршрутизации с помощью команды `show ip route`;

на каждом компьютере выполнить команду трассировки `tracert` других компьютеров;

исследовать параметры протокола OSPF с помощью команд `show ip ospf interface`, `show ip ospf database` и `debug ip ospf events`.

## **5. Содержание отчета**

- 6.1 Титульный лист.
- 6.2 Исходные данные в соответствии с индивидуальным вариантом.
- 6.3 Описание всех использованных команд.
- 6.4 Скриншоты топологии, реализованных настроек строек и результатов исследования функционирования сети.
- 6.5 Выводы.

## **6. Контрольные вопросы**

- 7.1 Что такое автономная система?
- 7.2 Что такое метрика связи и как она определяется?
- 7.3 Какие существуют классы протоколов динамической маршрутизации?
- 7.4 Объясните работу дистанционно-векторных протоколов.
- 7.5 Объясните работу протоколов состояния связи.
- 7.6 В чём преимущества и недостатки дистанционно-векторных протоколов и протоколов состояния связи?
- 7.7 Как узнать, какие протоколы маршрутизации запущены на маршрутизаторе?
- 7.8 Перечислите основные этапы установки маршрутизатора.
- 7.9 Опишите процесс функционирования протокола OSPF.
- 7.10 Как на маршрутизаторе запустить и настроить протокол маршрутизации OSPF?
- 7.11 Как получить информацию об источнике маршрута удаленных сетей?
- 7.12 Какие возможны ошибки при настройке динамической маршрутизации?
- 7.13 Как выявлять ошибки настройке динамической маршрутизации?
- 7.14 Как просмотреть таблицу соседних устройств? Какую информацию о ней можно получить?

## Лабораторная работа №3

**Исследование способов назначения списков контроля доступа в локальных компьютерных сетях****1 Цель работы**

Исследование методов контроля доступа к сетевым ресурсам и способов составления списков ограничения доступа, приобретение практических навыков составления стандартных и расширенных списков доступа, а также конфигурации сетевого оборудования.

**2 Краткие теоретические сведения****2.1 Общая характеристика списков контроля доступа**

В процессе реализации политики сетевой безопасности одной из важнейших задач является возможность закрытия доступа для некоторых пакетов. Для отсеечения нежелательных пакетов широко применяются списки доступа **ACL (Access Control Lists)**, которые являются своеобразными фильтрами пакетов. Они позволяют запретить или разрешить определенным хостам доступ к ресурсам сети. Например, в корпоративной сети администраторы могут запретить доступ к внутреннему серверу или в Интернет определенным пользователям, а остальным наоборот – разрешить.

Списки доступа позволяют фильтровать трафик на входе и выходе интерфейсов маршрутизаторов. На входе весь поступающий трафик подвергается фильтрации. Нежелательные пакеты отбрасываются и уже только потом остальные пакеты маршрутизируются. Если же ACL настроены на выходе интерфейса, то трафик фильтруется сразу же после процесса маршрутизации. Списки доступа позволяют использовать маршрутизатор как межсетевой экран (брандмауэр) для запрета или ограничения доступа к внутренней сети из внешней сети, например, Интернет. Брандмауэр, как правило, помещается в точку соединения между двумя сетями.

Списки доступа содержат набор инструкций (директив, предписаний) какие порты и адреса блокировать, а какие наоборот разрешить. Этих инструкций в списке может быть от единиц до нескольких десятков. В конце списка всегда содержится неявная инструкция по блокировке всего трафика. Данная инструкция добавляется автоматически самой системой. В настройках она не видна, но нужно учитывать ее наличие.

На настоящее время существуют 3 типа списков доступа: 1) стандартные; 2) расширенные и 3) именованные списки.

**Стандартные списки** позволяют проверять только IP адрес отправителя. Стандартные списки доступа рекомендуется устанавливать как можно ближе к

отправителю. Стандартные и расширенные списки доступа обязательно нумеруются. Номера стандартных списков могут принимать значение от 0 до 99.

**Расширенные списки управления доступом** *extended ACL* (*extended Access Control List*) используются чаще, чем стандартные, поскольку они обеспечивают большие возможности контроля. Расширенные списки предписывают проверку как адреса источника, так и адреса получателя. Они могут также проверять конкретные протоколы, номера портов и другие параметры. Это придает им большую гибкость в задании проверяемых условий. Пакету может быть разрешена отправка или отказано в передаче в зависимости от того, откуда он был выслан и куда направлен. Расширенным спискам доступа назначаются номера от 100 до 199.

**Именованные списки** аналогичны стандартным и расширенным ACL, но вместо нумерации используются названия списков. Стандартные и расширенные списки редактировать нельзя. К примеру, нельзя в середину списка вставить команду или удалить ее. Для этого нужно сначала **деактивировать список на самом интерфейсе**, а затем полностью его удалить и настроить заново. Именованные списки позволяют редактировать вновь созданные списки. Все введенные команды нумеруются, что позволяет легко добавлять и удалять команды.

## 2.2 Правила составления списков доступа

Правила построения и назначения списков доступа для различных протоколов имеют свою специфику, однако, можно выделить два этапа работы с любыми списками доступа. Сначала, создается список доступа, а затем выполняется привязка его к соответствующему интерфейсу, линии связи или логической операции, выполняемой маршрутизатором (роутером).

Каждое предписание в списке доступа записывается отдельной строкой. Список доступа в целом представляет собой набор строк с директивами, имеющих один и тот же номер (или имя). Порядок задания директив в списке играет важную роль. Проверка пакета на соответствие списку производится последовательным применением предписаний из данного списка (в том порядке, в котором они были внесены). В конце каждого списка системой IOS добавляется неявное правило, состоящее в том, что если пакет удовлетворяет какому-либо предписанию, то дальнейшие проверки его на соответствие следующим директивам в списке НЕ ПРОИЗВОДЯТСЯ. Таким образом, пакет, который не соответствует ни одному из введенных предписаний, отвергается.

Для одного списка можно определить несколько директив. Каждая из них должна ссылаться на имя или на номер списка для того, чтобы все они были связаны с одним и тем же списком. Количество директив может быть произвольным, и ограничено лишь объемом имеющейся памяти. Однако, чем больше в списке директив, тем труднее понять логику работы списка и контролировать ее. Поэтому рекомендуется тщательно заносить всю информацию о списках в специальный журнал.

Как уже упоминалось выше, порядок строк в списке доступа очень важен, поскольку невозможно изменить этот порядок или исключить какие-либо строки из существующего списка доступа. По этой причине целесообразно предварительно создавать списки доступа (например, на tftp-сервере) и загружать их целиком в маршрутизатор, а не пытаться редактировать их на маршрутизаторе. Если список доступа с данным номером (именем) существует, то строки списка с тем же номером (именем) будут добавляться к существующему списку в конец его.

Списки управления доступом представляют собой перечень особых **директив** (предписаний): «**разрешить**» (*permit*) и «**запретить**» (*deny*). Эти директивы применяются к адресам или протоколам верхних уровней (3-7). Предписание «разрешить» означает, что все пакеты, отвечающие определенным условиям, будут пропущены, т.е. им будет разрешено дальнейшее перемещение по сети. Предписание «запретить» указывает, что пакет, имеющий определенные характеристики, необходимо удалить. Списки доступа могут применяться для запрещения продвижения пакетов через определенный интерфейс маршрутизатора в ту или другую сторону, для ограничения доступа некоторых пользователей и устройств к сетевым ресурсам, для указания способа шифрования, а также для указания приоритетности обработки пакетов.

Каждая из директив в списке доступа читается процессором маршрутизатора по порядку, т.е. очередной пакет, проходящий через соответствующий порт, будет последовательно сравниваться со всеми критериями (адресом источника, адресом получателя или номером порта) **в списке доступа с начала списка до конца**. Если пакет не соответствует условию первой директивы, то он проверяется на соответствие второй директиве из списка управления доступом. А если параметры пакета соответствуют следующему условию, которое представляет собой директиву разрешения доступа, то ему разрешается отправка на интерфейс получателя. Таким образом, при первом обнаружении соответствия остальные директивы не рассматриваются. Поэтому, если была записана директива, разрешающая передачу всех данных, то все последующие директивы не проверяются. Следует особо подчеркнуть, что **в конце каждого списка выполняется неявное правило "deny all"** (запретить все), поэтому при назначении списков на интерфейс нужно следить, чтобы явно разрешить все виды необходимого трафика через интерфейс (не только пользовательского, но и служебного, например, обмен информацией по протоколам динамической маршрутизации).

Для создания **стандартного списка доступа** для маршрутизаторов Cisco применяется команда **access-list**, которая вводится в следующем формате:

```
access-list <номер_списка> {deny|permit} <адрес отправителя> [маска шаблона адреса] [log]
```

**Маска шаблона** (англ. *wildcard mask*) указывает маршрутизатору на те биты в шаблоне адреса отправителя, которые следует сравнивать с поступившим в порт маршрутизатора адресом отправителя, и те, которые нужно проигнорировать. Как и маска в схеме адресации протокола IP, маска шаблона в списке доступа состоит из 32-х битов, записанных в точечно-десятичной форме.

Например, маска шаблона 0.0.0.255 соответствует двоичному представлению 00000000.00000000.00000000.11111111. Однако запись маски шаблона в списках доступа, в отличие от метода записи маски адреса на сетевых интерфейсах, **записана инверсно**, т.е. единицами отмечены биты адреса, которые НЕ будут проверяться. Нулевые биты в маске списка доступа предписывают маршрутизатору необходимость сравнения соответствующих битов IP-адреса в проверяемом пакете с аналогичными битами в шаблоне адреса. Соответственно, единичные биты указывают на то, что сравнение производить не нужно. Таким образом, маска 0.0.0.0 вынуждает маршрутизатор сравнивать все 32 бита адреса пакета на соответствие их битам шаблона, заданного в списке доступа.

Ключевое слово **"log"** инициирует выдачу записи о совпадении пакета с данным предписанием на консоль и в системный лог-файл. Часто используемое описание фильтра, которому удовлетворяет любой адрес 0.0.0.0 255.255.255.255, имеет специальное обозначение "any":

```
access-list <access-list-number> {deny | permit} any
```

В расширенном списке перед полем адреса источника можно указывать **тип протокола**, а после адреса источника указываются (при необходимости) адрес хоста назначения и порт. Общий формат расширенного списка доступа имеет следующий вид:

```
access-list номер-списка {permit | deny} {протокол} {адрес источника}
[маска-источника] [адрес получателя] [маска-получателя] [оператор номер порта]
[established] [log]
```

Параметры списка могут принимать значения:

оператор — *it*, *gt*, *eq*, *neq* (меньше, чем, больше, чем, равно, не равно);  
*established* — разрешает прохождение TCP-потока если он использует установленное соединение (т. е. если бит АСК в заголовке сегмента установлен). Частные случаи записи могут иметь вид:

```
access-list access-list-number {deny|permit} протокол any any
```

или

```
access-list access-list-number {deny | permit} протокол host source host
destination
```

Если в качестве протокола указано "tcp" или "udp", то описания source- и destination-wildcard могут включать номера портов для данных протоколов с ключевыми словами "eq" (*equal*) — равно, "neq" (*not equal*) — не равно, "lt" (*less than*) — меньше чем, "gt" (*greater then*) — больше чем, "range" — указание диапазона номеров портов. Для протокола "tcp", возможно также применение слова "established" для выделения только установленных tcp-сессий. Ключевое слово "host source" эквивалентно записи: "source 0.0.0.0".

При использовании именованных списков в список добавляется оператор, указывающий на стандартный (*standard*) или расширенный (*extended*) список доступа. Синтаксис такого списка имеет вид:

```
ip access-list {standard | extended} {<номер ACL> и <имя ACL>}
```

**Пример.** Запретить прохождение через маршрутизатор пакетов с рабочей станции с IP-адресом 235.12.60.23 и пропустить все остальные.

В связи с тем, что запрет осуществляется только по адресу источника, используем стандартный список доступа, присвоив ему номер 4.

```
access-list 4 deny host 235.12.60.23
access-list 4 permit any
```

Для удаления списка доступа необходимо сначала ввести команду по `ip access-group` с номером списка для каждого интерфейса, на котором он использовался, а затем команду по `access-list` с номером списка.

При составлении сценариев конфигурации маршрутизаторов следует помнить, что команды всех видов списков доступа вводятся в режиме конфигурирования маршрутизатора, который индицируется промптом: `Router(config)#`.

Для того, чтобы список доступа начал выполнять свою работу, он должен быть применен к интерфейсу с помощью команды

```
Router(config-if)#ip access-group номер-списка-доступа in|out
```

Список доступа может быть применен либо как входной (`in`) либо как выходной (`out`). Когда список доступа применяется как входной, то маршрутизатор получает входной пакет и сверяет содержащийся в нем адрес с элементами списка. Маршрутизатор разрешает пакету маршрутизироваться на интерфейс назначения, если пакет удовлетворяет разрешающим элементам списка, либо отбрасывает пакет, если он соответствует условиям запрещающих элементов списка. Если список доступа применяется как выходной, то маршрутизатор получает входной пакет, пересылает его на интерфейс назначения и только тогда проверяет содержащийся в пакете адрес согласно элементам списка доступа этого интерфейса. Далее маршрутизатор либо разрешает отправку пакета через выходной интерфейс, либо отбрасывает его согласно разрешающим и запрещающим директивам списка соответственно. Так, созданный ранее список, например, с номером 77 применяется к интерфейсу Ethernet 0 маршрутизатора как входной список следующими командами:

```
Router(config)#int Ethernet 0
Router(config-if)#ip access-group 77 in
```

Этот же список применяется к интерфейсу Ethernet 0 маршрутизатора как выходной список с помощью команд

```
Router(config-if)#ip access-group 77 out
```

Отменяется список на интерфейсе с помощью команды **no**

```
Router(config-if)#no ip access-group 77 out
```

Рассмотрим принцип создания более сложных списков доступа. Пусть имеем сеть, изображенную на рисунке 2.1.

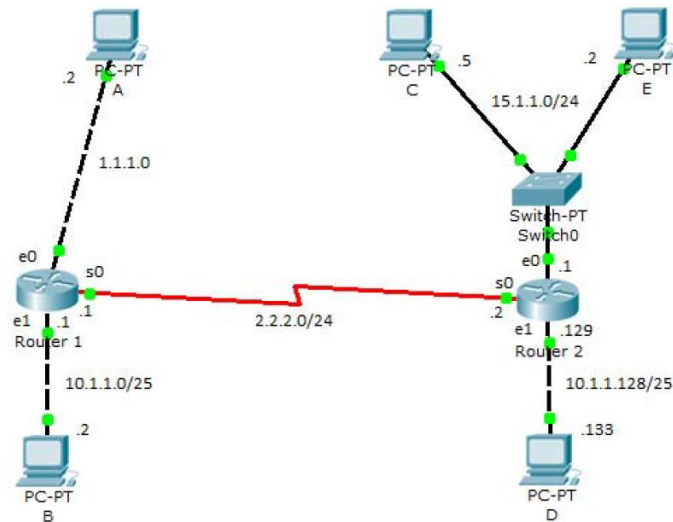


Рисунок 2.1 – Пример топологии сети для создания сложных списков доступа

Разрешим все пакеты, исходящие из сети 10.1.1.0/25 (10.1.1.0 255.255.255.128), но запретим все пакеты, поступающие из сети 10.1.1.128 /25 (10.1.1.128 255.255.255.128) по последовательному интерфейсу S0. Пусть также необходимо запретить все пакеты, исходящие из сети 15.1.1.0 /24 (15.1.1.0 255.255.255.0), за исключением пакетов от единственного хоста с адресом 15.1.1.5. Все остальные пакеты разрешаем. Списку присвоим номер 2. Последовательность команд для выполнения поставленной задачи будет следующая

```
Router(config)#access-list 2 deny 10.1.1.128 0.0.0.127
Router(config)#access-list 2 permit 15.1.1.5 0.0.0.0
Router(config)#access-list 2 deny 15.1.1.0 0.0.0.255
Router(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

Отметим отсутствие разрешающего элемента для сети 10.1.1.0 255.255.255.128. Его роль выполняет последний элемент **access-list 2 permit 0.0.0.0 255.255.255.255**.

Удостоверимся, что поставленная задача выполнена.

1. Разрешить все пакеты, исходящие из сети 10.1.1.0 255.255.255.128. Последняя строка в списке доступа удовлетворяет этому критерию. Нет необходимости в явном виде разрешать эту сеть в нашем списке доступа так, как в списке нет строк, соответствующей этой сети за исключением последней разрешающей строки **permit 0.0.0.0 255.255.255.255**.

2. Запретить все пакеты, исходящие из сети 10.1.1.128 255.255.255.128. Первая строка в списке выполняет этот критерий. Важно отметить вид инверсной маски 0.0.0.127 для этой сети. Эта маска предписывает, что не нужно брать в рассмотрение последние семь бит четвертого октета адреса, которые назначены для адресации компьютера в данной подсети. Маска для этой сети 255.255.255.128, которая указывает, что последние семь бит четвертого октета определяют адресацию компьютера в данной сети.

3. Запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0, за исключением пакетов от единственного хоста с адресом 15.1.1.5. Это требование удовлетворяется второй и третьей строкой нашего списка доступа. Важно отметить, что список доступа осуществляет это требование не в том порядке как оно

определено. Обязательно следует помнить, что список доступа обрабатывается сверху вниз и при первом совпадении обработка пакетов прекращается. Поэтому вначале запрещаются все пакеты, исходящие из сети 15.1.1.0 255.255.255.0 и лишь затем разрешаются пакеты с адресом 15.1.1.5. Если в командах, определяющих список доступа, переставить вторую и третью команды, то вся сеть 15.1.1.0 будет запрещена до разрешения хоста 15.1.1.5. То есть, адрес 15.1.1.5 сразу же в начале будет запрещен более общим критерием **deny 15.1.1.0 0.0.0.255**.

4. Разрешить все остальные пакеты. Последняя команда разрешает все адреса, которые не соответствуют первым трем командам.

Таким образом, последовательность действий для реализации списка доступа может быть записана в следующем виде.

1. Определить критерии и ограничения для доступа.

2. Реализовать их с помощью команд access-list, создав список доступа с определенным номером.

3. Применить список к определенному интерфейсу либо как входящий, либо как исходящий.

Следует заметить, что в общем случае стандартный список доступа нужно помещать как можно ближе к точке назначения, а не к источнику пакетов. Если список помещен вблизи источника пакетов, то очень вероятно, что доступ к устройствам, на которых не осуществляется никакая конфигурация доступа, будет затруднен.

Конкретизируем политику безопасности для сети на рисунке 2.1. Наша цель создать политику для компьютера А (адрес 1.1.1.2 сеть 1.1.1.0/24), которая изо всех устройств локальной сети 15.1.1.0/24 в которую входит компьютер С (15.1.1.5) разрешит доступ к компьютеру А лишь самого компьютера С. Мы также хотим создать политику, запрещающую удаленный доступ к компьютеру А из любого устройства локальной сети 10.1.1.128 / 25 компьютера D (10.1.1.133). Весь остальной трафик мы разрешаем. На рисунке 2.1 компьютер PC5 (15.1.1.5) играет роль произвольного, отличного от компьютера С, представителя локальной сети 15.1.1.0/24.

Размещение списка критично для реализации такой политики. Возьмем созданный ранее список с номером 2. Если список сделать выходным на последовательном интерфейсе S0 маршрутизатора 2, то задача для компьютера А будет выполнена, однако возникнут ограничения на трафик между другими локальными сетями. Аналогичную ситуацию получим, если сделаем этот список входным на последовательном интерфейсе маршрутизатора 1. Если мы поместим этот список как выходной на интерфейс Ethernet0 маршрутизатора 1, то задача будет выполнена безо всяких побочных эффектов.

## Именованные ACL

К именованным ACL обращаются по имени, а не по номеру, что дает наглядность и удобство для обращения. Для создания именованного ACL имеется команда

```
Router(config)#ip access-list standard|extended ACL_name
```

и далее команды для создания элементов списка

```
Router(config-ext-nacl)#permit|deny IP_protocol
source_IP_address
wildcard_mask [protocol_information] destination_IP_address
wildcard_mask [protocol_information] [log]
```

Для завершения создания списка следует дать команду **exit**.

Имя именованного списка чувствительно к регистру. Команды для создания неименованного списка аналогичные командам для создания элементов нумерованного списка, но сам процесс создания отличен. Вы должны использовать ключевое слово **ip** перед главным ACL оператором и тем самым войти в режим конфигурации именно для этого именованного списка. В этом режиме вы начинаете с ключевых слов **permit** или **deny** и не должны вводить **access-list** в начале каждой строки.

Привязка именованных ACL к интерфейсу осуществляется командой

```
Router(config)#interface type [slot_№] port_№
Router(config-if)#ip access-group ACL_name in|out
```

Именованный ACLs разрешает себя редактировать. Для этого надо набрать команду, которая была использована для его создания

```
Router(config)#ip access-list standard|extended ACL_name
```

С помощью клавиш с вертикальными стрелками следует найти строку списка, которую нужно изменить. Изменить ее, используя горизонтальные стрелки. Нажать ввод. Новая строка добавится в конец списка. Старая не уничтожится. Для ее удаления следует ввести **no** в начале строки.

Для редактирования же числовых ACLs необходимо его уничтожить и создать заново или изменить список офлайн и загрузить в устройство с помощью **telnet**.

### Пример именованного списка доступа

Создается стандартный список доступа с именем **Internet\_filter** и расширенный список доступа с именем **marketing\_group**:

```
Router(config)#interface Ethernet0/5
Router(config-if)#ip address 2.0.5.1 255.255.255.0
Router(config)#ip access-group Internet_filter out
Router(config-if)#ip access-group marketing_group in
Router(config)#ip access-list standard Internet_filter
Router(config-ext-nacl)#permit 1.2.3.4
Router(config-ext-nacl)#deny any
Router(config)#ip access-list extended marketing_group
Router(config-ext-nacl)#permit tcp any 171.69.0.0 0.0.255.255 eq
telnet
Router(config-ext-nacl)#deny tcp any any
Router(config-ext-nacl)#permit icmp any any
Router(config-ext-nacl)#deny udp any 171.69.0.0 0.0.255.255 lt 1024
Router(config-ext-nacl)#deny ip any any log
```

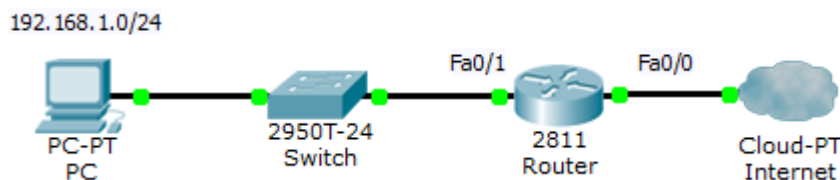
### 5.1.1 Reflexive ACL – зеркальные списки контроля доступа

Reflexive ACL – зеркальные списки контроля доступа, позволяют запоминать, кто обращался из нашей сети наружу (с каких адресов, с каких портов, на какие адреса, на какие порты) и автоматически формировать зеркальный ACL, который будет пропускать обратный трафик извне вовнутрь только в том случае, если изнутри было обращение к данному ресурсу.

Зеркальные (reflexive) ACL – это расширение технологии extended ACL, которое позволяет организовать пропуск трафика из Интернета в локальную сеть только в ответ на предварительно сделанный запрос из локальной сети в Интернет. Зеркальные списки ACL можно задать **только** с помощью расширенных именованных списков ACL для протокола IP. Их нельзя определить с помощью нумерованных или стандартных списков ACL для протокола IP или с помощью списков ACL для других протоколов.

Суть фильтрации пакетов состоит в следующем: на выходной интерфейс локальной сети прикрепляется ACL, который пропускает исходящий трафик. Одновременно автоматически формируется встречный ACL для пропуска входящего трафика. Благодаря этому разрешается получать ответы из Интернета только на свои запросы. Ключевые слова, используемые в Reflexive ACL – это **reflect** для исходящего трафика и **evaluate** для входящего.

**Пример.** Пусть имеется локальная компьютерная сеть с адресом 192.168.1.0/24. Из нее нужно организовать доступ в Интернет всем клиентским компьютерам сети по протоколам http, pop и smtp.



Вначале необходимо создать именованный расширенный список для исходящего трафика, например, с именем IN-TO-OUT, а затем для входящего трафика список именем OUT-TO-IN. При этом следует учитывать, что сообщения для указанных сервисов (http, pop и smtp) передаются по протоколу TCP. Сценарий, реализующий заданные условия, имеет следующий вид.

```

R1(config)#ip access-list extended IN-TO-OUT
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq www
reflect BACK-WWW
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq pop3
reflect BACK-POP
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq smtp
reflect BACK-SMTP
R1(config-ext-nacl)#exit
R1(config)#ip access-list extended OUT-TO-IN
R1(config-ext-nacl)#evaluate BACK-WWW
R1(config-ext-nacl)#evaluate BACK-POP
R1(config-ext-nacl)#evaluate BACK-SMTP
  
```

Здесь BACK-WWW, BACK-POP и BACK-SMTP – имена зеркальных списков доступа. Параметр **reflect name** используется для создания рефлексивного списка доступа.

Затем эти списки связываются с внешним fa0/0 и внутренним fa0/1 интерфейсами маршрутизатора.

```
R1(config)#interface fa0/0
R1(config-if)#ip access-group OUT-TO-IN in
R1(config)#interface fa0/1
R1(config-if)#ip access-group IN-TO-OUT out
R1(config-if)#
```

Список IN-TO-OUT разрешает выход трафика изнутри наружу. Пропускается трафик на порты 25, 80 и 110, параллельно формируются зеркальные ACL с именами BACK-WWW, BACK-POP и BACK-SMTP, которые пропускают обратный трафик. Весь трафик извне фильтруется ACL с именем OUT-TO-IN, который по умолчанию ничего не пропускает, но когда появляются зеркальные записи, то трафик начинает пропускаться.

Предположим, что пользователь обращается с адреса 192.168.1.100 к веб-страничке на сервере 123.123.123.123 при обращении выбирается случайный порт отправителя (например, 1235), порт получателя используется стандартный – 80. Когда пакет проходит через маршрутизатор, он проверяется IN-TO-OUT и на основании первой строчки списка выводится в канал. Одновременно в ACL с именем BACK-WWW автоматически на время добавляется зеркальная запись:

```
permit tcp host 123.123.123.123 eq 80 host 192.168.1.100 eq 12345
```

То есть, в настоящий момент весь трафик из интернета вовнутрь будет заблокирован, за исключением ответа от веб-сервера на наш запрос. Преимущество Reflexive ACL перед established заключается в том, что при established используется только флагом в TCP сегменте, а Reflexive реально отслеживает соединения. Флаг можно подделать, в этом случае входящий трафик начнет пропускаться. Конечно, его вряд ли кто-то примет, но можно устроить, например, DOS атаку. Но самое важное преимущество, с помощью established в принципе нельзя организовать пропуск протоколов, отличных от TCP. Например, протоколов, базирующихся на UDP, или ICMP трафик. Зеркальные же ACL успешно справляются с этими задачами.

ACL обрабатываются сверху вниз. Наиболее часто повторяющийся трафик должен быть обработан в начале списка. Как только обрабатываемый список пакет удовлетворяет элементу списка, обработка этого пакета прекращается. Стандартные ACLs следует помещать ближе к точке назначения, где трафик должен фильтроваться. Выходные (out) расширенные ACLs следует помещать как можно ближе к источнику фильтруемых пакетов, а входные следует помещать ближе к точке назначения, где трафик должен фильтроваться.

### Ограничение доступа к VTY при помощи ACL

ACL можно применять не только для фильтрации трафика, но и для ограничения адресов, с которых можно подключиться к маршрутизатору по

telnet или ssh. Сначала создается стандартный ACL, в котором перечисляются адреса и сети, из которых доступ по telnet надо разрешить. Теперь его необходимо применить непосредственно на **line vty 0 4**, то есть, на линии виртуального терминала, к которым происходит подключение. Таким образом, не важно, через какой интерфейс маршрутизатора telnet-пакеты попадут на роутер, они будут отфильтрованы когда доберутся собственно до vty.

На маршрутизаторе создается стандартный список доступа **VTY\_ACCESS**:

```
Router(config)#ip access-list standard VTY_ACCESS
Router(config-std-nacl)#permit 15.15.1.0 0.0.0.255
```

Устанавливается ограничение доступа к **VTY** на маршрутизаторе:

```
Router(config)#line vty 0 4
Router(config-line)#access-class VTY_ACCESS in
```

Теперь по telnet можно подключиться только из сети 15.15.1.0.

Обратите внимание, что ACL применяется на интерфейсе командой `access-group`, а на vty – командой `access-class`.

### 3. ОПИСАНИЕ ЛАБОРАТОРНОЙ УСТАНОВКИ

В качестве лабораторной установки используется персональный компьютер с установленной программой Packet Tracer, позволяющей осуществлять моделирование компьютерных сетей, построенных на оборудовании корпорации Cisco. Подробно описание пакета моделирования и работы с ним приведено в лабораторной работе №1. Исследуемая схема компьютерной сети изображена на рисунке 3.1.

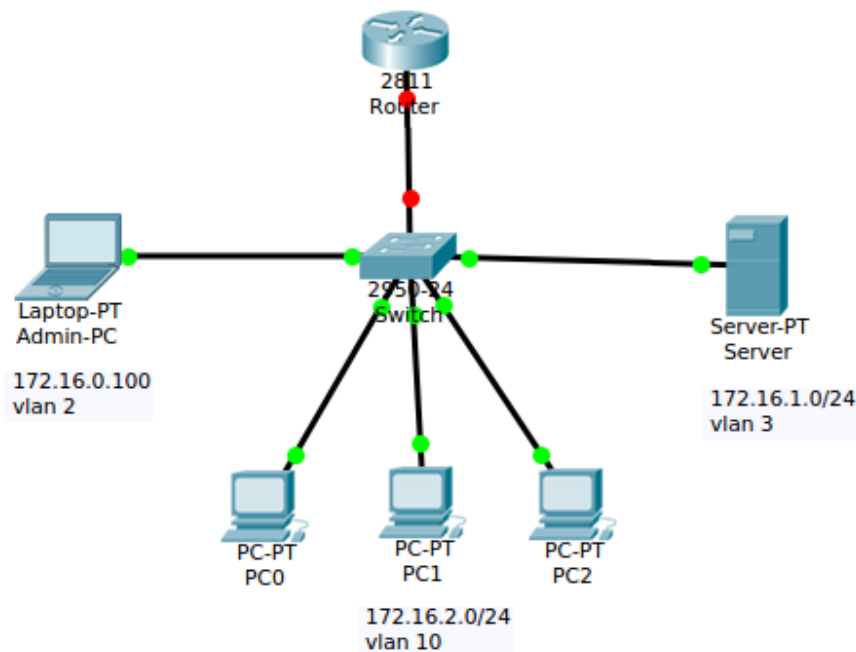


Рисунок 3.1 – Схема исследуемой компьютерной сети

## 4. ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ

4.1. Изучить теоретический материал, относящийся к составлению и применению списков доступа (выполняется в процессе домашней подготовки).

4.2. Создать в рабочем окне Packet Tracer схему сети, изображенную на рисунке 3.1.

4.3. Сконфигурировать коммутатор таким образом, чтобы компьютер администратора с адресом 172.16.0.100 находился в vlan 2, сервер с адресом 172.16.1.0/24 размещался в vlan 3, а рабочие станции представляли собой подсеть vlan 10 с адресом 172.16.2.0/24. Конфигурацию оборудования выполнить с командной строки.

4.4. Сконфигурировать оборудования т.о., чтобы доступ к серверу имел только администратор.

4.5. Проверить путем пингования, что требования, изложенные в п.4.3 и 4.4 выполнены.

4.6. Переконфигурировать оборудования т.о., чтобы пользователи рабочих станций PC0-PC2 имели доступ к файл-серверу и к HTTP (порт 80) и FTP (порт 21) серверам. При этом предусмотреть функционирование DNS (порт 53) сервера.

4.7. Сформулировать выводы по результатам исследований.

**Примечание:** проверить правильность конфигурации телекоммуникационного оборудования и обнаружить ошибки конфигурации можно путем использования приложения А.

## 5. СОДЕРЖАНИЕ ОТЧЕТА

Отчет о выполненной работе должен содержать:

1. Титульный лист.
2. Схему исследуемой сети и программу работы.
3. Скрипты настроек сетевого оборудования.
4. Скриншоты результатов исследования функционирования сети.
5. Выводы.

## 6 КОНТРОЛЬНЫЕ ВОПРОСЫ

- 6.1. Что представляют собой списки контроля доступа?
- 6.2. Какой адрес является критерием для разрешения/запрещения пакета?
- 6.3. Где применяются ACL?
- 6.4. Как задать элемент ACL и что такое инверсная маска?
- 6.5. Как маршрутизатор обрабатывает элементы ACL?
- 6.6. Какой элемент всегда неявно присутствует в ACL?
- 6.7. Как ACL применить к интерфейсу и затем его отменить?
- 6.8. Чем отличается входной ACL от выходного?
- 6.9. Где в сети рекомендуется размещать ACL?

- 6.10. Какими тремя командами можно проверить содержимое ACL и привязку к интерфейсу.
- 6.11. Что фильтруют расширенные ACL?
- 6.12. Какую дополнительную функциональность имеют расширенные ACL по сравнению со стандартными?
- 6.13. Можно ли, используя расширенные ACL, наложить ограничения на трафик к определённой TCP/IP службе?
- 6.14. Опишите процедуру создания именованного ACL.
- 6.15. Как отредактировать конкретную строку в числовом ACL?
- 6.16. Как отредактировать конкретную строку в именованном ACL?
- 6.17. Чем отличаются форматы команд для ввода элементов числового и именованного ACL?

**Приложение А.** Сценарий реализации пунктов программы 4.3 и 4.4.  
[<https://pcradar.ru/nastroyka-acl-v-cisco/>]

```

Switch>enable - переходим в расширенный режим
Switch#configure terminal - переходим в режим конфигурации
Switch(config)#vlan 2 - создаем vlan 2
Switch(config-vlan)#name Admin - название для vlan 2
Switch(config)#vlan 3 - создаем vlan 3
Switch(config-vlan)#name Server - название для vlan 3
Switch(config)#vlan 10 - создаем vlan 10
Switch(config-vlan)#name User's - название для vlan 10
Switch(config)#interface range fa0/1 - fa0/9 - настраиваем интерфейсы
                                         в сторону Пользователей
Switch(config-if-range)#description User's - описание интерфейса
Switch(config-if-range)#switchport mode access - настраиваем порт на
                                                  тегированный режим
Switch(config-if-range)#switchport access vlan 10 - тегуем кадры
                                                  10-й VLAN
Switch(config-if-range)#exit
Switch(config)#interface fa0/10 - настраиваем интерфейсы в сторону
                                   Сервера
Switch(config-if)#description Server - описание интерфейса
Switch(config-if)#switchport mode access - настраиваем порт на
                                                  тегированный режим
Switch(config-if)#switchport access vlan 3 - тегуем кадры 3 VLAN
Switch(config-if)#exit
Switch(config)#interface fa0/20 - настраиваем интерфейсы в сторону
                                   Админа
Switch(config-if)#description Admin - описание интерфейса
Switch(config-if)#switchport mode access - настраиваем порт на
                                                  тегированный режим
Switch(config-if)#switchport access vlan 2 - тегуем кадры 2 VLAN
Switch(config-if)#exit
Switch(config)#interface fa0/24 - настраиваем интерфейсы в сторону
                                   маршрутизатора
Switch(config-if)#description Router - описание интерфейса
Switch(config-if)#switchport mode trunk - настраиваем порт на
                                                  магистральный режим
Switch(config-if)#switchport trunk allowed vlan 2-3,10 – разрешаем
                                                         кадры VLAN 2-3,10
Switch(config-if)#exit
Switch(config)#do write - сохраняем конфигурацию

```

### Конфигурация для маршрутизатора:

```

Router>enable - переходим в расширенный режим
Router#configure terminal - переходим в режим конфигурации
Router(config)#interface fa0/0 - настраиваем порт в сторону коммутатора
Router(config-if)#description Switch - описание интерфейса
Router(config-if)#no shutdown - включаем интерфейс физически
Router(config-if)#exit
Router(config)#interface fa0/0.2 - настраиваем подинтерфейс для подсети
                                Админа
Router(config-subif)#description Admin - описание интерфейса
Router(config-subif)#encapsulation dot1q 2 - тегируем 2 VLAN'ом
Router(config-subif)#ip address 172.16.0.1 255.255.255.0 - задаем шлюз
                                                            по умолчанию для Админа

Router(config-subif)#exit
Router(config)#interface fa0/0.3 - настраиваем подинтерфейс для подсети
                                Серверов
Router(config-subif)#description Server - описание интерфейса
Router(config-subif)#encapsulation dot1q 3 - тегируем 3 VLAN'ом
Router(config-subif)#ip address 172.16.1.1 255.255.255.0 - задаем шлюз
                                                            по умолчанию для Серверов

Router(config-subif)#exit
Router(config)#interface fa0/0.10 - настраиваем подинтерфейс для подсети
                                пользователей
Router(config-subif)#description User's - описание интерфейса
Router(config-subif)#encapsulation dot1q 10 - тегируем 10 VLAN'ом
Router(config-subif)#ip address 172.16.2.1 255.255.255.0 - задаем шлюз по
                                                            умолчанию для Серверов

Router(config-subif)#exit

```

Запускаем пинг с пользовательского компьютера до сервера

```

PC>ping 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time=1ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127

Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 1ms, Average = 0ms

```

Как видим доступ есть. Нам же необходимо, чтобы доступ имел только админ. Для этого нам необходимо создать список доступа (пусть он будет иметь порядковый номер 10), в котором мы разрешим всем пакетам от администратора (172.16.0.100) доступ в подсеть серверов (172.16.1.0/24). После чего применим это правило на подинтерфейсе fa0/0.3 (для серверов) для всех исходящих пакетов.

```
Router(config)#access-list 10 permit host 172.16.0.100 - создаем список доступа,
                                         в котором разрешаем хосту админа
Router(config)#interface fa0/0.3 - настраиваем подинтерфейс для Серверов
Router(config-subif)#ip access-group 10 out - применяем настройки списка
                                         доступа на подинтерфейсе
```

Тестируем настройки. Запускаем пинг с пользовательского компьютера в сторону сервера.

```
PC>ping 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.

Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Выдается сообщение Destination host unreachable – хост назначения недоступен. Запускаем пинг с компьютера администратора.

```
PC>ping 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time=1ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=5ms TTL=127

Ping statistics for 172.16.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 5ms, Average = 1ms
```

Пинг проходит – значит ACL настроили правильно. Что происходит, когда мы пингуем сервер с ноутбука администратора? Пакет сначала поступает на подинтерфейс fa0/0.2 маршрутизатора. На данном интерфейсе не настроены списки доступа значит пакет проходит далее. Маршрутизатор анализирует свою таблицу маршрутизации и видит, что подсеть серверов находится на подинтерфейсе fa0/0.3. Перед отправкой пакета маршрутизатор обнаруживает, что к данному интерфейсу прикреплен ACL 10. В данном списке доступа всего одна запись – разрешить отправку пакетов только хосту 172.16.0.100 (ноутбук админа). Маршрутизатор анализирует IP-пакет и обнаруживает адрес отправителя

172.16.0.100 после чего отправляет пакет в подсеть серверов. IP-пакет с любым отличным от 172.16.0.100 будет отбрасываться, так как в конце ACL 10 стоит неявный deny any – запретить все.

В связи с тем, что пользователям в заданной сети необходимо иметь доступ к файловому хранилищу и веб-сайту, требуется использовать расширенные списки доступа. Однако перед этим был полностью ограничен доступ клиентам к серверу. Для исправления ситуации необходимы расширенные списки доступа, которые могут проверять IP-адреса источника/отправителя, тип протокола, UDP/TCP-порты. В заданной ситуации необходимо будет проверять номера портов. Если пользователь обращается к серверу по разрешенному порту, то маршрутизатор пропускает такой пакет. Разрешенные порты: 80 (HTTP – доступ к веб-сайту), 21 (FTP – доступ к файловому хранилищу). Протоколы HTTP и FTP работают поверх TCP. Также для распознавания доменных имен на сервере необходимо включить (поднять) DNS, который работает на порту 53.

Размещать расширенный список доступа будем на подинтерфейсе fa0/0.3. Но на этом интерфейсе уже размещен список доступа. Следует помнить правило: нельзя разместить более одного списка доступа на интерфейс. По этой причине придется удалить созданный ранее список доступа. Правило, созданное для администратора, перенесем в новый расширенный список с именем *Server-out*.

Конфигурация для маршрутизатора:

```
Router(config)#no access-list 10 permit host 172.16.0.100
- удаляем предыдущий список доступа
Router(config)#interface fa0/0.3
- настраиваем сабинтерфейс для Серверов
Router(config-subif)#no ip access-group 10 out
- удаляем предыдущие настройки списка доступа
Router(config-subif)#exit
Router(config)#ip access-list extended Server-out
- создаем расширенный список доступа
Router(config-ext-nacl)#permit ip host 172.16.0.100 host 172.16.1.2 - даем админу
полный доступ к серверу
Router(config-ext-nacl)#permit tcp any host 172.16.1.2 eq 80
- разрешаем любому хосту доступ по HTTP к серверу
Router(config-ext-nacl)#permit tcp any host 172.16.1.2 eq 21
- разрешаем любому хосту доступ по FTP к серверу
Router(config-ext-nacl)#permit tcp any host 172.16.1.2 eq 53
- разрешаем любому хосту доступ по DNS к серверу
Router(config-ext-nacl)#exit
Router(config)#interface fa0/0.3
```

```
Router(config-if)#ip access-group Server-out out
```

С компьютера админа пинг до сервера есть:

```
PC>ping -t 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127
Reply from 172.16.1.2: bytes=32 time=0ms TTL=127

Ping statistics for 172.16.1.2:
    Packets: Sent = 5, Received = 5, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

С компьютера пользователя ответа на пинг нет:

```
PC>ping -t 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.
Reply from 172.16.2.1: Destination host unreachable.

Ping statistics for 172.16.1.2:
    Packets: Sent = 8, Received = 0, Lost = 8 (100% loss),
```

Проверим с компьютера пользователя проходят ли DNS-запросы до сервера. Для этого запустим утилиту *nslookup* – которая определяет IP-адрес до доменному имени.

```
PC>nslookup test.site

Server: [172.16.1.2]
Address: 172.16.1.2

Non-authoritative answer:
Name: test.site
Address: 172.16.1.2
```

DNS-запросы проходят без проблем. Проверим доступ к нашему условному Web-сайту через браузер:



Напоследок подключимся к FTP-серверу:

```
PC>ftp 172.16.1.2
Trying to connect...172.16.1.2
Connected to 172.16.1.2
220- Welcome to PT Ftp server
Username:user
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>
```

Подключение прошло успешно!

## Лабораторная работа №4

### Исследование способов конфигурации сетевых серверных служб стека протоколов TCP/IP

#### 1 Цель работы

Исследование особенностей использования основных сетевых серверных служб стека протоколов TCP/IP и конфигурации серверов, реализующих эти службы, приобрести практические навыки по конфигурации серверного сетевого оборудования.

#### 2 Краткие теоретические сведения

В простейшем понимании служба — это пара программ, взаимодействующих между собой согласно определенным протоколами по схеме клиент-сервер. Одна из программ этой пары называется сервером, а другая — клиентом. Соответственно, когда говорят о работе сетевых служб, речь идет о взаимодействии оборудования и программного обеспечения сервера с оборудованием и программным обеспечением клиента, обеспечивающих функционирование компьютерной сети.

Протоколы, реализующие сетевые службы, относятся к протоколам прикладного уровня. Существует большое количество протоколов этого уровня и выполняют они совершенно различные функции. К наиболее часто используемым протоколам прикладного уровня относятся протоколы HTTP, DNS, DHCP, SMTP и POP3, Telnet, SSH, FTP и TFTP.

В данной работе исследуются протоколы, предназначенные для облегчения администрирования компьютерных сетей, в частности: HTTP, DNS и DHCP.

1) **Протокол передачи гипертекста HTTP** (*HyperText Transport Protocol*), используемый обычно для получения информации с веб-сайтов. Построен на основе клиент-серверной модели, то есть существуют клиенты, формирующие и отправляющие запрос и серверы, которые принимают запросы и, соответственно, на них отвечают. Передача данных по протоколу HTTP обычно происходит через TCP/IP-соединения. Серверное программное обеспечение при этом использует TCP-порт 80 (и, если порт не указан явно, то обычно клиентское программное обеспечение по умолчанию использует именно 80-й порт для открываемых HTTP-соединений), хотя может использовать и любой другой.

В качестве клиентов выступают веб-браузеры: Internet Explorer, Mozilla Firefox, Google Chrome и т.д, а в качестве серверного ПО используются серверы Apache, IIS (*Internet Information Server*), nginx (engine x) и др.

Кроме протокола HTTP имеется расширенная версия HTTPS, согласно которой все данные передаются в зашифрованном виде.

2) **Протокол разрешения доменных имен DNS** (*Domain Name System*) относится к прикладному уровню эталонной модели. При этом используется 53-й TCP- или UDP-порт. Чаще всего применяется для получения IP-адреса по

имени хоста (компьютера или устройства). Имеется распределённая база данных DNS, которая поддерживается с помощью иерархии DNS-серверов, взаимодействующих по DNS-протоколу.

3) **Протокол DHCP** (*Dynamic Host Configuration Protocol*) — это одна из служб поддержки протокола TCP/IP, разработанная для упрощения администрирования IP-сети за счет использования специально настроенного сервера для централизованного управления IP-адресами и другими параметрами протокола TCP/IP, необходимыми сетевым узлам. Сервер DHCP избавляет сетевого администратора от необходимости ручного выполнения операций. С его помощью реализуется:

- автоматическое назначение сетевым узлам IP-адресов и прочих параметров протокола TCP/IP (например, маска подсети, адрес основного шлюза подсети, адреса серверов DNS и WINS);
- защита от дублирования IP-адресов, назначаемых различным узлам сети;
- освобождение IP-адресов узлов, удаленных из сети;
- ведение централизованной БД выданных IP-адресов.

При загрузке компьютера, настроенного на автоматическое получение IP-адреса, или при смене статической настройки IP-конфигурации на динамическую, а также при обновлении IP-конфигурации сетевого узла происходят следующие действия:

- 1) компьютер посылает широковещательный запрос на обнаружение доступного DHCP-сервера, (DHCP Discover);
- 2) DHCP-серверы, получившие данный запрос, посылают данному сетевому узлу свои предложения IP-адреса (DHCP Offer);
- 3) клиент отвечает на предложение, полученное первым, соответствующему серверу запросом на выбор арендуемого IP-адреса (DHCP Request);
- 4) DHCP-сервер регистрирует в своей БД выданную IP-конфигурацию (вместе с именем компьютера и физическим адресом его сетевого адаптера) и посылает клиенту подтверждение на аренду IP-адреса (DHCP Acknowledgement).

При планировании серверов DHCP:

- желательно в каждой IP-сети установить отдельный DHCP-сервер;
- если нет возможности установить свой сервер в каждой IP-сети, необходимо на маршрутизаторах, объединяющих IP-сети, запустить и настроить агент ретрансляции DHCP-запросов (*DHCP Relay Agent*) таким образом, чтобы он пересылал широковещательные запросы DHCP из подсети, в которой нет DHCP-сервера, на соответствующий DHCP-сервер, а на самом DHCP-сервере создать области для всех обслуживаемых IP-сетей;
- для повышения отказоустойчивости следует установить несколько серверов DHCP, при этом на каждом DHCP-сервере, кроме областей для "своих" IP-сетей, необходимо создать области для других подсетей (при этом диапазо-

ны IP-адресов в таких резервных областях не должны пересекаться с основными областями, созданными на серверах DHCP в "своих" подсетях);

- в больших IP-сетях DHCP-серверы должны иметь мощные процессоры, достаточно большие объемы оперативной памяти и быстродействующие дисковые подсистемы, т.к. обслуживание большого количества клиентов требует интенсивной работы с базой данных DHCP-сервера.

### 3 Описание лабораторной установки

В качестве лабораторной установки используется персональный компьютер с установленной программой Packet Tracer, позволяющей осуществлять моделирование компьютерных сетей, построенных на оборудовании корпорации Cisco. В программе имеется возможность включать в состав моделируемых сетей серверы практически всех типов и осуществлять их конфигурацию. К наиболее широко используемым серверами, моделируемыми системой Packet Tracer, относятся следующие типы серверов.

**Cisco HTTP (WEB) сервер** – позволяет создавать простейшие веб-странички и проверять прохождение пакетов на 80-ый порт сервера. Эти серверы предоставляют доступ к веб-страницам и сопутствующим ресурсам, например, изображением.

**DHCP сервер** – позволяет организовывать пулы сетевых настроек для автоматического конфигурирования сетевых интерфейсов. *Dynamic Host Configuration Protocol* обеспечивает автоматическое распределение IP-адресов между компьютерами в сети. Такая технология широко применяется в локальных сетях с общим выходом в *Интернет*.

**DNS сервер** – позволяет организовать службу разрешения доменных имён. *Функция DNS-сервера* заключается в преобразовании доменных имен серверов в IP-адреса.

**Cisco EMAIL** – *почтовый сервер*, для проверки почтовых правил. Электронное письмо нельзя послать непосредственно получателю – сначала оно попадает на сервер, на котором зарегистрирована учетная запись отправителя. Тот, в свою очередь, отправляет "посылку" серверу получателя, с которого последний и забирает сообщение.

**FTP** – *файловый сервер*. В его задачи входит хранение файлов и обеспечение доступа к ним клиентских ПК, например, по протоколу *FTP*. Ресурсы *файл-сервера* могут быть либо открыты для всех компьютеров в сети, либо защищены системой идентификации и правами доступа.

В настоящей работе будут исследоваться процессы конфигурации Web-, DHCP- и DNS-серверов.

### 4 Программа выполнения работы

4.1 Повторить теоретический материал по иерархии протоколов стека TCP/IP, по протоколам прикладного уровня и составу полей кадров и пакетов этих протоколов (выполняется в процессе домашней подготовки).

4.2 Составить в рабочем окне эмулятора схему исследуемой сети, изображенной на рисунке 4.1.

4.3 Установить для всех серверов сети статический режим адресации и задать их адреса в следующем виде: XY.0.0.10 – DHCP-сервер; XY.0.0.100 – DNS-сервер; XY.0.0.100 – HTTP-сервер [www.sevgu.ru](http://www.sevgu.ru); XY.0.0.200 – HTTP-сервер [www.kaf.is](http://www.kaf.is). Здесь X-предпоследняя цифра зачетной книжки, а Y-предпоследняя.

4.4 Задать режим динамической адресации для оконечных устройств сети, и провести установку и настройку DHCP-сервера на компьютере XY.0.0.10.

4.5 Установить на серверный компьютер XY.0.0.100 DNS-сервер и осуществить его настройку.

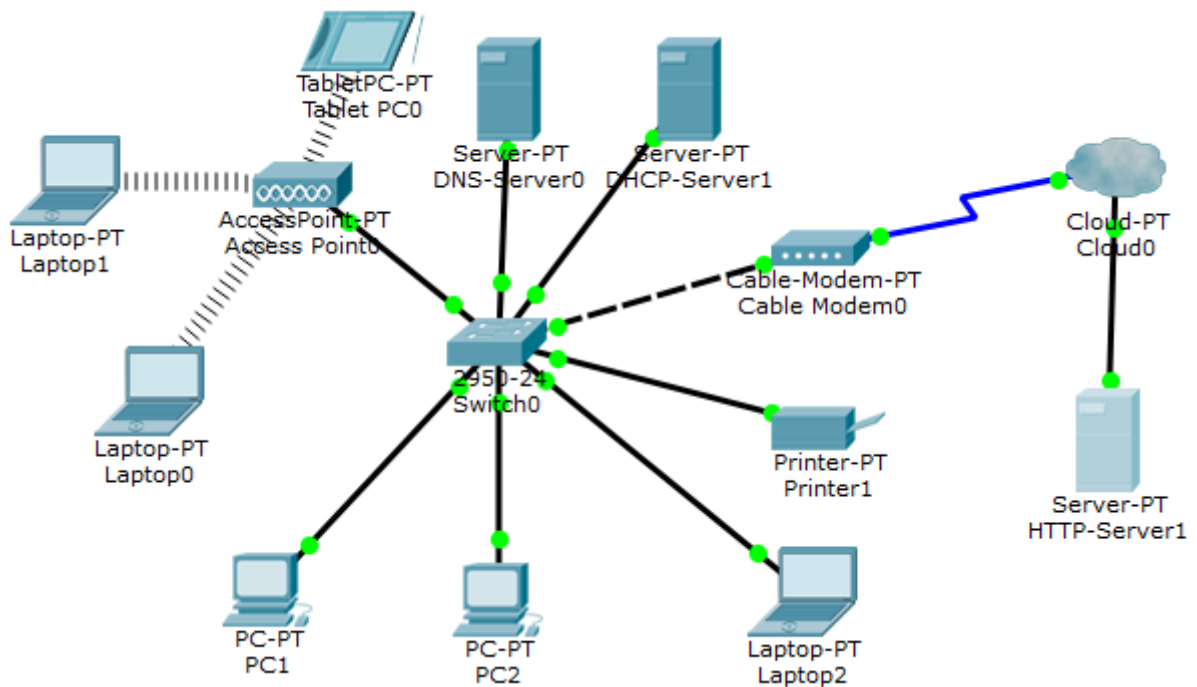


Рисунок 4.1 – Схема исследуемой сети с сетевыми службами

4.6 Установить на серверный компьютер XY.0.0.100 HTTP-сервер, и разместить на нем страничку сайта [www.sevgu.ru](http://www.sevgu.ru) с информацией о университете.

4.7 Установить на серверный компьютер XY.0.0.200 HTTP-сервер, и разместить на нем страничку сайта [www.kaf.is](http://www.kaf.is) с рекламной информацией о кафедре ИС.

4.8 Провести проверку связи оконечных устройств друг с другом и доступа к страницам сайтов по их IP-адресам и по доменным символическим именам в реальном режиме и режиме симуляции.

4.9 Исследовать структуру пакетов при обращении к странице одного из сайтов.

4.1 Оформить отчет и сделать выводы по работе.

## 5 Методические рекомендации по выполнению лабораторной работы

### 5.1 Подключение удаленного сервера через облако Интернета

Связь коммутатора с облачным устройством осуществляется с помощью коаксиального кабеля и кабельного модема. Чтобы пакеты могли передаваться через облако нужно настроить порты со стороны модема и удаленного сервера. Для этого нужно выполнить следующие действия.

Щелкните пиктограмму облака Интернета в логическом рабочем пространстве Packet Tracer и выберите вкладку **Physical** (Физические). Для облачного устройства потребуются два модуля, если они еще не установлены. Модуль **PT-CLOUD-NM-1CX** необходим для подключения службы кабельного модема, а модуль **PT-CLOUD-NM-1CFE** — для подключения медного кабеля Ethernet. Если эти модули отсутствуют, отключите физические облачные устройства, нажав кнопку питания, и перетащите оба этих модуля в пустые порты для модулей устройства. После этого снова включите питание.

Находясь на вкладке **Config** (Конфигурация), выберите Ethernet в разделе **INTERFACE** (Интерфейс) в левой панели. В окне конфигурации Ethernet выберите **Cable** (Кабель) в поле Provider Network.

Для определения выходного и входного портов облачного пространства откройте вкладку **Config** в окне Cloud device. В левой панели выберите **Cable** в разделе **CONNECTIONS** (Подключения). В первом раскрывающемся списке выберите пункт Coaxial (Коаксиальный), а во втором — Ethernet. Затем нажмите кнопку **Add** (Добавить), чтобы добавить их в качестве выходного и входного портов.

### 5.2 Создание служб и конфигурация серверов

Для установки любой службы на серверном компьютере щелкните левой кнопкой мыши по компьютеру и выберите вкладку **Services** (Службы).

При установке службы DHCP Выберите **DHCP** в списке **SERVICES** в панели слева. В окне конфигурации DHCP настройте следующие параметры DHCP:


- нажмите **On**, чтобы включить службу DHCP;
- задайте имя пула адресов Pool name, например DHCPpool;
- задайте адреса: Шлюза по умолчанию, DNS-сервера, Начальный IP-адрес, Маску подсети и Максимальное число пользователей в сети;
- Нажмите **Add**, чтобы добавить пул.

Для установки и конфигурации DNS-сервера необходимо активировать Server0 и открыть закладку Services, на которой выбрать службу DNS. В рабочем окне нужно задать две ресурсные записи (Resource Records) в прямой зоне DNS, которая представляет собой часть дерева доменных имен (включая ресурсные записи), размещаемая как единое целое на сервере доменных имен (DNS-сервере). Сначала в ресурсной записи типа A Record необходимо связать доменное имя компьютера с его IP-адресом. Для этого в окошке Name задать имя DNS -сервера (например, server0.mail.ru), а в окошке Address занести его

IP-адрес (например, 10.0.0.100). Затем в окошке Type следует выбрать тип записи «A Record» и нажать на кнопку Add, а также активировать переключатель On.

Далее в окошко Name внести имя сайта (например, www.mail.ru), а в окошко Host Name – имя сервера. После этого нужно связать название сайта с сервером, для чего в окошке **Type** выбрать тип ресурсной записи «CNAME» и нажать кнопку **Add** (добавить).

Для создания HTTP-сервера необходимо открыть на серверном компьютере вкладку Services, активировать протокол HTTP и выбрать режим редактирования (edit) шаблона страницы сайта с названием **index.html**.

В этом окне можно добавить новую страницу или удалить текущую кнопкой. Переключение между несколькими страницами осуществляется кнопками .

В окне html кода нужно сформировать текст первой страницы сайта **index.html**. Текст можно создать в текстовом редакторе и переносить в это окно через буфер обмена. Следует учесть, что текст должен быть только на английском языке!

### 5.3 Проверка подключения

Перед началом пингования убедитесь, что ПК получает конфигурационные данные IPv4 от DHCP. Нажмите **PC** в рабочем окне Packet Tracer и выберите вкладку **Desktop**. Щелкните пиктограмму **Command Prompt (Командная строка)**. В командной строке обновите настройки IP-адреса, выполнив команды **ipconfig /release** и **ipconfig /renew**. В выходных данных должно быть указано, что ПК имеет IP-адрес из заданного Вами диапазона, маску подсети, шлюз по умолчанию (при его наличии) и адрес DNS-сервера.

После пингования оконечных устройств по IP-адресам проверьте ответ сервера при запросе на его доменное имя. Для этого в командной строке выполните команду **ping имя сервера**. На получение ответа от команды ping может уйти несколько секунд.

Для проверки работоспособности созданного HTTP сервера нужно открыть окно настройки одного из клиентских компьютеров и на вкладке **Desktop** запустить приложение **Web Browser**. Затем в строке URL задать IP-адрес созданного WEB-сервера и нажать на кнопку **GO**. Появление на экране текстового сообщения, которое было подготовлено на страничке index.html, свидетельствует о работоспособности Web-сервера.

Для исследования структуры пакетов при обмене данными между сетевыми устройствами нужно запустить Packet Tracer в режиме симуляции (иконка в правом нижнем углу) и открыть лист событий (нажать Event List). В окне Event List отображаются все пакеты, которые передаются в сети, с указанием источников и получателей пакетов и их типов. При нажатии на цветной квадратик Info исследуемого типа пакета данных (PDU) открывается окно с форматом пакета.

Для более детального анализа содержимого пакета необходимо нажать кнопку Inbound PDU Details (Сведения о входящем PDU) или Outbound PDU Details (Сведения об исходящем PDU).

## 6 Содержание отчета

- 6.1 Титульный лист.
- 6.2 Схема моделируемой сети.
- 6.3 Скриншоты топологии, реализованных настроек и результатов исследования функционирования сети с пояснениями полученных результатов.
- 6.4 Выводы.

## 7 Контрольные вопросы

- 7.1 Что представляют собой сетевые службы и зачем они предназначены?
- 7.2 Назовите протоколы прикладного уровня стека TCP/IP и поясните для чего используется тот или иной протокол.
- 7.3 Приведите формат IP-адреса протокола IPv4, назовите его принципиальное отличие от MAC-адреса.
- 7.4 С какой целью разработан протокол ARP и каков основной состав полей заголовка ARP-пакета?
- 7.5 Для чего предназначен протокол DNS и как решается проблема, если в данном DNS-сервере отсутствует запись соответствия символического и сетевого адресов?
- 7.6 Расскажите об особых (выделенных под специальные нужды) IP-адресах и их назначениях.
- 7.7 Что представляют собой локальные IP-адреса, назовите диапазоны сетей таких адресов. Для чего служит протокол сетевой трансляции адресов?
- 7.8 Опишите формат и использование маски подсети. Как по значению маски определить количество адресов, которое она выделяет? Перечислите известные Вам маски и их характеристики для сети класса C.
- 7.9 Что представляет собой технология бесклассовой междоменной маршрутизации? Запишите адрес и маску суперсети для 2000 хостов.
- 7.10 Для чего предназначен протокол DHCP и может ли компьютерная сеть функционировать без этого протокола?
- 7.11 Что представляет собой маршрут по умолчанию, для чего он используется? Каким образом маршрут по умолчанию указывается в таблице маршрутизации?
- 7.12 Чем отличается статическая маршрутизация от динамической? Приведите названия используемых протоколов динамической маршрутизации.
- 7.13 Какие действия происходят в сети при смене статической адресации на динамическую?
- 7.14 Как рационально спланировать установку DHCP-серверов в локальных и глобальных сетях?
- 7.15 Как на практике в эмуляторе Packet Tracer проверить содержимое заголовков пакетов?

**Приложение 1.** Таблица П1 – Варианты заданий для индивидуального моделирования локальных сетей и серверных служб

Вариант	Пользователи	Сервер HTTP	Сервер DNS	Сервер DHCP
1	2ПК+2ЛТ	Server0	Server1	Server2
2	3ПК+1ЛТ	Server1	Server0	Server0
3	2ПК+3ЛТ	Server2	Server2	Server0
4	4ПК+1ЛТ	Server0	Server0	Server1
5	3ПК+3ЛТ	Server2	Server1	Server1
6	4ПК+2ЛТ	Server1	Server1	Server2
7	3ПК+4ЛТ	Server0	Server0	Server2
8	4ПК+2ЛТ	Server0	Server0	Server1
9	5ПК+1ЛТ	Server2	Server1	Server1
10	5ПК+3ЛТ	Server1	Server1	Server2
11	4ПК+4ЛТ	Server0	Server0	Server2
12	3ПК+5ЛТ	Server0	Server1	Server2
13	4ПК+3ЛТ	Server1	Server0	Server0
14	5ПК+4ЛТ	Server2	Server2	Server0
15	2ПК+5ЛТ	Server0	Server0	Server1
16	3ПК+1ЛТ	Server2	Server1	Server1
17	2ПК+3ЛТ	Server1	Server1	Server2
18	4ПК+1ЛТ	Server0	Server0	Server2
19	3ПК+3ЛТ	Server0	Server0	Server1
20	4ПК+2ЛТ	Server2	Server1	Server1
21	3ПК+4ЛТ	Server1	Server1	Server2
22	5ПК+1ЛТ	Server0	Server0	Server2

### Библиографический список

1. Баскаков И. Построение коммутируемых компьютерных сетей / И. Баскаков, А. Пролетарский, Е. Смирнова, Р. Федотов. Национальный Открытый Университет "ИНТУИТ": <http://www.intuit.ru/studies/courses/3591/833/info>.
2. Дибров М.В. Сети и телекоммуникации. Маршрутизация в IP–сетях. В 2 ч. Часть 2: учебник и практикум для академического бакалавриата / М.В. Дибров. – М.: Изд-во Юрайт, 2019. – 351 с. <https://biblio-online.ru/book/seti-i-telekommunikacii-marshrutizaciya-v-ip-setyah-v-2-ch-chast-2-437865>
3. Олифер В., Олифер Н. Компьютерные сети. Принципы, технологии, протоколы / В.Олифер, Н.Олифер. 5-е изд. — СПб.: Питер, 2016. — 992 с.
4. Сети и телекоммуникации: учебник и практикум для академического бакалавриата / Под ред. К.Е. Самуйлова, И.А. Шалимова, Д.С. Кулябова. – М.: Изд-во Юрайт, 2016. – 363 с. <https://biblio-online.ru/book/seti-i-telekommunikacii-432824>
5. Чернега В.С. Компьютерные сети / В.С. Чернега, Б. Платтнер. — Севастополь: Изд-во СевНТУ, 2006. — 500 с.

