Министерство образования Республики Беларусь
Учреждение образования
«Белорусский государственный университет
информатики и радиоэлектроники»

Факультет инфокоммуникаций

Кафедра инфокоммуникационных технологий

В. А. Ковшик, В. Н. Мищенко, В. В. Рабцевич

ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ В ИНФОКОММУНИКАЦИОННЫХ СИСТЕМАХ

Рекомендовано УМО по образованию в области информатики и радиоэлектроники в качестве учебно-методического пособия для специальности 1-45 80 01 «Системы и сети инфокоммуникаций»

Минск БГУИР 2021

УДК 621.391(075.8)
ББК 32.88я73
К56

Р е ц е н з е н т ы:

кафедра связи учреждения образования
«Военная академия Республики Беларусь»
(протокол №14 от 24.02.2020);

заведующий кафедрой телекоммуникационных систем
учреждения образования
«Белорусская государственная академия связи»
кандидат технических наук, доцент С. И. Половеня

Ковшик, В. А.

К56 Технологии передачи данных в инфокоммуникационных системах : учеб.-метод. пособие / В. А. Ковшик, В. Н. Мищенко, В. В. Рабцевич. – Минск : БГУИР, 2021. – 148 с. : ил.

ISBN 978-985-543-609-7.

Представлены технологии передачи данных в инфокоммуникационных системах.

Предназначено для магистрантов очной и заочной форм обучения специальности 1-45 80 01 «Системы и сети инфокоммуникаций».

УДК 621.391(075.8)
ББК 32.88я73

ISBN 978-985-543-609-7

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Основная задача учебно-методического пособия – дать студентам общие систематизированные сведения об организации и особенностях технологий передачи данных в инфокоммуникационных системах. Инфокоммуникационные сети представляют собой комплекс аппаратных и программных средств, обеспечивающих передачу информационных сообщений между абонентами с заданными параметрами качества. При этом формируется определенный маршрут, по которому передается сообщение. Процесс формирования маршрута сопровождается процедурами коммутации и маршрутизации информации в транзитных узлах, при которых происходит передача (продвижение) сообщения с входного интерфейса на выходной. Рассматриваются общие принципы межсетевого взаимодействия инфокоммуникационных систем.

В учебном издании представлен комплекс лабораторных работ, который дает возможность изучить построение инфокоммуникационных сетей передачи данных, методы и особенности функционирования коммутаторов, их конструктивное исполнение и примеры реализации интерфейсов при создании сетей различного назначения. Развитие сетей передачи данных неразрывно связано с использованием технологии Ethernet, которая и в настоящее время остается одной из самых распространенных.

В первой лабораторной работе рассмотрена организация удаленного доступа к коммутатору по протоколу Telnet, команды первичной настройки коммутатора, во второй лабораторной работе – особенности формирования виртуальных локальных сетей по протоколу 802.1Q, в третьей лабораторной работе – протоколы связующего дерева. В четвертой лабораторной работе приводятся правила и процедуры адресации сетевого уровня, маршрутизации при передаче данных между информационными системами. В пятой лабораторной работе описана настройка статической и динамической маршрутизации на коммутаторах третьего уровня, в шестой лабораторной

работе – функции повышения качества обслуживания в современных сетях, которые необходимы для обеспечения гарантированного и дифференцированного уровня обслуживания сетевого трафика, запрашиваемого теми или иными приложениями на основе различных механизмов распределения ресурсов, ограничения интенсивности трафика, обработки очередей и приоритизации. В седьмой лабораторной работе изучаются функции обеспечения безопасности и ограничения доступа к сети, их достоинства и недостатки.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

RIP (Routing Information Protocol) – протокол обмена информацией

OSPF (Open Shortest Path First) – открытый маршрут передачи информации

BGP (Border Gateway Protocol) – протокол граничного узла

VLAN (Virtual Local Area Network) – виртуальные локальные сети

QoS (Quality of Service) – качество обслуживания

ACL (Access Control List) – списки управления доступом

CIST (CIST Regional Root) – региональный корневой мост

TDM (Time Division Multiplexing) – принцип мультиплексирования с разделением по времени

IP (Internet Protocol) – протокол Интернета

 – коммутатор

 – маршрутизатор

 – компьютер

 – сетевая среда

 – сервер

 – пользователь

Лабораторная работа №1

ОРГАНИЗАЦИЯ УДАЛЕННОГО ДОСТУПА К КОММУТАТОРУ ПО ПРОТОКОЛУ TELNET. ИЗУЧЕНИЕ КОМАНД ПЕРВИЧНОЙ НАСТРОЙКИ КОММУТАТОРА

Цель работы: изучить организацию доступа к коммутатору и основные команды управления через Command Line Interface (CLI), а также функции программного обеспечения Wireshark.

1.1. Особенности функционирования устройств на канальном уровне

Работа коммутаторов сетей передачи данных начинается с построения таблицы коммутации. Первоначально она пуста. При подключении устройств и начале передачи данных коммутатор изучает подключенные к нему сетевые устройства с помощью MAC-адресов, которые он получает от источников принимаемых кадров.

Записи в таблице коммутации создаются динамически, т. е. при прочтении коммутатором нового MAC-адреса он будет занесен в таблицу коммутации вместе с ассоциированным с ним портом. Дополнительно в таблицу коммутации для записей записывается время старения, которое позволяет в автоматическом режиме реагировать на перемещение, добавление или удаление сетевых устройств [1].

Вторым типом построения таблиц коммутации является статическая форма записи, которая позволяет повышать безопасность за счет гарантированного подключения устройств только с разрешенными MAC-адресами. Процесс пересылки кадров между портами коммутатора при добавлении записи в таблицу коммутации показан на рис. 1.1.

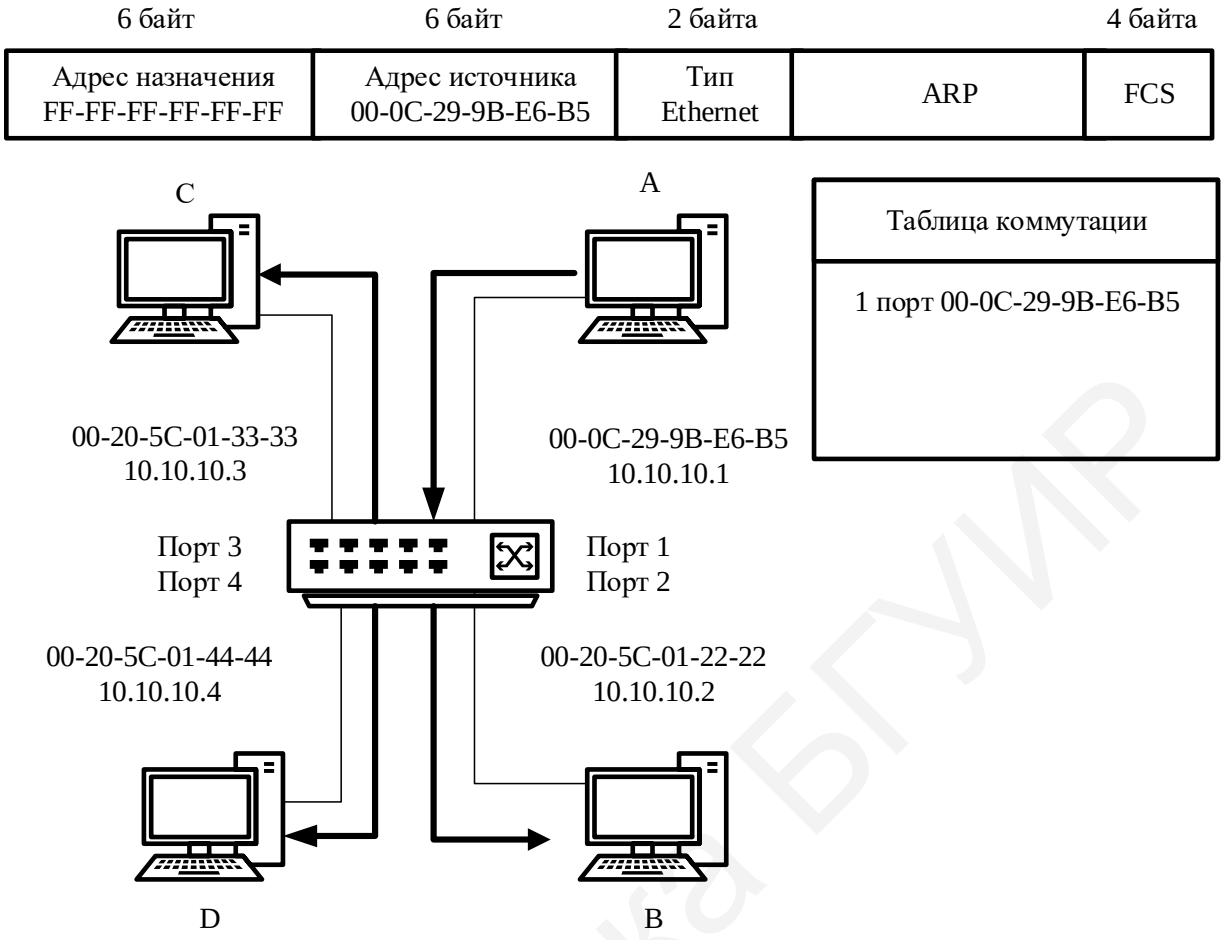


Рис. 1.1. Построение таблицы коммутации

В том случае если MAC-адрес приемника в таблице коммутации отсутствует, коммутатор создает копии этого кадра и рассылает их на все свои порты, кроме порта источника. Этот процесс называется *лавинной передачей*. После получения каждого кадра производится анализ его содержимого, и только после этого коммутатор принимает решение о возможности передачи.

В настоящее время в коммутаторах могут быть три сценария коммутации при получении кадра: с промежуточным хранением, при котором производится проверка содержимого пакетов, сквозной метод, при котором происходит считывание только адреса назначения, и гибридный.

В зависимости от выполняемых задач коммутаторы могут быть оборудованы различным количеством и типом портов. В табл. 1.1 приведены

типы наиболее часто используемых интерфейсов и их основные характеристики в соответствии со стандартом IEEE 802.3-2008.

Все компьютерные сети строятся в соответствии с иерархической моделью, которая определяет подходы к проектированию различных компьютерных сетей и состоит из трех логических уровней (рис. 1.2): уровень доступа, уровень распределения, уровень ядра [1].

Таблица 1.1

Типы кабелей

Стандарт	Тип кабеля	Максимальное расстояние передачи, м
1	2	3
10Base-T	Витая пара (категория 3 или 5)	100
100Base-TX	Витая пара (категория 5)	100
100Base-FX	Многомодовый оптический кабель	412 (полудуплекс) 2000 (дуплекс)
100Base-BX 10	Одноволоконный одномодовый оптический кабель (длина волны: 1310 нм – восходящий поток, 1550 нм – нисходящий)	10 000
100Base-LX 10	Одноволоконный одномодовый оптический кабель (длина волны – 1310 нм)	10 000
1000Base-T	Витая пара (категория 5, 5e, 6 или 7)	100
1000 Base-SX	Многомодовый оптический кабель (62,5/125 мк /50/125 мк)	220/550
1000 Base-LX	Многомодовый оптический кабель или одномодовый оптический кабель	550 5000
1000 Base-LX10	Многомодовый оптический кабель (длина волны – 1310 нм) или одномодовый оптический кабель (длина волны – 1310 нм)	550 10 000

Окончание табл. 1.1

1	2	3
1000 Base-BX10	Одноволоконный одномодовый оптический кабель (длина волны: 1310 нм – восходящий поток, 1550 нм – нисходящий)	10 000
1000 Base-ZX	Одномодовый оптический кабель (длина волны 1310 нм)	80 000
1000 Base-LH	Одномодовый оптический кабель	50 000
10GBase-CX4	Экранированный сбалансированный медный кабель	15
10GBase-SR	Многомодовый оптический кабель	300
10GBase-LR	Одномодовый оптический кабель	10 000
10GBase-ER	Одномодовый оптический кабель	40 000
10GBase-ZR	Одномодовый оптический кабель	70 000

Каждый уровень выполняет определенные функции. Для организации модели из трех уровней необязательно использовать три различных устройства, можно применить одно устройство, выполняющее все функции двух соседних уровней.

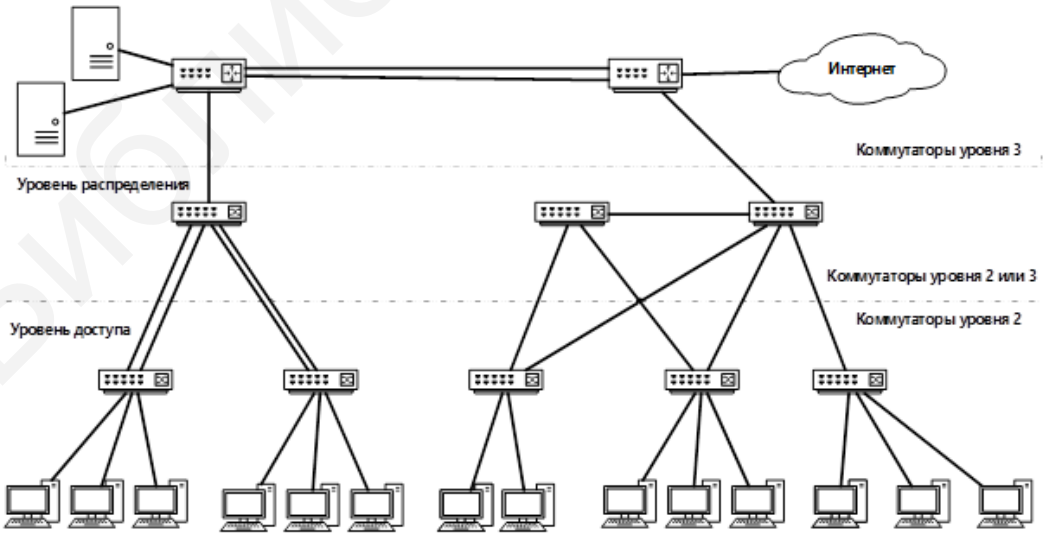


Рис. 1.2. Трехуровневая модель сети

На уровне ядра происходит передача больших объемов трафика, являющегося общим для пользователей. Передаваемые данные обрабатываются на уровне распределения, который при необходимости пересылает запросы к ядру. Уровень ядра является единой точкой отказа, которая может привести к потере связности между уровнями.

Уровень распределения отвечает за организацию маршрутизации, безопасности и необходимого качества обслуживания, также на нем возможно использование агрегирования каналов и переходов между различными технологиями (например, от 100Base-TX к 1000Base-T).

Уровень доступа представляет пользователям и рабочим группам доступ к ресурсам объединенной сети. Основная задача уровня – создание точек входа или выхода пользователей в сеть. Уровень выполняет следующие функции [1]:

- управление доступом пользователей и политиками сети;
- создание отдельных доменов коллизий (сегментация);
- подключение рабочих групп к уровню распределения;
- использование технологии коммутируемых локальных сетей.

1.2. Указания по выполнению лабораторной работы

Управление коммутатором возможно через интерфейс командной строки и Web-интерфейс. В данном цикле лабораторных работ настройка оборудования будет происходить с помощью протокола Telnet. Оперативная система (ОС) Windows имеет уже встроенный Telnet-клиент.

Для выполнения лабораторной работы произведите подключение к коммутатору с помощью протокола Telnet. В операционной системе Windows для подключения к коммутатору по протоколу Telnet можно использовать встроенный Telnet-клиент. Перед настройкой коммутатора необходимо произвести настройку сетевой карты компьютера. Компьютеры,

используемые в лабораторных работах, имеют две сетевые карты. Для выполнения лабораторного задания достаточно настроить одну из них.

По умолчанию в коммутаторах D-link DES 1210 используется IP-адрес 10.90.90.90 с маской подсети 255.255.255.0. Соответственно сетевая карта может получить любой IP-адрес из диапазона 10.90.90.1–10.90.90.254 за исключением IP-адреса по умолчанию, который уже зарезервирован за коммутатором.

Telnet-клиент в ОС Windows 10 по умолчанию выключен. Чтобы его активировать, нажмите *Пуск → Параметры → Приложения → Программы и компоненты → Включение или отключение компонентов Windows*. Установите флажок рядом с компонентом Клиент Telnet и нажмите кнопку *OK*.

Подключите рабочую станцию к коммутатору с помощью Ethernet-кабеля и настройте статический IP-адрес в заданном диапазоне, значение последнего октета укажите в соответствии с номером ПК, за которым выполняется лабораторная работа. На рабочей станции запустите анализатор трафика Wireshark. Выберите интерфейс локальной сети для захвата трафика. Подключитесь к интерфейсу командной строки коммутатора через Telnet. Откройте командную строку Windows (для включения командной строки Windows необходимо одновременно нажать сочетание клавиш win+R) (рис. 1.3) и введите `telnet 10.90.90.90`.

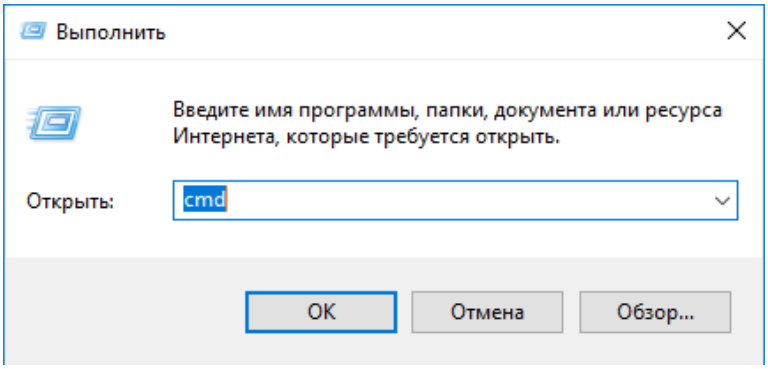


Рис. 1.3. Открытие командной строки в Windows

В окне командной строки после приглашения *UserName* нажмите кнопку *Enter*, после приглашения *PassWord* снова нажмите кнопку *Enter*.

Захват сетевого трафика с помощью Wireshark

Запустите программный модуль Wireshark. В открывшемся диалоговом окне выберите активный интерфейс либо необходимый вам, если активных несколько. Выполните настройку отображения захвата трафика через команду меню *Захват → Опции*. В открывшемся диалоговом окне устанавливаются следующие параметры захвата пакетов:

- прописать место сохранения будущего захваченного файла в окне *Вывод → Файл*;
- выбрать интерфейс, с которого будет происходить захват сетевого трафика;
- при необходимости добавить разрешение имен и дополнительные опции.

Захват трафика происходит через команду меню *Захват → Старт*.

Фильтрация пакетов

После выбора интерфейса и захвата трафика выводится окно захвата пакетов (рис. 1.4). Описание полей окна представлено в табл. 1.2.

В целях создания нового профайла для анализа захваченного трафика в командной строке выбираем *Редактирование → Конфигурационные профили → New* и даем название профиля.

При необходимости с помощью командной строки *Редактирование → Параметры → Layout* выбираем в левом поле списка нужные функции нового профиля.

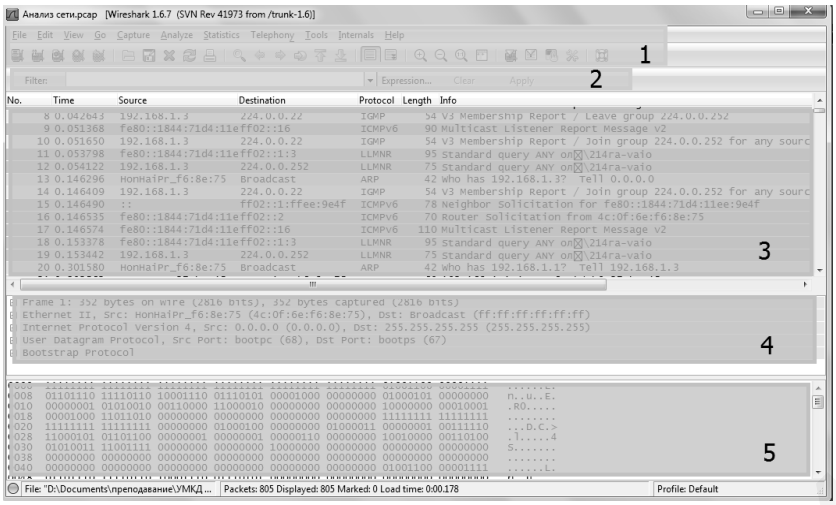


Рис. 1.4. Окно захваченных пакетов

Таблица 1.2

Описание полей окна захвата

Выделенная область	Описание
1	Меню программы и панель инструментов наиболее часто используемых функций программы
2	Фильтр захваченных пакетов
3	Поле списка краткой информации по всем захваченным PDU (Packet list). Столбцы поля 3 показывают: **No.** – номер пакета в файле захвата; **Time** – временная отметка пакета; **Source** – адрес отправителя (откуда пришел пакет); **Destination** – адрес получателя (куда пакет пойдет); **Protocol** – название протокола в сокращенной версии; **Info** – дополнительная информация о содержании пакета
4	Информационное поле отображения подробной информации по конкретно выбранному PDU (Packet Details)
5	Поле отображения данных, выделенных в информационном поле в шестнадцатеричной и текстовой форме (Packet Bytes)

Фильтрация пакетов по определенным протоколам может применяться как при захвате трафика в реальном времени, так и при его анализе в случае сохранения в файле захвата.

Для применения фильтра необходимо:

1) ввести фильтр в поле ввода;

2) нажать кнопку *Apply*.

Поле для ввода фильтра может менять цвет в зависимости от того, что было набрано:

- зеленый цвет означает, что все в порядке;

- красный – допущена ошибка;

- желтый – получен неожиданный результат, потому что существуют другие варианты написания фильтра.

Осуществим фильтрацию по протоколам TCP, аналогично произведем фильтрацию по протоколам HTTP, DNS.

Также может производиться фильтрация пакетов по определенным значениям полей в заголовках протоколов. Каждое поле из панели деталей пакетов может использоваться как строка фильтра, тогда Wireshark покажет только те пакеты, в которых есть это поле.

Операторы сравнения и некоторые обозначения полей, которые могут использоваться при построении фильтров, представлены в табл. 1.3–1.5.

Таблица 1.3

Фильтры сравнения значений

Оператор	Значение	Описание и пример
1	2	3
eq	==	*Равный:* ip.addr==192.168.1.1 Отображать только те пакеты протокола IP, в которых сетевой адрес отправителя или получателя равен 192.168.1.1
ne	!=	*Не равный:* ip.addr!=10.0.0.5

Окончание табл. 1.3

1	2	3
gt	>	*Больше:* cp.dstport>10000 Отображать только те сегменты протокола TCP, в которых порт получателя больше 10 000
lt	<	*Меньше:* tcp.dstport<1024 Отображать только те дейтаграммы протокола UDP, в которых порт получателя меньше 1024
ge	>=	*Больше либо равный:* frame.pkt_len ge 0x100
le	<=	*Меньше либо равный:* frame.pkt_len <= 0x20

Таблица 1.4

Фильтры логических операций

Оператор	Значение	Описание и пример
1	2	3
and	&&	*Логическое И:* ip.src==192.168.1.1 && ip.dst==192.168.1.10 Отображать только сообщения, отправленные устройством с сетевым адресом 192.168.1.1 для устройства с сетевым адресом 192.168.1.10
or	||	*Логическое ИЛИ:* eth.dst==ff:ff:ff:ff:ff:ff || ip.dst==255.255.255.255 Отображать только широковещательные кадры протокола Ethernet или пакеты протокола IP
xor	^^	*Исключающее ИЛИ:* tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29
not	!	*Логическое отрицание:* !arp Не отображать PDU протокола ARP

Таблица 1.5

Обозначения часто используемых полей фильтров

Поле	Описание
eth.addr	Физический адрес источника или получателя в кадре протокола Ethernet
eth.len	Длина кадра протокола Ethernet
ip.addr	Сетевой адрес источника или получателя в пакете протокола IP
ip.dst	Сетевой адрес получателя в пакете протокола IP
ip.src	Сетевой адрес источника в пакете протокола IP
tcp.ack	Подтверждения (ACK) протокола TCP
tcp.port	Порт источника или получателя в сегменте протокола TCP
tcp.dstport	Порт получателя в сегменте протокола TCP
tcp.srcport	Порт источника в сегменте протокола TCP
tr.dst	Физический адрес получателя в кадре протокола Token Ring

Для создания фильтра по определенным параметрам пакета необходимо нажать правой кнопкой мыши на выбранный параметр и применить его.

Выбрав функцию *Редактировать → Найти пакет*, можно выбрать фильтры поиска (рис. 1.5):

1) Дисплейный фильтр: ip.addr==192.168.0.1.

2) Шестнадцатеричное значение: 00:00 (поиск определенной последовательности байтов в данных пакетах).

3) Строка (поиск строки в данных пакета с различными опциями).

4) По регулярному выражению.

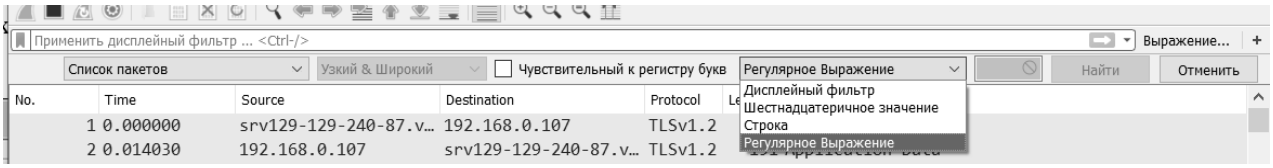


Рис. 1.5. Диалоговое окно «Поиск пакета»

Выбрав меню *Запуск,* можно легко перейти на определенный пакет при помощи одного из подменю:

- Переход на предыдущий пакет.

- Переход на следующий пакет.

- Перейти к пакету, позволяет ввести номер пакета, как только нажмете *ОК*.

- Перейти к соответствующему пакету.

- Перейти к первому пакету.

- Перейти к последнему пакету.

Можно настроить Wireshark так, чтобы он выделял пакеты разными цветами согласно цвету фильтра. Это позволяет акцентировать внимание на пакетах, которые часто используются.

Для этого необходимо зайти в *Просмотр → Цветовые правила* (рис. 1.6), выбрать любые цвета для шрифта и заливки пакета.

При нажатии на любой выбранный фильтр откроется диалоговое окно редактирования цвета. Чтобы появилось окно для выбора цвета, нажмите кнопку «Передний план» или «Фон».

Для извлечения файлов или картинок из захваченных файлов необходимо произвести фильтрацию по пакетам протокола HTTP.

Далее для извлечения информации следует перейти в меню *Файл → Экспортировать объекты → HTTP*. Появится окно, которое покажет все захваченные http-объекты – текстовые файлы, картинки и т. д. (рис. 1.7).

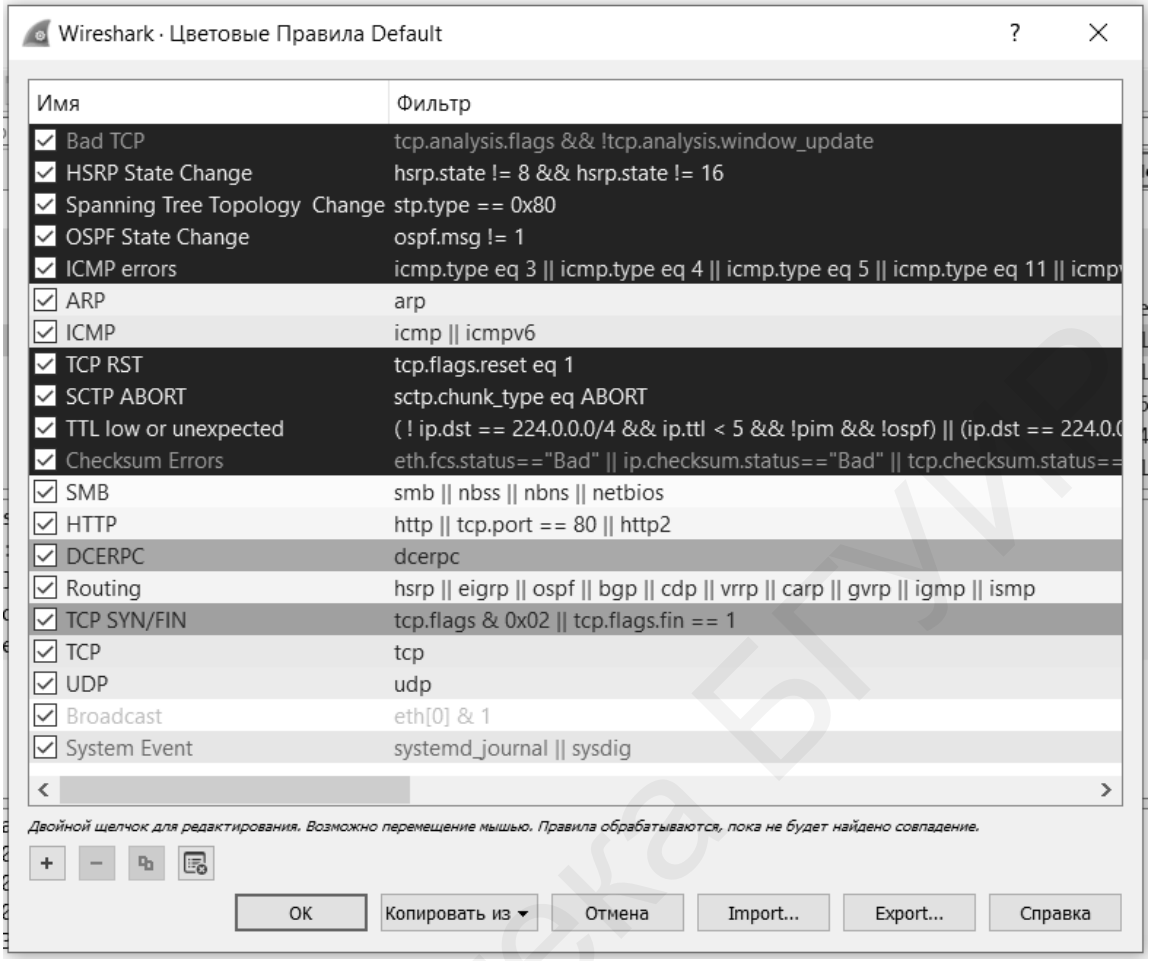


Рис. 1.6. Диалоговое окно «Правила закраски»

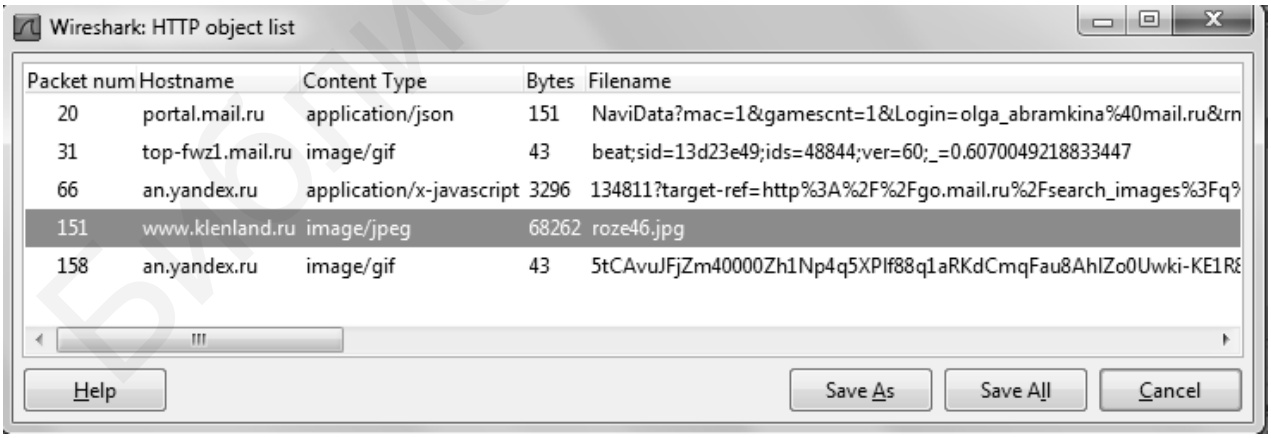


Рис. 1.7. Список захваченных файлов

Для того чтобы вытащить любой файл из этого списка, достаточно просто выделить его и нажать *Сохранить как*. Таким же способом можно извлекать и потоковое видео/аудио.

Чтобы быстро просмотреть передаваемые данные в рамках того или иного сеанса, используют команду меню *Анализ → Следовать → Поток TCP*.

После выполнения команды на экране появится диалоговое окно, в котором разными цветами будут отображены как запросы клиента, так и ответы сервера.

Для использования функции *Следовать → Поток TCP* необходимо знать, в каком пакете находятся искомые данные. Определить это можно, «увидев» передачу данных в потоке TCP как пакеты протокола FTP-data. На пакете, в котором предполагается передача данных, щелкаем правой кнопкой мыши и выбираем *Следовать → Поток TCP*. Если вы точно не знаете расширение сохраняемого файла, то заголовок файла может помочь, например jpg-файл. Файлы, начинающиеся с MZ, являются исполняемыми. Также важно выбрать направление, в котором велась передача файла, – либо от сервера к клиенту, либо наоборот. Определить это можно по IP-адресам.

Анализ TCP-сеансов

Для определения количества сеансов TCP в буфере захваченных пакетов необходимо выполнить команду меню *Статистика → Диалоги* (рис. 1.8).

Выберите первый сеанс и с помощью контекстного меню *Применить как фильтр → Выбрано → A-B* отобразите в буфере кадры, принадлежащие этому сеансу (рис. 1.9). Выбрав меню *Анализ → Информация эксперта*, можно просмотреть сообщения об ошибках и флаги предупреждения (такие, как «потерянный» или «не в очереди сегмент») для быстрого обнаружения проблемы.

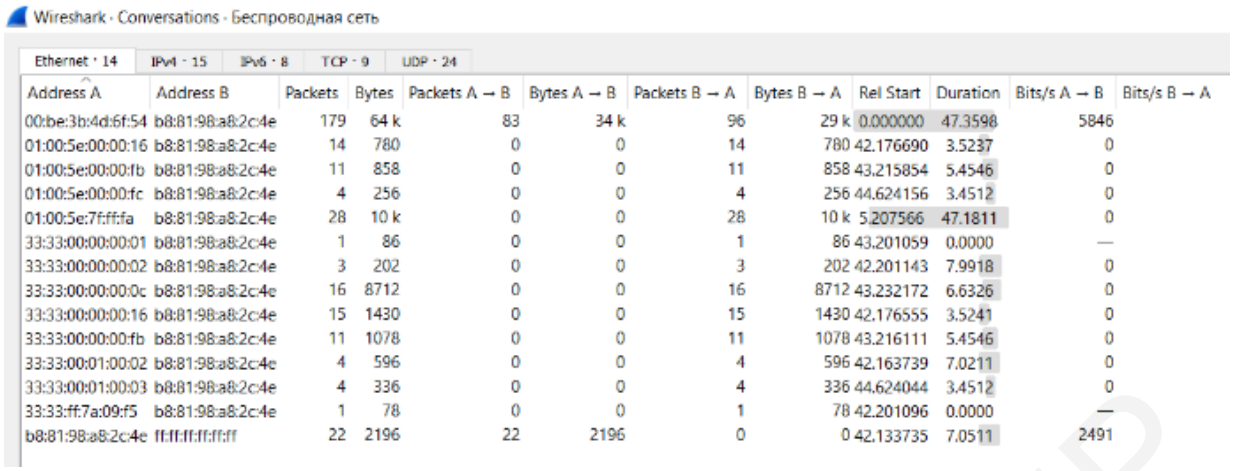


Рис. 1.8. Количество сеансов TCP

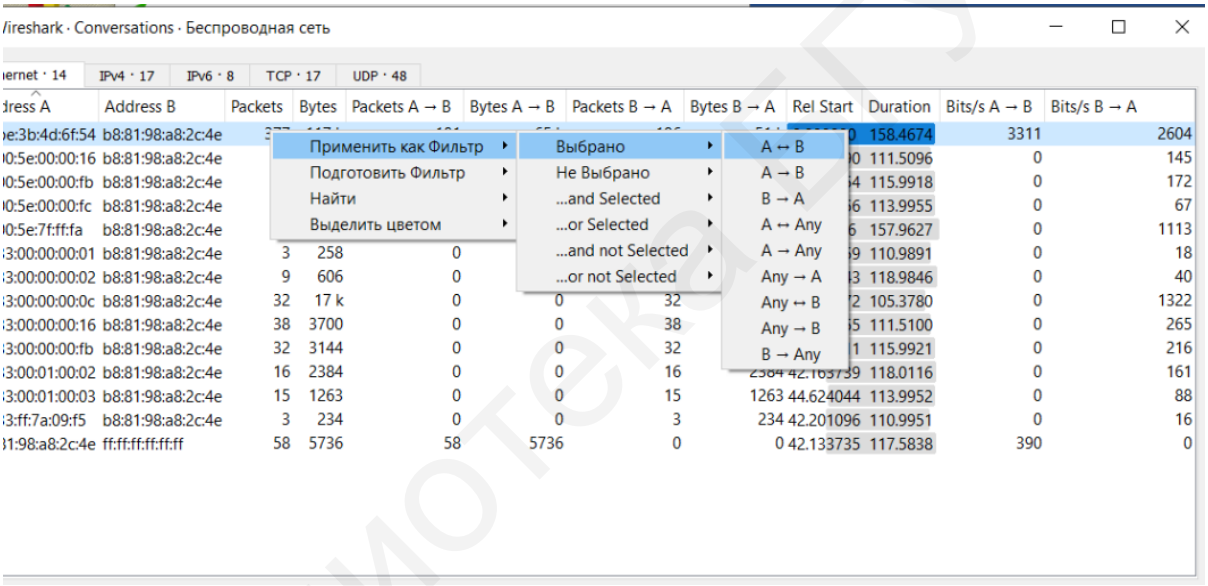


Рис. 1.9. Отображение кадров только одного сеанса

Настройка коммутатора

Посмотрите общую информацию о коммутаторе. В командной строке введите `show switch`

На рабочей станции остановите захват трафика. Установите фильтр для отображения пакетов протокола Telnet: в строке фильтра введите telnet и

нажмите *Apply*. Выберите любой пакет Telnet. Щелкните на нем правой кнопкой мыши и выберите *Следовать → Поток TCP* (рис. 1.10).

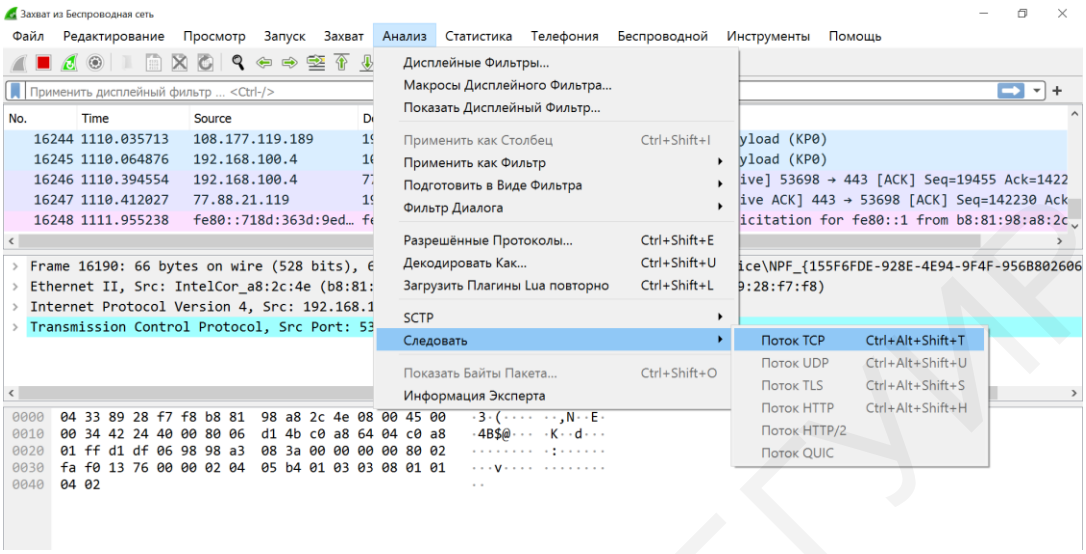


Рис. 1.10. Просмотр TCP-потока

В окне *Следовать → Поток TCP* показаны данные, передаваемые при установлении Telnet-соединения с коммутатором. Обратите внимание, имя пользователя и пароль передаются в открытом виде. Протокол Telnet не предусматривает шифрование и не обеспечивает безопасность передаваемых данных, поэтому вместо него рекомендуется использовать протокол SSH (рис. 1.11).

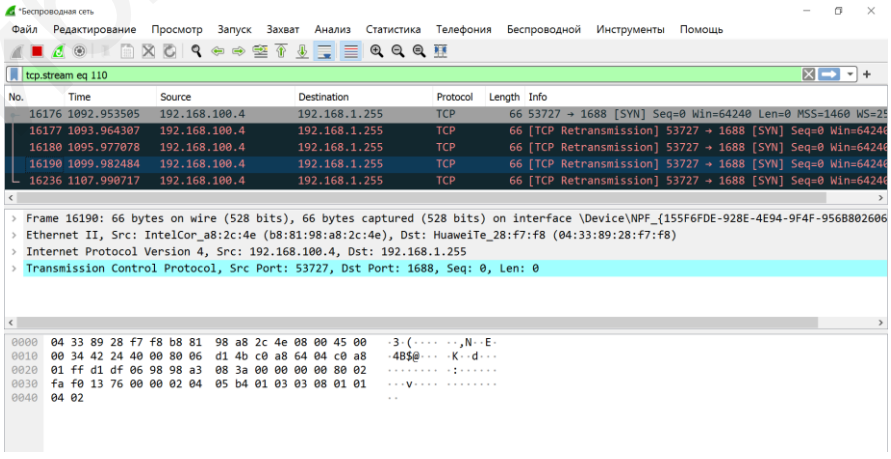


Рис. 1.11. Захваченный трафик Telnet

Подключение к интерфейсу командной строки коммутатора

Подключите рабочую станцию ко второму порту коммутатора с помощью Ethernet-кабеля.

На рабочей станции настройте статический IP-адрес, как показано на рис. 1.12.

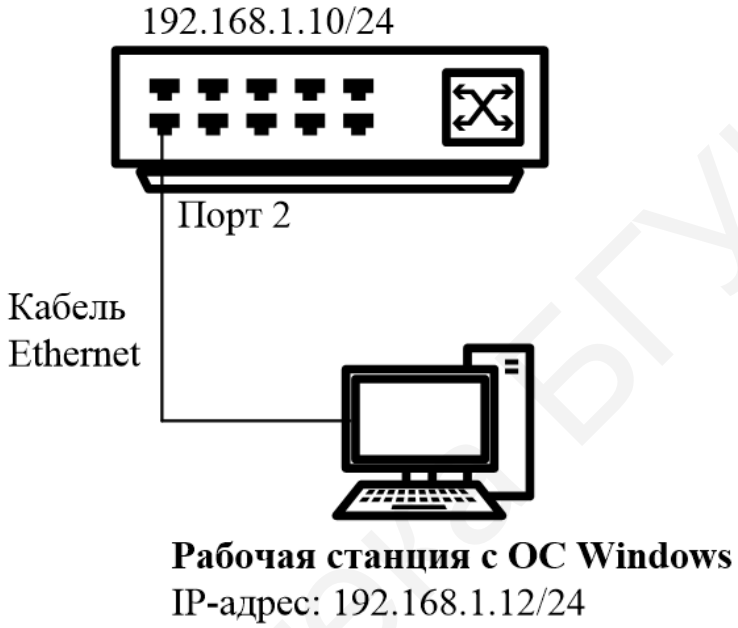


Рис. 1.12. Схема подключения к коммутатору для первоначальной настройки

Изучение команд настройки коммутатора через CLI

В дальнейшем для настройки некоторых функций коммутатора будет использоваться интерфейс командной строки (Command Line Interface, CLI). В связи с этим рассмотрим процесс подключения к интерфейсу командной строки через консольный порт и основные команды коммутатора.

Все команды CLI являются чувствительными к регистру, поэтому прежде чем вводить команду, следует убедиться, что отключены все функции, которые могут привести к изменению регистра текста.

При работе в CLI можно вводить сокращенный вариант команды. Например, если ввести команду sh sw, то коммутатор интерпретирует ее как show switch.

Для описания ввода команд, ожидаемых значений и аргументов при настройке коммутатора через CLI используются символы, описанные в табл. 1.6 [4].

Настройка IP-адреса управления

Для удаленного управления коммутатором через Web-интерфейс или Telnet ему необходимо назначить IP-адрес из адресного пространства сети, в которой планируется его использовать. Посмотрите значение IP-адреса интерфейса управления коммутатора:

```
show ipif
```

Измените IP-адрес интерфейса управления коммутатора (*System – имя управляющего интерфейса коммутатора*):

```
config ipif System ipaddress 192.168.1.10/24
```

Проверьте настройки коммутатора:

```
show switch
```

На рабочей станции проверьте соединение с коммутатором. В командной строке введите

```
ping 192.168.1.10
```

Настройка времени на коммутаторе

Проверьте время:

```
show time
```

Введите дату и время на момент выполнения лабораторной работы, например:

```
config time 01sep2019 15:45:30
```

Установите часовой пояс Минска (GMT +3:00):

```
config time_zone operator + hour 4 min 0
```

Проверьте время:

```
show time
```

Управление учетными записями пользователей

При создании конфигурации коммутатора важно обеспечить его защиту от доступа неавторизованных пользователей. Самая простая схема обеспечение безопасности – создание учетных записей пользователей с соответствующими правами. Создавая учетную запись, можно задать один из уровней привилегий: *admin, operator, power_user* или *user*. Учетная запись с уровнем привилегий *admin* имеет наивысший уровень привилегий.

Создайте учетную запись администратора:

```
create account admin admin
```

Укажите пароль и подтверждение пароля администратора:

```
Enter a case-sensitive new password: admin
Enter the new password again for confirmation: admin
```

Проверьте настройки учетных записей:

```
show account
```

Для выхода из режима с текущими правами введите

```
logout
```

Подключитесь к интерфейсу командной строки коммутатора, используя учетную запись администратора. Создайте учетную запись пользователя:

```
create account user user
```

Укажите пароль и подтверждение пароля пользователя:

```
Enter a case-sensitive new password: user
```

```
Enter the new password again for confirmation: user
```

Проверьте настройки учетных записей:

```
show account
```

Измените пароль пользователя:

```
config account user
```

Укажите старый пароль пользователя и два раза введите новый пароль:

```
Enter a old password:****
Enter a case-sensitive new password:****
Enter the new password again for confirmation:****
```

Посмотрите список пользователей, подключенных к CLI коммутатора:

```
show session
```

Посмотрите текущую конфигурацию коммутатора, хранящуюся в RAM, и проверьте, зашифрованы ли пароли:

```
show config current_config
```

Удалите учетную запись пользователя и администратора:

```
delete account user
delete account admin
```

Убедитесь в удалении учетной записи пользователя:

```
show account
```

Управление возможностью доступа к коммутатору через Web-интерфейс и Telnet

Для повышения безопасности сети в том случае, если для доступа к коммутатору не используются Web-интерфейс и Telnet, рекомендуется их отключить (по умолчанию Web-интерфейс и Telnet на коммутаторе включены). Отключите возможность подключения к коммутатору по Telnet:

```
disable telnet
```

Проверьте выполненные настройки:

```
show switch
```

Убедитесь, что доступ по Telnet отключен. Выполните на рабочей станции ПК1 команду:

```
telnet <IP-адрес коммутатора>
```

Включите функцию подключения к коммутатору по Telnet: откройте любой браузер на компьютере и в адресной строке наберите 10.90.90.90; После этого зайдите в настройках коммутатора в пункт меню *SNMP Settings* и откройте *Telnet Settings,* выберите *Telnet State → enable*.

Проверьте выполненные настройки и убедитесь в возможности подключения к коммутатору по Telnet.

Протокол Telnet не предусматривает шифрование и не обеспечивает безопасность передаваемых данных. Протокол SSH обеспечивает безопасное соединение путем шифрования передаваемых данных, включая пароли.

Включите возможность подключения к коммутатору по SSH:

```
enable ssh
```

Проверьте выполненные настройки:

```
show switch
```

Отключите возможность подключения к коммутатору по SSH:

```
disable ssh
```

Проверьте выполненные настройки:

```
show switch
```

Настройка параметров баннера приветствия

С целью упрощения идентификации пользователями активного сетевого оборудования или создания его уникальных логотипов возможно изменение баннера приветствия, который появляется в момент загрузки

коммутатора. Также возможно изменение приглашения Command Prompt в командной строке CLI [4].

Измените приглашение Command Prompt:

```
config command_prompt Test_switch
```

Установите приглашение по умолчанию:

```
config command_prompt default
```

Посмотрите баннер приветствия:

```
show greeting_message
```

Войдите в режим редактирования баннера приветствия:

```
config greeting_message
```

Добавьте строку в приветствие:

```
Labaratory work 1
Group 908578
Student Ivanov F.
```

Для редактирования приветствия используйте следующие команды:

<Function Key>	*<Control Key>*
Ctrl+C	*Выйти без сохранения*
Ctrl+W	*Сохранить и выйти*
left/right/up/down	*Переместить курсор*
Ctrl+D	*Удалить линию*
Ctrl+X	*Стереть все настройки*
Ctrl+L	*Перезагрузить первоначальные настройки*

Сохраните изменения в приветствии и выйдите из режима редактирования: Ctrl+W. Проверьте измененный баннер приветствия (рис. 1.13):

```
show greeting_message
```

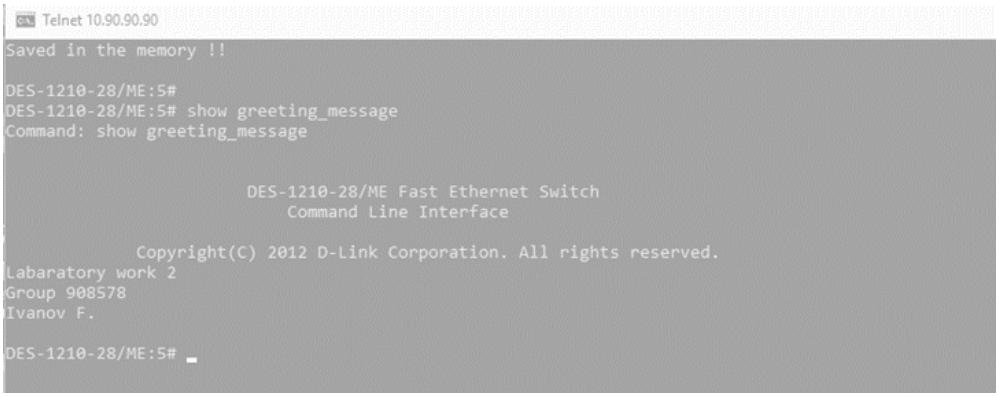


Рис. 1.13. Измененный баннер приветствия

Восстановите настройки баннера по умолчанию:

```
config greeting_message default
```

Проверьте баннер приветствия:

```
show greeting_message
```

Настройка параметров портов коммутатора

По умолчанию порты всех коммутаторов D-Link поддерживают автоматическое определение скорости и режима работы (дуплекса). Может возникнуть ситуация, когда автоопределение будет действовать некорректно и потребуется ручная установка скорости и режима работы порта. В этом случае настройку параметров необходимо выполнить на обоих концах канала связи [4]. Посмотрите текущие настройки портов:

```
show ports
```

Измените скорость и режим работы порта 2:

```
config ports 2 speed 10_half
```

Проверьте выполненные настройки:

```
show ports
```

Отключите работу порта 2:

```
config ports 2 state disable
```

Проверьте выполненные настройки:

```
show ports
```

На рабочей станции проверьте соединение с коммутатором. В командной строке введите

```
ping 192.168.1.10
```

Включите работу порта 2 (предварительно подключитесь к другому, рабочему порту):

```
config ports 2 state enable
```

Задайте описание для порта 2:

```
config ports 2 description PC_PORT
```

Проверьте описание портов:

```
show ports description
```

Команды мониторинга сети

Команды show являются удобным средством проверки состояния и параметров коммутатора, предоставляя информацию, необходимую для мониторинга и поиска неисправностей в работе коммутатора [4].

Посмотрите статистику о пакетах, передаваемых и принимаемых портом 2 (данная команда определяет количественные характеристики передаваемых одноадресных, многоадресных и широковещательных пакетов). В случае возникновения в сети большого количества широковещательного трафика (более 15 % от передаваемого) необходимо проверить наличие DOS-атаки или неисправности [4]:

```
show packet ports 2
```

Посмотрите статистику об ошибках для порта 2 (данная команда определяет ошибки передаваемых данных и локализует проблемы в коммутируемой сети):

```
show error ports 2
```

Посмотрите загрузку центрального процессора коммутатора (ЦП):

```
show utilization cpu
```

В случае длительной загрузки ЦП более 90–100 % необходимо проверить следующие характеристики:

- возможные атаки на коммутатор, неправильная настройка сети: данная проблема может быть решена путем включения функции Safeguard Engine;

- неправильная настройка ACL или других функций коммутатора, влияющих на производительность и работу ЦП;

- некорректная работа ПО коммутатора при работе некоторых функций: данная проблема может быть решена путем обновления ПО коммутатора.

Посмотрите загрузку портов коммутатора (с помощью приведенной команды можно посмотреть загрузку портов коммутатора и объем принимаемого и передаваемого ими трафика в секунду):

```
show utilization ports
```

Посмотрите журнал работы коммутатора:

```
show log
```

Очистите журнал работы коммутатора:

```
clear log
```

Сброс к заводским установкам

Сбросьте текущие настройки коммутатора к настройкам по умолчанию:

```
reset
```

На коммутаторе восстановятся все заводские настройки по умолчанию, за исключением IP-адреса интерфейса управления, учетных записей пользователей и журнала регистраций. Коммутатор не сохранит сброшенные настройки в энергонезависимой памяти NVRAM и не перезагрузится [4].

Если указано ключевое слово config, на коммутаторе восстановятся все заводские настройки по умолчанию, включая IP-адрес интерфейса управления, учетные записи пользователей и журнал регистраций. Коммутатор не сохранит сброшенные настройки в энергонезависимой памяти NVRAM и не перезагрузится:

```
reset config
```

Если указано ключевое слово system, на коммутаторе восстановятся все заводские настройки по умолчанию в полном объеме. Коммутатор сохранит эти настройки в энергонезависимой памяти NVRAM и перезагрузится:

```
reset system
```

Перезагрузите коммутатор:

```
Reboot
```

1.3. Содержание отчета

1. Цель лабораторной работы.

2. Схема соединения лабораторной установки с указанием подключенных портов и настроенных IP-адресов.

3. Выводы по проделанной работе.

1.4. Контрольные вопросы и задания

1. Что такое блок PDU?

2. Какими параметрами характеризуется пакет в системе Wireshark?

3. Что такое «кадр»?

4. Напишите фильтр, который выбирал бы только пакеты с сетевым адресом источника, равным 192.168.1.12.

5. Проанализируйте третий кадр в рамках любого выбранного сеанса TCP и ответьте на следующие вопросы:

а) Какие порты используются клиентом и сервером?

б) Какой начальный последовательный номер выбран клиентом?

в) Присутствует ли в этом кадре поле подтверждения, каково его значение?

г) Какая длина заголовка TCP, присутствуют ли данные в этом кадре?

д) Какой бит флагов установлен и для чего он служит?

6. Какие характеристики коммутатора необходимо проверить в случае длительной загрузки?

7. Какой протокол обеспечивает более безопасное соединение при передаче данных?

Лабораторная работа №2
НАСТРОЙКА ВИРТУАЛЬНЫХ ЛОКАЛЬНЫХ СЕТЕЙ ПО ПРОТОКОЛУ 802.1Q

Цель работы: изучить настройку виртуальных локальных сетей с помощью протокола 802.1Q на коммутаторах D-Link.

2.1. Использование виртуальных локальных сетей

Применение виртуальных локальных сетей позволяет разграничить пользователей сетевых устройств внутри предприятия. Устройства, работающие на канальном уровне и рассылающие широковещательные кадры, через все свои порты, кроме порта приемника во всю сеть, при большом своем количестве способны значительно увеличить нагрузку на сеть. Под широковещательными кадрами понимаются кадры, предназначенные всем узлам сети. Такой тип кадров может быть использован при работе таких протоколов, как ARP, DHCP и другие. Персональный компьютер отправляет сообщения другим устройствам о своем появлении в сети. Иногда при неправильной настройке оборудования может возникать неконтролируемая рассылка широковещательных кадров, что может привести к нерациональному использованию сетевых ресурсов или возникновению широковещательного шторма, особенно в крупных сетях. Одним из способов для устранения этого нежелательного эффекта является ограничение области передачи широковещательных кадров (т. е. ограничение широковещательного домена) с помощью настройки виртуальных локальных сетей [1].

Под виртуальной локальной сетью, или VLAN, понимается группа устройств, принадлежащих одной сети, широковещательный трафик которых на канальном уровне не доступен для других устройств в этой же сети. Другими словами, передача кадров на канальном уровне (вне зависимости от

типа используемого адреса: широковещательного, группового или индивидуального) возможна только внутри локальной сети.

К преимуществам VLAN следует отнести:

- простоту и эффективность группировки сетевых пользователей в виртуальные рабочие группы несмотря на их физическое размещение в сети;
- обеспечение возможности контроля широковещательных сообщений, что увеличивает полосу пропускания, доступную для пользователя;
- возможность повышения безопасности сети при помощи фильтров, настроенных на коммутаторе или маршрутизаторе и определяющих политику взаимодействия пользователей из разных виртуальных сетей [1].

На рис. 2.1 показан пример использования логической сегментации сетей с помощью VLAN для разграничения доступа в сеть Интернет различных отделов предприятия. При обычном подходе к решению этой задачи для каждого обособленного отдела требуется его подключение к коммутатору и маршрутизатору, предоставляющему доступ к сети Интернет. При этом каждое устройство должно обладать достаточным количеством портов для подключения всех сетевых устройств. Минусом такого решения является плохая масштабируемость и высокая стоимость.

При логической сегментации с использованием технологии VLAN можно сократить количество применяемых коммутаторов для разделения трафика различных отделов, что позволяет сократить количество используемых устройств.

В коммутаторах D-Link могут быть реализованы следующие типы VLAN: на основе MAC-адресов, на основе стандарта IEEE 802.1Q, на основе портов (Q-in-Q VLAN), на основе стандарта IEEE 802.1ad, на основе портов и протоколов IEEE 802.1v, асимметричные.

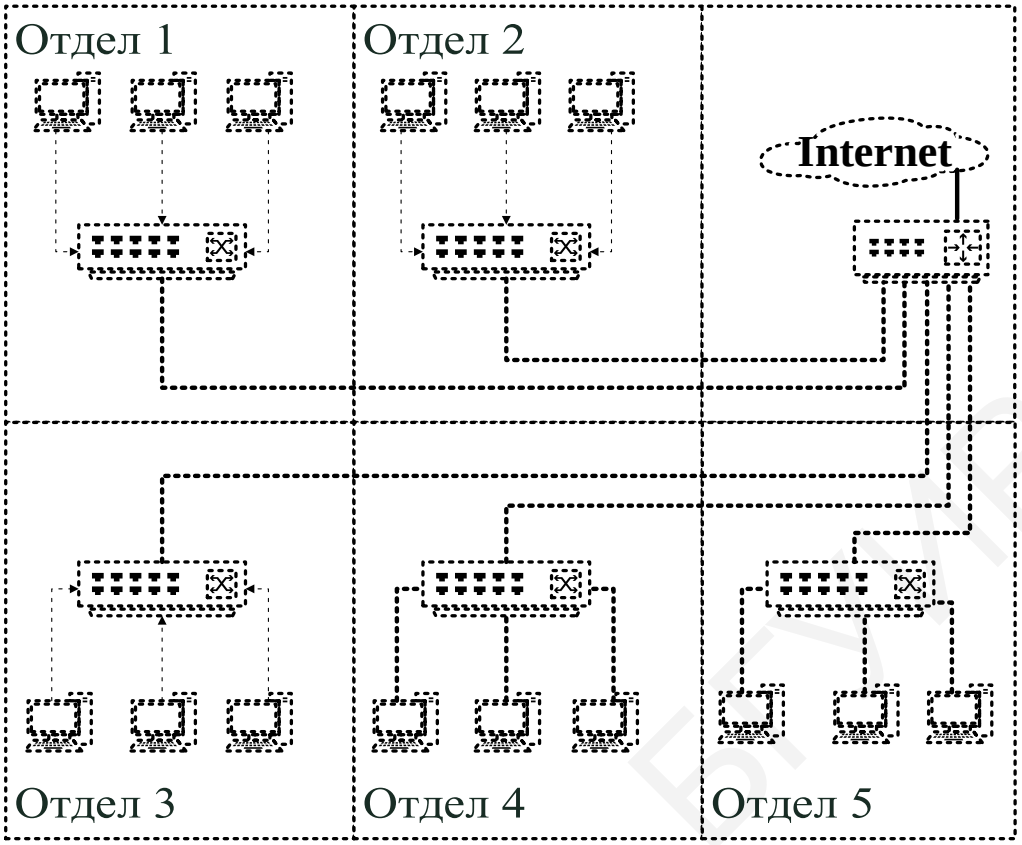


Рис. 2.1. Физическая сегментация сети

2.2. Построение виртуальной локальной сети на основе стандарта IEEE 802.1Q

В соответствии со стандартом 802.1Q кадры получают отметку в зависимости от порта, на который они пришли. Для применения этого метода все машины одного порта должны принадлежать одной виртуальной локальной сети. 802.1Q – открытый стандарт, который описывает процедуру маркировки или тегирования передаваемых кадров с помощью дополнительных полей кадров, в которых находится информация о принадлежности к определенной VLAN.

Достоинства применения стандарта IEEE 802.1Q:

- удобство настройки и конфигурации VLAN: использование тегов позволяет информации о VLAN проходить через все устройства,

поддерживающие стандарт 802.1Q, по одному магистральному каналу (рис. 2.2);

- использование протоколов связующего дерева (STP) на всех портах при работе в обычном режиме;

- использование VLAN IEEE 802.1Q дает возможность добавлять и извлекать теги из заголовков кадров, допускает использование сетевых устройств, которые не поддерживают стандарт IEEE 802.1Q.

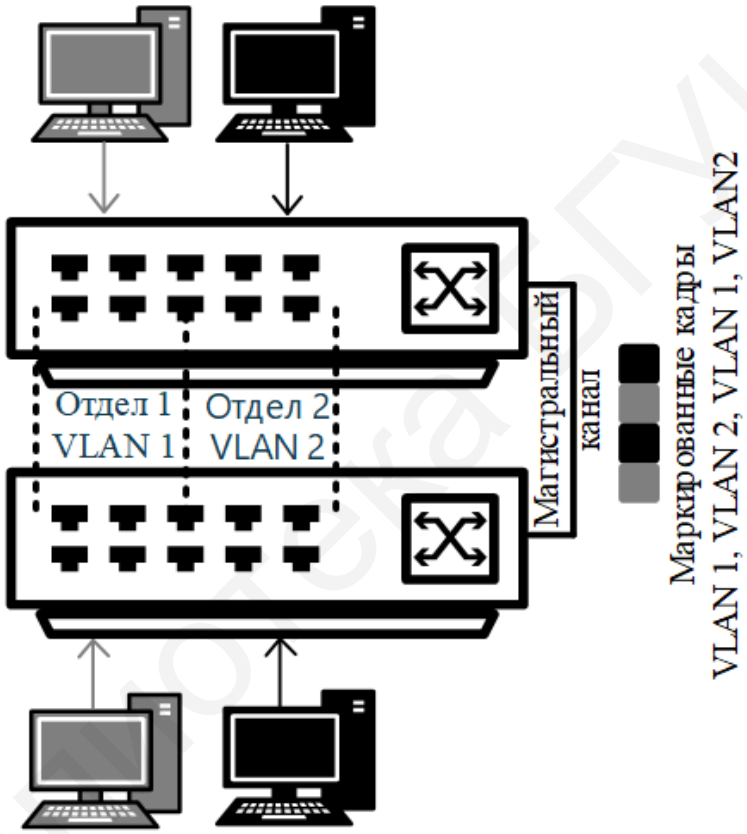


Рис. 2.2. Передача кадров нескольких VLAN
по магистральному кабелю связи

Для организации связи на сетевом уровне необходимо использовать маршрутизатор или коммутатор с функциями сетевого уровня. Но для организации доступа к разделяемым ресурсам из различных VLAN

достаточно добавить необходимый порт коммутатора во все подсети, для которых организуется доступ [1].

В протоколе 802.1Q определены следующие понятия:

- Tagging – маркировка кадра путем добавление тега, содержащего данные о принадлежности к протоколу 802.1Q.
- Untagging – извлечение тега, содержащего данные о принадлежности к протоколу 802. 1Q.
- VLAN ID (VID) – идентификатор виртуальной локальной сети VLAN.
- Port VLAN ID (PVID) – идентификатор порта VLAN.
- Ingress port – входной порт коммутатора, который принимает решение о принадлежности к VLAN.
- Egress port – выходной порт коммутатора, который принимает решение о добавлении тега.

Каждый порт настраивается или как *tagged*, или как *untagged*. Настройка порта *untagging* оставляет возможность работы ему с устройствами, которые не поддерживают стандарт IEEE 802.1Q (рис. 2.3).

Использование стандарта 802.1Q может быть возможным только при внесении в кадр Ethernet изменения путем добавления 32 бит (рис. 2.4). Поле *TPID* проверяет кадр на содержание в нем тега протокола 802.1Q. Поле *Priority (Приоритет)* отвечает за приоритет передаваемого кадра (уровни от 0 до 7, где 7 – наивысший приоритет), поле *Canonical Format Indicator (CFI)* содержит информацию, если кадр принадлежит сетям других видов. *VID (VLAN ID)* является идентификатором виртуальной локальной сети, показывающим, какой VLAN принадлежит трафик (VID 0 и VID 4095 зарезервированы) (см. рис. 2.4).

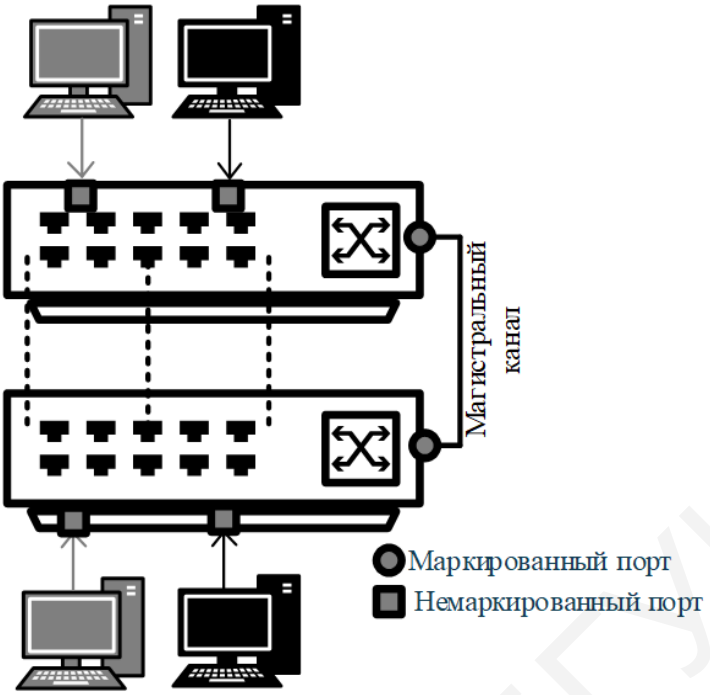


Рис. 2.3. Маркированные и немаркированные порты VLAN

Обычный (немаркированный) кадр

Адрес назначения (DA)	Адрес источника (SA)	Длина/тип (Length/Type)	Данные (Data)	Контрольная сумма кадра (CRC)

Маркированный кадр 802.1p/802.1Q

Адрес назначения (DA)	Адрес источника (SA)	**Тег (Tag)**	Длина/тип (Length/Type)	Данные (Data)	Контрольная сумма кадра (CRC)

Идентификатор протокола тега (TPID) 0×8100	Приоритет (Priority)	Индикатор канонического формата (CFI)	**Идентификатор VLAN (VID)**
16 бит	3 бита	1 бит	12 бит

Рис. 2.4. Маркированный кадр Ethernet

PVID – идентификатор физического порта коммутатора. На основе PVID принимается решение о том, в какую VLAN будет направлен немаркированный кадр, полученный из части сети, подключенной к данному порту (внутри коммутатора в заголовки всех немаркированных кадров добавляется идентификатор VID, равный PVID порта, на который они были

приняты). Коммутаторы хранят таблицу, в которой каждому идентификатору портов PVID ставится в соответствие идентификатор VID сети. Каждый PVID может ассоциироваться с таким количеством VID, которое поддерживает коммутатор. Если на коммутаторе не настроено ни одной VLAN, то порты по умолчанию входят в одну VLAN с PVID = 1 [1].

Алгоритм принятия решения о передаче кадра внутри VLAN основан на следующих правилах:

- Правила входящего трафика – алгоритм принятия решения о принадлежности к той или иной виртуальной локальной сети.
- Правила продвижения между портами – алгоритм для принятия решения о передаче или фильтрации кадра.
- Правила исходящего трафика – принятие решения о сохранении или удалении тега 802.1Q в заголовке кадра.

Если пришедший кадр содержит тег о принадлежности к протоколу 802.1Q, то он является неизменным, т. е. в заголовок кадра будет добавлен VID, равный PVID входного порта.

Принятие решения о принадлежности к определенной виртуальной локальной сети осуществялется по следующему алгоритму: если пришедший кадр не имеет маркировки, то в его заголовок будет добавлен тег с идентификатором VID, идентичным PVID порта, через который был передан кадр. Если кадр *маркирован*, то его принадлежность к виртуальной локальной сети определится по идентификатору VID. Тег в этом случае останется неизменным.

Алгоритм пересылки кадров между портами принимает решение об удалении или передаче кадра на порт назначения, используя данные о принадлежности к определенной локальной сети и MAC-адреса устройства назначения.

Если пришедший кадр содержит тег, то коммутатор решает, является ли порт, через который был получен кадр, членом той же виртуальной локальной сети по идентификатору VID в заголовке кадра и набору VID,

ассоциированных с портом. Если идентификаторы не совпадают, то кадр удаляется. Если кадр не содержит тег, анализ не выполняется. После этого порт назначения проверяется на принадлежность к той же виртуальной локальной сети. Если выходной порт не принадлежит заданной VLAN, то кадр удаляется, если принадлежит, то происходит передача кадра в подключенный к нему участок сети.

Функция сегментации трафика (Traffic Segmentation) используется для разделения доменов на канальном уровне с помощью настройки портов или группы портов для доступа к общим сетевым ресурсам через разделяемые порты, например, для подключения серверов или магистрали сети [1]. Сегментация трафика также позволяет уменьшить нагрузку внутри сетей VLAN 802.1Q, разбивая их на меньшие группы. При этом правила VLAN имеют более высокий приоритет при передаче трафика. Правила Traffic Segmentation применяются после них.

2.3. Указания по выполнению лабораторной работы

Порядок настройки VLAN на основе стандарта IEEE 802.1Q

Перед выполнением лабораторной работы необходимо сбросить настройки коммутаторов к заводским настройкам по умолчанию командой

```
reset config
```

Настройка VLAN на основе стандарта IEEE 802.1Q производится по схеме, представленной на рис. 2.5.

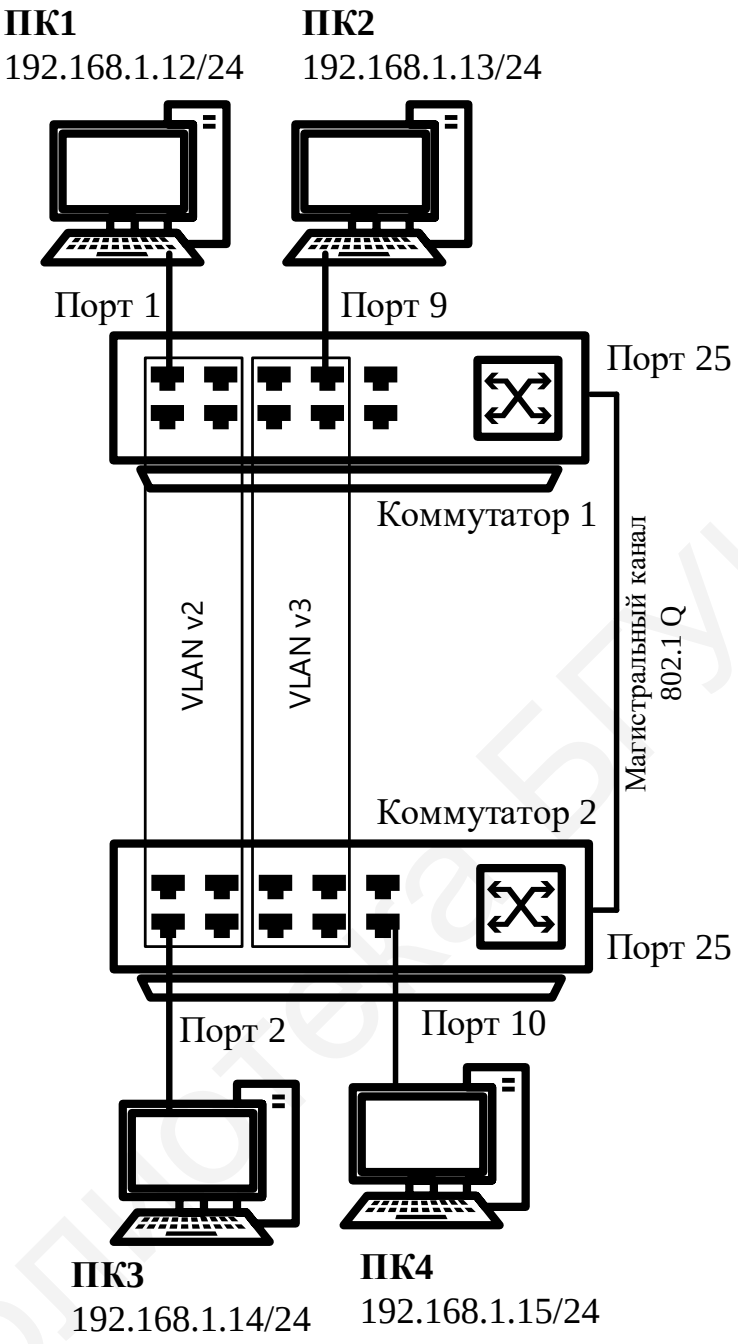


Рис. 2.5. Схема лабораторной сети для настройки VLAN

Последние две цифры для настройки IP-адреса рабочей станции зависят от ее номера в лаборатории, который указан на экране.

После сброса к заводским настройкам все коммутаторы в рабочей сети имеют IP-адрес по умолчанию 10.90.90.90, поэтому перед началом подключения сети необходимо произвести следующую настройку:

SW1: 192.168.0.1/24;

SW2:192.168.0.2/24;

SW3: 192.168.0.3/24;

ПК1: 192.168.0.12/24;

ПК 2: 192.168.0.13/24;

ПК 3: 192.168.0.14/24;

ПК 4: 192.168.0.15/24.

После настройки адресного пространства проверьте доступность соединения между рабочими станциями с помощью команды `ping <IP-address>`:

- от ПК1 к ПК 3 и ПК 4 ;

- от ПК2 к ПК 3 и ПК 4.

В данной лабораторной работе необходимо настроить две виртуальные локальные сети – Gruppa2, Gruppa3. Коммутаторы SW1 и SW3 будут иметь один маркированный порт – 25, коммутатор SW2 – два маркированных и порта – 24 и 25. При настройке следует учесть, что порты 1–8 принадлежат VLAN Gruppa2, а порты 9–16 – VLAN Gruppa 3 на каждом коммутаторе.

Настройте порты 24 и 25 маркированными в виртуальной локальной сети по умолчанию (данная сеть настроена на всех коммутаторах и по умолчанию в нее добавлены все порты устройства) vlan default на каждом коммутаторе:

```
config vlan default add tagged 25
```

Создайте VLAN Gruppa2 и Gruppa3, добавьте в соответствующие VLAN порты, которые необходимо настроить немаркированными. Настройте порты 24 и 25, где необходимо, маркированными:

```
create vlan Gruppa2 tag 2
config vlan Gruppa2 add untagged 1-8
config vlan Gruppa2 add tagged 25
create vlan Gruppa3 tag 3
```

```
config vlan Gruppa3 add untagged 9-16
config vlan Gruppa3 add tagged 25
```

Проверьте настройки VLAN:

```
show vlan
```

Повторите процедуру настройки для коммутатораов SW2 и SW3. Проверьте доступность соединения между рабочими станциями командой `ping <IP-address>`:

- от ПК1 к ПК3 и ПК4;
- от ПК2 к ПК3 и ПК4.

Настройка сегментации трафика внутри VLAN

Используя функцию сегментации трафика, настройте порты 9–16 коммутатора 1, находящиеся в VLAN Gruppa3, таким образом, чтобы рабочие станции, подключенные к ним, не могли обмениваться данными между собой, но при этом могли передавать данные через магистральный канал.

Настройте сегментацию трафика:

```
config traffic_segmentation 9-16 forward_list 25
```

Проверьте выполненные настройки:

```
show traffic_segmentation
```

Подключите ПК1 к порту 9 коммутатора 1. Проверьте доступность соединения между рабочими станциями командой `ping <IP-address>`:

- от ПК1 к ПК 2;
- от ПК1 к ПК3.

2.4. Содержание отчета

1. Цель лабораторной работы.

2. Схема подключения с указанием настроенных IP-адресов, настройка портов и их принадлежность к VLAN, указание настроенных VLAN.

3. Пример выполненной настройки коммутатора.

4. Выводы по проделанной работе.

2.5. Контрольные вопросы и задания

1. Назовите типы виртуальных локальных сетей.

2. Каковы преимущества использования VLAN?

3. Назовите отличия маркированных и немаркированных портов.

4. Назовите правила прохождения пакетов через маркированные и немаркированные порты.

5. Каковы преимущества использования функции Traffic Segmentation?

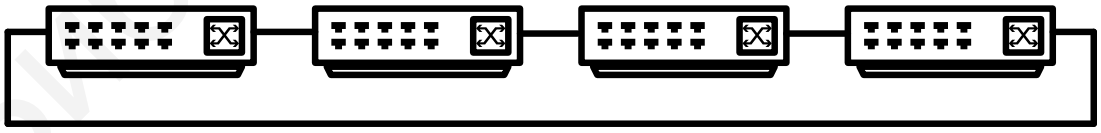
Лабораторная работа №3
ПРОТОКОЛЫ СВЯЗУЮЩЕГО ДЕРЕВА

Цель работы: понять функционирование протоколов связующего дерева и изучить их настройку на коммутаторах D-Link.

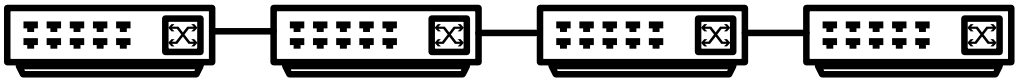
1.1. Применение протоколов связующего дерева в коммутируемых локальных сетях

В коммутируемых локальных сетях проблема обеспечения надежности сети имеет свою специфику: базовый протокол прозрачного моста корректно работает только в сети с древовидной топологией, в которой между любыми двумя узлами сети существует единственный маршрут. Однако для надежной работы сети необходимо наличие альтернативных маршрутов между узлами, которые можно использовать при отказе основного маршрута.

Также при создании сети иногда применяется физическое стекирование, которое используется для объединения коммутаторов в одно логическое устройство для увеличения количества портов и облегчения управления. Стек имеет общие таблицы маршрутизации и коммутации, а также может иметь две топологии: «*кольцо*» и «*цепь*» (рис. 3.1) [1].



Топология стекирования «кольцо»



Топология стекирования «цепь»

Рис. 3.1. Технологии стекирования «кольцо» и «цепь»

Для построения стека типа «кольцо» данные передаются от одного устройства к другому, пока не достигнут заданного порта. Преимуществом является защита топологии при выходе одного устройства из строя (остальные продолжат работу), однако для корректной работы такой схемы необходимо использовать дополнительные протоколы.

Алгоритм связующего дерева, разработанный в 1983 году, включен в спецификацию 802.1D, в которой описывается алгоритм прозрачного моста. Фактически протокол STP является специфической упрощенной версией протокола маршрутизации. Упрощение заключается в направлении кадров по активному маршруту независимо от их адреса назначения, в то время как в протоколах маршрутизации активный маршрут выбирается для каждого адреса индивидуально. Сегодня протокол STP широко применяется в наиболее массовых устройствах современных локальных сетей – коммутаторах. Версия протокола STP, получившая название RSTP (Rapid STP, т. е. быстрый протокол покрывающего дерева), затрачивает на поиск новой топологии несколько секунд [12].

Протоколы связующих деревьев Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP) и Multiple Spanning Tree Protocol (MSTP) являются протоколами второго уровня модели OSI. Протоколы позволяют строить древовидные связи между устройствами локальных сетей, исключая возникновение петель, и в то же время они обеспечивают возможность создания резервных связей между устройствами.

3.2. Протокол связующего дерева STP

Как было сказано в подразд. 3.1, при построение большой коммуникационной сети необходимо предусмотреть возможность создания резервных маршрутов, которые в свою очередь приводят к возникновению петель. Создание дополнительных маршрутов позволяет повысить отказоустойчивость сети, однако наличие петель приводит к возникновению

следующих проблем: широковещательные штормы, множественные копии кадров и петли. *Широковещательный шторм* возникает, если в сеть, содержащую петлю, попадает широковещательный кадр. В таком случае коммутаторы продолжат рассылать его бесконечно, как показано на рис. 3.2.

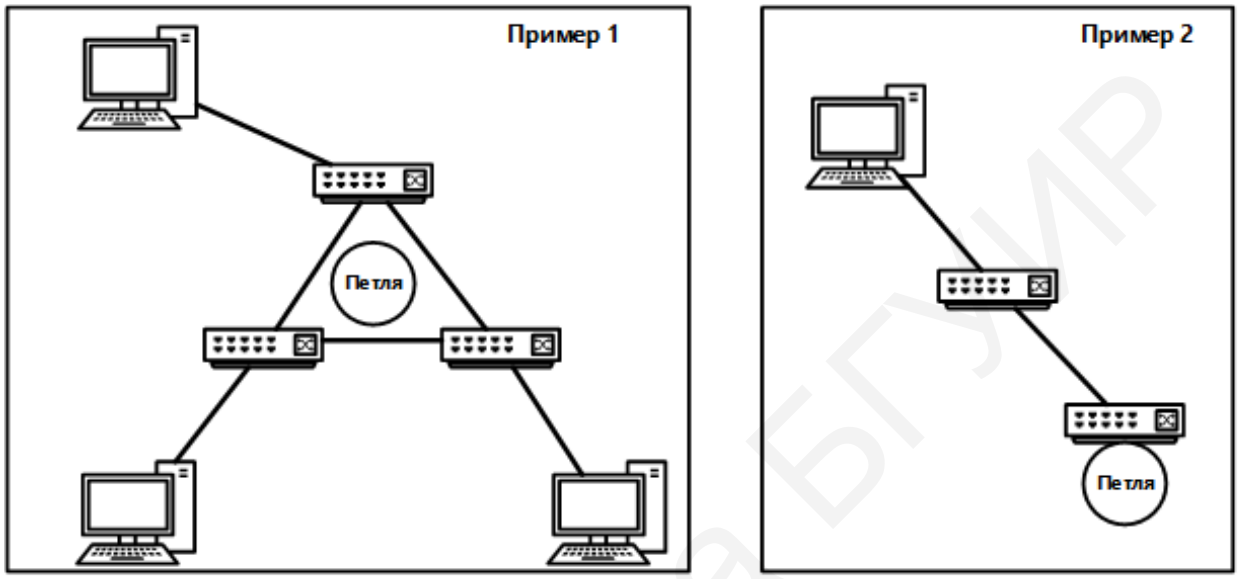


Рис. 3.2. Примеры петель между коммутаторами

Множественные копии кадров возникают при получении коммутатором нескольких копий одного кадра, пришедших из разных портов. В этом случае невозможно распознать расположение отправителя из-за получения кадра от нескольких источников, что приведет к бесконечному обновлению таблицы коммутации.

Петли в сети могут возникать при объединении обширных сетей, которые могут появиться внутри других петель, что может привести к образованию широковещательных штормов. Для решения данной проблемы был разработан протокол связующего дерева, описанный в стандарте IEEE 802.1D-1998.

Протокол STP позволяет создать древовидную конфигурацию связей на компьютерной сети без петель путем построения связующего (покрывающего) дерева, т. е. представляет сеть в виде графа, вершинами

которого являются коммутаторы и сегменты сети. Сегментом сети в свою очередь является связная часть сети, не содержащая коммутаторов (маршрутизаторов). Сегмент может быть разделяемым и включать устройства физического уровня – повторители/концентраторы, существование которых коммутатор, будучи устройством канального уровня, «не замечает». Конфигурация связующего дерева строится коммутаторами автоматически с использованием обмена служебными кадрами, называемыми Bridge Protocol Data Units (BPDU). Данные кадры являются специальными пакетами, с помощью обмена которыми коммутаторы в автоматическом режиме определяют конфигурацию связующего дерева. Пакеты BPDU переносят данные об идентификаторах коммутаторов и портов, а также о расстоянии до корневого коммутатора. Существуют два типа сообщений, которые переносят пакеты BPDU: конфигурационные, называемые также сообщениями Hello (с интервалом 2 с), и сообщения с уведомлениями об изменении конфигурации. Для доставки BPDU используется групповой адрес 01:80:C2:00:00:00, позволяющий организовать эффективный обмен данными [12].

3.3. Алгоритм построения активной топологии связующего дерева

Алгоритм построения топологии по протоколу STP начинается с получения каждым коммутатором в сети уникального *идентификатора моста*, а каждым портом коммутатора – *стоимости пути* и идентификатора порта (рис. 3.3, 3.4). Активная топология строится в три этапа [1]:

1. Определение корневого коммутатора или моста, от которого строится дерево. В качестве корневого коммутатора выбирается коммутатор с наименьшим значением идентификатора. Идентификатором коммутатора является 8-байтное поле, содержащее две части: значение приоритета и MAC-адрес. В исходном состоянии каждый коммутатор, считая себя корневым, генерирует и передает своим соседям сообщения Hello, в которых

помещает свой идентификатор в качестве идентификатора корневого коммутатора. Коммутатор, получив от соседа сообщение Hello, содержащее идентификатор корневого коммутатора, меньший его собственного, перестает считать себя корневым коммутатором и генерировать свои сообщения Hello, начиная ретранслировать сообщения Hello, получаемые от соседей. Значение идентификатор по умолчанию – 32 768. Если идентификаторы одинаковы, то корневой мост будет выбран по наименьшему MAC-адресу.

2. Выбор корневого порта для каждого коммутатора. Корневым портом коммутатора является тот порт, расстояние от которого до корневого коммутатора является минимальным. Сам корневой коммутатор корневых портов не имеет. Для определения корневого порта каждый коммутатор использует пакеты Hello, ретранслируемые ему другими коммутаторами. На основании этих пакетов каждый коммутатор определяет минимальные расстояния от всех своих портов до корневого коммутатора и выбирает порт с наименьшим значением в качестве корневого. При равенстве расстояний выбирается порт с наименьшим значением идентификатора порта (это порядковый номер порта в коммутаторе). После определения корневого моста все остальные коммутаторы, включенные в общую топологию, определяют стоимость пути от себя до корневого моста.

3. Выбор назначенных коммутаторов и портов для каждого сегмента сети. Назначенным является коммутатор, у которого расстояние до корневого моста является минимальным. Назначенные порты для сегментов исполняют ту же роль, что и корневые порты для коммутаторов, располагаясь на кратчайшем пути до корневого коммутатора. Как и при выборе корневого порта, здесь используется распределенная процедура. Каждый коммутатор сегмента прежде всего исключает из рассмотрения свой корневой порт (для сегмента, к которому он подключен, всегда существует другой коммутатор, расположенный ближе к корню). Для каждого из оставшихся портов выполняется сравнение принятых по ним минимальных расстояний до корня

(еще до наращивания на метрику сегмента) с расстоянием до корня корневого порта данного коммутатора.

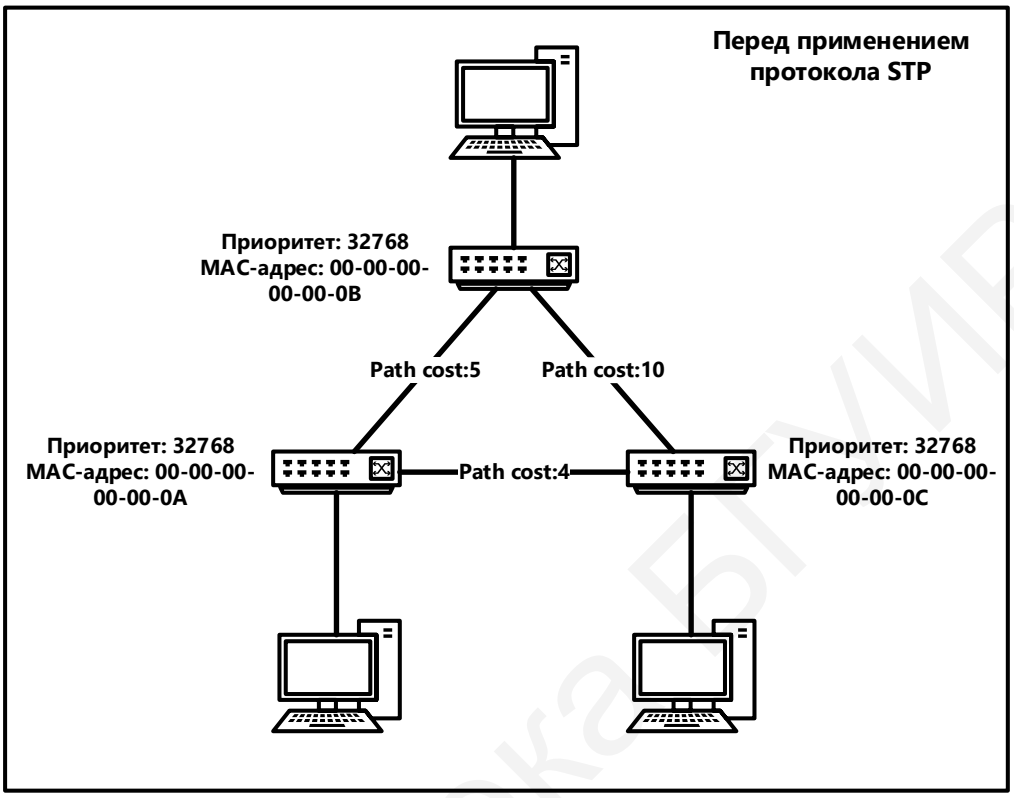


Рис. 3.3. Конфигурация сети перед процедурой STP

Если все принятые на этом порту расстояния оказываются больше, чем расстояние от собственного корневого порта, то это значит, что для сегмента, к которому подключен порт, кратчайший путь к корневому коммутатору проходит через него, и он становится назначенным. Коммутатор делает все свои порты, для которых такое условие выполняется, назначенными. Когда имеется несколько портов с одинаковым кратчайшим расстоянием до корневого коммутатора, выбирается порт с наименьшим идентификатором.

Выбор корневого порта на коммутаторах определяется по стоимости наименьшего маршрута до корня. Если минимальные стоимости нескольких маршрутов будут равны, то корневым портом будет объявлен тот, идентификатор которого меньше [1].

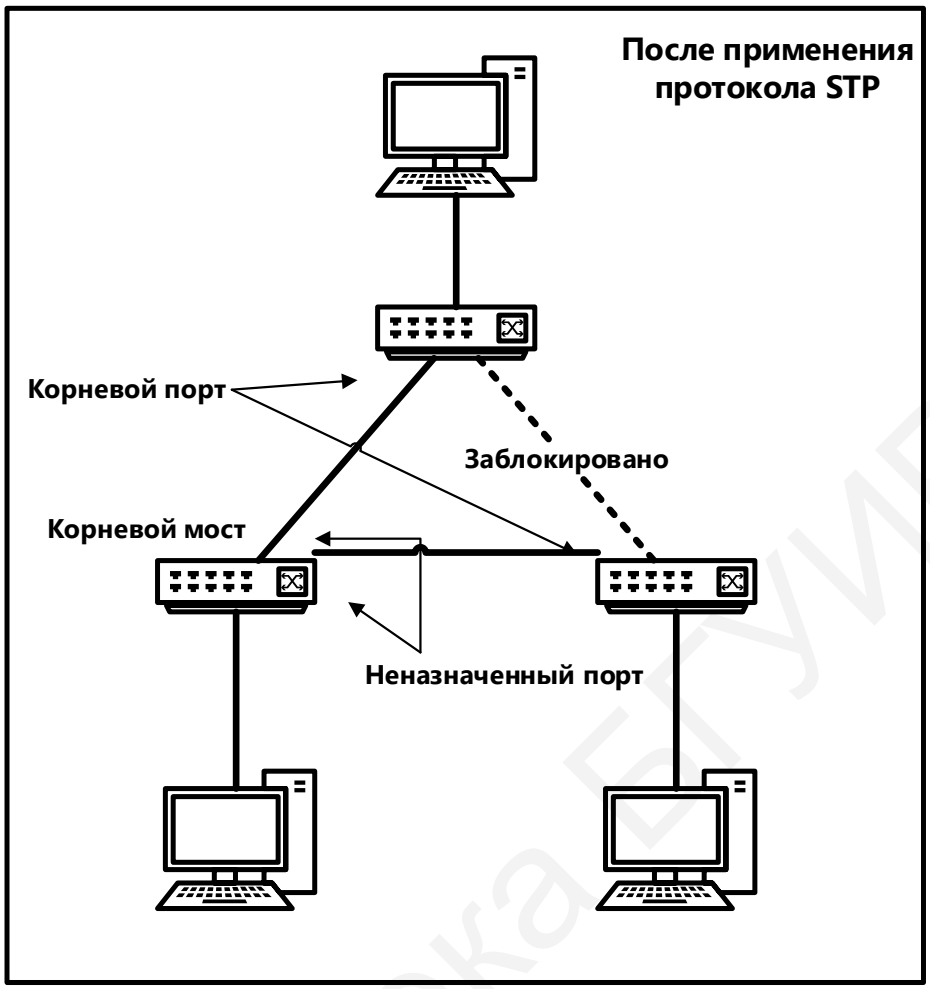


Рис. 3.4. Конфигурация сети после процедуры STP

Следующим этапом построения связующего дерева является определение назначенных портов – тех портов коммутатора, которые связаны с определенным сегментом сети и корневым мостом. Назначенным мостом объявляется коммутатор, которому принадлежит назначенный порт сегмента сети. Назначенный порт выбирается путем сравнения стоимости маршрутов до указанного сегмента от корневого моста. Если несколько путей имеют равные минимальные значения стоимостей, то выбор происходит на основе идентификаторов мостов и идентификаторов портов. Все порты корневого моста являются назначенными, а корневой порт отсутствует.

После определения назначенных и коревых портов все остальные переходят в состояние блокировки, при котором возможны прием и передача только BPDU-кадров.

На выполнение всех трех этапов коммутаторам сети отводится по умолчанию 15 с. Эта стадия работы портов называется стадией прослушивания (listening) – порты слушают только сообщения BPDU и не передают пользовательские кадры. Считается, что порты находятся в заблокированном состоянии, которое относится только к пользовательским кадрам, в то время как кадры BPDU обрабатываются. Предполагается, что в стадии прослушивания каждый коммутатор получает столько пакетов Hello, сколько требуется для определения состояния своих портов. Все остальные порты, кроме корневых и назначенных, каждым коммутатором блокируются и не могут передавать пользовательские кадры. Математически доказано, что при таком выборе активных портов из сети исключаются петли, а оставшиеся связи образуют связующее дерево (если оно вообще может быть построено при существующих связях в сети) [12].

3.4. Формат кадра BPDU

Построение связующего дерева начинается при включении коммутатора или при изменении топологии. Для корректной работы протокола необходима периодическая пересылка информации между коммутаторами, что происходит за счет пересылки специальных кадров – блоков данных протокола моста – BPDU (Bridge Protocol Data Unit). Рассылка кадров осуществляется с помощью MAC-адреса порта в качестве отправителя и широковещательного MAC-адреса в качестве получателя. Также имеется возможность отключить рассылку BPDU на граничные коммутаторы для повышения безопасности путем исключения передачи кадров BPDU по клиентским сегментам сети. Формат кадра BPDU представлен в табл. 3.1 [1].

Существует три типа кадров BPDU:

- CBPDU (Configuration BPDU) – конфигурационный кадр BPDU, используемый для построения дерева.
- TCN (Topology Change Notification BPDU) – кадр с информацией об изменении топологии.
- TCNA (Topology Change Notification Acknowledgement) – кадр, содержащий подтверждение о получении TCN.

Таблица 3.1

Формат кадра BPDU

Название раздела	Английский вариант названия раздела	Размер (байт)
1	2	3
Идентификатор протокола	Protocol Identifier	2
Версия протокола	Protocol Version Identifier	1
Тип BPDU	BPDU Type	1
Флаги	Flags	1
Идентификатор корневого моста	Root Identifier	8
Расстояние до корневого моста	Root Path Cost	2
Идентификатор моста	Bridge Identifier	8
Идентификатор порта	Port Identifier	2
Время жизни сообщения	Message Age	2
Максимальное время жизни сообщения	Max Age	2
Время приветствия	Hello Time	2
Задержка смены состояний	Forward Delay	2

При построении топологии связующего дерева все порты коммутаторов проходят последовательно через различные состояния (рис. 3.5) [1]:

- Блокировка – первое состояние, в котором находятся все порты при включении. В состоянии блокировки происходит только прием и обработка кадров BPDU.
- Прослушивание – при переходе в этот режим порт коммутатора принимает, обрабатывает и передает кадры BPDU. Порт может перейти в состояние блокировки при получении кадра BPDU с лучшими параметрами (идентификатор моста, порта или стоимость пути). Если этого не происходит, то через время, установленное таймером смены состояний (Forward Delay), порт перейдет в стадию обучения.
- Обучение – порт принимает все кадры и начинает строить таблицу коммутации. В этом состоянии при получении кадра с лучшими параметрами порт перейдет в состояние блокировки. После истечения срока, установленного таймером смены состояний, порт перейдет в стадию продвижения.
- Продвижение – в этом состоянии происходит обработка и передача всех поступающих кадров.
- Отключен – режим, назначаемый портам вручную. При отключении этого режима порт перейдет в состояние блокировки.

Для корректной работы протокола STP предусмотрено использование следующих таймеров: Hello Time, Forward Delay и Max Age.

Hello Time – устанавливает промежуток времени, через который будут отправляться CBPDU. Установка этого таймера на корневом мосту определяет его значение на всех коммутаторах, так как они просто пересылают полученные кадры. Значение таймера по умолчанию – 2 с. Диапазон рекомендованных значений – от 1 до 10 с.

Forward Delay – промежуток времени, в течение которого порт находится в режимах обучения и прослушивания. Эта задержка смены состояний необходима для исключения возможности появления параллельных маршрутов при смене состояний. Значение по умолчанию – 15 с. Диапазон рекомендованных значений – от 4 до 30 с.

Max Age – таймер, установленный для определения максимального времени, в течение которого будут храниться настройки связующего дерева. По истечении таймера и при неполучении кадра BPDU от корневого моста коммутатор начинает считать себя корневым мостом и запускает процесс построение дерева. Значение таймера по умолчанию – 20 с. Диапазон рекомендованных значений – от 6 до 40 с.

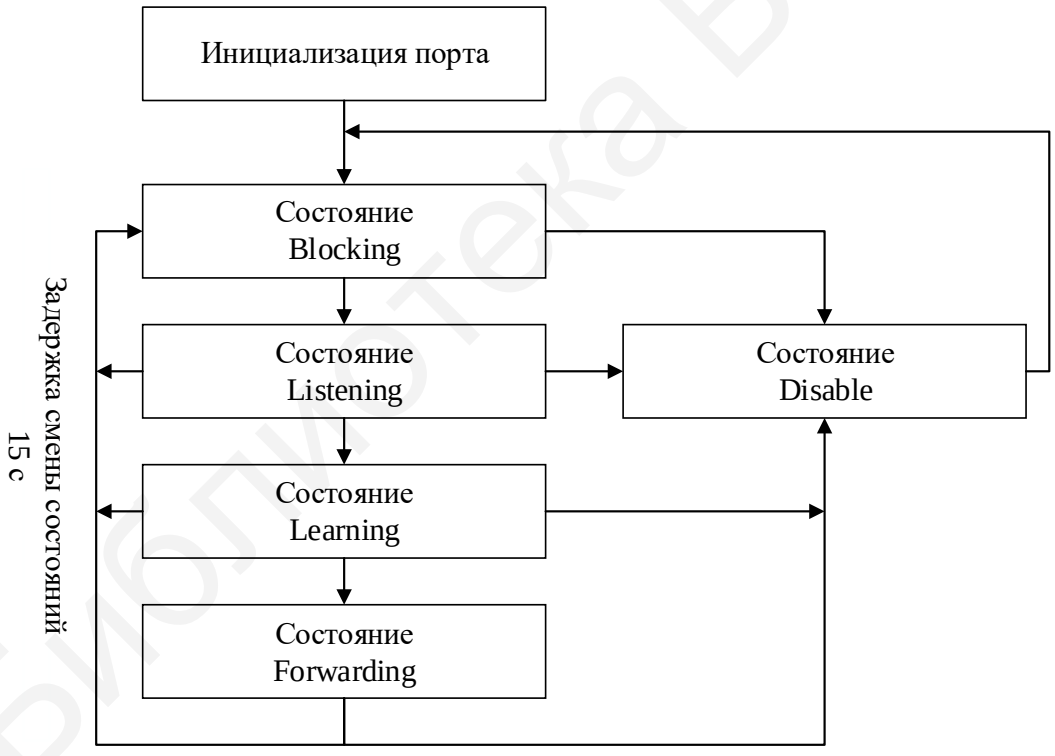


Рис. 3.5. Состояние портов при работе STP

Все таймеры могут быть выставлены вручную, однако следует помнить, что установка слишком больших значений может негативно сказаться на производительности всей сети [1].

В случае истечения максимального времени жизни сообщения корневой порт, не получивший служебного кадра BPDU, начинает процесс построения нового связующего дерева.

3.5. Протокол связующего дерева RSTP

Главным недостатком протокола STP является низкая скорость построения дерева, особенно для сетей с большим количеством устройств. В 2001 году была разработана стандартная ускоренная версия протокола – RSTP (спецификация IEEE 802.lw), которая затем вошла в общий стандарт 802.1D-2004.

В версии протокола RSTP (Rapid Spanning Tree Protocol) значительно сокращается время построения связующего дерева за счет уменьшения количества состояния ролей портов. Основные понятия и терминология протоколов STP и RSTP представлены на рис. 3.6.

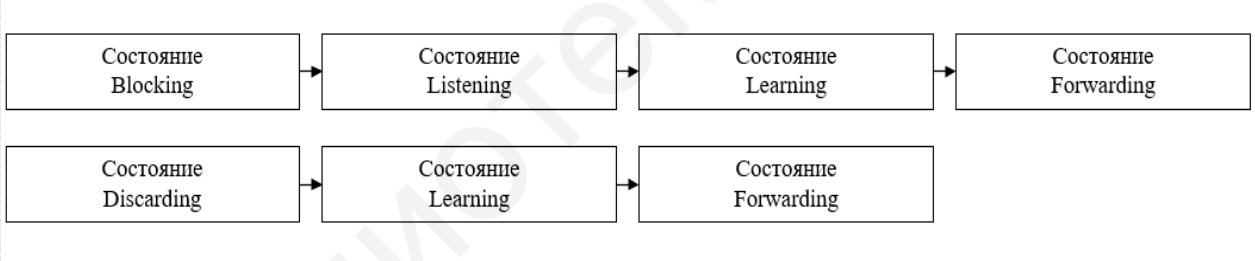


Рис. 3.6. Состояние портов протоколов STP и RSTP

В версии RSTP для сокращения времени построения активной топологии использовано несколько новых механизмов и приемов. Коммутаторы стали учитывать тип сегмента, подключенного к порту. Различаются следующие типы сегментов:

- Двухточечный сегмент. В коммутируемых сетях это единственный тип сегмента; для него у порта существует единственный порт-сосед.

- Разделяемая среда. Стандарт RSTP по-прежнему учитывает существование разделяемой среды, так как формально ее никто не отменял для скоростей ниже 10 Гбит/с.
- Тупиковая связь (edge port). Связь, которая соединяет порт коммутатора с конечным узлом сети; по этому сегменту нет смысла ожидать прихода сообщений протокола RSTP. Тупиковая связь конфигурируется администратором [12].

Различия между состояниями портов в STP и RSTP представлены в табл. 3.2.

В случае подключения к порту тупикового сегмента этот порт не участвует в работе протокола RSTP, а сразу после включения переходит в стадию продвижения кадров. Следует отметить, что в стандарте RSTP начальное заблокированное состояние портов переименовано в состояние отбрасывания. Для портов со связями остальных типов переход в состояние продвижения по-прежнему достижим только после прохождения стадии обучения. Исключается стадия прослушивания. Коммутаторы не делают паузу в 15 с для того, чтобы зафиксировать соответствующую роль порта, например, корневого или назначенного. Вместо этого порты переходят в стадию обучения сразу же после назначения им роли корневого или назначенного порта. Сокращается период фиксации отказа в сети – вместо 10 периодов неполучения сообщений Hello он стал равен трем таким периодам, т. е. 6 с вместо 20. Введены новые роли портов – появились альтернативный (alternative) и резервный (backup) порты. Альтернативный порт является портом-дублером корневого порта коммутатора, т. е. он начинает продвигать кадры в том случае, когда корневой порт отказывает либо перестает принимать сообщения Hello в течение трех периодов. Резервный порт является портом-дублером назначенного порта сегмента; однако такая роль порта имеет смысл только для сегментов, представляющих собой разделяемую среду [12].

Таблица 3.2

Различия между состояниями портов в STP и RSTP

Состояние порта STP	Административное состояние порта коммутатора	Наличие возможности изучать MAC-адреса	Состояние порта RSTP	Роль порта в активной топологии
Disable	Disable	Нет	Discarding	Исключен (Disable)
Disable	Enable	Нет	Discarding	Исключен (Disable)
Blocking	Enable	Да	Discarding	Исключен (Altermate, Backup)
Listening	Enable	Да	Discarding	Включен (Root, Designated)
Learning	Enable	Да	Learning	Включен (Root, Designated)
Forwarding	Enable	Да	Forwarding	Включен (Root, Designated)

Альтернативные и резервные порты находятся в состоянии отбрасывания кадров, так как они не должны продвигать кадры до тех пор, пока их роль не изменится на роль корневого или назначенного порта. Как альтернативные, так и резервные порты выбираются одновременно с корневыми и назначенными портами. Такой подход значительно ускоряет реакцию сети на отказы, так как переход, например, на альтернативный порт, происходит сразу же после фиксации отказа и не связан с ожиданием истечения тайм-аутов. За счет новых механизмов и новых ролей портов протокол RSTP строит новую активную топологию существенно быстрее, чем протокол STP, – за несколько секунд вместо минуты или даже нескольких минут. Протокол RSTP совместим с протоколом STP, так что сеть, построенная из коммутаторов, часть из которых поддерживает RSTP, а часть – STP, будет работать нормально [1].

Построение активной топологии оканчивается присвоением каждому порту своей роли: корневой, назначенный, альтернативный или резервный порт.

Корневой порт – порт коммутатора с минимальной стоимостью пути до корневого моста. Порт включен в активную топологию (рис. 3.7).

Назначенный порт – порт, передающий данные в определенный сегмент сети. Порт включен в активную топологию (рис. 3.8).

Альтернативный порт может заменить корневой при его поломке (рис. 3.9).

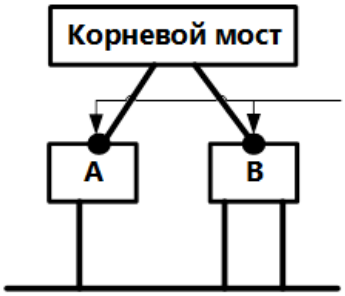


Рис. 3.7. Корневой порт

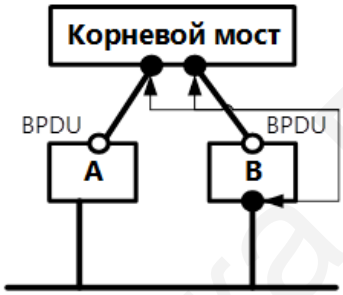


Рис. 3.8. Назначенный порт

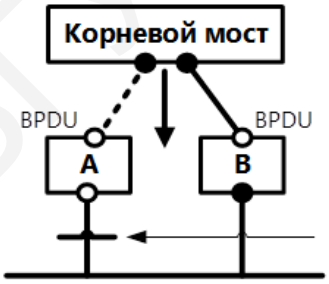


Рис. 3.9. Альтернативный порт

Резервный порт является заменой для назначенного порта при наличии соединения данного моста с сетью (рис. 3.10).

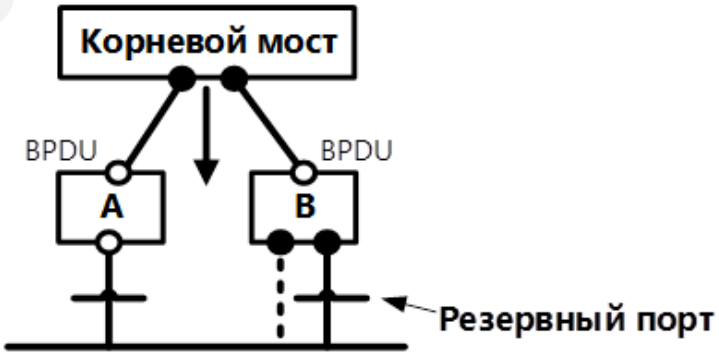


Рис. 3.10. Резервный порт

Формат кадра BPDU протокола RSTP представлен в табл. 3.3.

Таблица 3.3

Формат кадра BPDU протокола RSTP

Название раздела	Английский вариант названия раздела	Размер (байт)
Идентификатор протокола	Protocol Identifier	2
Версия протокола	Protocol Version Identifier	1
Тип BPDU	BPDU Type	1
Флаги	Flags	1
Идентификатор корневого моста	Root Identifier	8
Расстояние до корневого моста	Root Path Cost	2
Идентификатор моста	Bridge Identifier	8
Идентификатор порта	Port Identifier	2
Время жизни сообщения	Message Age	2
Максимальное время жизни сообщения	Max Age	2
Время приветствия	Hello Time	2
Задержка смены состояний	Forward Delay	2
Длина версии 1	Version 1 Length	1

Построение связующего дерева у протокола RTSP отличается от его предшественника быстрым переходом порта в состояние Forwarding с помощью механизма предложений и соглашений. Протокол RSTP предоставляет механизм предложений и соглашений, с помощью которого обеспечивается быстрое переключение портов в режим передачи. Также протокол RSTP использует понятие граничного порта, ведущего к сегменту сети, в которой подключены различные сетевые устройства. Такой порт сразу же переходит в состояние передачи.

На рис. 3.11 показана работа алгоритма предложений и соглашений. Все коммутаторы соединены каналом типа «точка – точка». Коммутатор А является мостом, коммутатор В получает кадр BPDU от коммутатора А с предложением стать новым назначенным мостом. После этого коммутатор В назначает корневой порт (порт, через который был получен BPDU порт p2) и переводит все неграничные порты в состояние блокировки. Все остальные порты получат новую информацию о топологии сети (происходит синхронизация портов) [1].

Порт является синхронизированным «*in-sync*» (шаг 2), если он удовлетворяет следующим критериям:

- находится в заблокированном состоянии;

- является граничным портом.

После синхронизации всех портов коммутатор В разблокирует свой новый корневой порт (шаг 3 на рис. 3.11) и отправит через него коммутатору А согласие на предложение, которое является копией BPDU-предложения, в котором вместо бита Proposal установлен бит Agreement. После этого коммутатор А переведет свой назначенный порт p1 в состояние передачи.

В заблокированном состоянии порт p4 коммутатора В начнет отсылать предложения нижележащему коммутатору и пытаться быстро перейти в состояние Forwarding (шаг 4 на рис. 3.11).

Механизм изменения топологии протокола RSTP также получил некоторые изменения. Изменение топологии могут инициировать только неграничные порты. При переходе порта в заблокированное состояние не генерируется TCN BPDU.

При получении корневым мостом сообщения об изменении топологии инициализируется следующий процесс:

1) устанавливается значение While Timer (таймер, за время которого происходит рассылка информации об изменении топологии), превышающее таймер Hello как минимум в два раза;

2) удаляются MAC-адреса, связанные с назначенными и корневыми портами;

3) рассылаемые BPDU имеют бит TC, равный 1.

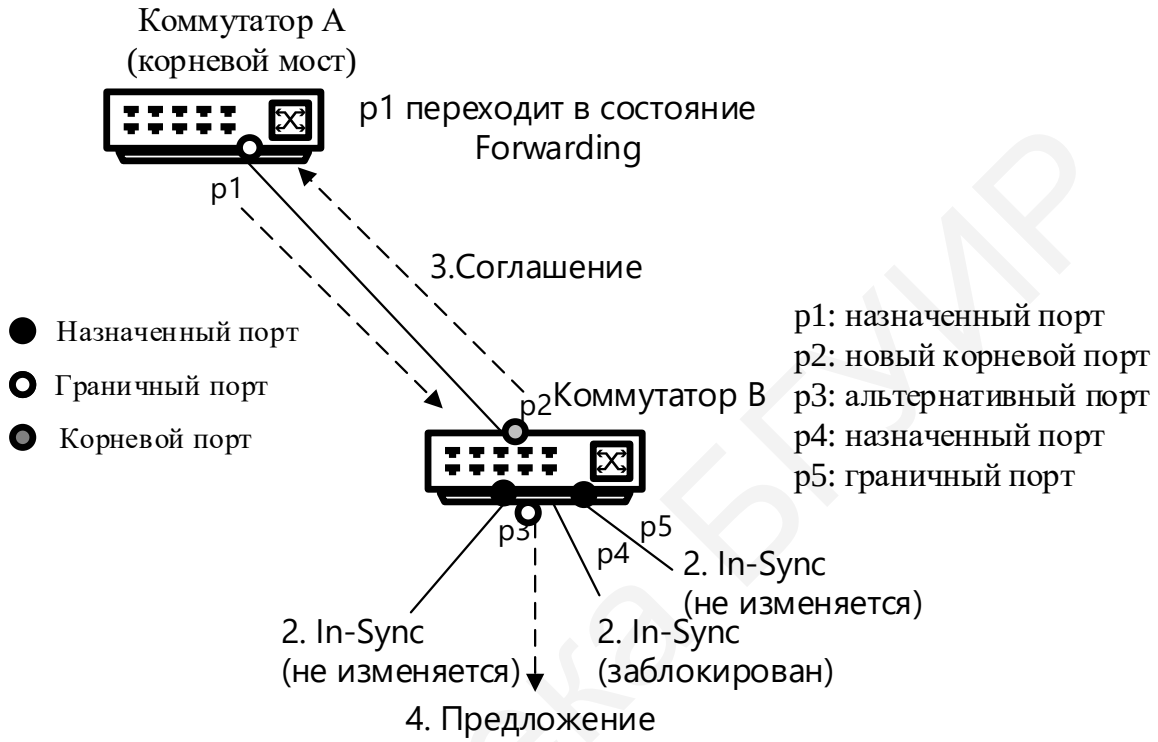


Рис. 3.11. Механизм предложений и соглашений

Распространение информации об изменении топологии происходит при получении коммутатором BPDU с установленным битом TC:

1) удаляются все MAC-адреса, изученные портами, кроме того, и тот, на который пришла информация об изменении топологии;

2) запускается While Timer и отправляется кадр BPDU через все порты, т. е. происходит рассылка информации об изменении по всей сети.

Такой алгоритм рассылки информации гораздо быстрее, чем его аналог в протоколе STP (рис. 3.12). Также протокол RSTP определяет стоимости пути (табл. 3.4), которые зависят от скорости связи.

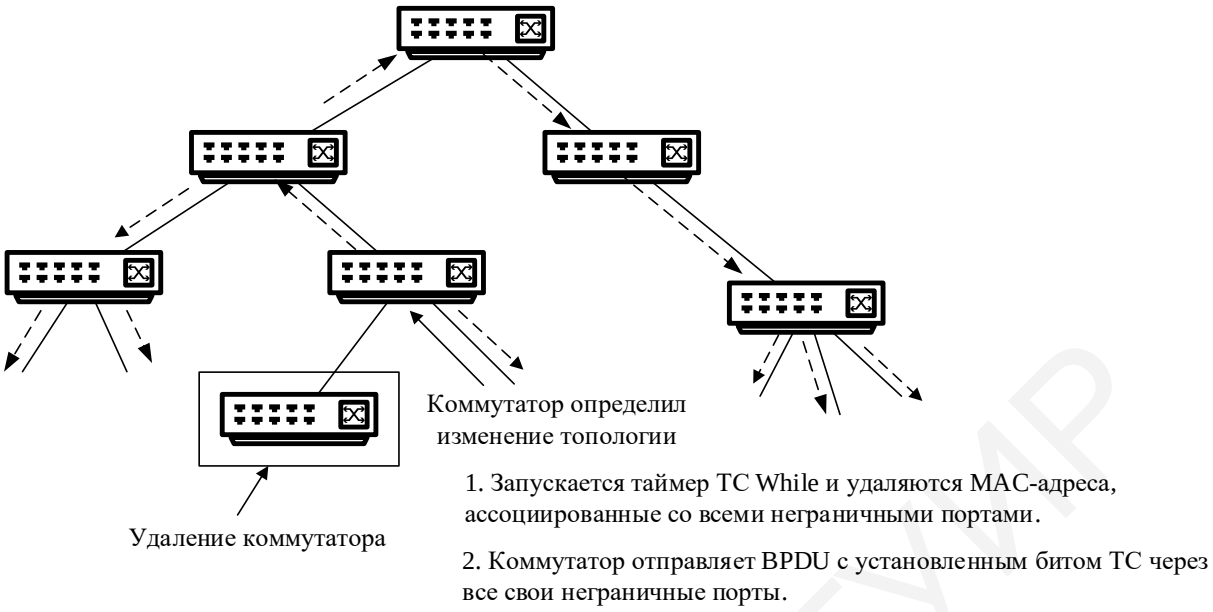


Рис. 3.12. Новый механизм изменения топологии

Одним из преимуществ работы протокола RSTP является его возможность преобразовывать кадры BPDU в формат 802.1D для работы с протоколом STP. Для организации совместной работы каждый порт хранит переменную, определяющую тип протокола, используемого в соответствующем сегменте [1].

Таблица 3.4

Стоимость пути в протоколе RSTP

Скорость канала	Рекомендованное значение	Рекомендованный диапазон
10 Мбит/с	2 000 000	200 000–20 000 000
100 Мбит/с	200 000	20 000–2 000 000
1 Гбит/с	20 000	2000–200 000
10 Гбит/с	2000	200–20 000

3.6. Агрегирование каналов связи

Агрегирование каналов – технология, с помощью которой возможно объединение нескольких физических каналов в один логический, что

позволит значительно увеличить надежность и пропускную способность. Большая часть протоколов позволяет объединять только каналы между двумя устройствами. Без использования специальных протоколов для агрегирования каналов все порты, кроме одного, будут заблокированы протоколами STP. Протоколы агрегирования позволяют использовать пропускную способность всех портов одновременно, при этом контролируя распространение широковещательных кадров, которые будут передаваться только через один порт [4].

Для включения технологии агрегирования каналов используется протокол Link Aggregation Control Protocol (LACP) – протокол управления агрегированным каналом, позволяющий объединить физические порты в одну логическую группу.

Кадры протокола LACP распространяются устройством через все порты, на которых он включен. Используемые порты могут иметь настройку в двух режимах: **активном** и **пассивном**. В пассивном режиме порты только обрабатывают управляющие кадры протокола LACP, в активном – происходит передача кадров. Для настройки автосогласования рекомендуется настраивать порты с одной стороны канала как пассивные, с другой – как активные.

Для настройки агрегированного канала порты должны иметь одинаковые характеристики, такие, как режим работы, скорость, среда передачи, метод управления потоком. На таких портах не допускается настройка блокировки, зеркалирования трафика и функции аутентификации 802.1x.

3.7. Указания по выполнению лабораторной работы

Перед выполнением лабораторной работы производятся мониторинг и диагностика сети во время широковещательного шторма. Соедините схему без подключения петли, как показано на рис. 3.13.

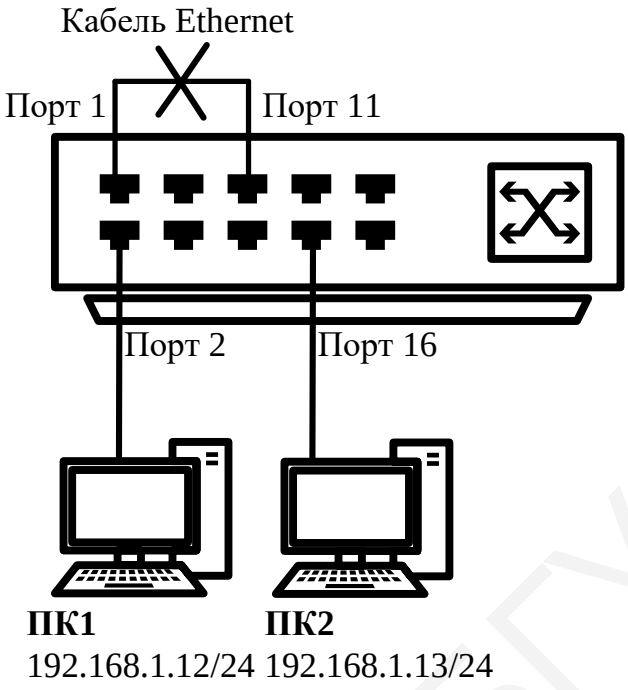


Рис. 3.13. Схема лабораторной сети для диагностики широковещательного шторма

Настройка протоколов связующего дерева STP, RSTP

Сбросьте настройки коммутатора к заводским по умолчанию командой

```
reset config
```

Просмотрите статистику о пакетах, передаваемых через порты коммутатора:

```
show packet ports 1
show packet ports 2
show packet ports 16
```

Соберите схему и соедините кабелем Ethernet порты 1 и 11 коммутатора. Выполните на рабочей станции ПК2 команду ping и не останавливайте ее до окончания выполнения задания:

```
ping 192.168.1.1 -t
```

Повторно просмотрите статистику о пакетах, передаваемых через порты коммутатора. Проверьте, возник ли широковещательный шторм.

```
show packet ports 1
```

Посмотрите загрузку центрального процессора коммутатора:

```
show utilization cpu
```

Просмотрите загрузку портов коммутатора:

```
show utilization ports
```

Запишите процент загрузки портов, используемых в схеме. Просмотрите загрузку ЦП на ПК1 и ПК2. Выполните на рабочей станции ПК 1 команду

```
ping 192.168.1.13
```

Отсоединив кабель от портов 1 и 11, удалите петлю.

Поместите порты 2 и 16 в новую VLAN:

```
config vlan default delete 2,16
create vlan v2 tag 2
config vlan v2 add untagged 2,16
```

Проверьте настройки VLAN:

```
show vlan
```

Просмотрите статистику о пакетах, передаваемых через порт 1:

```
show packet ports 1
```

Соедините кабелем порты 1 и 11 для повторного создания петли. Просмотрите загрузку портов:

```
show utilization ports
```

Выполните на рабочей станции ПК 1 команду

```
ping 192.168.1.13
```

Настройка протокола RSTP

При настройке связующего дерева не соединяйте коммутаторы одновременно двумя кабелями до ее окончания.

Перед выполнением задания необходимо сбросить настройки коммутаторов к заводским настройкам по умолчанию командой

```
reset config
```

Настройка коммутатора 1

Включите протокол связующего дерева на коммутаторе:

```
enable stp
```

Проверьте текущую конфигурацию протокола связующего дерева:

```
show stp
```

Протокол RSTP используется по умолчанию. Если нет, активизируйте его:

```
config stp version rstp
```

Установите на коммутаторе меньшее значение приоритета, чтобы он был выбран корневым мостом:

```
config stp priority 8192 instance_id 0
```

Просмотрите выполненные изменения:

```
show stp instance 0
```

Назначьте порты 1–24 граничными портами:

```
config stp ports 1-24 edge true
```

Активизируйте протокол связующего дерева на портах:

```
config stp ports 1-24 state enable
```

Настройка коммутатора 2

Активизируйте функцию связующего дерева:

```
enable stp
```

Проверьте текущую конфигурацию протокола связующего дерева:

```
show stp
```

Протокол RSTP используется по умолчанию. Если нет, включите его:

```
config stp version rstp
```

Назначьте порты 1–24 граничными портами:

```
config stp ports 1-24 edge true
```

Активизируйте протокол связующего дерева на портах:

```
config stp ports 1-24 state enable
```

Соедините между собой коммутаторы 1 и 2 с помощью двух кабелей, как показано на рис. 3.14. Проверьте настройки RSTP, состояние портов и их роли у обоих коммутаторов:

```
show stp ports x, где x - номер порта
```

Посмотрите, какой коммутатор стал корневым, какие порты стали заблокированными. Выполните от компьютера ПК1 до ПК2 и наоборот команду ping, и не останавливайте ее до окончания выполнения задания:

- на ПК1: ping 192.168.1.13 –t;
- на ПК2: ping 192.168.1.12 –t.

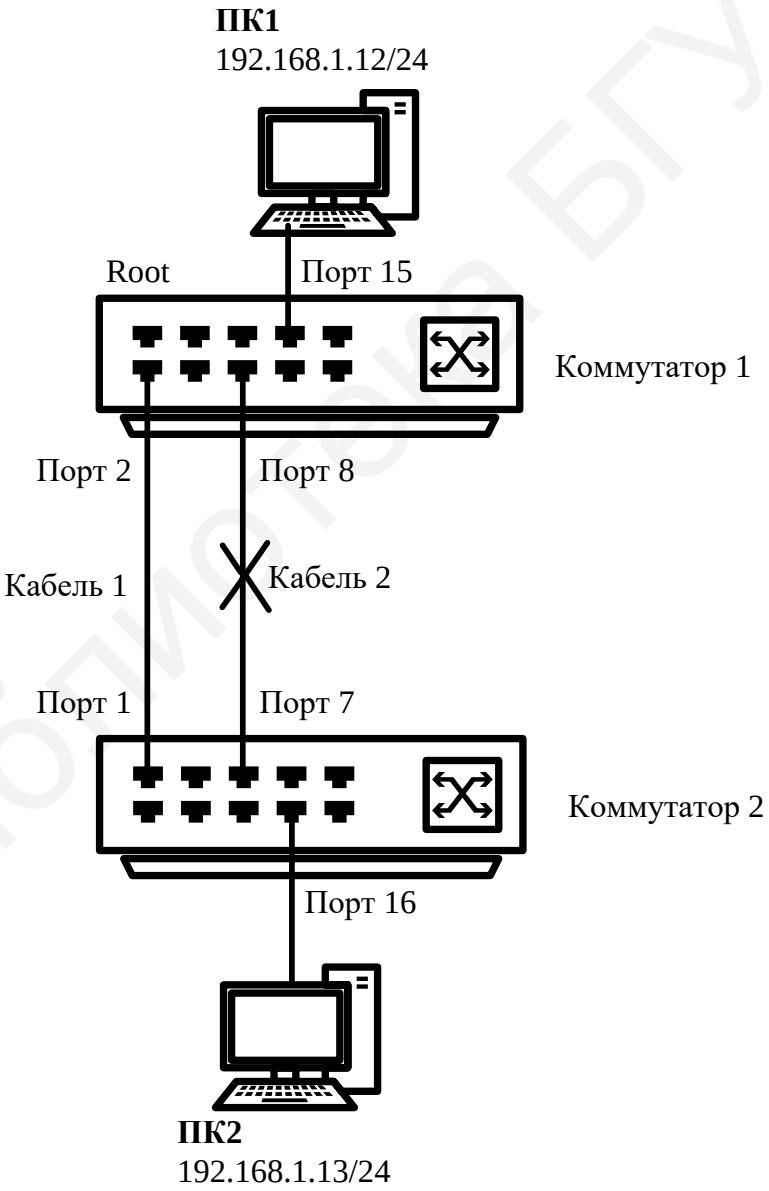


Рис. 3.14. Схема лабораторной сети для настройки протокола связующего дерева RSTP

Отсоедините кабель от корневого порта и посмотрите, что происходит с тестом ping. Проверьте состояние заблокированного порта и его роль в настоящий момент.

Снова подключите обратно кабель. Поменяйте версию протокола связующего дерева с RSTP на STP на обоих коммутаторах командой

```
config stp version stp
```

Настройка защиты от несанкционированного подключения корневых коммутаторов

Настройте на коммутаторе 1 защиту от несанкционированного подключения корневых коммутаторов. Отключите кабели, соединяющие коммутаторы.

Настройка коммутатора 1

Включите на портах 1–8 защиту от перевыборов корневого коммутатора, активизировав параметр restricted_role:

```
config stp ports 1-8 restricted_role true
```

Настройка коммутатора 2

Измените значение приоритета коммутатора 2 так, чтобы оно стало ниже значения приоритета коммутатора 1:

```
config stp priority 4096 instance_id 0
```

Соедините порты обоих коммутаторов кабелем 1, как показано на рис. 3.14. На коммутаторе 1 посмотрите log-файл:

```
show log
```

Определите, какой коммутатор является корневым. На коммутаторе 1 переключите кабель из порта 2 в порт 9, на коммутаторе 1 посмотрите

log-файл. Выясните, какой коммутатор является корневым и какая роль у порта 9 коммутатора 1.

Настройка защиты от получения ложных кадров об изменении топологии

Настройте на коммутаторе 1 защиту от получения ложных кадров об изменении топологии (TCN BPDU). Отключите кабели, соединяющие коммутаторы.

Настройка коммутатора 1

Включите на портах 1 – 8 коммутатора функцию защиты от получения ложных TCN BPDU:

```
config stp ports 1-8 restricted_tcn true
```

Очистите log-файл:

```
clear log
```

Настройка коммутатора 2

Настройте на коммутаторе приоритет по умолчанию:

```
config stp priority 4096 instance_id 0
```

Проверьте выполненные настройки:

```
show stp instance 0
```

Очистите log-файл:

```
clear log
```

Соедините между собой коммутаторы 1 и 2 с помощью двух кабелей, как показано на рис. 3.14. Соедините порты 10 и 12 коммутатора кабелем Ethernet. На коммутаторах 1 и 2 посмотрите log-файл:

```
show log
```

Отключите на коммутаторе 1 функцию защиты от получения ложных TCN BPDU:

```
config stp ports 1-8 restricted_tcn false
```

Отключите кабель, соединяющий порты 10 и 12 коммутатора 2. На коммутаторе 1 посмотрите log-файл.

3.8. Содержание отчета

1. Цель лабораторной работы.

2. Схема подключения с настроенными IP-адресами, обозначением ролей портов, VLAN, настроенными регионами и т. д.

3. Выводы по проделанной работе.

3.9. Контрольные вопросы и задания

1. Объясните понятия широковещательного шторма и лавинной передачи, причины их появления.

2. Перечислите способы борьбы с широковещательными штормами.

3. Опишите процесс построения связующего дерева по протоколу STP.

4. Каковы роли портов, назначаемые по протоколу STP?

5. Опишите процесс построения связующего дерева по протоколу RSTP.

6. Каковы роли портов, назначаемые по протоколу RSTP?

7. Опишите процесс построения связующего дерева по протоколу MSTP.

8. Каковы роли портов, назначаемые по протоколу MSTP?

Лабораторная работа №4

АДРЕСАЦИЯ СЕТЕВОГО УРОВНЯ

Цель работы: научиться определять адрес сети и адрес узла по маске подсети, количество узлов и диапазон адресов в заданной сети, а также научиться формировать подсети с использованием маски подсети.

4.1. Сетевой уровень

Важнейшей функцией сетевого уровня является поиск наилучшего маршрута для пересылки пакетов. Маршруты могут быть заданы вручную или меняться динамически в зависимости от ситуации на сети. Также они могут задаваться для каждого отдельного соединения и вычисляться для каждого нового пакета отдельно. При нахождении в подсети большого количества пакетов они могут переполнять буферы сетевых устройств в узких местах, поэтому одной из функций сетевого уровня является недопущение такого переполнения. В общем смысле сетевой уровень занимается предоставлением определенного уровня сервиса (это касается задержек, времени передачи, вопросов синхронизации). Основным протоколом сетевого уровня является протокол *IP (Internet Protocol)*.

Основная задача IP-протокола – передача данных между устройствами сети. С каждым сетевым устройством обычно ассоциирован физический адрес (MAC-адрес) на канальном уровне и логический адрес (IP-адрес) на сетевом уровне (рис. 4.1). Главным отличием адресов сетевого уровня является то, что они не привязываются к определенному устройству и могут изменяться динамически [1].

На сетевом уровне происходит передача данных с помощью пакетов. Заголовок пакета, использующего протокол IPv4, равен 20 байт.

Для обеспечения взаимодействия на сетевом уровне необходимо присвоить уникальный IP-адрес каждому интерфейсу для обеспечения гарантированной доставки пакетов нужному получателю.

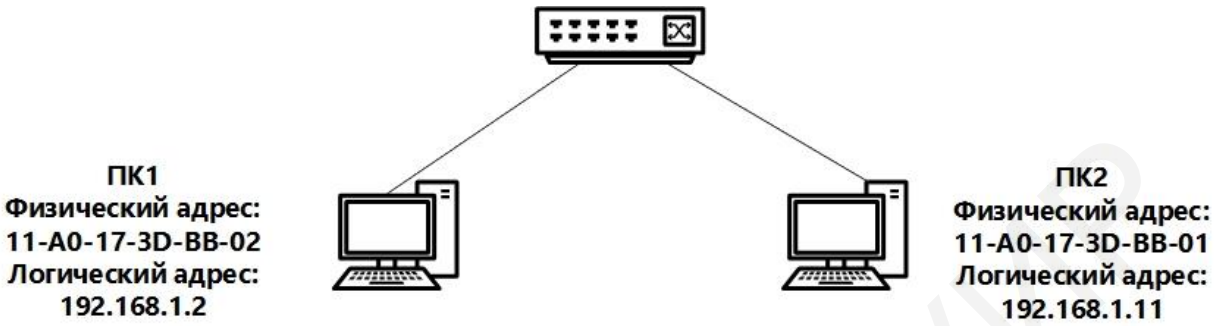


Рис. 4.1. Физические и логические адреса

4.2. Формат пакета IPv4

В заголовке содержится различная сервисная информация и данные о получателе и отправителе пакета. В поле «Данные» расположена пользовательская информация. В отличие от формата других протоколов IPv4-пакет не содержит контрольной суммы всего пакета (рис. 4.2).

Рис. 4.2. Форма пакета IPv4

IPv4-пакет состоит из следующих полей:

- *Версия (Version)* – указывает на версию протокола IP.
- *Длина заголовка (Internet Header Length)* – указывает на начало блока данных в пакете.
- *Тип сервиса (Differentiated Services Code Point)* – указывает на разделение трафика на классы обслуживания, например, для установки чувствительного к задержкам трафика большего приоритета.
- *Общая длина* – полный размер пакета в байтах, включая заголовок и данные.
- *Идентификатор пакета* – используется для идентификации фрагментов пакета, если он был фрагментирован.
- *Флаги (Flag)* – поле размером три бита, содержащее флаги контроля над фрагментацией. Биты, от старшего к младшему, означают: 0 – зарезервирован, 1 – не фрагментировать, 2 – у пакета еще есть фрагменты. Если установлен флаг «не фрагментировать», то в случае необходимости фрагментации такой пакет будет уничтожен.
- *Смещение фрагмента* – значение, определяющее позицию фрагмента в потоке данных.
- *Время жизни (Time to Live, TTL)* – определяет максимальное количество маршрутизаторов на пути следования пакетов. При прохождении коммутатора значение TTL уменьшается на единицу.
- *Протокол* – указывает, данные какого протокола верхнего уровня размещены в поле данных пакета.
- *Контрольная сумма заголовка* – используется для проверки целостности заголовка.
- *Адрес источника (SA) и адрес назначения (DA)* – содержат 32-битные адреса отправителя и получателя пакета.
- *Опции* – необязательное поле дополнительных опций. При использовании данного поля длина заголовка может быть более 20 байт в зависимости от количества опций, но всегда остается кратной 32 битам [1].

4.3. Структура адреса IPv4

Адрес формата IPv4 состоит из 32 бит или 4 байт. Адрес обычно разбивают на 4 части – по 8 бит каждая (октеты). Октеты представляют в виде десятичных чисел, разделенных точками. Преобразование IP-адреса из двоичного (бинарного) представления в десятичное показано на рис. 4.3.

В двоичной системе представления максимальное значение может быть равно 11111111, что в десятичной системе исчисления равняется 255. Соответственно адрес не может превышать заданного значения.

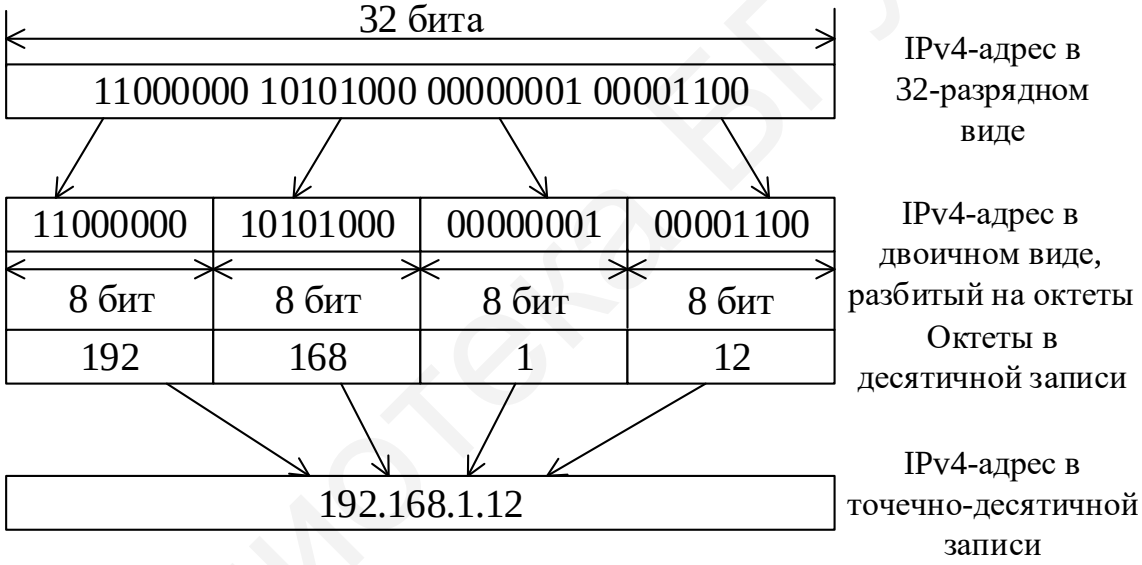


Рис. 4.3. Представление IPv4-адреса в бинарном и десятичном виде

Преобразование двоичного кода в десятичный представлено в табл. 4.1. Каждое десятичное число можно представить суммой цифр, соответствующих ненулевым битам в октете.

IP-адрес можно разделить на две логические части: *идентификатор сети (Net ID)* – сетевая часть адреса и *идентификатор узла (Host ID),* который однозначно определяет устройство в сетевом сегменте. Такое разделение является иерархическим и представляет возможность отправлять

пакеты не только в определенную сеть, но и конкретному интерфейсу в данной сети (рис. 4.4).

Таблица 4.1

Преобразование двоичного кода в десятичный

Двоичное значение октета	Значение бит октета	Десятичное значение октета
00000000	0	0
10000000	128	128
11000000	128+64	192
11100000	168+64+32	224
11110000	128+64+32+16	240
11111000	128+64+32+16+8	248
11111100	128+64+32+16+8+4	252
11111110	128+64+32+16+8+4+2	254
11111111	128+64+32+16+8+4+2+1	255

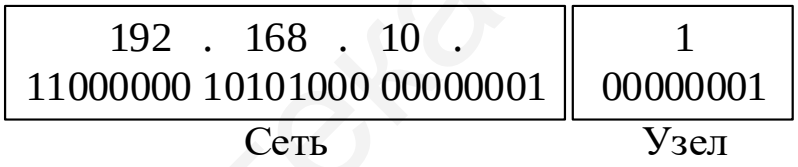


Рис. 4.4. Иерархическая структура IPv4-адреса

4.4. Классовая адресация IPv4

При размере IPv4-адреса в 32 бита можно адресовать 2^{32} = 4,3 млрд устройств. Данного количества адресного пространства в современных реалиях недостаточно для адресации в сети Интернет, что приводит к необходимости оптимизации выдаваемых адресов. Первым способом разделения адресного пространства является классовая модель, согласно которой все пространство делится на пять классов (A, B, C, D, E), которые определяются значениями первых четырех бит [1].

Первые три класса используются для индивидуальной адресации сетей и интерфейсов, класс D – для многоадресной рассылки, а класс E

зарезервирован для экспериментов. Классы A, B и C имеют различную длину части, отведенной под идентификацию адреса сети.

Сеть класса A идентифицирует сеть в первом октете, старший бит которого равен нулю. Остальные октеты используются для идентификации узлов. Такое разбиение позволяет создать 128 (2^7) сетей, в каждой из которых 16 777 214 ($2^{24} – 2$) узлов. Два адреса вычитаются вследствие того, что они используются в специальных целях и не могут быть назначены устройству (первый – адрес сети, последний – широковещательный адрес) (рис. 4.5).

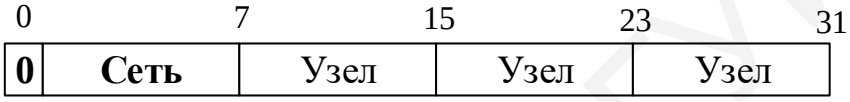


Рис. 4.5. Формат IPv4-адреса класса A

Для сетей класса B первые два октета являются идентификаторами сети. Два старших бита первого октета всегда равны 10 в двоичной системе исчисления. Такое разбиение позволяет создать 16 384 (2^{14}) сетей, в каждой из которых 65 534 ($2^{16} – 2$) узлов (рис. 4.6).

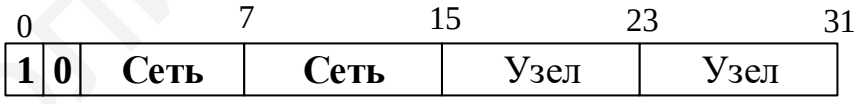


Рис. 4.6. Формат IPv4-адреса класса B

Сети класса C отводят первые три октета для идентификации сети и один октет для идентификации узлов. Три старших бита первого октета равны 110 в двоичной системе исчисления. Такое разбиение позволяет организовать 2 097 152 (2^{21}) сетей, в каждой из которых находится 254 ($2^8 – 2$) узла (рис. 4.7).

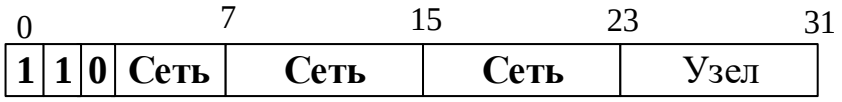


Рис. 4.7. Формат IPv4-адреса класса C

Сети класса D определяются значениями 1110 в двоичной системе исчисления старших бит первого октета. Адресное пространство класса D зарезервировано для групповой рассылки и используется для адресации группы узлов (рис. 4.8).

Рис. 4.8. Формат IPv4-адреса класса D

Сети класса E являются экспериментальными, старшие биты первого октета имеют значение 1111 в двоичной системе исчисления (рис. 4.9).

Рис. 4.9. Формат IPv4-адреса класса E

Для идентификации получателей сообщений в глобальной сети Интернет используются уникальные IPv4 адреса, которые являются публичными, но так как количество таких адресов ограничено, в каждом из классов IP-сетей определено так называемое частное пространство IP-адресов, предназначенных для работы в локальных компьютерных сетях. Для локальных сетей используются любые уникальные в ее пределах адреса.

Адресное пространство IPv4-адресов состоит из трех классов:

- 10.0.0.0 – 10.255.255.255 (класс A);
- 172.16.0.0 – 172.31.255.255 (класс B);
- 192.168.0.0 – 192.168.255.255 (класс C).

Также определены специальные IPv4-адреса (табл. 4.2).

Таблица 4.2

Специальные IP-адреса

Идентификатор сети	Идентификатор узла	Описание
Все «0»	Все «0»	Адрес узла, сгенерировавшего пакет, используется устройством для ссылки на самого себя, если оно не знает свой IPv4-адрес
Все «0»	Идентификатор узла	Узел назначения принадлежит той же сети, что и узел-отправитель
Идентификатор сети	Все «0»	Адрес IPv4-сети
Идентификатор сети	Все «1»	Ограниченный широковещательный адрес
Все «1»	Все «1»	Глобальный широковещательный адрес
127.0.0.0		Адрес интерфейса обратной петли предназначен для тестирования оборудования без реального отправления пакета

Со временем для увеличения адресного пространства была добавлена новая процедура разбиения сетей на подсети и в структуру IPv4-адреса был добавлен новый уровень – *подсеть*. IPv4-адрес остался 32-разрядным, но часть, отвечающая ранее только за идентификацию узлов, стала отвечать еще и за идентификацию подсети (рис. 4.10).

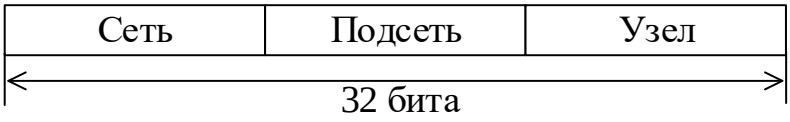


Рис. 4.10. Трехуровневая иерархия IPv4-адреса

Такое разбиение позволяет более рационально использовать адресное пространство (не целиком весь класс, а только его часть), а также повысить управляемость и безопасность сети [1].

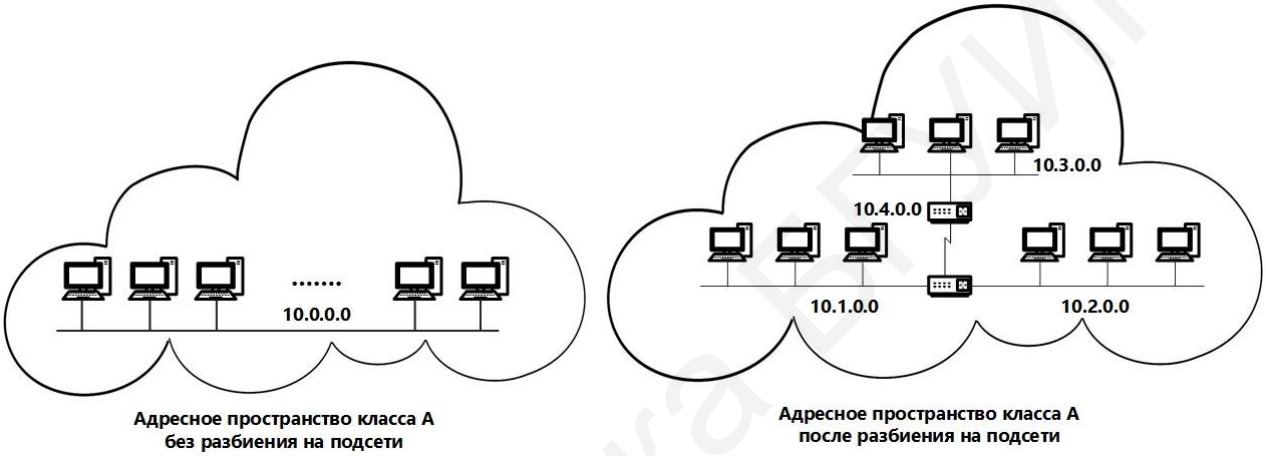


Рис. 4.11. Пример разбиения на подсети

С новым подходом к разделению появился идентификатор подсети, который представляет собой битовую маску (маска подсети), отделяющую адресное пространство, отвечающее за идентификацию узлов в сети, от идентификаторов подсети. Маска подсети представляет собой 32-битное число, в двоичной записи которого нули содержатся в тех разрядах, которые отвечают за идентификацию узлов, остальные разряды содержат единицы, отвечающие за идентификацию сети. Последовательность единиц должна быть непрерывной (рис. 4.12). Использование маски позволяет создать большое количество сетей с меньшим количеством узлов для более эффективного использования адресного пространства.

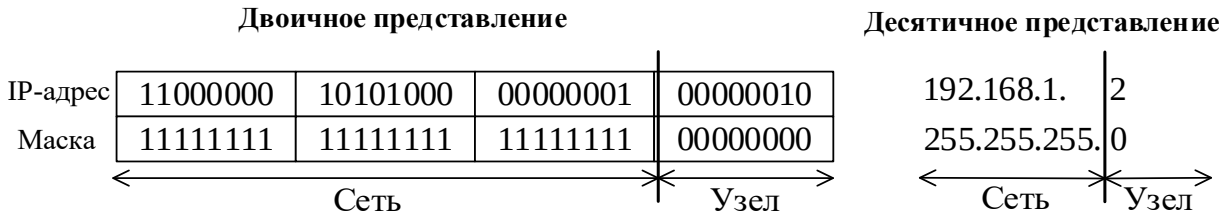


Рис. 4.12. Формирование маски подсети

Для расчета адреса сети по известному IPv4-адресу используется операция Логическое И для IP-адреса и маски подсети (рис. 4.13) [4].

IP-адрес	11000000	10101000	00000001	00000010	192.168.1.2
Маска	11111111	11111111	11111111	00000000	& 255.255.255.0
Адрес сети	11000000	10101000	00000001	00000010	= 192.168.1.0

Рис. 4.13. Получение адреса сети из IP-адреса и маски подсети

Для сетей класса A, B и C маски подсети определены заранее (табл. 4.3.)

Чтобы узнать возможное количество подсетей, используется формула 2^s, где s – количество бит, занятых идентификатором сети из части, отданной под идентификатор узла. Количество узлов определяется по формуле $2^n - 2$, где n – количество бит для идентификации узлов, два вычитаемых адреса – это широковещательный адрес и адрес подсети. Пример разбиения сети представлен на рис. 4.14 [4].

Таблица 4.3

Маски подсети для стандартных классов сетей

Класс сети	Маска подсети	Количество бит под идентификатор сети
Класс A	255.0.0.0	8
Класс B	255.255.0.0	16
Класс C	255.255.255.0	24

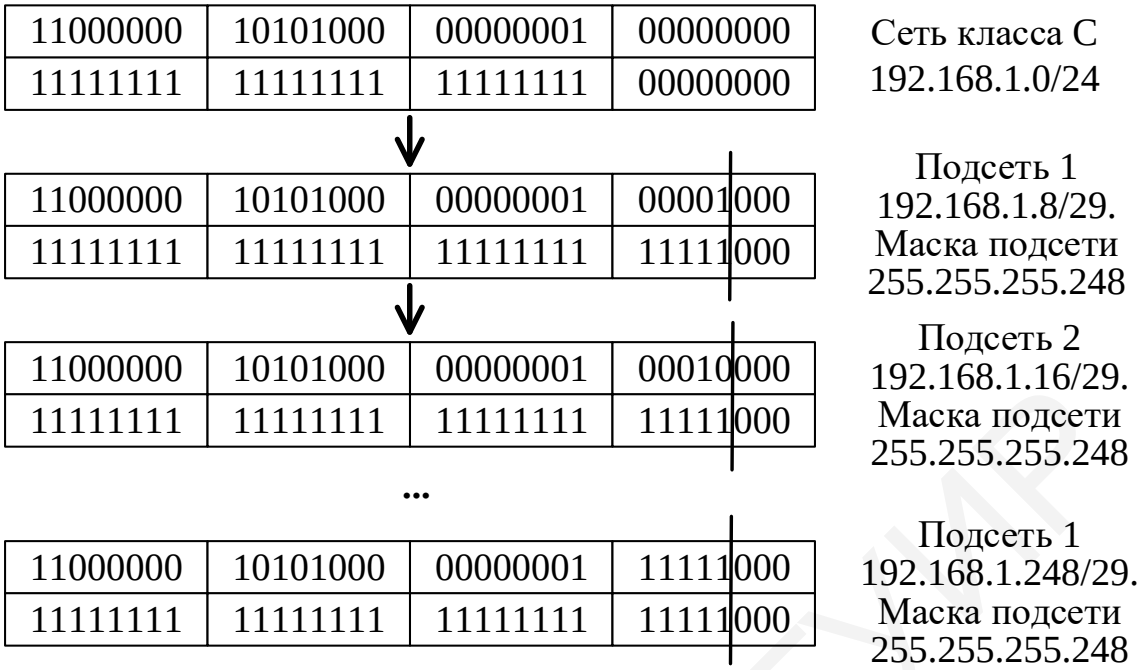


Рис. 4.14. Пример разбиения сети 192.168.1.0/24 на подсети

Для корректной работы необходимо обеспечить одну и ту же маску подсети в одном сегменте сети.

4.5. Бесклассовая адресация IPv4

Несмотря на все способы оптимизации адресного пространства IPv4, классовая модель быстро исчерпала свои возможности. Также классовая модель нерационально использует адресное пространство для слишком маленьких или слишком больших сетей. Еще одним недостатком классовой адресации является разбиение на подсети одинаковых размеров, что в свою очередь также приводит к нерациональному использованию адресного пространства.

Со временем появилась *бесклассовая модель IPv4-адресации, использующая маски подсети переменной длины (Variable Length Subnet Mask, VLSM)* и *технологию бесклассовой междоменной маршрутизации*

(Classless Inter Domain Routing, CIDR). Термин «маска переменной длины» означает, что сеть может быть разбита на подсети с различными масками подсети, т. е. сеть можно разбить на подсети столько раз, сколько необходимо, чтобы использовать различные маски. Такие маски записываются в виде нотации «IP-адрес/длина префикса». Число после «/» означает количество единичных разрядов в маске подсети. Например, сетевой адрес 192.168.1.8 с маской подсети 255.255.255.248 также может быть записан как 192.168.1.8/29. Число 29 указывает, что маска подсети 255.255.255.248 состоит из 29 единичных бит.

На рис. 4.15 показано разделение адресного пространства на шесть подсетей с различным количеством адресов с помощью масок переменной длины.

Если организация желает разделить свою сеть класса С 192.168.1.0/24 на шесть подсетей, то в четырех из них должно быть 10 узлов, в оставшихся двух – 50 и 100. Максимальное количество узлов в сети класса С – 254, разбить сеть можно с заданными требованиями только при помощи технологии VLSM.

Первоначально сеть 192.168.1.0/24 разделяется на две подсети: 192.168.1.0/25 и 192.168.1.128/25. Сеть 192.168.1.0/25, в которой 126 узлов, оставим в первоначальном виде. Сеть 192.168.1.128/25 еще раз разделим на две подсети: 192.168.1.128/26 и 192.168.1.192/26, в каждой из которых будет по 62 узла. Далее сеть 192.168.1.192/26 разделим на четыре, в каждой из которых будет по 14 узлов.

IPv4-адреса могут быть заданы динамически с помощью протокола динамического назначения адресов DHCP (Dynamic Host Configuration Protocol) [4].

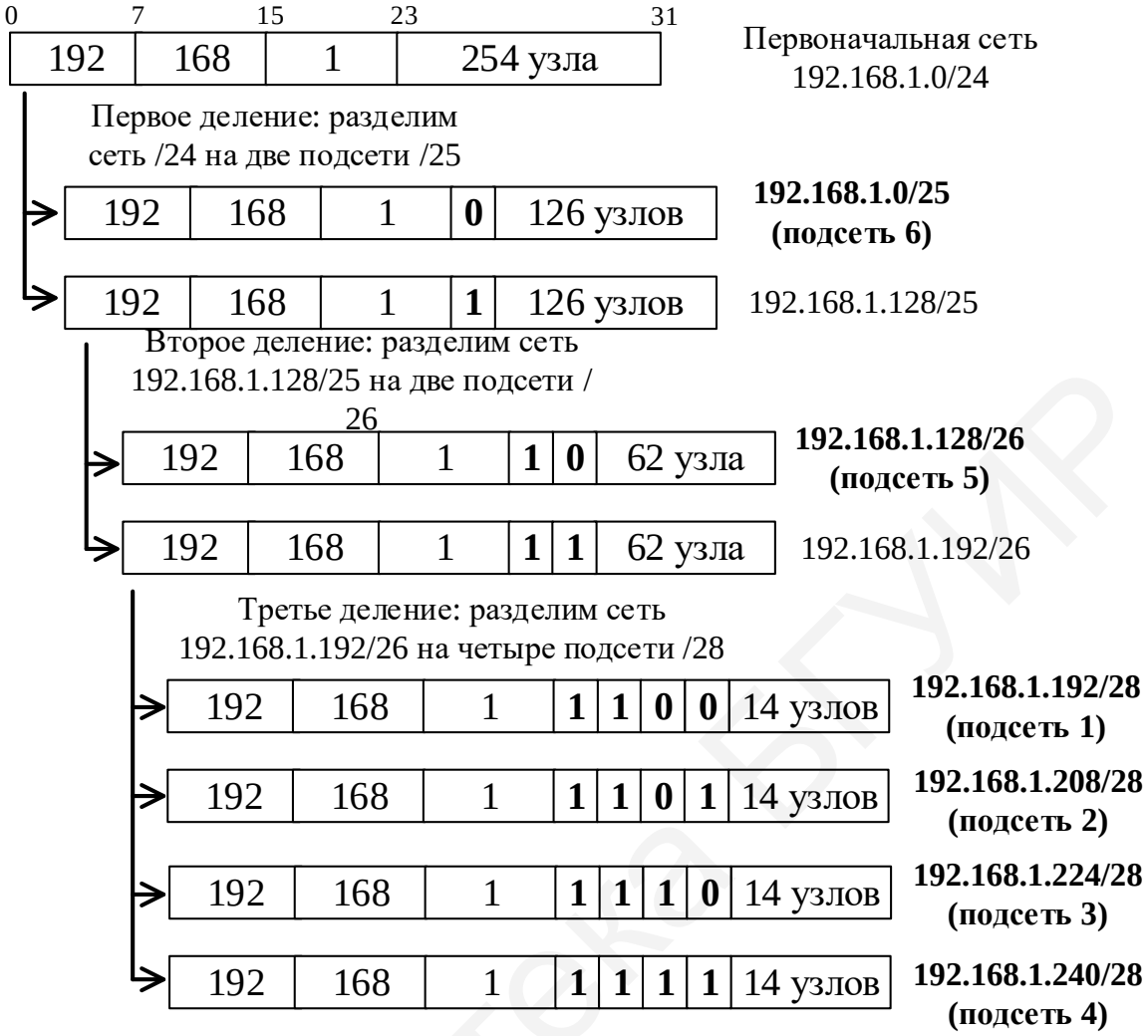


Рис. 4.15. Пример разбиения сети 192.168.1.0/24 на подсети при помощи VLSM

4.6. Указания по выполнению лабораторной работы

Лабораторная работа выполняется индивидуально каждым студентом по выданному преподавателем варианту.

Для того чтобы устройство могло участвовать в сетевом взаимодействии с помощью протокола IP, ему должен быть присвоен уникальный IP-адрес (логический адрес), который позволяет однозначно идентифицировать интерфейс между устройством и сетью. Это требуется для обеспечения гарантии передачи пакета конкретному получателю. Отметим,

что IP-адрес присваивается не конкретному устройству, а его интерфейсу. Любое устройство, которое передает данные, используя сетевой уровень, будет иметь как минимум один уникальный IP-адрес для сетевого интерфейса.

Определение адреса сети, широковещательного адреса и количества узлов по заданному IP-адресу и маске подсети

По IP-адресу узла 10.193.68.59 и маске подсети 255.255.248.0 определите:

- десятичное представление сети;
- широковещательный адрес в десятичном и двоичном представлении;
- IP-адрес первого узла подсети в десятичном и двоичном представлении;
- IP-адрес последнего узла подсети в десятичном и двоичном представлении.

Шаг 1. Переведите IP-адрес узла и маску подсети в двоичный вид.

Шаг 2. Определите адрес сети. Для этого примените к IP-адресу и маске подсети операцию Логическое И (&).

Шаг 3. Определите широковещательный адрес подсети. Маска подсети позволяет определить, какая часть адреса указывает на идентификатор подсети, а какая – на идентификатор узла. Широковещательный адрес содержит единицы в тех разрядах, которые должны определяться как идентификатор узла.

Шаг 4. Определите IP-адрес первого узла подсети. Этот IP-адрес всегда на единицу больше адреса сети.

Шаг 5. Определите IP-адрес последнего узла подсети. Этот IP-адрес всегда на единицу меньше широковещательного адреса подсети.

Шаг 6. Определите количество узлов в подсети. Количество узлов в подсети вычисляется по формуле $2^n - 2$, где n – количество бит, оставшихся в части, идентифицирующей узел, а два адреса – адрес сети и широковещательный адрес – не могут быть назначены узлу.

Формирование подсетей

Организации необходимо разбить сеть 152.79.0.0 на пять подсетей с одинаковым количеством узлов в каждой.

Шаг 1. Определите, к какому классу относится адрес 152.79.0.0. 152.79.0.0. Это класс B, соответственно, стандартная маска подсети для класса B равна 255.255.0.0 и под идентификатор узла отведены последние два октета.

Шаг 2. Определите количество бит, которое необходимо занять от идентификатора узла для формирования пяти подсетей. Так как найти число, при котором степень 2 будет равна 5, невозможно, выбираем ближайшее большее число $2^3 = 8$. Таким образом, три первых бита идентификатора узла будут использованы для идентификации подсети, а оставшиеся 13 бит – для идентификации узлов в них. Результат разбиения представлен в табл. 4.4.

Формирование подсетей с использованием масок переменной длины (VLSM)

Организации выделена сеть 204.15.5.0/24. Требуется разделить данную сеть на пять подсетей: в первой и второй должно быть 28 узлов, в третьей – 14 узлов, в четвертой – 7 узлов, в пятой – 2 узла.

Таблица 4.4

Результат разбиения сети 152.79.0.0 на пять подсетей

Номер подсети	Адрес подсети	Маска подсети	Количество узлов
1	152.79.0.0	255.255.224.0	8190
2	152.79.32.0		
3	152.79.64.0		
4	152.79.96.0		
5	152.79.128.0		

Шаг 1. Определите количество бит, необходимое для адресации 28 узлов. Количество узлов в подсети определяется по формуле $2^n - 2$, где n – количество бит, оставшихся в идентификаторе узла. Выбираем ближайшее большее число $2^5 = 32$. Таким образом, три первых бита идентификатора узла будут использованы для идентификации подсети, а оставшиеся пять бит – для идентификации узлов в них.

Шаг 2. Три первых бита идентификатора узла позволяют разделить сеть 204.15.5.0/24 на восемь подсетей, в каждой из которых может быть по 30 узлов (не забывайте про два зарезервированных адреса, которые не могут быть назначены узлам – это адрес сети и широковещательный адрес). Первые две подсети оставьте, так как требуется, чтобы в первой и второй подсети было 28 узлов, а третью (204.15.5.64/32) разделите на подсети с меньшим количеством узлов. Полученные подсети и количество узлов в каждой представлены в табл. 4.5.

Шаг 3. Разделите подсеть 204.15.5.64/27 на две подсети. Для этого займите один бит из оставшихся пяти, отведенных под идентификатор узла. Таким образом, получится две подсети 204.15.5.64/28 и 204.15.5.80/28, в каждой из которых допустимое количество узлов равно 14 ($2^4 - 2$). Две полученные подсети позволяют адресовать требуемое количество узлов, необходимое для подсетей 3 и 4.

Шаг 4. Для получения пятой подсети разделите сеть 204.15.5.96/27 на подсети, в каждой из которой должно быть по два узла. Для этого займите три бита из оставшихся пяти, отведенных под идентификатор узла. В результате получится восемь подсетей.

Таблица 4.5

Количество узлов в каждой подсети

Номер подсети	Адрес подсети/префикс	Количество узлов
1	204.15.5.0/27	30
2	204.15.5.32/27	30
3	204.15.5.64/28	14
4	204.15.5.80/28	14
5	204.15.5.96/30	2

4.7. Содержание отчета

1. Цель лабораторной работы.

2. Задание для самостоятельного выполнения в соответствии с вариантом.

3. Выводы по проделанной работе.

4.8. Задания для самостоятельного выполнения и контрольные вопросы

Вариант №1

IP-адрес узла – 10.128.100.32, маска подсети – 255.128.0.0.

1. Найти адрес сети.

2. Разбить на равные восемь подсетей, указать маску в десятичной системе, первый и последний IP-адрес для каждой подсети.

3. Четвертую, полученную в п. 2 сеть, разбить на четыре сети, указать маску, первый и последний IP-адрес.

4. Третью, полученную в п. 3 сеть, разбить на сети, в каждой из которых не менее 15 000 пользователей, указать маску, первый и последний IP-адрес.

5. Из двенадцатой, полученной в п. 4 сети, выделить две сети по 3500 пользователей, указать маску, первый и последний IP-адрес .

6. Последнюю сеть, полученную в п. 5, разбить для использования p2p-подключений между устройствами. Записать пять полученных подсетей, маску, первый и последний IP-адрес.

7. Из шестой полученной в п. 4 сети выделить одну сеть на 2000 пользователей и две, каждая из которых рассчитана на 400 пользователей.

Вариант №2

IP-адрес узла – 172.230.128.64, маска подсети – 255.252.0.0.

1. Найти адрес сети.

2. Разбить на равные шесть подсетей, указать маску в десятичной системе, первый и последний IP-адрес для каждой подсети.

3. Четвертую, полученную в п. 2 сеть, разбить на четыре сети, указать маску, первый и последний IP-адрес.

4. Третью, полученную в п. 3 сеть, разбить на сети, в каждой из которых не менее 4000 пользователей, указать маску, первый и последний IP-адрес.

5. Из двенадцатой, полученной в п. 4 сети, выделить две сети по 1000 пользователей, указать маску, первый и последний IP-адрес.

6. Последнюю сеть, полученную в п. 5, разбить для использования p2p-подключений между устройствами. Записать пять полученных подсетей, маску, первый и последний IP-адрес.

7. Из шестой, полученной в п. 4 сети, выделить две сети на 200 пользователей и три сети, каждая из которых рассчитана на 50 пользователей.

Контрольные вопросы и задания

1. По IP-адресу узла 172.30.1.33 и маске подсети 255.255.224.0 определите:

- адрес сети (десятичное представление);
- адрес сети (двоичное представление);
- широковещательный адрес (десятичное представление);
- широковещательный адрес (двоичное представление);
- Ip-адрес первого узла подсети (десятичное представление);
- Ip-адрес последнего узла подсети (десятичное представление);
- количество узлов в подсети (десятичное представление).

2. По Ip-адресу узла 192.168.100.234 и маске подсети 255.255.192.0 определите:

- адрес сети (десятичное представление);
- адрес сети (двоичное представление);
- широковещательный адрес (десятичное представление);
- широковещательный адрес (двоичное представление);
- IP-адрес первого узла подсети (десятичное представление);
- IP-адрес последнего узла подсети (десятичное представление);
- количество узлов в подсети (десятичное представление).

3. По IP-адресу узла 172.17.99.171 и маске подсети 255.255.240.0 определите:

- адрес сети (десятичное представление);
- адрес сети (двоичное представление);
- широковещательный адрес (десятичное представление);

- широковещательный адрес (двоичное представление);

- IP-адрес первого узла подсети (десятичное представление);

- IP-адрес последнего узла подсети (десятичное представление);

- количество узлов в подсети (десятичное представление).

4. Организации требуется создать подсеть 172.16.0.0, в которой должно быть 1000 узлов. Какую маску подсети необходимо использовать?

5. Организации требуется создать подсеть 192.168.12.0, в которой должно быть 55 узлов. Какую маску подсети необходимо использовать?

6. Разделите сеть 185.210.0.0 на 256 подсетей и определите количество узлов в каждой подсети. Запишите адреса подсетей 1 и 256.

7. Сколько бит необходимо занять от идентификатора узла, чтобы организовать 256 подсетей? Как определить максимальное число узлов в каждой подсети?

8. Разделите сеть 172.16.0.0 на восемь подсетей. Запишите адрес каждой подсети, маску и количество узлов.

9. Сколько бит необходимо занять от идентификатора узла, чтобы организовать восемь подсетей?

10.Организации выделена сеть 212.100.54.0/24. Требуется разделить данную сеть на семь подсетей. В первой, второй, третьей и четвертой подсетях должно быть по два узла, в пятой – 10 узлов, в шестой – 26 узлов, в седьмой – 58 узлов. Запишите адрес каждой подсети с префиксом и укажите количество узлов, доступных для каждой подсети.

11. Можно ли сеть 212.100.254.124/30 разделить на две подсети? Ответ обоснуйте.

12. Может ли маска подсети быть 255.254.128.0? Ответ обоснуйте.

13. Можно ли назначить рабочей станции IP-адрес 160.54.255.255? Ответ обоснуйте.

Лабораторная работа №5

НАСТРОЙКА СТАТИЧЕСКОЙ И ДИНАМИЧЕСКОЙ МАРШРУТИЗАЦИИ НА КОММУТАТОРАХ ТРЕТЬЕГО УРОВНЯ

Цель работы: изучить настройку статической и динамической маршрутизации на коммутаторах D-Link.

5.1. Общие правила работы алгоритмов маршрутизации

Задача маршрутизации решается на сетевом уровне, она включает в себя две подзадачи: определение маршрута и распространение информации по сети о выбранном маршруте.

Маршрут строится через определенную последовательность узлов и интерфейсов. Часто на сети возникает несколько возможных маршрутов, выбор оптимального производится на основании нескольких критериев. В качестве критериев оптимальности могут выступать, например, номинальная пропускная способность и загруженность каналов связи, задержки, вносимые каналами, количество промежуточных транзитных узлов, надежность каналов и транзитных узлов.

Маршруты могут устанавливаться вручную администратором сети или автоматически. После определения маршрута происходит передача сервисной информации на все устройства сети о том, что делать каждому транзитному узлу с данными для дальнейшего их продвижения. При обработке получаемых сообщений на каждом транзитном устройстве создаются записи в таблице маршрутизации (табл. 5.1) [1].

В таблицах маршрутизации все записи можно разделить на четыре вида: статический маршрут (определяемый вручную администратором сети), динамический (создается устройствами с помощью различных протоколов), маршрут по умолчанию (задается как путь для передачи данных, если другой маршрут к конечному пункту не определен) и локальный (адрес непосредственно подключенной к интерфейсам маршрутизатора локальной сети).

Таблица 5.1

Пример таблицы маршрутизации

Routing Table				
IP Adress/Netmask	Gateway	Interface	Cost	Protocol
10.0.0.0/8	0.0.0.0	System	1	Local
172.16.0.0/23	172.16.8.2	iptv-1	2	RIP
172.16.2.0/23	172.16.8.2	iptv-1	2	RIP
172.16.8.0/30	0.0.0.0	iptv-1	1	Local
172.16.8.12/30	0.0.0.0	stream	1	Local

Каждая запись в таблице содержит следующую информацию: адрес назначения, маска сети, адрес шлюза (информацию о том, что получатель пакета подключен непосредственно или доступен через другой маршрутизатор), идентификатор интерфейса, метрика, определяющая стоимость маршрута, и тип протокола, используемый для маршрутизации.

Обработка пакета при маршрутизации представлена на рис. 5.1. ПК1, принадлежащий сети 192.168.1.0/24, отправляет запрос серверу, находящемуся в сети 172.11.10.0/16

Коммутатор SW1, выполняющий функции маршрутизатора, получает кадр на свой интерфейс Int1. Если после проверки кадра выяснится, что он поврежден, то устройство его отбросит, если с кадром все в порядке, то коммутатор SW1 удаляет его концовик и заголовок. Адрес назначения, полученный из заголовка кадра, сравнивается с таблицей маршрутизации. При нахождении соответствия данные пересылаются на указанный интерфейс (т. е. Int2 SW2 в нашем случае). Если соответствий найдено не было, данные будут отправлены на шлюз по умолчанию или отброшены, если такой шлюз не назначен. Отправитель в таком случае получает сообщение ICMP о том, что получатель недостижим [1].

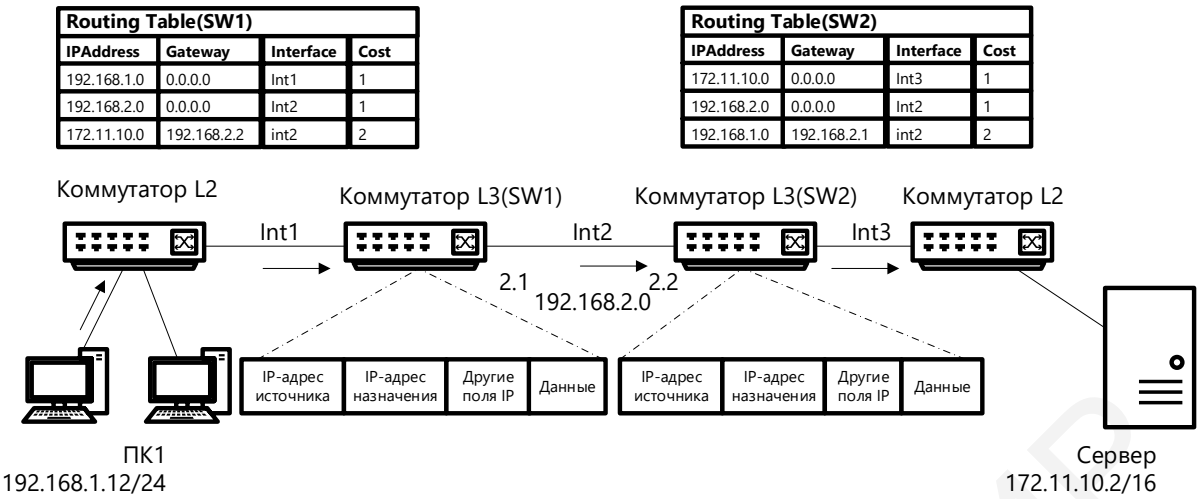


Рис. 5.1. Процесс обработки пакета маршрутизации

После получения кадра интерфейс Int2 сформирует новый кадр, инкапсулируя в него пакет, и передаст следующему транзитному узлу, т. е. маршрутизатору SW2.

Следует отметить, что коммутаторы третьего уровня имеют отличия от коммутаторов второго уровня и маршрутизаторов, так как они реализуют функции и коммутации, и маршрутизации.

Как говорилось ранее, таблицы маршрутизации могут строиться статически и динамически. При динамическом построении таблицы записи добавляются с помощью протоколов маршрутизации, при статической маршрутизации таблица заполняется вручную. Каждый из методов имеет свои достоинства и недостатки. Например, статическая маршрутизация применима на небольшой сети и в случае необходимости скрыть информацию о маршруте, например, для повышения безопасности сети. Однако такое построение сети создает единую точку отказа.

При использовании динамической маршрутизации в случае обрыва соединения сетевая топология будет обновлена в автоматическом режиме.

Все протоколы маршрутизации можно разделить на две большие группы: внутренние (используемые для передачи данных внутри автономных систем) и внешние (используемые для передачи данных между автономными системами).

К внутренним протоколам маршрутизации можно отнести: RIPv1 (Routing Information Protocol version 1); RIPv2 (Routing Information Protocol version 2); RIPng (Routing Information Protocol next generation); OSPF (Open Shortest Path First). Внешним протоколом является BGP (Border Gateway Protocol).

Под алгоритмом маршрутизации понимается метод, используемый протоколами маршрутизации для поиска оптимального маршрута и занесения его в таблицу маршрутизации. Для поиска маршрута используются различные *метрики*, отражающие информацию о пропускной способности каналов, задержке передачи пакета, скорости, числе промежуточных узлов или переходов, надежности линии связи, средней загруженности каналов связи, стоимости и др. Чем меньше метрика, тем оптимальнее маршрут [1].

На основе используемых алгоритмов все протоколы маршрутизации подразделяются на три вида:

- дистанционно-векторные протоколы;
- протоколы с учетом состояния канала;
- гибридные протоколы маршрутизации.

Дистанционно-векторные протоколы используют в своей работе алгоритм Беллмана – Форда. Каждый возможный маршрут описывается двумя параметрами (рис. 5.2) – числом переходов и *вектором*, т. е. направлением к сети назначения. При работе алгоритма маршрутизатор с заданным переводом будет рассылать свою таблицу маршрутизации подключенным маршрутизаторам. Получив таблицу от соседнего устройства, маршрутизатор будет обновлять свою, добавляя при этом к метрикам расстояния единицу и передавая эту информацию дальше. Такая рассылка происходит постоянно, даже если нет смены топологии.

При включении маршрутизатора в работу с использованием дистанционно-векторного алгоритма устройство начинает исследовать всех своих соседей, т. е. непосредственно подключенные устройства. Все они в таблице маршрутизации будут иметь число переходов, равное единице.

После проверки соседние устройства получают широковещательный запрос с просьбой прислать их таблицы маршрутизации вновь подключенному маршрутизатору. Соседние маршрутизаторы получают друг от друга полные таблицы маршрутизации и сравнивают их со своей таблицей маршрутизации. Таблица маршрутизации добавляет все новые пути или обновляет старые, если они имеют лучшие метрики. Все новые записи получают увеличение метрики расстояния на единицу для корректного отображения дальности расстояния от исходного маршрутизирующего устройства [1].

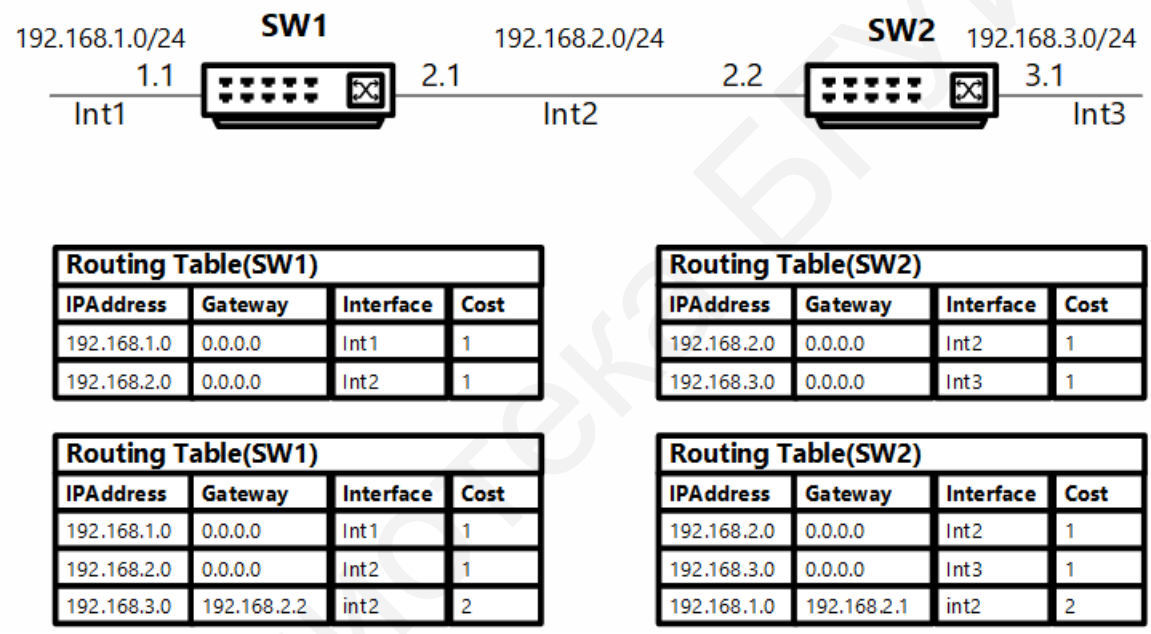


Рис. 5.2. Принцип работы дистанционно-векторного протокола

Главным недостатком дистанционно-векторного протокола является медленная сходимость, вследствие чего на сети могут возникнуть петли. Под *сходимостью сети* понимают получение всеми маршрутизаторами информации о топологии сети, под *петлей маршрутизации* – ситуацию, когда маршрутизаторы начинают пересылать пакеты по замкнутому маршруту, без возможности быть доставленными в пункт назначения.

Для устранения возможности образования петли маршрутизации используется несколько подходов: метод расщепления горизонта; ограничение максимального числа переходов; испорченный обратный маршрут; триггерные обновления; установка таймеров удержания.

Для обозначения недоступности некоторого маршрута дистанционно-векторные протоколы используют рассылку обновлений таблицы маршрутизации с метрикой бесконечности, т. е. максимальное значение метрики. Например, для протокола RIP максимальное значение метрики равно 15, если значение равно 16, то это означает, что сеть является недостижимой.

Еще одним подходом к решению проблемы возникновения петель маршрутизации является метод *расщепления горизонта*, заключающийся в ограничении передачи информации о маршруте тому устройству, от которого она получена, т. е. маршрутизатор рассылает информацию об обновлении сети, но не передает данные о маршрутах, полученных в этой сети (рис. 5.3). Однако этот метод не всегда срабатывает, особенно если большое количество маршрутизаторов подключено не напрямую друг к другу.

Метод испорченного обратного маршрута является продолжением метода расщепления горизонта. В данном случае, если устройство получает информацию о том, что какой-то маршрут недоступен, то оно отсылает в обратном направлении маршрут с недостижимой метрикой, т. е. маршрут «портится» (рис. 5.4) [1].

Следующим является *механизм установки таймера удержания,* позволяющий маршрутизирующему устройству запускать таймер при получении сообщения о недоступности пути от соседнего маршрутизатора и игнорировать в течение установленного времени все приходящие обновления о маршрутах с худшей метрикой.

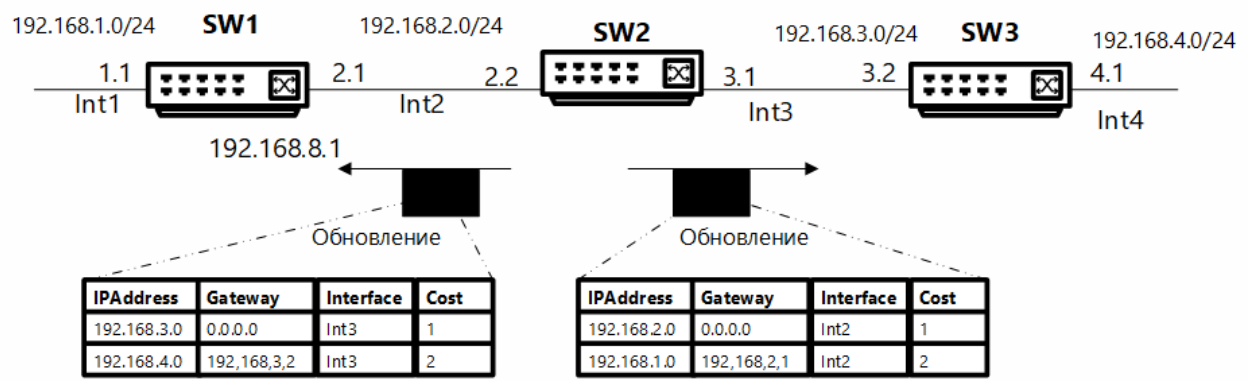


Рис. 5.3. Механизм расщепления горизонта

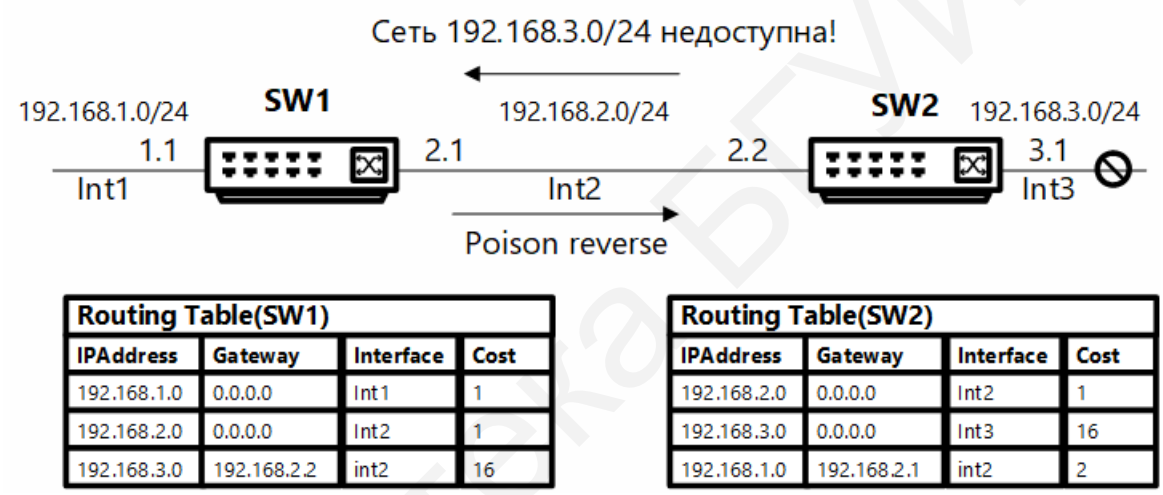


Рис. 5.4. Метод испорченного обратного маршрута

Устранение проблемы петель маршрутизации также возможно при увеличении скорости передачи информации о недоступности маршрута, что снижает вероятность зацикливания пакетов. В данном случае применяется *метод триггерных обновлений*, построенный на мгновенной передаче новой маршрутной информации, без прохождения цикла обновления. Такой подход может привести к перегрузке сети служебными сообщениями, поэтому отправка триггерных обновлений происходит с небольшой задержкой. Однако применение только триггерных обновлений, без дополнительных методов, не исключает полностью возможность возникновения петель маршрутизации.

5.2. Протокол маршрутизации RIP

Протокол RIP (Routing Information Protocol) основан на дистанционно-векторном алгоритме, в качестве метрики для построения маршрута использует *количество переходов* (остальные метрики, такие как загруженность канала, задержка или надежность, не рассматриваются). Он используется в небольших однородных сетях, где максимальный путь составляет 15 переходов. В случае если маршрутизатор подключен к сети, количество переходов до нее равно единице. При работе протокола широковещательная рассылка информации о маршрутах повторяется каждые 30 с. При получении информации от соседнего устройства маршрутизатор заносит все новые записи в таблицу маршрутизации и увеличивает число переходов к соответствующей сети на единицу. Протокол RIP может привести к петлям маршрутизации, поэтому в нем предусмотрено использование механизмов испорченного обратного маршрута, расщепления горизонта или таймеров удержания [1].

В настоящее время существует три версии протокола: RIPv1 используется для поддержки классовой адресации протокола IPv4, RIPv2 – для поддержки бесклассовой адресации IPv4, RIPng – для протокола IPv6.

5.3. Протокол маршрутизации RIPv1

Протокол RIPv1 может работать только с *классовой адресацией*, т. е. он не передает информацию о маске подсети. В RIPv1 есть два типа сообщений: *запрос*, отправляемый маршрутизатору с просьбой прислать таблицу маршрутизации, и *ответ* – сообщение, содержащее запрашиваемую ранее информацию (рис. 5.5).

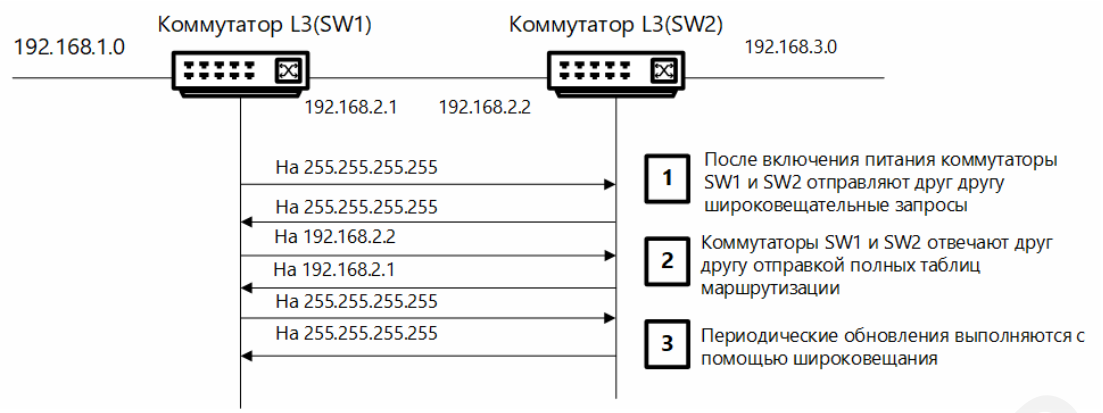


Рис. 5.5. Обмен RIP-сообщениями между коммутаторами

Для передачи сообщений протокол RIP использует протокол UDP (порт 520). При включении маршрутизатора в сеть отправляются RIP-запросы в непосредственно подключенные к нему сети, чтобы запросить таблицы маршрутизации своих соседей. При получении сообщения с запросом маршрутизатор обрабатывает его и отправляет RIP-ответ с таблицей маршрутизации. При штатной работе устройства широковещательно отправляют соседям RIP-ответ по *таймеру обновлений* (по умолчанию 30 с). Это гарантирует регулярное обновление маршрутной информации. Формат сообщения RIPv1 показан на рис. 5.6 [1].

Сообщение RIPv1 состоит из следующих полей:

- *Команда:* 1 – запрос на получение таблицы маршрутизации; 2 – ответ, содержащий таблицу маршрутизации.
- *Версия* – 1 для RIPv1.

Далее идут записи о маршрутах (максимальное количество – 25), которые содержат следующие поля:

- *Идентификатор типа адреса* – тип протокола, используемого в соответствующей сети (для протокола IP значение равно двум).
- *IP-адрес* – IP-адрес сети назначения.
- *Метрика* – число переходов до сети, указанной в поле IP-адреса.

Рис. 5.6. Формат сообщения протокола RIPv1

Также каждая запись снабжается таймером (время старения), который обнуляется при получении информации о маршруте. Если информация о маршруте отсутствует в получаемых обновлениях, то по истечении времени, установленного таймером, маршрут помечается как недостижимый. После этого запускается таймер *«сборщик мусора»,* отсчитывающий время, по прошествии которого недостижимый маршрут полностью удаляется из таблицы маршрутизации. Значение таймеров, используемых по умолчанию протоколом RIP, приведено в табл. 5.2 [1].

Так как протокол RIPv1 может работать только с классовой маршрутизацией, то он не включает в маршрутные обновления информацию о маске подсети.

Таблица 5.2

Значения таймеров, применяемых в протоколе RIP

Название таймера	Значение по умолчанию
Update Time	30 с
Timeout Time	180 с
Garbage Collection Time	120 с

Если к одному из интерфейсов маршрутизатора подключена сеть, разбитая на подсети, то маршрутизатор будет автоматически создавать в таблице маршрутизации суммарный маршрут, основанный на классовой маске подсети. И этот суммарный маршрут будет передаваться в обновлениях (рис. 5.7) [1].

Класс адреса	Наиболее значащие биты	Диапазон значений
Class A	0000	0.0.0.0 – 126.255.255.255
Class B	1000	128.0.0.0 – 191.255.255.255
Class C	1100	192.0.0.0 – 223.255.255.255
Class D	1110	224.0.0.0 – 239.255.255.255
Class E	1111	240.0.0.0 – 255.255.255.255

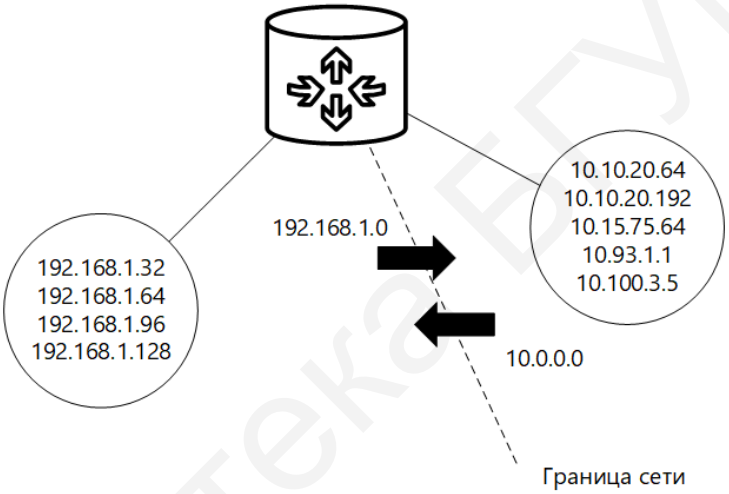


Рис. 5.7. Суммирование маршрутов
на граничном маршрутизирующем устройстве

5.4. Протокол маршрутизации RIPv2

Протокол RIPv2 является расширением RIPv1, принцип его работы такой же, как и в RIPv1, но имеются некоторые нововведения, а именно:

- *Поддержка бесклассовой адресации,* что позволяет передавать данные о маске подсети каждого IP-адреса с поддержкой масок переменной длины (VLSM) и бесклассовой маршрутизации (CIDR).
- *Указание адреса следующего маршрутизатора.* Каждая запись включает IP-адрес следующего транзитного маршрутизатора, что позволяет повысить эффективность маршрутизации, избежав лишних пересылок.

- *Аутентификация.* В протокол RIPv2 встроен базовый механизм аутентификации, идентифицирующий другие маршрутизаторы, прежде чем принимать RIP-сообщения от них.
- *Метки маршрута*. Все записи содержат поле *Route Tag (метка маршрута)*, содержащее дополнительную информацию о маршруте.
- *Использование многоадресной рассылки.* Для рассылки маршрутных обновлений используется не широковещательный метод, а специальный групповой адрес 224.0.0.9.

Формат сообщения протокола RIPv2 аналогичен формату RIPv1 (рис. 5.8), но в него были добавлены новые поля:

- *метка маршрута* – для корректной работы с протоколами внешней маршрутизации;
- *маска подсети*;
- *следующий шаг* – информация об IP-адресе следующего транзитного маршрутизатора.

Маршрутизаторы, работающие с протоколом RIPv1, могут также принимать маршрутные обновления RIPv2 [1].

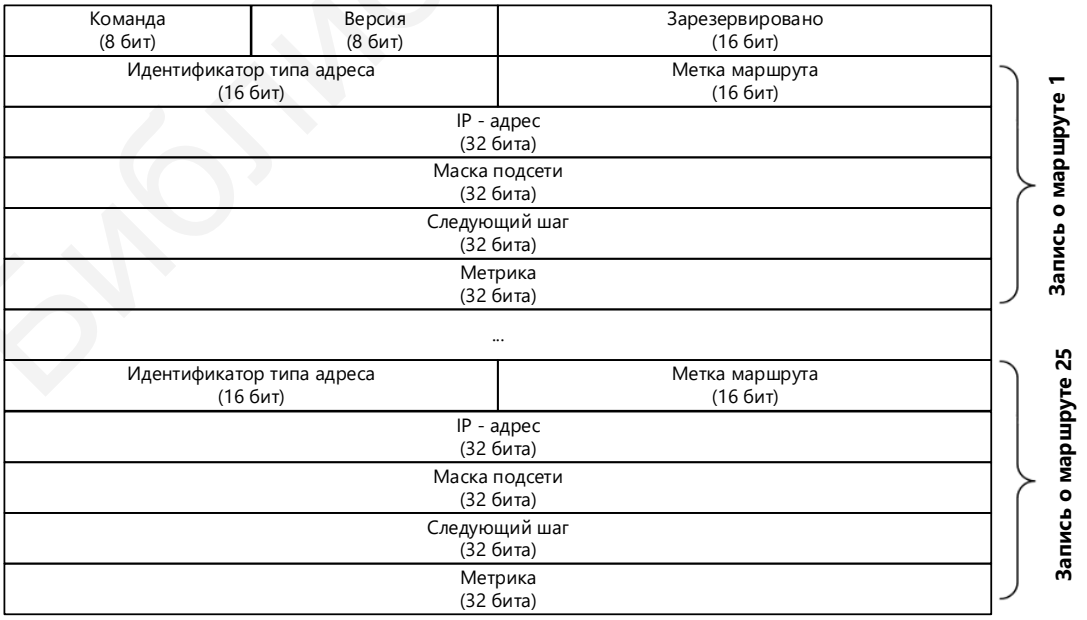


Рис. 5.8. Формат сообщения RIPv2

5.5. Указания по выполнению лабораторной работы

Настройка статической и динамической маршрутизации IPv4

Для выполнения лабораторной работы необходимо настроить статическую маршрутизацию между VLAN V5 и VLAN V6. Перед выполнением задания необходимо сбросить настройки коммутаторов к заводским настройкам по умолчанию командой

```
reset config
```

Схема подключения коммутаторов 1, 2 и 3 приведена на рис. 5.9.

Примечание. Не соединяйте коммутаторы одновременно несколькими кабелями во время настройки до особого указания [4].

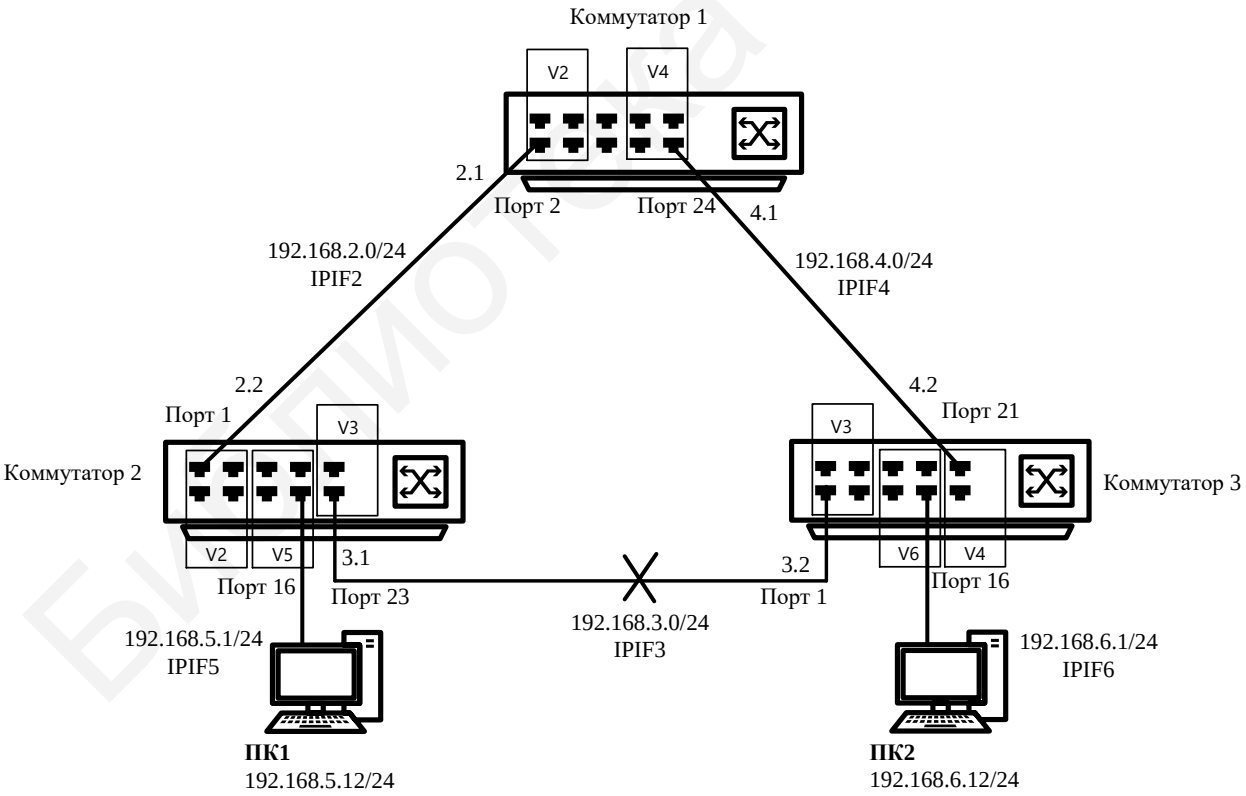


Рис. 5.9. Схема лабораторной сети для настройки статической маршрутизации

Настройка коммутатора 1

Создайте VLAN v2 и v4, добавьте в соответствующие VLAN порты, которые необходимо настроить немаркированными:

```
create vlan v2 tag 2
config vlan v2 add tagged 1-4
create vlan v4 tag 4
config vlan v4 add tagged 21-24.
```

Проверьте настройки VLAN:

```
show vlan
```

Создайте IP-интерфейс для VLAN v2 и v4 с именами IPIF2 и IPIF4 соответственно:

```
create ipif IPIF2 192.168.2.1/24 v2 state enable
create ipif IPIF4 192.168.4.1/24 v4 state enable
```

Проверьте выполненные настройки IP-интерфейсов:

```
show ipif
```

Настройка коммутатора 2

Удалите порты коммутатора из VLAN по умолчанию для их использования в других VLAN:

```
config vlan default delete 1-24
```

Создайте VLAN v2, v3 и v5, добавьте в соответствующие VLAN порты, которые необходимо настроить немаркированными:

```
create vlan v2 tag 2
config vlan v2 add tagged 1-4
create vlan v3 tag 3
config vlan v3 add tagged 21-24
create vlan v5 tag 5
config vlan v5 add untagged 7-18
```

Проверьте настройки VLAN:

```
show vlan
```

Создайте IP-интерфейс для VLAN v2, v3 и v5 с именами IPIF2, IPIF3 и IPIF5 соответственно:

```
create ipif IPIF2 192.168.2.2/24 v2 state enable
create ipif IPIF3 192.168.3.1/24 v3 state enable create
ipif IPIF5 192.168.5.1/24 v5 state enable
```

Проверьте выполненные настройки IP-интерфейсов:

```
show ipif
```

Настройка коммутатора 3

Удалите порты коммутатора из VLAN по умолчанию для их использования в других VLAN:

```
config vlan default delete 1-24
```

Создайте VLAN v3, v4 и v6, добавьте в соответствующие VLAN порты, которые необходимо настроить немаркированными:

```
create vlan v3 tag 3
config vlan v3 add tagged 1-4
create vlan v4 tag 4
config vlan v4 add tagged 21-24
create vlan v6 tag 6
config vlan v6 add untagged 7-18
```

Проверьте настройки VLAN:

```
show vlan
```

Создайте IP-интерфейс для VLAN v3, v4 и v6 с именами IPIF3, IPIF4 и IPIF6 соответственно:

```
create ipif IPIF3 192.168.3.2/24 v3 state enable
create ipif IPIF4 192.168.4.2/24 v4 state enable create
ipif IPIF6 192.168.6.1/24 v6 state enable
```

Проверьте выполненные настройки IP-интерфейсов:

```
show ipif
```

Соедините между собой коммутаторы 1, 2 и 3 с помощью Ethernet-кабелей и задайте рабочим станциям ПК1 и ПК2 IP-адреса в соответствии с рис. 5.9. В качестве шлюза по умолчанию (default gateway) укажите адрес IP-интерфейса маршрутизирующего коммутатор соответствующей VLAN. Проверьте соединение между рабочими станциями командой `ping <IP-address>`:

- от ПК1 к ПК2;
- от ПК2 к ПК1.

Настройка коммутатора 1

Проверьте таблицу маршрутизации:

```
show iproute
```

Отметьте на схеме, сколько записей в таблице маршрутизации вы наблюдаете. Создайте статический маршрут к сети 192.168.5.0/24:

```
create iproute 192.168.5.0/24 192.168.2.2
```

Примечание. Команда означает, что сеть 192.168.5.0/24 доступна через интерфейс 192.168.2.2 коммутатора 2.

Создайте статический маршрут к сети 192.168.6.0/24:

```
create iproute 192.168.6.0/24 192.168.4.2
```

Создайте статический маршрут к сети 192.168.3.0/24:

```
create iproute 192.168.3.0/24 192.168.2.2
```

Проверьте таблицу маршрутизации:

```
show iproute
```

Отметьте, сколько записей в таблице маршрутизации вы наблюдаете.

Настройка коммутатора 2

Проверьте таблицу маршрутизации:

```
show iproute
```

Запишите, сколько записей в таблице маршрутизации вы наблюдаете. Создайте статический маршрут к сети 192.168.6.0/24:

```
create iproute 192.168.6.0/24 192.168.3.2
```

Создайте статический маршрут к сети 192.168.4.0/24:

```
create iproute 192.168.4.0/24 192.168.3.2
```

Проверьте таблицу маршрутизации:

```
show iproute
```

Отметьте, как изменились записи в таблице маршрутизации. Сравните с предыдущими результатами.

Настройка коммутатора 3

Проверьте таблицу маршрутизации:

```
show iproute
```

Запишите, сколько записей в таблице маршрутизации вы наблюдаете. Создайте статический маршрут к сети 192.168.2.0/24:

```
create iproute 192.168.2.0/24 192.168.4.1
```

Создайте статический маршрут к сети 192.168.5.0/24:

```
create iproute 192.168.5.0/24 192.168.3.1
```

Проверьте таблицу маршрутизации:

```
show iproute
```

Сколько записей в таблице маршрутизации вы наблюдаете? Сравните с предыдущими результатами. Проверьте соединение между рабочими станциями командой `ping <IP-address>`:

- от ПК1 к ПК2;

- от ПК2 к ПК1.

Проверьте маршрут от ПК1 к ПК2 командой tracert. В командной строке ПК1 введите

```
tracert 192.168.6.12
```

Запишите количество переходов. Проверьте маршрут от ПК2 к ПК1 командой tracert. В командной строке ПК2 введите

```
tracert 192.168.5.12
```

Запишит количество переходов.

Настройка коммутатора 1

Удалите статический маршрут:

```
delete iproute 192.168.5.0/24 192.168.2.2.
delete iproute 192.168.6.0/24 192.168.4.2.
delete iproute 192.168.3.0/24 192.168.2.2.
```

Настройка коммутатора 2

Удалите статический маршрут:

```
delete iproute 192.168.6.0/24 192.168.3.2.
delete iproute 192.168.4.0/24 192.168.3.2.
delete iproute 192.168.3.0/24 192.168.2.2.
```

Настройка коммутатора 3

Удалите статический маршрут:

```
delete iproute 192.168.2.0/24 192.168.4.1.
delete iproute 192.168.5.0/24 192.168.3.1.
```

Настройте маршрут по умолчанию в VLAN V5 и VLAN V6.

Настройка коммутатора 1

Создайте маршрут по умолчанию:

```
create iproute default 192.168.4.2
```

Примечание. Маршрут по умолчанию используется в том случае, если другой маршрут к сети назначения неизвестен.

Проверьте таблицу маршрутизации:

```
show iproute
```

Настройка коммутатора 2

Создайте маршрут по умолчанию:

```
create iproute default 192.168.2.1
```

Настройка коммутатора 3

Создайте маршрут по умолчанию:

```
create iproute default 192.168.3.1
```

Проверьте соединение между рабочими станциями командой `ping <IP-address>`:

- от ПК1 к ПК2;

- от ПК2 к ПК1.

Проверьте маршрут от ПК1 к ПК2 командой tracert. В командной строке ПК1 введите

```
tracert 192.168.6.12
```

Запишите количество переходов. Проверьте маршрут от ПК2 к ПК1 командой tracert. В командной строке ПК2 введите

```
tracert 192.168.5.12
```

Настройка коммутатора 1

Удалите статический маршрут по умолчанию:

```
delete iproute default 192.168.4.2
```

Настройка коммутатора 2

Удалите статический маршрут по умолчанию:

```
delete iproute default 192.168.2.1
```

Настройка коммутатора 3

Удалите статический маршрут по умолчанию:

```
delete iproute default 192.168.3.1
```

Настройка протокола динамической маршрутизации RIP v2

Настройка коммутатора 1

Включите работу протокола RIP глобально на коммутаторе:

```
enable rip
```

Настройте параметры протокола RIP для всех интерфейсов:

```
config rip all tx_mode v2_only rx_mode v2_only state enable
```

Повторите процедуру настройки для коммутаторов 2 и 3. Проверьте таблицу маршрутизации:

```
show iproute
```

Запишите количество записей в таблице маршрутизации и проверьте версию и статус протокола RIP:

```
show rip
```

Проверьте соединение между рабочими станциями командой `ping <IP-address>`:

- от ПК1 к ПК2;

- от ПК2 к ПК1.

Проверьте маршрут от ПК1 к ПК2 командой tracert. В командной строке ПК1 введите

```
tracert 192.168.6.12
```

Запишите в отчет количество переходов. Проверьте маршрут от ПК2 к ПК1 командой tracert. В командной строке ПК2 введите

```
tracert 192.168.5.12
```

Отключите кабель Ethernet, соединяющий коммутаторы 2 и 3 (см. рис. 5.9). Проверьте таблицу маршрутизации:

```
show iproute
```

Запишите, сколько записей в таблице маршрутизации вы наблюдаете. Сравните с предыдущими результатами. Выключите работу протокола RIP глобально на коммутаторе:

```
disable rip
```

5.6. Содержание отчета

1. Цель лабораторной работы.

2. Схема подключения, настроенные порты, VLAN, IP-адреса, количество переходов, таблицы маршрутизации для каждого устройства

3. Выводы по проделанной работе.

5.7. Контрольные вопросы и задания

1. Опишите протоколы маршрутизации.

2. Объясните понятие маршрутизации.

3. Назовите типы записей таблицы маршрутизации.

4. Каковы особенности протокола RIP?

5. Каковы особенности протокола OSPF?

6. Каковы особенности протокола VRRP?

Лабораторная работа №6
КАЧЕСТВО ОБСЛУЖИВАНИЯ НА СЕТИ

Цель работы: изучить настройку приоритизации трафика, управление полосой пропускания на коммутаторах D-Link, исследовать эффективность работы приоритизации.

6.1. Модели QoS

Из-за разнородности трафика, передаваемого в сети, возникает вопрос о дифференциальном подходе к обеспечению различных приложений сетевыми ресурсами, так как существуют типы трафика реального времени (Voice over IP (VoIP), видеоконференции, онлайн-игры и др.), при передаче которых важнее отсутствие временных задержек, чем достоверность. При передаче данных достоверность, как правило, является основным требованием, а задержка передачи и ее вариации – не критичными.

Традиционно IP-трафик передается по методу *Best Effort Service,* что является негарантированным методом доставки. Сеть старается обработать поступающий трафик как можно быстрее, но при этом никаких гарантий относительно результатов доставки не дает. Для решения этой проблемы было введено понятие *качества обслуживания* (*Quality of Service, QoS).*

Главными функциями при предоставлении качества обслуживания являются обеспечение дифференцированной и гарантированной передачи сетевого трафика различных приложений с учетом всех ограничений скорости передачи, приоритизации, обработка очередей и механизмов распределения ресурсов.

Можно выделить три модели реализации QoS в сети [1]:

- **Негарантированная доставка данных (Best Effort Service)** – обеспечение связи без гарантии надежной доставки данных, времени доставки, пропускной способности и определенного приоритета.

- **Дифференцированное обслуживание (Differentiated Service, DiffServ)**, **мягкий QoS (Soft QoS).** Эта модель предполагает деление трафика на классы согласно требованиям к качеству обслуживания. При передаче каждого пакета в него добавляется дополнительная информация, с помощью которой все последующие узлы сети будут принимать решение о его продвижении в соответствии с политикой обслуживания трафика данного класса (Per-Hop Behavior, PHB). Данная модель не предполагает обеспечение гарантий предоставляемых услуг.
- **Интегрированные услуги (Integrated Services, IntServ), жесткий QoS (Hard QoS).** Данная модель предполагает предварительное резервирование сетевых ресурсов в целях соблюдения необходимых параметров сети для приложений, требующих гарантированной выделенной полосы пропускания на всем пути следования трафика, для обеспечения их корректной работы.

6.2. Приоритизация пакетов

Обеспечение QoS на канальном уровне происходит благодаря коммутаторам, поддерживающим стандарт IEEE 802.1p, позволяющий задать до восьми уровней приоритетов (где семь – наивысший), которые определяют способ обработки кадра, используя 3 бита поля приоритета тега IEEE 802.1Q (рис. 6.1).

Обеспечение QoS на сетевом уровне происходит за счет 8-битного поля ToS (Type of Service) (рис. 6.2) в заголовке протокола IPv4, которое может содержать либо значение приоритета IP Precedence (3 бита, диапазон значений от 0 до 7, используется для указания относительного приоритета обработки пакета на сетевом уровне), либо значение DSCP, Differentiated Services Code Point (занимает 6 старших бит байта ToS и позволяет задать до 64 уровней приоритетов от 0 до 63) [1].

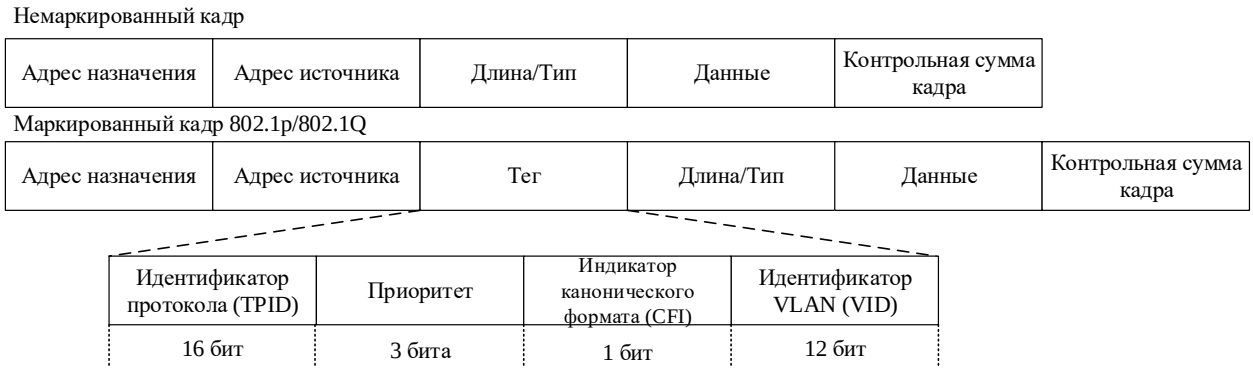


Рис. 6.1. Формат кадра 802.1Q битами приоритета 802.1p

Рис. 6.2. Байт ToS заголовка IPv4

6.3. Классификация пакетов

Для обеспечения дифференцирования при обслуживании трафика коммутаторами может поддерживаться от четырех до восьми аппаратных очередей приоритетов на каждом порту (при поддержке четырех очередей, наивысшем приоритетом обладает третья, при поддержке восьми – седьмая). Для корректной работы очередности передачи пакетов на коммутаторе необходимо настроить алгоритм обслуживания очередей и карту привязки приоритетов 802.1p, ToS, DSCP к очередям (рис. 6.3) [1].

Для того чтобы определить, к какой очереди нужно отнести пакет, в соответствии с настроенной политикой QoS происходит классификация пакетов, во время которой коммутатор анализирует содержимое одного или нескольких полей его заголовка, а именно: IP-приоритет или поле DCSP в байте ToS, приоритет 802.1p. При таком анализе коммутатор не может

изменить приоритет внутри пакетов, а только определяет очередность и способ их обработки выходным портом. При поступлении на входной порт коммутатора немаркированного кадра (т. е. заголовок кадра не содержит информации о приоритете) его классификация производится на основе значений приоритета 802.1p по умолчанию, настроенного данному порту.

Также классификация пакетов может производиться на основании таких параметров, как тег, VLAN, MAC-адрес, IP-адрес, номер порта TCP/UDP и др.

После классификации осуществляется маркировка пакетов, которая определяет способ записи значений бит приоритета (DSCP, 802.1p или IP Precedence) входящих пакетов данных [1].

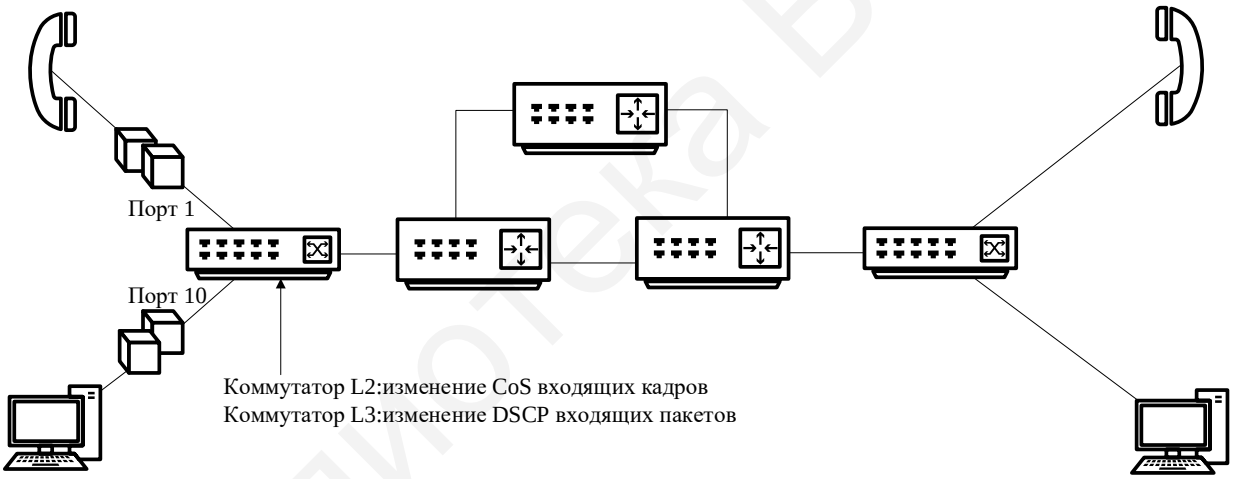


Рис. 6.3. Маркировка пакетов

Как правило, маркировка выполняется на граничные устройства, что позволяет следующим маршрутизаторам или коммутаторам применить новое значение приоритета в пакете для определения его к одному из классов обслуживания, поддерживаемых в сети. Для изменения значений бит приоритета в заголовках необходимо использовать списки управления доступом.

6.4. Управление перегрузками и механизмы обслуживания очередей

При соединении сетей с разной полосой пропускания может возникнуть перегрузка, которая приводит к буферизации пакетов, с их распределением по очередям. Для определения порядка их отправки через выходной интерфейс используется *механизм обслуживания очередей*, позволяющий управлять пропускной способностью сети. Он включает в себя следующие механизмы:

- настраиваемые очереди (Custom Queuing);
- механизм FIFO (First-In, First-Out), который передает пакеты, попавшие в порядке, в котором они поступили в очередь, однако такой подход не классифицирует пакеты и рассматривает их как принадлежащие одному классу;
- взвешенный алгоритм кругового обслуживания (Weighted Round Robin, WRR), который исключает недостаток очередей приоритетов, предоставляя полосу пропускания для пакетов из низкоприоритетных очередей и обеспечивая обработку в соответствии с назначенным весом;
- очереди приоритетов (Priority Queuing), в которых предусмотрено четыре вида очередей (с высоким, средним, обычным и низким приоритетами). Очередь с высоким приоритетом обслуживается быстрее остальных, все остальные пакеты начнут передаваться только после того, как опустеет очередь с более высоким приоритетом. Такой механизм может привести к значительной задержке при обслуживании пакетов низкоприоритетного трафика [1].

6.5. Механизм предотвращения перегрузок

Даже в случаях применения механизмов, указанных в подразд. 6.4, может возникнуть момент, когда выходные очереди будут переполнены. В этом случае на сетевых устройствах предусмотрена процедура *предотвращения перегрузок (Congestion avoidance)*, которая инициирует

выборочное удаление пакетов, пока длина всех очередей не уменьшится. Такой алгоритм управления длиной выходных очередей называется «*отбрасывание хвоста*».

При удалении пакетов источник TCP-соединения, при отсутствии подтверждения о доставке пакета, уменьшит скорость передачи до одного сегмента и перезапустит алгоритм *медленного старта*. Однако это может привести к эффекту глобальной синхронизации, когда тысячи источников TCP-соединений при возникновении перегрузки снизят свои скорости передачи и через одинаковое значение времени, установленное таймерами задержки перед увеличением скорости, что вновь приведет к увеличению интенсивности трафика и переполнению очередей в буферах сетевых устройств.

Для устранения этого недостатка был предложен *алгоритм произвольного раннего обнаружения (Random Early Detection, RED)*, который отбрасывает поступающие пакеты на основании среднего размера очередей. При превышении порогового значения пакеты начнут отбрасываться с некоторой заданной вероятностью, что поможет избежать эффекта глобальной синхронизации. Скорость отбрасывания пакетов будет возрастать пропорционально скорости роста среднего размера очереди. В случае превышения очередью максимального порогового значения будут отбрасываться все пакеты, предназначенные для постановки в очередь [1].

6.6. Контроль полосы пропускания

Современные сетевые устройства позволяют регулировать интенсивность трафика для обеспечения требуемого качества обслуживания. Для этого используются механизмы выравнивания трафика (*Traffic Policing*) и ограничения трафика (*Traffic Shapin*).

Выравнивание трафика используется для ограничения скорости передачи на интерфейсе коммутатора. При активации этой функции

администратор сети сможет установить пороговые значения скорости передачи на всех выходных портах коммутатора. При превышении порогового значения трафик будет обрабатываться в соответствии с настроенной политикой, например, отбрасываться или маркироваться новым значением приоритета.

Главным алгоритмом при ограничении трафика является «*корзина маркеров*» *(token bucket*). Данный алгоритм устанавливает следующие параметры [1]:

- **CIR, Committed Information Rate (согласованная скорость передачи)** – средняя скорость интенсивности трафика через интерфейс коммутатора или маршрутизатора, а также скорость помещения маркеров в корзину.
- **CBS, Committed Burst Size (согласованный размер всплеска)** – это объем трафика, на который может быть превышен размер корзины маркеров в отдельно взятый момент всплеска, т. е. данный параметр определяет стандартный размер корзины.
- **EBS, Extended Burst Size (расширенный размер всплеска)** – это объем трафика, на который может быть превышен размер корзины маркеров в экстренном случае.

На рис. 6.4 показана схема реализации алгоритма «корзина маркеров» в рамках механизма Traffic Policing.

Для передачи пакета из корзины вынимается некоторое число маркеров, соответствующее размеру пакета в битах. Если корзина в данный момент времени содержит достаточное число маркеров, то пакет пересылается дальше. При недостаточном количестве маркеров пакет будет считаться несоответствующим настроенному профилю. Для такого избыточного пакета могут применяться два способа обработки: перемаркировка и отбрасывание.

Механизм ограничения трафика, в отличие от механизма Traffic Policing, помещает избыточные пакеты в буфер.

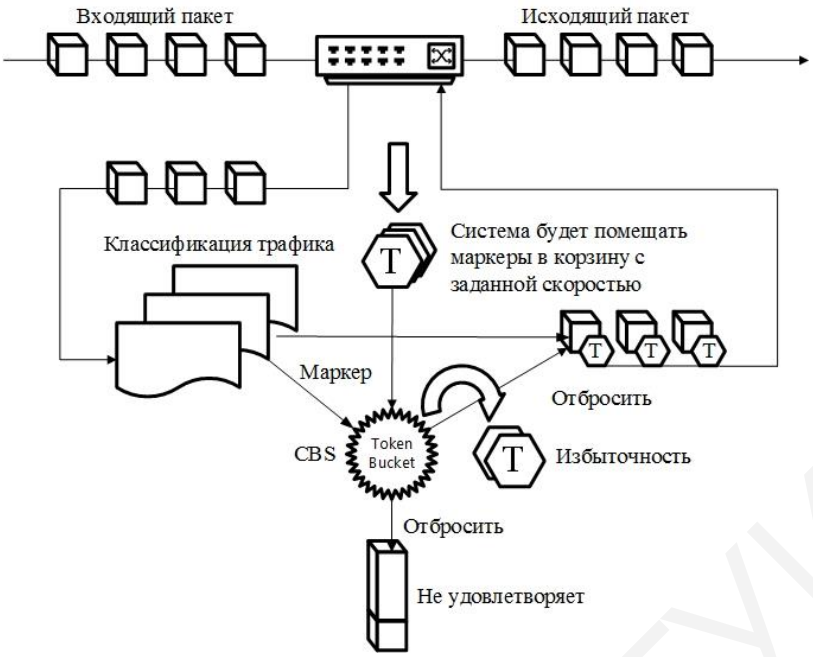


Рис. 6.4. Алгоритм «корзина маркеров»
в рамках механизма Traffic Policing

Для работы этого механизма также применяется алгоритм «корзина маркеров», однако при недостаточном количестве маркеров избыточный пакет ставится в очередь и буферизируется для последующей передачи при накоплении достаточного количества маркеров в корзине [1].

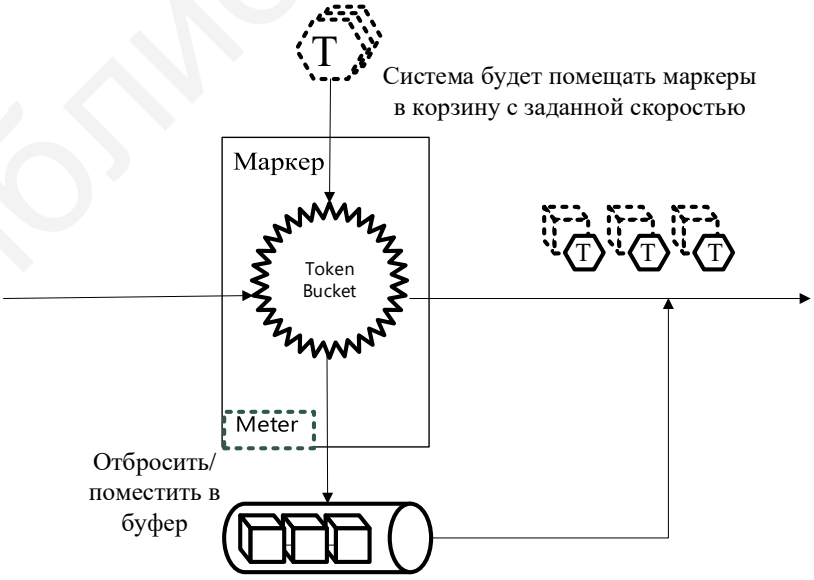


Рис. 6.5. Алгоритм «корзина маркеров»
в рамках механизма Traffic Shapin

Такой подход к избыточным пакетам вносит задержку при передаче трафика, что может привести к некорректной работе приложений и сервисов, чувствительных к задержкам, однако этот механизм более дружественен к TCP-потокам, так как благодаря буферизации уменьшается количество отбрасываемых пакетов и число их повторных передач [1].

6.7. Указания по выполнению лабораторной работы

Лабораторная работа выполняется двумя подгруппами.

Сети с коммутацией пакетов на основе протокола IP не обеспечивают гарантированной пропускной способности, поскольку не обеспечивают гарантированной доставки.

В лабораторной работе рассматривается следующий пример: в сети, имеющей явное «узкое» место, на рабочих станциях ПК1 и ПК3 выполняется тест ping друг на друга. Этому трафику необходимо обеспечить высокий приоритет обработки по сравнению с приложениями остальных станций, которые создают искусственную нагрузку на канал связи между коммутаторами с помощью программы iperf (рис. 6.6).

Возможность управления полосой пропускания на портах в коммутаторах D-Link реализуется с помощью функции Bandwidth control, которая использует для ограничения скорости механизм Traffic Policing. Можно задать вручную требуемую скорость соединения на портах в диапазоне от 64 кбит/с до максимально поддерживаемой скорости интерфейса с шагом 64 кбит/с. Более гибким решением ограничения полосы пропускания является функция per-flow, которая использует механизм списков управления доступом для просмотра определенного типа трафика и ограничения для него полосы пропускания [4].

Перед выполнением задания необходимо сбросить настройки коммутаторов к заводским настройкам по умолчанию командой

```
reset config
```

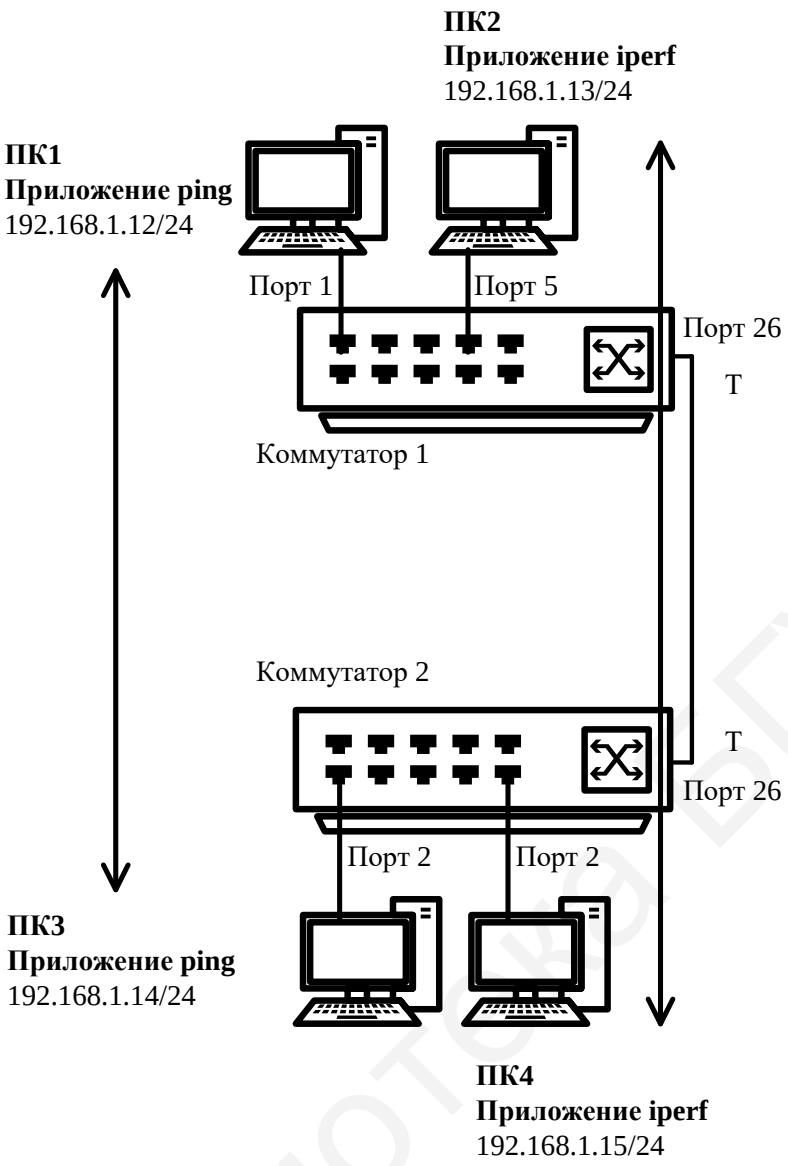


Рис. 6.6. Схема подключения для настройки приоритизации трафика

Настройка коммутатора 1

Для создания «узкого» места настройте на порте 26 функцию bandwidth_control, ограничивающую прием и передачу данных со скоростью 64 кбит/с:

```
config bandwidth_control 26 rx_rate 64 tx_rate 64
```

Настройка коммутатора 2

Для создания «узкого» места настройте на порте 26 функцию bandwidth_control, ограничивающую прием и передачу данных со скоростью 64 кбит/с:

```
config bandwidth_control 26 rx_rate 64 tx_rate 64
```

Для выполенения задания назначьте на всех ПК IP-адреса из одной подсети. Запустите продолжительный тест ping между ПК1 и ПК3, а также между ПК2 и ПК4. Собрав в течение 20–30 с статистику, запишите примерное среднее время откликов и количество потерь (запросов без ответов), если они существуют:

- между ПК1 и ПК3;

- между ПК2 и ПК4.

Запустите продолжительный тест ping между ПК1 и ПК3, а также между ПК2 и ПК4. Для создания нагрузки на линию связи между коммутаторами запустите программу iperf:

- на ПК2 с ключом «-s» (в роли сервера):

```
iperf –s –u
```

- на ПК4 с ключами «-c ip-сервера -i 1 -t 10000 -r -u -b10M -P5» (в роли клиента):

```
iperf -c 192.168.1.13 -i 1 -t 10000 -r -u -b10M -P5
```

Не останавливайте запущенные программы ping и iperf. Собранная с помощью них статистика понадобится для выполнения лабораторного задания. Собрав в течение 20–30 с статистику, запишите примерную среднюю скорость, выводимую программой iperf на ПК2 и ПК4. Посмотрите на ПК1 и ПК3, ПК2 и ПК4 информацию и запишите примерное среднее время откликов и количество потерь (запросов без ответов), если они есть: от ПК1 к ПК3, от ПК3 к ПК1, от ПК2 к ПК4 и от ПК4 к ПК2 [4].

Для этого поменяйте на порте 1, к которому подключена рабочая станция ПК1, значение приоритета по умолчанию на 7:

```
config 802.1p default_priority 1 7
```

Пользовательский приоритет и метод обработки остаются по умолчанию. Поменяйте на порте 2, к которому подключена рабочая станция ПК3, значение приоритета по умолчанию на 7:

```
config 802.1p default_priority 2 7
```

Благодаря изменению значения приоритета портов, к которым подключены компьютеры с приоритетным трафиком на 7, все кадры, передаваемые ими, получат наивысший приоритет по сравнению с кадрами, поступающими от других компьютеров на остальные неприоритизированные порты обоих коммутаторов. Посмотрите текущие настройки приоритета по умолчанию на портах коммутаторов 1 и 2:

```
show 802.1p default_priority
```

Отметьте на схеме, какой приоритет назначен по умолчанию порту 3.

Посмотрите карту привязки пользовательских приоритетов 802.1p к очередям класса обслуживания:

```
show 802.1p user_priority
```

При включении приоритизации посмотрите, как изменились условия прохождения трафика.

6.8. Содержание отчета

1. Цель лабораторной работы.

2. Схема подключения с настроенными IP-адресами, узкие места и настроенные приориты.

3. Выводы по проделанной работе.

6.9. Контрольные вопросы и задания

1. Опишите способы приоритизации трафика.

2. Назовите функции качества обслуживания в современных сетях.

3. Каковы отличия гарантированного и дифференцированного уровня обслуживания сетевого трафика?

4. Назовите особенности стандарта IEEE 802.1p.

5. Какому классу обслуживания соответствует приоритет по умолчанию = 0?

Лабораторная работа №7
ФУНКЦИИ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ И ОГРАНИЧЕНИЯ ДОСТУПА К СЕТИ

Цель работы: на коммутаторе D-Link настроить списки управления доступом, используя в качестве критериев фильтрации MAC- и IP-адреса, научиться управлять подключением узлов к портам коммутатора и изучить настройку функции Port Security на коммутаторах D-Link.

7.1. Функции обеспечения безопасности внутри локальной сети

Задачу обеспечения безопасности обычно решают с помощью межсетевых экранов, однако этого недостаточно. Поэтому современные коммутаторы имеют большое количество функций для обеспечения безопасности. Большая часть функций направлена на защиту от атак внутри сети, например, таких, как ARP Spoofing, неавторизованный доступ, подмена DHCP-сервера, атаки типа DoS и т. д.

Одной из функций повышения безопасности на сети являются *списки управления доступом (Access Control List, ACL)*, которые позволяют фильтровать потоки данных без потери производительности, так как проверка содержимого пакетов данных выполняется на аппаратном уровне (рис. 7.1). При фильтрации есть возможность ограничить доступ пользователей или устройств к различным ресурсам, ограничить типы используемых приложений, также можно использовать списки управления доступом для определения политики QoS путем классификации трафика и переопределения его приоритета [1].

Списки управления доступом по своей сути – набор условий для проверки различных параметров передаваемых пакетов. Критерии фильтрации могут быть определены на основе следующей информации, содержащейся в пакете данных: MAC- или IP-адрес, тип протокола, VLAN и др. Возможные критерии для фильтрации могут отличаться у разных моделей коммутаторов.

Списки управления доступом состоят из *профилей доступа (определяют типы критериев)* и *правил (определяют значения критериев)*. Каждый профиль может иметь множества правил. Если ни одно из правил при анализе трафика не подходит, то применяется политика по умолчанию, разрешающая прохождение всего трафика [1].

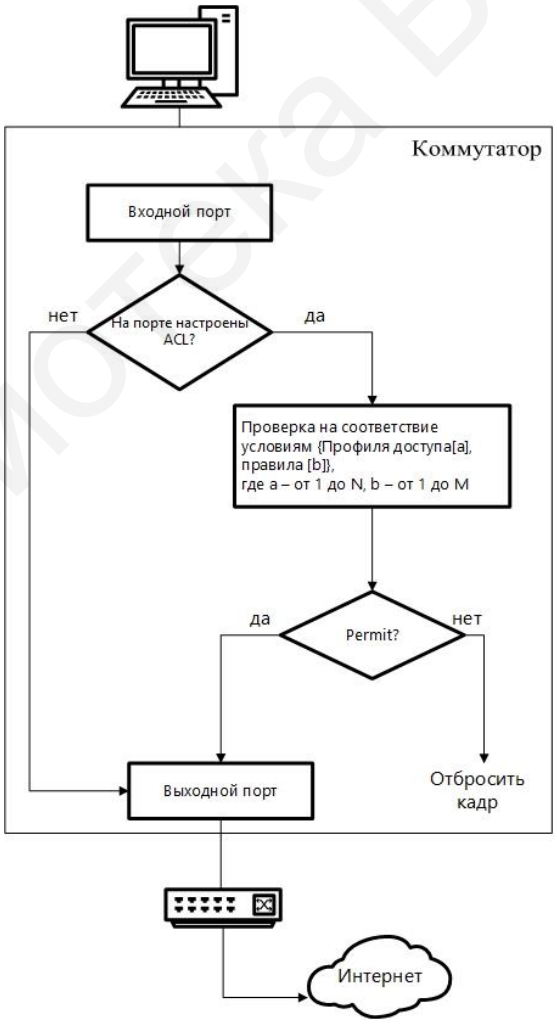


Рис. 7.1. Принцип работы ACL

Еще одной функцией ограничения доступа является функция Port Security, которая позволяет настроить любой порт коммутатора таким образом, чтобы доступ к сети через него мог осуществляться только определенными устройствами с указанными MAC-адресами. MAC-адреса могут быть изучены динамически или настроены вручную. Также эта функция позволяет ограничивать количество изучаемых портом MAC-адресов, ограничивая этим количество возможных подключений.

Функция Port Security имеет три режима работы [1]:

- Постоянный (*Permanent*) – MAC-адреса, занесенные в таблицу коммутации, никогда не устаревают.
- Удалить при истечении времени (*Delete on Timeout*) – MAC-адреса, занесенные в таблицу коммутации, устареют после истечения таймера FDB Aging Time.
- Удалить при сбросе настроек (*Delete on Reset*) – MAC-адреса, занесенные в таблицу коммутации, будут удалены после перезагрузки.

Функция IP-MAC-Port Binding (IMPB), реализованная в коммутаторах D-Link, позволяет контролировать доступ компьютеров в сеть на основе их IP- , MAC-адресов и порта подключения (рис. 7.2). Существует возможность создания «белого листа», который свяжет MAC- и IP-адреса компьютеров с портами подключения коммутатора. На основе этих записей, в случае совпадения всех составляющих, клиенты будут получать доступ к сети. В том случае, если при подключении клиента связка MAC-IP-порт будет отличаться от параметров заранее сконфигурированной записи, коммутатор заблокирует MAC-адрес соответствующего узла с занесением его в «черный лист» [1].

Функция IP-MAC-Port Binding может работать в трех режимах: DHCP Snooping mode, ARP mode (по умолчанию) и ACL mode [1]. При работе в режиме *ARP mode* коммутатор анализирует ARP-пакеты и сопоставляет параметры IP-MAC ARP-пакета с предустановленной администратором

связкой IP-MAC. Если хоть один параметр не совпадает, то MAC-адрес узла будет занесен в таблицу коммутации с отметкой Drop (Отбрасывать). Если все параметры совпадают, MAC-адрес узла будет занесен в таблицу коммутации с отметкой Allow (Разрешен) [1].

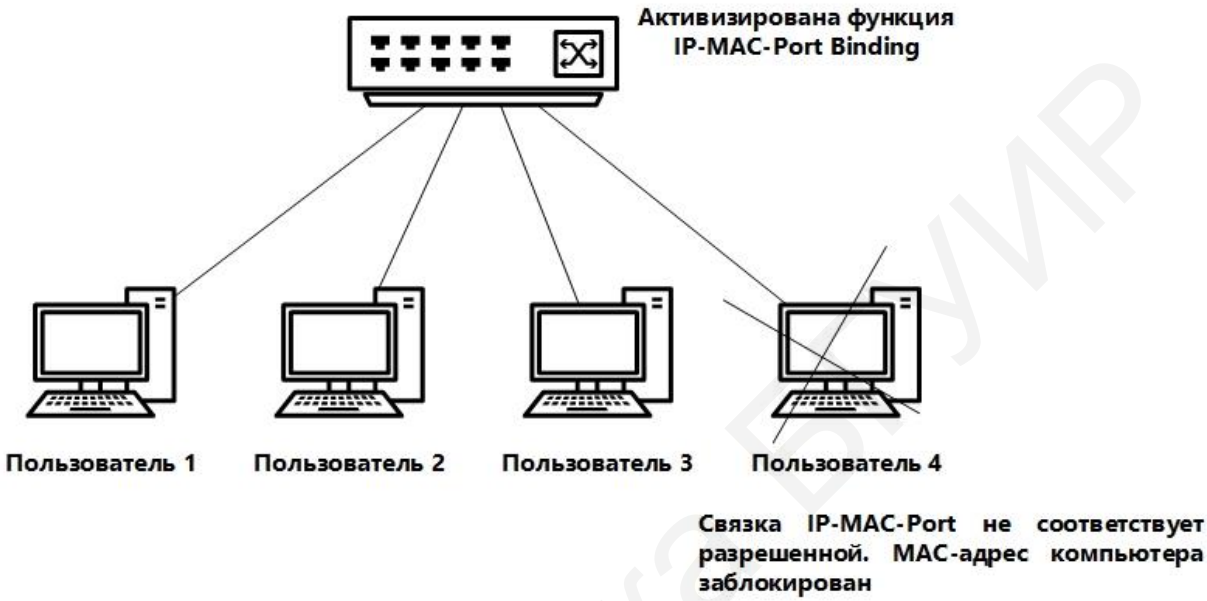


Рис. 7.2. Функция IP-MAC-Port-Binding

При функционировании в *ACL mode* коммутатор на основе предустановленного администратором «белого листа» IMPB создает правила ACL. Любой пакет, связка IPMAC которого отсутствует в «белом листе», будет блокироваться ACL. Если режим ACL отключен, правила для записей IMPB будут удалены из таблицы ACL [1].

Режим *DHCP Snooping* используется коммутатором для динамического создания записей IP-MAC на основе анализа DHCP-пакетов и привязки их к портам с включенной функцией IMPB (администратору не требуется создавать записи вручную). Таким образом, коммутатор автоматически создает «белый лист» IMPB в таблице коммутации или аппаратной таблице ACL (если режим ACL включен). При этом для обеспечения корректной работы сервер DHCP должен быть подключен к доверенному порту с

выключенной функцией IMPB. Администратор может ограничить максимальное количество создаваемых в процессе автоизучения записей IP-MAC на порт, т. е. ограничить для каждого порта с активизированной функцией IMPB количество узлов, которые могут получить IP-адрес с DHCP-сервера. При работе в режиме DHCP Snooping коммутатор не будет создавать записи IP-MAC для узлов с IP-адресом, установленным вручную.

При активизации функции IMPB на порте администратор должен указать режим его работы:

- **Strict Mode** – в этом режиме порт по умолчанию заблокирован. Прежде чем передавать пакеты, он будет отправлять их на центральном процессоре для проверки совпадения их параметров IP-MAC с записями в «белом листе». Таким образом, порт не будет передавать пакеты до тех пор, пока не убедится в их достоверности. Порт проверяет все IP- и ARP-пакеты.
- **Loose Mode** – в этом режиме порт по умолчанию открыт. Порт будет заблокирован, как только через него пройдет первый недостоверный пакет. Порт проверяет только пакеты ARP и IP Broadcast [1].

7.2. Указания по выполнению лабораторной работы

Для выполнения лабораторной работы необходимо разрешить доступ к серверу пользователям, имеющим IP-адреса с 192.168.1.1/24 по 192.168.1.63/24 (рис. 7.3). Остальным пользователем сети 192.168.1.0/24 с адресами, не входящими в разрешенный диапазон, доступ к серверу запретить [4].

Правила

Правило 1

Если IP-адрес источника равен IP-адресам из диапазона с 192.168.1.1 по 192.168.1.63 (подсеть 192.168.1.0/26), – разрешить (permit);

Правило 2

Если IP-адрес источника принадлежит сети 192.168.0.0/24, но не входит в разрешенный диапазон адресов, – запретить (deny).

Правило 3

Иначе, по умолчанию разрешить доступ всем узлам.

Перед выполнением задания необходимо сбросить настройки коммутатора к заводским настройкам командой

```
reset config
```

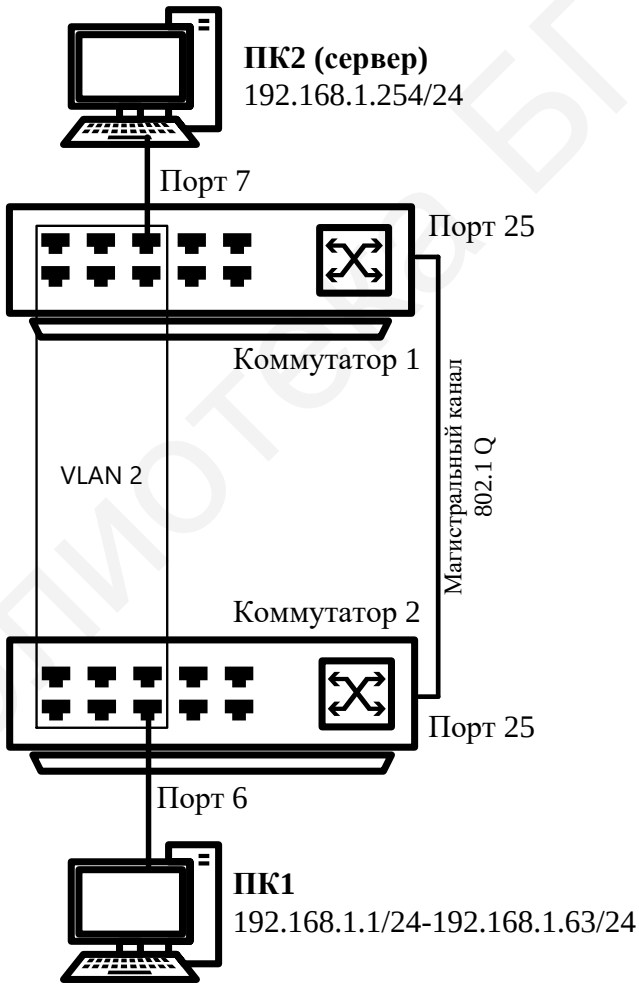


Рис. 7.3. Схема подключения с настройкой ограничения доступа пользователей к серверу по IP-адресам

Настройка коммутатора 2

Удалите порты коммутатора из VLAN по умолчанию для их использования в других VLAN:

```
config vlan default delete 1-16
```

Создайте VLAN 2 и добавьте соответствующие порты, которые необходимо настроить немаркированными. Настройте порт 25 маркированным:

```
create vlan v2 tag 2
config vlan v2 add untagged 1-16 config vlan v2 add tagged 25
```

Проверьте настройки VLAN:

```
show vlan
```

Повторите процедуру настройки для коммутатора 1.

Проверьте доступность соединения между ПК1 и ПК2 командой `ping <IP-address>`:

- от ПК1 к ПК2.

Настройка коммутатора 1

Правило 4

Создайте профиль доступа с номером 5, разрешающий доступ для подсети 192.168.1.0/26 (к узлам с 1 по 63):

```
create access_profile profile_id 5 profile_name 5 ip source_ip_mask 255.255.255.192
```

Создайте правило для профиля доступа 5:

```
config access_profile profile_id 5 add access_id 1 ip source_ip 192.168.1.0 port 25 permit
```

Примечание. Созданное правило разрешает прохождение трафика IP-подсети 192.168.1.0/26 через порт 25.

Правило 5

Создайте профиль доступа с номером 15, запрещающий остальным станциям доступ к серверу:

```
create access_profile profile_id 15 profile_name 15 ip source_ip_mask 255.255.255.0
```

Создайте правило для профиля доступа 15:

```
config access_profile profile_id 15 add access_id 1 ip source_ip 192.168.1.0 port 25 deny
```

Примечание. Созданное правило запрещает прохождение через порт 25 трафика, который принадлежит сети 192.168.1.0/24, но не входит в разрешенный диапазон.

Правило 6

Разрешите все остальное: выполняется по умолчанию.

Проверьте созданные профили:

```
show access_profile
```

Укажите в отчете, сколько профилей создано и сколько в них правил.

Подключите рабочую станцию ПК1, как показано на рис. 7.3 (адрес из диапазона 192.168.1.1–192.168.1.63/24), к коммутатору 2. Протестируйте командой ping соединение с сервером 192.168.1.254/24. Измените IP-адрес рабочей станции ПК1 (адрес из диапазона 192.168.1.64–192.168.1.254/24). Протестируйте командой ping соединение с сервером 192.168.1.254/24. Удалите профиль ACL (например, профиль 15).

```
delete access_profile profile_id 15
```

Проверьте соединение с сервером командой ping:

```
ping 192.168.1.254
```

Настройка фильтрации кадров по MAC-адресам

Для выполнения работы необходимо настроить профиль доступа так, чтобы кадры, принимаемые на любой порт коммутатора от ПК3 (с MAC-адресом 00-50-ba-22-22-22), зеркалировались (копировались) на целевой порт коммутатора, к которому подключено устройство мониторинга сети (рис. 7.4).

Рис. 7.4. Схема подключения для настройки фильтрации кадров

Правило 7

Если MAC-адрес источника равен MAC-адресу ПК3 (00-50-ba-22-22-22), следует копировать кадры на целевой порт. Перед выполнением задания необходимо сбросить настройки коммутатора к заводским настройкам командой

```
reset config
```

Замените указанные в командах MAC-адреса на реальные MAC-адреса рабочих станций.

Создайте профиль доступа 5:

```
create access_profile profile_id 5 profile_name 5 ethernet source_mac FF-FF-FF-FF-FF-FF
```

Создайте правило для профиля доступа 5, в результате выполнения которого кадры, принимаемые на любой порт коммутатора с ПК3, будут зеркалироваться на целевой порт:

```
config access_profile profile_id 5 add access_id 1 ethernet source_mac 00-50-ba-22-22-22 port all mirror
```

Проверьте созданный профиль:

```
show access_profile
```

Включите функцию зеркалирования портов глобально на коммутаторе:

```
enable mirror
```

Укажите целевой порт:

```
config mirror port 26
```

Проверьте настройки функции:

```
show mirror
```

Подключите рабочие станции ПК2 и ПК3, как показано на рис. 7.5. Выполните тестирование соединения между ПК2 и ПК3 с помощью команды `ping <IP address>`:

- от ПК2 к ПК3;
- от ПК3 к ПК2.

Запустите на рабочей станции ПК1 анализатор протоколов Wireshark (настройка программы описана в лабораторной работе №1).

Захватите и проанализируйте пакеты с помощью анализатора протоколов.

Подключите рабочую станцию ПК3 к порту 10 коммутатора.

Выполните тестирование соединения между ПК2 и ПК3 и наоборот командой ping.

Захватите и проанализируйте пакеты с помощью анализатора протоколов.

Удалите все профили ACL:

```
delete access_profile all
```

Отключите функцию зеркалирования портов:

```
disable mirror
```

Управление количеством подключаемых к портам коммутатора пользователей осуществляется путем ограничения максимального количества изучаемых MAC-адресов.

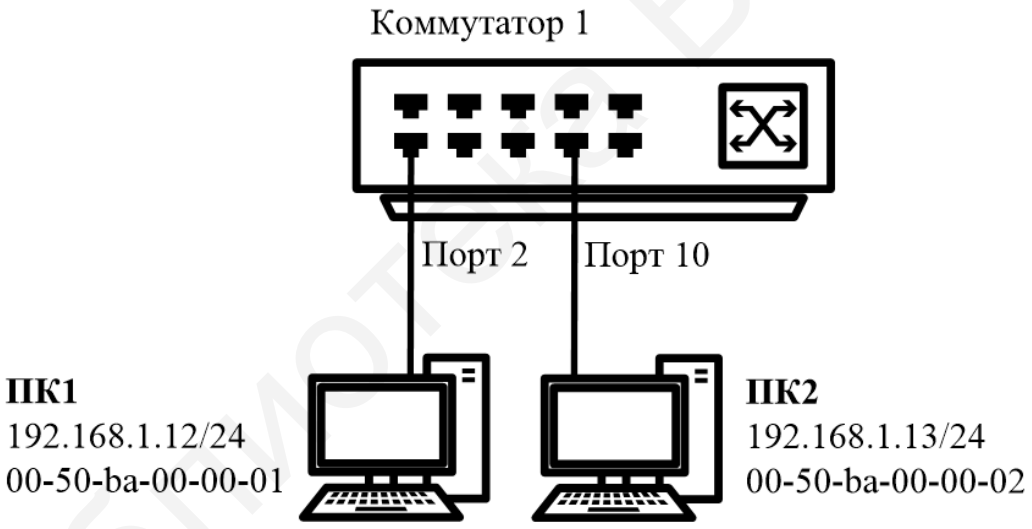


Рис. 7.5. Схема подключения для настройки Port Security

Сбросьте настройки коммутатора к заводским настройкам по умолчанию командой

```
reset config
```

Проверьте информацию о настройках Port Security:

```
show port_security
```

Установите максимальное количество изучаемых каждым портом MAC-адресов равным единице, и включите функцию на всех портах:

```
config port_security ports all admin_state enable max_learning_addr 1
```

Подключите ПК1 и ПК2 к портам 2 и 10 коммутатора соответственно.

Посмотрите MAC-адреса, которые стали известны портам 2 и 10:

```
show fdb port 2 show fdb port 10
```

Проверьте, соответствуют ли зарегистрированные адреса адресам рабочих станций.

Проверьте информацию о настройках Port Security на портах коммутатора:

```
show port_security ports 1-24
```

Включите запись в журнал работы коммутатора MAC-адресов, подключающихся к порту станций, и отправку сообщений SNMP Trap:

```
enable port_security trap_Log
```

Выполните тестирование доступности узлов командой ping от ПК1 к ПК2 и наоборот.

Подключите ПК1 к порту 10, а ПК2 к порту 1.

Повторите тестирование соединения между рабочими станциями командой ping.

Проверьте информацию в журнале работы коммутатора:

```
show log
```

Сохраните конфигурацию и перезагрузите коммутатор:

```
save reboot
```

Выполните тестирование соединения между рабочими станциями командой ping. Сохраняется ли информация о привязке MAC-порта?

Настройте на порте 2 работу функции Port Security в режиме Permanent и максимальное количество изучаемых адресов, равное единице:

```
config port_security ports 2 admin_state enable
max_learning_addr
1 lock_address_mode permanent
```

Сохраните конфигурацию и перезагрузите коммутатор:

```
save reboot
```

Проверьте информацию о настройках Port Security на портах коммутатора:

```
show port_security ports 1-24
```

Проверить, сохраняется ли информация о привязке MAC-порта.

Очистите информацию о привязке MAC-порта на порте 2:

```
clear port_security_entry port 2
```

Отключите работу функции Port Security на порте 2 и приведите настройки в исходное (по умолчанию) состояние:

```
config port_security ports 2 admin_state disable
max_learning_addr
1 lock_address_mode deleteonreset
```

Посмотрите время таймера блокирования (он соответствует времени жизни MAC-адреса в таблице коммутации):

```
show fdb aging_time
```

Изменить время действия таймера можно с помощью настройки времени жизни MAC-адреса в таблице коммутации (время указано в секундах):

```
config fdb aging_time 20
```

Измените режим работы функции Port Security на Delete on Timeout:

```
config port_security ports 2 admin_state enable
max_learning_addr
1 lock_address_mode deleteontimeout
```

Проверьте MAC-адреса, которые стали известны порту 2:

```
show fdb port 2
```

Проверьте информацию о настройках Port Security на портах коммутатора:

```
show port_security ports 1-24
```

Выполните тестирование соединения между ПК1 и ПК2 командой ping. Проверить, сохраняется ли информация о привязке MAC-порта.

Отключите работу функции Port Security на портах:

```
config port_security ports 1-24 admin_state disable
```

Отключите функцию записи в log-файл и отправки SNMP Trap:

```
disable port_security trap_Log
```

После выполнения обучения имеется возможность отключить функцию динамического изучения MAC-адресов, тогда в таблице коммутации сохранятся изученные адреса. Таким образом, текущая конфигурация сети будет сохранена, и дальнейшее подключение новых устройств без ведома администратора будет невозможно. Новые устройства можно добавить путем создания статических записей в таблице коммутации.

Настройка защиты от подключения к портам, основанная на статической таблице MAC-адресов

Отключите рабочие станции от коммутатора.

Сбросьте настройки коммутатора к заводским настройкам командой

```
reset system
```

Активизируйте функцию Port Security на всех портах и запретите изучение MAC-адресов, установив параметр max_learning_addr равным нулю (команда вводится в одну строку):

```
config port_security ports 1-24 admin_state enable max_learning_addr 0
```

Проверьте состояние портов:

```
show ports
```

Проверьте соединение между ПК1 и ПК2 командой ping. Проверьте состояние таблицы коммутации:

```
show fdb
```

В таблице коммутации вручную создайте статические записи для MAC-адресов рабочих станций, подключенных к портам 2 и 10.

Внимание. Замените указанные в командах MAC-адреса на реальные адреса рабочих станций, подключаемых к коммутатору.

```
create fdb default 00-50-ba-00-00-01 port 2 create fdb default 00-50-ba-00-00-02 port 10
```

Проверьте созданные статические записи в таблице коммутации:

```
show fdb
```

Проверьте информацию о настройках Port Security на портах коммутатора:

```
show port_security ports 1-24
```

Проверьте соединение между ПК1 и ПК2 командой ping.

Подключите ПК1 к порту 8, а ПК2 к порту 2.

Повторите тестирование командой ping.

Удалите ранее созданную статическую запись из таблицы MAC-адресов на порте 2:

```
delete fdb default 00-50-ba-00-00-02 port 2
```

7.3. Содержание отчета

1. Цель лабораторной работы.

2. Схема подключения лабораторной работы с настроенными протоколами.

3. Выводы по проделанной работе.

7.4. Контрольные вопросы и задания

1. Назовите средства фильтрации потоков данных, используемые в компьютерных сетях.

2. Опишите алгоритм работы списков управления доступом (Access Control List, ACL).

3. Объясните понятие профилей доступа и правила в списках управления доступом.

4. Опишите работу функции Port Security.

ЗАКЛЮЧЕНИЕ

Инфокоммуникационная сеть позволяет соединять оборудование абонентов в одну систему, внутри которой может происходить обмен данными. Главным элементом таких сетей является оборудование абонентов, которое строится с использованием компьютеров и других устройств – принтеров, телефонов и т. д. В сети также присутствует общее сетевое оборудование – это различные маршрутизаторы, роутеры, модемы, маршрутизаторы, серверы. Это оборудование соединяется между собой с помощью так называемой сетевой среды – оптических и металлических кабелей и беспроводных технологий доступа. Передача данных в сетях различного уровня рассматривается на базе эталонной базовой модели, разработанной Международной организацией по стандартам взаимодействия открытых сетей. Эволюция сетей передачи данных неразрывно связана с использованием главным образом технологии Ethernet, которая по сегодняшний день остается одной из самых распространенных технологий.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Построение коммутируемых компьютерных сетей : учеб. пособие / Е. В. Смирнова [и др.]. – М. : Нац. открытый ун-т «ИНТУИТ», 2012.

2. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 2-е изд. – СПб. : Питер, 2000.

3. Принципы коммутации сегментов и узлов локальных сетей, использующих традиционные технологии. D-Link [Электронный ресурс]. – Режим доступа : http://citforum.ru/nets/lsok/glava_4.shtml.

4. Цикл методических материалов D-Link [Электронный ресурс]. – Режим доступа : http://learn.dlink.ru/.

5. Гольдштейн, Б. С. Сети связи : учебник для магистрантов вузов / Б. С. Гольдштейн, Н. А. Соколов, Г. Г. Яновский. – СПб. : БХВ-Петербург, 2011.

6. Курицын, С. А. Телекоммуникационные технологии и системы : учеб. пособие / С. А. Курицын. – М. : Академия, 2008.

7. Таненбаум, Э. Компьютерные сети / Э. Таненбаум, Д. Уэзеролл. – 5-е изд. – СПб. : Питер, 2020.

8. Платунов, С. М. Технические средства коммутации Zyxel : учеб. пособие / С. М. Платунов [Электронный ресурс]. – Режим доступа : http://window.edu.ru/catalog/pdf2txt/574/78574/59403?p_page=2.

9. Коммутация (вычислительные сети) [Электронный ресурс]. – Режим доступа : https://ru.bmstu.wiki/index.php?title=%D0%9A%D0%BE%D0%BC%D0%BC%D1%83%D1%82%D0%B0%D1%86%D0%B8%D1%8F_(%D0%B2%D1%8B%D1%87%D0%B8%D1%81%D0%BB%D0%B8%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5_%D1%81%D0%B5%D1%82%D0%B8)&mobileaction=toggle_view_mobile.

10. Построение коммутируемых компьютерных сетей [Электронный ресурс]. – Режим доступа : https://www.intuit.ru/studies/courses/3591/833/lecture/14251?page=3.

11. Типы интерфейсов коммутаторов [Электронный ресурс]. – Режим доступа : https://infopedia.su/12x83aa.html.

12. Олифер, В. Г. Компьютерные сети. Принципы, технологии, протоколы / В. Г. Олифер, Н. А. Олифер. – 6-е изд. – СПб. : Питер, 2020.

13. Типы VLAN [Электронный ресурс]. – Режим доступа : https://www.intuit.ru/studies/courses/3591/833/lecture/ 14258?page=2.

14. Семёнов, Ю. А. Алгоритмы телекоммуникационных сетей : учеб. пособие. В 3 ч. Ч. 2 : Протоколы и алгоритмы маршрутизации в Internet / Ю. А. Семёнов. – М. : Интернет-Университет Информационных Технологий : БИНОМ. Лаборатория знаний, 2014.

15. Агрегирование каналов [Электронный ресурс]. – Режим доступа : https://bogachev.biz/2015/06/28/agregirovanie-kanalov-cisco/.

Св. план 2021, поз. 35

Учебное издание

Ковшик Виктория Анатольевна
Мищенко Валерий Николаевич
Рабцевич Виолетта Викторовна

ТЕХНОЛОГИИ ПЕРЕДАЧИ ДАННЫХ В ИНФОКОММУНИКАЦИОННЫХ СИСТЕМАХ

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *М. А. Зайцева*
Корректор *Е. Н. Батурчик*
Компьютерная правка, оригинал-макет *В. М. Задоля*

Подписано в печать 17.05.2021. Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».
Отпечатано на ризографе. Усл. печ. л. 8,72. Уч.-изд. л. 8,0. Тираж 50 экз. Заказ 245.

Издатель и полиграфическое исполнение: учреждение образования
«Белорусский государственный университет информатики и радиоэлектроники».
Свидетельство о государственной регистрации издателя, изготовителя,
распространителя печатных изданий №1/238 от 24.03.2014,
№2/113 от 07.04.2014, №3/615 от 07.04.2014.
Ул. П. Бровки, 6, 220013, г. Минск