МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Севастопольский государственный университет

ИССЛЕДОВАНИЕ СПОСОБОВ ПОСТРОЕНИЯ ВИРТУАЛЬНЫХ ЛОКАЛЬНЫХ КОМПЬЮТЕРНЫХ СЕТЕЙ

Методические указания
к выполнению лабораторной работы №3
по дисциплине

“Архитектура

инфокоммуникационных систем и сетей”

Для студентов, обучающихся по направлению 09.03.02
"Информационные системы и технологии"
по учебному плану подготовки бакалавров
дневной и заочной форм обучения

Севастополь
2017

Рецензент:
Моисеев Д.В., канд. техн. наук, доцент кафедры ИТиКС

Составители: В.С. Чернега, А.В. Волкова.

Исследование способов построения виртуальных локальных компьютерных сетей. Методические указания к лабораторным занятиям по дисциплине «Архитектура (структуры и протоколы) инфокоммуникационных систем и сетей» / Сост., В.С. Чернега, А.В. Волкова – Севастополь: Изд-во СевГУ, 2017 – 22 с.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Архитектура (структуры и протоколы) инфокоммуникационных систем и сетей». Целью методических указаний является помощь студентам в исследовании принципов построения виртуальных локальных сетей и работы протокола VTP. Излагаются теоретические и практические сведения необходимые для выполнения лабораторной работы, требования к содержанию отчета.

Методические указания рассмотрены и утверждены на заседании кафедры информационных систем и методическом семинаре института информационных технологий и управления в технических системах (протокол № 2 от 11 мая 2017 г.)

Издательский номер 65/17

1 Цель работы

Исследование принципов работы коммутаторов и виртуальных локальных сетей, способов конфигурировании коммутаторов cisco для построения виртуальных локальных сетей, изучение принципов работы протокола VTP.

2 Теоретическая часть

2.1 Коммутация в локальных сетях. Основные определения

Структурированная кабельная система (СКС) – это все провода, розетки, патчпанели и патчкорды, то есть грубо говоря, это физика вашей сети в узком смысле, в широком – это совокупность сетей: ЛВС, телефонные сети, системы видеонаблюдения и прочее. Это отдельный очень большой и порой сложный пласт знаний и технологий, который вообще не имеет точек пересечения с настройкой.

Локальная сеть (LAN – Local Area Network) – в первом приближении – это сеть вашей организации. При втором приближении, локальной называют сеть, которая находится под управлением одного сетевого администратора. То есть, например, вы отвечаете за районный сегмент сети провайдера, в таком случае ваша районная сеть со всеми подсетями будет являться локальной, в то время, как вышестоящая сеть и сети других районов уже нет, так как за них отвечает уже другие люди. С точки зрения абонента какого-то провайдера, локальная сеть – это все, что находится до домашнего роутера.

Широковещательный домен – область сети, в которой происходит обмен широковещательными сообщениями, и устройства могут отправлять друг другу сообщения непосредственно, без участия маршрутизатора.

Например, компьютер отправляет широковещательный запрос в сеть в поисках DHCP-сервера. Этот *фрейм* (*кадр*) адресован всем устройствам и имеет MAC-адрес получателя FF:FF:FF:FF:FF:FF. Сначала он попадает на коммутатор, с которого его копии рассылаются на все порты. Потом часть попадает на другие компьютеры, часть уходят в соседние коммутаторы, кто-то доходит до маршрутизатора, а одну копию принимает-таки DHCP-сервер. Участок сети, внутри которого могут жить эти кадры и называется широковещательным доменом. А заканчивают свою жизнь они на конечных хостах (компьютеры, серверы) или на маршрутизаторах, которые их отбрасывают, если они им не предназначены.

Если же на коммутаторе заведены VLAN'ы, то они также разделяют широковещательные домены, потому что пакет между ними обязательно должен проходить через маршрутизатор, который отбросит широковещательные сообщения. Таким образом, один VLAN – это один широковещательный домен.

Существует три способа разграничить широковещательные домены:

- поставить маршрутизатор и разнести хосты в разные подсети;
- разделить сеть VLAN'ами;
- порвать кабель.

2.2 Принцип работы коммутатора

Рассмотрим подробно работу двух нижних уровней модели OSI.

Первый уровень – *физический* – это представление информации в виде сигналов – битов. Задача этого уровня сгенерировать электрический, оптический или радиосигнал, передать его в среду и принять его. К нему относится вся физика: интерфейсы,

кабели, антенны, медиаконвертеры (конвертеры среды), репитеры, старые хабы. В общем, это низкоуровневая работа. Это первый уровень модели OSI и стека TCP/IP.

Второй уровень – *канальный*. На этом уровне работают коммутаторы. Идентификатор устройства здесь, это MAC-адрес. У каждого узла (компьютер, маршрутизатор, ноутбук, IP-телефон, любой Wi-Fi-клиент) есть этот уникальный адрес, который однозначно определяет устройство в локальной сети. В теории MAC-адреса не должны повторяться вообще, но на практике такое случается и в рамках одного широковещательного домена может приводить к сложно отлавливаемым проблемам.

Наиболее известным протоколом этого уровня является Ethernet. Данные на этом уровне передаются кусками, каждый из которых называется Ethernet-фрейм (Ethernet-кадр или PDU канального уровня).

При попытке пропинговать, например, адрес соседнего компьютера командой ping в пакет ICMP инкапсулируется информация от приложения – это означает, что к данным 5-го уровня добавляется заголовок со служебной информацией 4-го уровня. Его данные упаковываются (инкапсулируются) в IP-пакеты, где в заголовке указан IP-адрес получателя и IP-адрес отправителя – логические адреса. Затем всё это инкапсулируется в Ethernet-кадры с MAC-адресами отправителя и получателя – физическими адресами. При формировании кадров в заголовке в качестве MAC-адреса источника (source) подставляется адрес компьютера-отправителя, а адресом получателя (destinantion) будет MAC-адрес компьютера – владельца IP-адреса получателя. В последнюю очередь сетевая карта вашего компьютера дробит фрейм на биты и отправляет их в кабель.

Коммутатор из поступивших битов собирает первоначальный кадр. Далее из заголовка извлекается адрес получателя, проверяется таблица MAC-адресов на предмет совпадения и, как только оно найдено, кадр без изменений отправляется в указанный порт. Если же адреса пока ещё нет или кадр пришёл широковещательный, то он направляется на все порты, кроме того, откуда пришёл. Если адреса отправителя в таблице до сих пор не было, то в этот момент коммутатор добавит его. Естественно, кадр опять передаётся в виде битов – это закон электроники, просто необходимо всегда иметь это в виду.

Конечный хост, получив поток битов, собирает из них кадр, ещё только предполагая, что он предназначается ему. Далее он сравнивает MAC-адрес получателя со своим и, если они совпадают, то заголовок второго уровня отбрасывается, а IP-данные передаются на обработку вышестоящему протоколу. Если адреса не совпадают, то кадр отбрасывается вместе со всем содержимым. Далее сравниваются IP-адрес получателя и этого устройства. Если совпадают, то заголовок сетевого уровня отбрасывается, и данные передаются транспортному уровню (ICMP).

Конечный хост обработал ICMP-запрос (echo-request) и готов послать ICMP-ответ (echo-reply) компьютеру-отправителю со своим IP-адресом и далее указанные выше действия повторяются уже для нового кадра.

2.3 Возможности и разновидности коммутаторов

Коммутаторы подразделяются на управляемые и неуправляемые.

Неуправляемый коммутатор автоматически распределяет скорость и трафик между всеми клиентами сети. Использование неуправляемого коммутатора является идеальным решением для малых и средних сетей с небольшим количеством подключенных пользователей. Достоинством является простота в управлении и подключении.

По сути, современные коммутаторы отличаются только тем, что в управляемых реализован интерфейс доступа к настройкам, а в неуправляемых они жестко прописы-

ваются производителем во время изготовления. Однако это не значит, что все коммутаторы одинаковы – большинство моделей адаптировано под конкретные цели. Например, часть портов имеет ограничение по скорости передачи трафика, для того, чтобы иметь возможность подключить к ним неприоритетное устройство, не нарушая при этом работу остальных клиентов сети. Каждый производитель представляет свой модельный ряд неуправляемых коммутаторов.

Управляемые коммутаторы позволяют управлять коммутацией на сетевом (третьем) уровне модели OSI. Обычно их именуют соответственно, например, «Layer 3 Switch» или сокращенно «L3 Switch». Управление коммутатором может осуществляться посредством Web-интерфейса, интерфейса командной строки (CLI), протокола SNMP, RMON и т.п.

Многие управляемые коммутаторы позволяют настраивать дополнительные функции: VLAN, QoS, агрегирование, зеркалирование. На данный момент многие коммутаторы уровня доступа обладают такими расширенными возможностями, как сегментация трафика между портами, контроль трафика на предмет штормов, обнаружение петель, ограничение количества изучаемых MAC-адресов, ограничение входящей/исходящей скорости на портах, функции списков доступа и т.п.

Примеры использования управляемых коммутаторов:

1. Есть небольшая локальная сеть, в которой компьютеры подключены к коммутатору, а тот в свою очередь подключен к серверу. Интернет раздается через этот сервер. На сервере, соответственно, нужно подсчитать, кто сколько трафика потратил. В подсчете трафика будем ориентироваться по IP-адресу компьютера (различные авторизаторы в данном рассматриваемом примере не используем). Однако, в этой небольшой локальной сети некто может взять и присвоить своему компьютеру IP-адрес соседа, тогда тот трафик, который этот некто потратит, будет записан на счет соседа (и при этом не сильно поможет даже проверка на сервере IP-адреса и MAC-адреса сетевой карты, потому как и MAC-адрес сменить тоже не проблема). Избежать возникновения такой ситуации позволит использование управляемого коммутатора. В данном случае, привязываем порт коммутатора к IP-адресу и если кто-то изменит IP, чтобы в интернете на халяву посидеть, то у него уже ничего не выйдет, потому как умный свитч его IP заблокирует (port-security).

2. С помощью управляемого коммутатора можно отделить одних пользователей от других, чтобы первые не видели вторых и наоборот, например, на работе можно отделить бухгалтерию от простых смертных ну, и т.д. Соответственно, если всех подключить к неуправляемому коммутатору, то все будут видеть всех, а с помощью управляемого коммутатора этого можно избежать (VLAN).

2.4 Общие сведения о VLAN

VLAN (Virtual Local Area Network – виртуальная локальная сеть) – это технология, при которой логические адреса устройств в сети делятся на сегменты в зависимости от функций, выполняемых устройствами, приложений и требований управления. Действуя таким образом, можно сформировать виртуальные локальные группы, не зависящие от физического расположения устройств, в них входящих. Анонсированный IEEE протокол IEEE 802.1Q стандартизует реализацию VLAN, функции VLAN коммутатора поддерживают стандарт IEEE 802.1Q.

Основная идея технологии VLAN состоит в динамическом разделении всей локальной сети на множество отдельных областей вещания (см. рисунок 1.1) в соответствии с требованиями, предъявляемыми к сети.

Функция VLAN позволяет осуществить взаимодействие двух и более сетевых устройств на канальном уровне, хотя физически данные устройства, могут быть подключены к разным коммутаторам или маршрутизаторам (а ведь канальный уровень отвечает за передачу данных между устройствами, подключенными к одному сетевому сегменту).

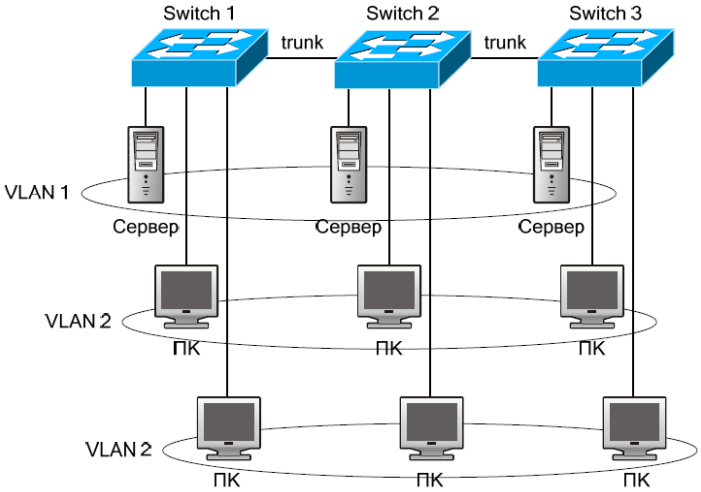

Рисунок 1.1 – Сеть VLAN, определенная логически

VLAN ведут себя так же, как и физически разделённые локальные сети. То есть после разбивки сети на VLAN мы получим несколько локальных сетей, которые далее необходимо объединить в единое целое с помощью маршрутизации на третьем сетевом уровне.

Каждая область вещания представляет собой VLAN. 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; вещательный, 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.

2.4.1 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 при-

сваивание 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.

2.4.2 Коммутатор и VLAN

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

2.5 Принцип коммутации

Внутрь фрейма после Source MAC-адреса добавляется ещё одно поле, очень грубо говоря, 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.

2.5.1 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, приходящий 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 переопределить 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.

2.5.2 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 1.2.

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. Портам 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 соеди-

нение) между коммутаторами, который представляет собой соединение двух нетегированных портов находящихся в области видимости каждого VLAN'а соответственно.

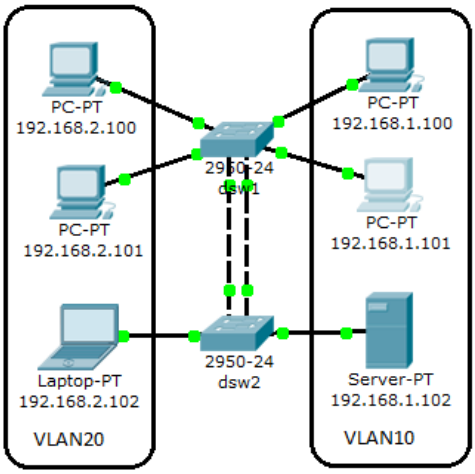

Рисунок 1.2 – Реализация нескольких VLAN

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 хосты в один широковещательный сегмент. Для решения этой проблемы используются тегированные порты (см. рисунок 1.3).

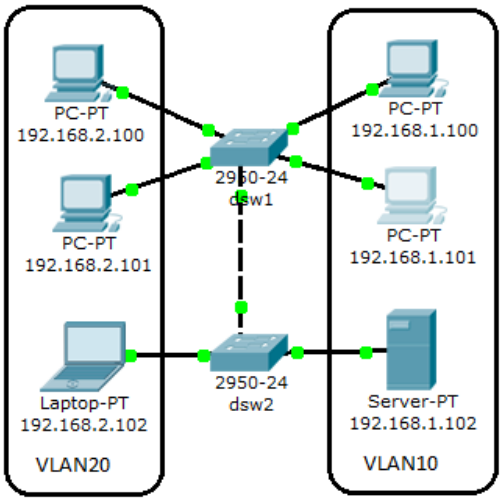

Рисунок 1.3 – Тегированный порт между коммутаторами

2.6 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'а есть номер. Существуют два типа VLAN:

– стандартный диапозон VLAN – от 1 до 1000;

– расширеный диапозон 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 необходиимый порт отметим как член 2-го VLAN'а 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, автоматический тегируется: на него вешается ленточка 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 ленточка с кадра отвязывается, 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 ленточка 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 прилетит 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, планшетам, телефонам) 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, свитч его отбросит из-за увеличенного размера кадра. 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
```

2.7 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'a. 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
```

2.8 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.

2.8.1 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'a. А кадры, входящие на физический интерфейс FastEthernet0/0 с тегом этого VLAN'a 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.

2.8.2 Пример настройки с использованием маршрутизатора

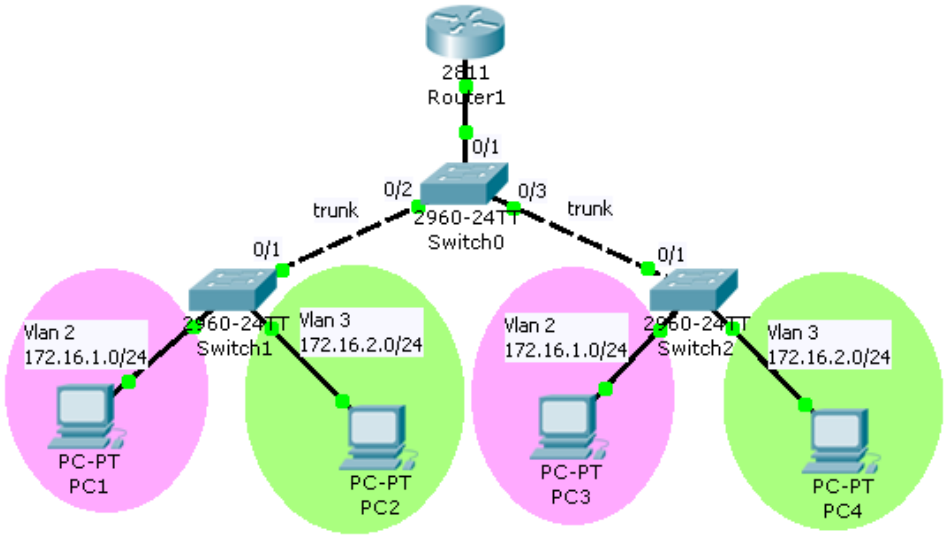

Рисунок 1.4 – InterVlan routing на маршрутизаторе

```
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
```

2.8.3 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
```

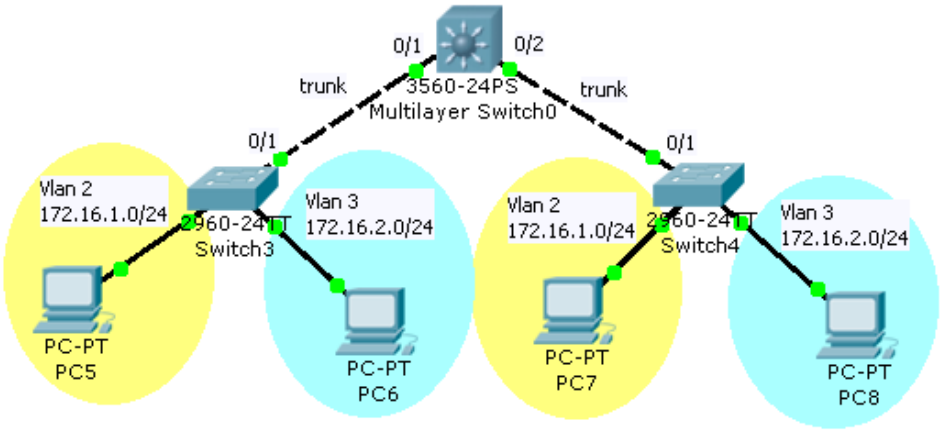

Рисунок 1.5 – InterVlan routing на коммутаторе L3-уровня

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
```

2.9 Изучение 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 десяток, 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.

2.9.1 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 `vtp 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.

2.9.2 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`, обзначающий, 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.

2.9.3 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).

2.9.4 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. Его можно просто разово заснифить и добавлять в VTP-сообщения для придания им «легитимности».

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 – прото-

кол 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.

2.9.5 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.

2.9.6 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 – A и B. Коммутатор A имеет роль VTP Server, а B – VTP Client. Между ними – транковый канал, 802.1Q. На коммутаторе B включен vtp pruning. Допустим, на коммутаторе A в базу VLAN добавлены VLAN 10

и VLAN 20. Соответственно, коммутатор A уведомит по протоколу VTP своего соседа – B – о новой ревизии базы VLAN'ов. Сосед B добавит эти VLAN'ы в базу и теперь, когда подключенный к коммутатору A клиент, например, передаст броадкаст в VLAN'е 10, этот броадкаст дойдёт и до коммутатора B. Невзирая на то, что у коммутатора B может вообще не быть ни одного порта и интерфейса в VLAN 10, а также не быть других транков (т.е. трафик 10-го VLAN'а коммутатору B совсем не нужен). Вот в данном случае механизм pruning сможет сэкономить полосу пропускания канала между коммутаторами A и B просто не отправляя трафик неиспользуемого VLAN'а коммутатору B.

2.9.7 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.

2.9.8 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`.

2.9.9 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). Дефолтная настройка коммутатора – это режим VTP Server (т.е. когда Вы достаёте коммутатор из коробки, Вы сразу можете на нём создавать VLAN'ы, в случае VTP Client это было бы невозможно).

Соответственно, возможна неприятная ситуация. Состоит она в том, что можно взять коммутатор, заранее его сконфигурировать, внеся больше изменений, чем есть сейчас в инфраструктурном VTP, задать правильные параметры домена и подключить к сети. Тогда коммутатор своей базой затрёт существующую. Почему так произойдёт и как это может быть? Рассмотрим подробнее.

Вы покупаете новый коммутатор и вводите его в эксплуатацию. Отдельно от других, которые работают в VTP. Ну, вот так сложилось – допустим, в филиале организа-

ции его ставите, где он не является непосредственно подключенным к другим коммутаторам. Начинаете с ним работать. Работаете интенсивно – добавляете на него VLAN'ы, удаляете их, переименовываете. Каждое действие плюсует единицу к revision number. Вдруг через некоторое время возникает ситуация – филиал закрывается. Коммутатор перевозят в основной офис, и Вы подключаете его к другим. И тут выясняется следующее. У данного коммутатора ревизия численно выше, чем у местного VTP Server. Имя домена и пароль совпадают. Как только транк поднимается, новый коммутатор выстреливает advertisement, который начинают слушать все остальные коммутаторы и ретранслировать дальше. Это штатный функционал – Вы не можете ограничить получение VTP-данных только от одного, «правильного» сервера. Соответственно, эта волна накрывает все коммутаторы в режиме Client и тот, который в Server. Это тоже штатное поведение – VTP Server, получив VTP-данные с большим номером ревизии, чем у себя, перезаписывает свою базу. И всё, Вы имеете большие проблемы – вместо Вашей базы VLAN'ов у Вас на всех коммутаторах та, которая была в филиале.

Чтобы избежать этого, можно поступить по-разному. Например, не делать у этого коммутатора имя домена и пароль, как в основной VTP-сети. Но можно и проще – ведь чтобы этого всего не произошло, надо просто сбросить номер ревизии. Для этого достаточно переименовать домен у коммутатора в какое-нибудь временное название и после вернуть назад. Ревизия сбросится. После не забудьте включить режим VTP Client.

Примечание: Как понятно, именно из-за этой ситуации использование VTP в production-сети с высоким уровнем безопасности является нежелательным. Злоумышленник может провести достаточно простую атаку – ему хватит доступа к транковому порту и возможности отправить, допустим, VTP-уведомление о том, что пришла база с версией 10000 и одним VLAN'ом, и всё – вся VTP-инфраструктура примет это как нормальное положение вещей и остальные VLAN'ы пропадут. Поэтому в безопасных сетях все коммутаторы работают в VTP transparent, где такая ситуация невозможна в принципе.

2.9.10 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.

Про этот стандарт и его основное применение (в протоколах семейства Spanning Tree), думается, лучше написать отдельно, а для нас будет достаточно то, что он меняет формат BID и что его надо включать, чтобы можно было переключиться в VTPv3. 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 изменят 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 1.6). При каждом добавлении нового коммутатора или новой команды необходимо просматривать текущую конфигурацию VLAN коммутаторов с помощью команды `show vlan`.

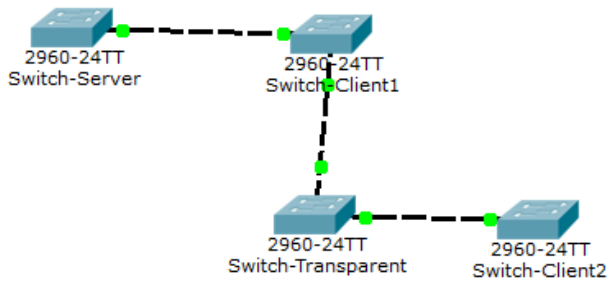

Рисунок 1.6 – Изучение работы протокола VTP

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

Реализовать схему, представленную на рис. 2.7 и 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.

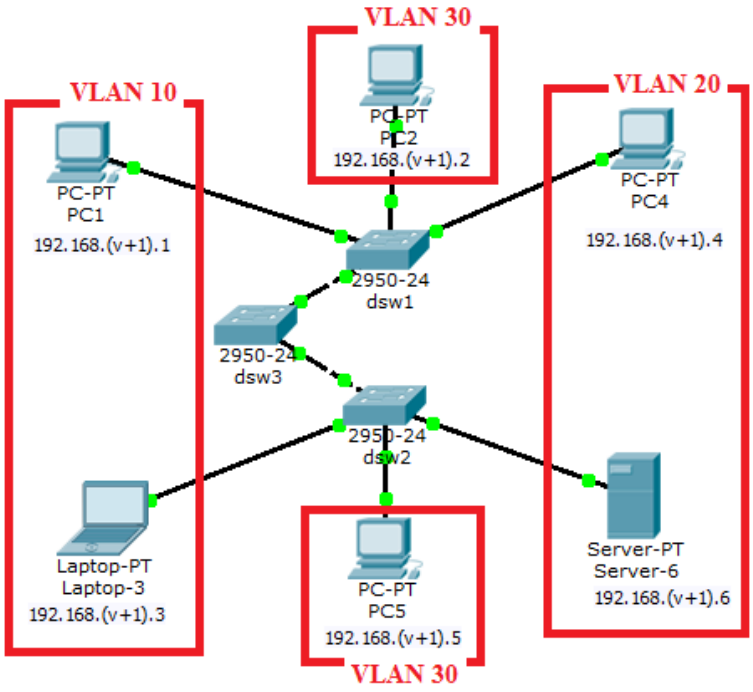

Рисунок 1.7 – Изучение работы протокола VTP

Задание №3. Настройка interVLAN routing с помощью маршрутизатора

Возьмите за основу топологию сети из задания №2. Измените ip-адрса хостов в соответствии с вариантом (v – номер по списку в журнале) и добавить роутер, как показано на рис. 2.8. 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.

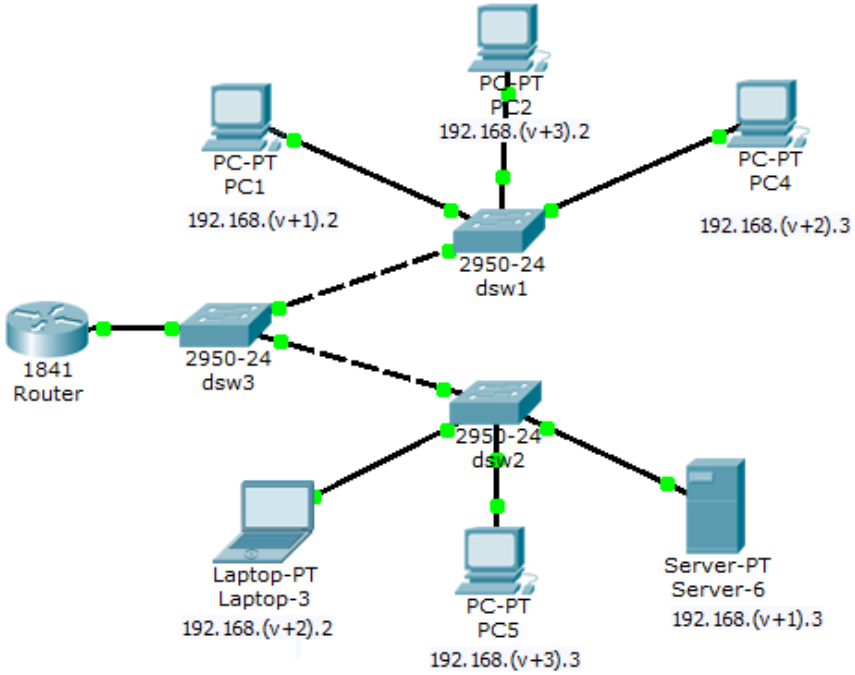

Рисунок 1.8 – InterVlan routing на маршрутизаторе

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.

4 Содержание отчета

1. Титульный лист.
2. Исходные данные в соответствии с индивидуальным вариантом.
3. Описание всех использованных команд.
4. Скриншоты получившихся 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. Что такое магистральные (trunk) порты и порты доступа (access)?
4. Что такое interVLAN routing?
5. Способы организации interVLAN routing?
6. Зачем использовать протокол VTP?
7. Принцип работы протокола VTP?