Технологии построения коммутируемых сетей Ethernet с маршрутизацией

Базовый курс D-Link

Версия 3

Москва, 2013

Обозначения, используемые в книге

В тексте книги используются следующие пиктограммы для обозначения сетевых устройств различных типов:

Синтаксис команд

Следующие символы используются для описания ввода команд, ожидаемых значений и аргументов при настройке коммутатора через интерфейс командной строки (CLI).

Таблица 1

<угловые скобки >	
Назначение	Содержат ожидаемую переменную или значение, которое должно быть указано.
[квадратные скобки]	
Назначение	Содержат требуемое значение или набор требуемых аргументов. Может быть указано одно значение или аргумент.
| вертикальная черта	
Назначение	Отделяет два или более взаимно исключающих пунктов из списка, один из которых должен быть введен/указан.
{ фигурные скобки }	
Назначение	Содержит необязательное значение или набор необязательных аргументов.

1. Основы коммутации

1.1 Эволюция локальных сетей

Эволюция локальных сетей неразрывно связана с историей развития технологии Ethernet, которая по сей день остается самой распространенной технологией локальных сетей.

Первоначально технология локальных сетей рассматривалась как времясберегающая и экономичная технология, обеспечивающая совместное использование данных, дискового пространства и дорогостоящих периферийных устройств. Снижение стоимости персональных компьютеров и периферии привело к их широкому распространению в бизнесе, и количество сетевых пользователей резко возросло. Одновременно изменились архитектура приложений (клиент/сервер) и их требования к вычислительным ресурсам, а также архитектура вычислений (распределенные вычисления). Стал популярным *downsizing* (разукрупнение) – перенос информационных систем и приложений с мэйнфреймов на сетевые платформы. Все это привело к смещению акцентов в использовании сетей: они стали обязательным инструментом в бизнесе, обеспечив наиболее эффективную обработку информации.

В первых сетях Ethernet (10Base-2 и 10Base-5) использовалась шинная топология, когда каждый компьютер соединялся с другими устройствами с помощью единого коаксиального кабеля, используемого в качестве среды передачи данных. Сетевая среда была разделяемой и устройства, прежде чем начать передавать пакеты данных, должны были убедиться, что она свободна. Несмотря на то, что такие сети были простыми в установке, они обладали существенными недостатками, заключающимися в ограничениях по размеру, функциональности и расширяемости, недостаточной надежности, а также неспособностью справляться с экспоненциальным увеличением сетевого трафика. Для повышения эффективности работы локальных сетей требовались новые решения.

Следующим шагом стала разработка стандарта 10Base-T с топологией типа «звезда», в которой каждый узел подключался отдельным кабелем к центральному устройству – *концентратору* (*hub*). Концентратор работал на физическом уровне модели OSI и повторял сигналы, поступавшие с одного из его портов на все остальные активные порты, предварительно восстанавливая их. Использование концентраторов позволило повысить надежность сети, т.к. обрыв какого-нибудь кабеля не влек за собой сбой в работе всей сети. Однако, несмотря на то, что использование концентраторов в сети упростило задачи ее управления и сопровождения, среда передачи оставалась разделяемой (все устройства находились в одном домене коллизий). Помимо этого общее количество концентраторов и соединяемых ими сегментов сети было ограничено из-за временных задержек и других причин.

Задача *сегментации сети*, т.е. разделения пользователей на группы (сегменты) в соответствии с их физическим размещением с целью уменьшения количества клиентов соперничающих за полосу пропускания была решена с помощью устройства, называемого *мостом* (*bridge*). Мост был разработан компанией Digital Equipment Corporation (DEC) в начале 1980-х годов и представлял собой устройство канального уровня модели OSI (обычно двухпортовое), предназначенное для объединения сегментов сети. В отличие от концентратора, мост не просто пересылал пакеты данных из одного сегмента в другой, а анализировал и передавал их только в том случае, если такая передача действительно была необходима, то есть адрес рабочей станции назначения принадлежал другому сегменту. Таким образом, мост изолировал трафик одного сегмента от трафика другого, уменьшая домен коллизий и повышая общую производительность сети.

Однако мосты были эффективны лишь до тех пор, пока количество рабочих станций в сегменте оставалось относительно невелико. Как только оно увеличивалось, в сетях

возникала перегрузка (переполнение приемных буферов сетевых устройств), которая приводила к потере пакетов.

Увеличение количества устройств, объединяемых в сети, повышение мощности процессоров рабочих станций, появление мультимедийных приложений и приложений клиент-сервер требовали большей полосы пропускания. В ответ на эти растущие требования фирмой Kalpana в 1990 г. на рынок был выпущен первый *коммутатор* (*switch*), получивший название EtherSwitch.

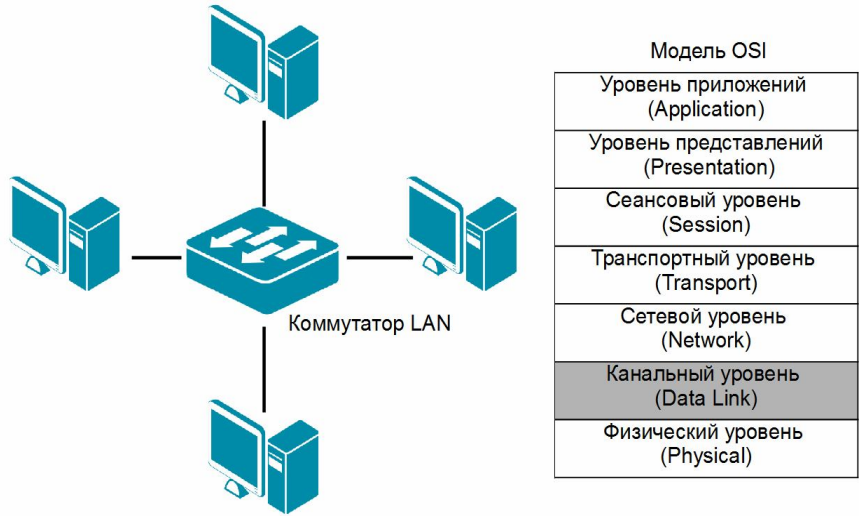


Рис. 1.1. Коммутатор локальной сети

Коммутатор представлял собой многопортовый мост и также функционировал на канальном уровне модели OSI. Основное отличие коммутатора от моста заключалось в том, что он мог устанавливать *одновременно несколько соединений* между разными парами портов. При передаче кадра через коммутатор в нем создавался отдельный виртуальный (либо реальный, в зависимости от архитектуры) канал, по которому данные пересылались «напрямую» от порта-источника к порту-получателю с максимально возможной для используемой технологии скоростью. Такой принцип работы получил название *микросегментация*. Благодаря микросегментации, коммутаторы получили возможность функционировать в *режиме полного дуплекса* (full duplex), что позволяло каждой рабочей станции одновременно передавать и принимать данные, используя всю полосу пропускания в обоих направлениях. Рабочей станции не приходилось конкурировать за полосу пропускания с другими устройствами, в результате чего не происходили коллизии, и повышалась производительность сети.

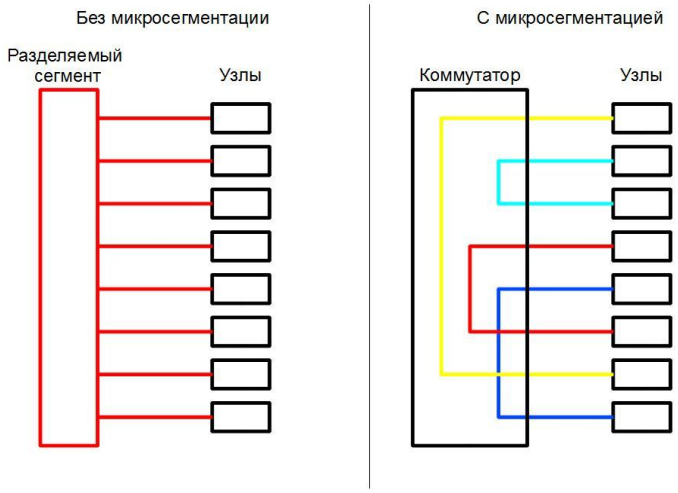


Рис. 1.2. Микросегментация

В настоящее время коммутаторы являются основным строительным блоком для создания локальных сетей. Современные коммутаторы Ethernet превратились в интеллектуальные устройства со специализированными процессорами для обработки и перенаправления пакетов на высоких скоростях, и реализации таких функций, как организация резервирования и повышения отказоустойчивости сети, агрегирование каналов, создание виртуальных локальных сетей (VLAN), маршрутизация, управление качеством обслуживания (Quality of Service, QoS), обеспечение безопасности и многих других. Также усовершенствовались функции управления коммутаторов, благодаря чему системные администраторы получили удобные средства настройки сетевых параметров, мониторинга и анализа трафика.

С появлением стандарта IEEE 802.3af-2003 PoE, описывающего технологию передачи питания по Ethernet (Power over Ethernet, PoE), разработчики начали выпускать коммутаторы с поддержкой данной технологии, что позволило использовать их в качестве питающих устройств для IP-телефонов, Интернет-камер, беспроводных точек доступа и другого оборудования.

С ростом популярности технологий беспроводного доступа в корпоративных сетях производители оборудования выпустили на рынок унифицированные коммутаторы с поддержкой технологии PoE для питания подключаемых к их портам точек беспроводного доступа и централизованного управления как проводной, так и беспроводной сетью.

Повышение потребностей заказчиков и тенденции рынка стимулируют разработчиков коммутаторов более или менее регулярно расширять аппаратные и функциональные возможности производимых устройств, позволяющие предоставлять в локальных сетях новые услуги, повышать их надежность, управляемость и защищенность.

1.2 Функционирование коммутаторов локальной сети

Коммутаторы локальных сетей обрабатывают кадры на основе алгоритма *прозрачного моста* (*transparent bridge*), который определен стандартом IEEE 802.1D. Процесс работы алгоритма прозрачного моста начинается с построения *таблицы коммутации* (*Forwarding DataBase, FDB*).

Изначально таблица коммутации пуста. При включении питания, одновременно с передачей данных, коммутатор начинает изучать расположение подключенных к нему сетевых устройств, путем анализа MAC-адресов источников получаемых кадров. Например, если на порт 1 коммутатора, показанного на рис. 1.3, поступает кадр от узла A, то он создает

в таблице коммутации запись, ассоциирующую MAC-адрес узла A с номером входного порта. Записи в таблице коммутации создаются *динамически*. Это означает, что, как только коммутатором будет прочитан новый MAC-адрес, то он сразу будет занесен в таблицу коммутации. Дополнительно к MAC-адресу и ассоциированному с ним порту в таблицу коммутации для каждой записи заносится *время старения* (*aging time*). Время старения позволяет коммутатору автоматически реагировать на перемещение, добавление или удаление сетевых устройств. Каждый раз, когда идет обращение по какому-либо MAC-адресу, соответствующая запись получает новое время старения. Записи, по которым не обращались долгое время, из таблицы удаляются. Это позволяет хранить в таблице коммутации только актуальные MAC-адреса, что уменьшает время поиска соответствующей записи в ней и гарантирует, что она не будет использовать слишком много системной памяти.

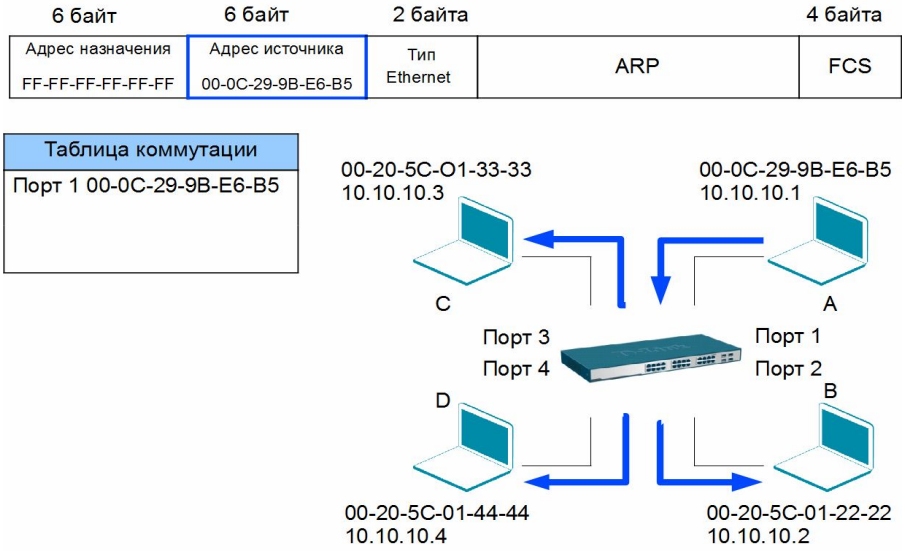


Рис. 1.3. Построение таблицы коммутации

Помимо динамического создания записей в таблице коммутации в процессе самообучения коммутатора, существует возможность создания статических записей таблицы коммутации вручную. Статическим записям, в отличие от динамических, не присваивается время старения, поэтому время их жизни не ограничено.

Статическую таблицу коммутации удобно использовать с целью повышения сетевой безопасности, когда необходимо гарантировать, что только устройства с определенными MAC-адресами могут подключаться к сети. В этом случае необходимо отключить автоизучение MAC-адресов на портах коммутатора.

Внимание: как правило, размер статической таблицы коммутации меньше размера динамической таблицы коммутации. Размеры обоих таблиц также зависят от модели коммутатора. Обычно производители указывают размеры таблиц коммутации в спецификациях на устройства.

Как только в таблице коммутации появляется хотя бы одна запись, коммутатор начинает использовать ее для пересылки кадров. Рассмотрим пример, показанный на рис. 1.4, описывающий процесс пересылки кадров между портами коммутатора.

Когда коммутатор получает кадр, отправленный компьютером A компьютеру B, он извлекает из него MAC-адрес приемника и ищет этот MAC-адрес в своей таблице коммутации. Как только в таблице коммутации будет найдена запись, ассоциирующая MAC-адрес приемника (компьютера B) с одним из портов коммутатора, за исключением порта-источника, кадр будет передан через соответствующий выходной порт (в приведенном примере – порт 2). Этот процесс называется *продвижением (forwarding)* кадра.

Если бы оказалось, что выходной порт и порт-источник совпадают, то передаваемый кадр был бы отброшен коммутатором. Этот процесс называется *фильтрацией (filtering)*.

В том случае, если MAC-адрес приемника в поступившем кадре неизвестен (в таблице коммутации отсутствует соответствующая запись), коммутатор создает множество копий этого кадра и передает их через все свои порты, за исключением того, на который он поступил. Этот процесс называется *лавинной передачей (flooding)*. Несмотря на то, что процесс лавинной передачи занимает полосу пропускания, он позволяет коммутатору избежать потери кадров, когда MAC-адрес приемника неизвестен, и осуществлять процесс самообучения.

Помимо лавинной передачи одноадресных кадров, коммутаторы также выполняют лавинную передачу многоадресных и широковещательных кадров, которые генерируются сетевыми мультимедийными приложениями.

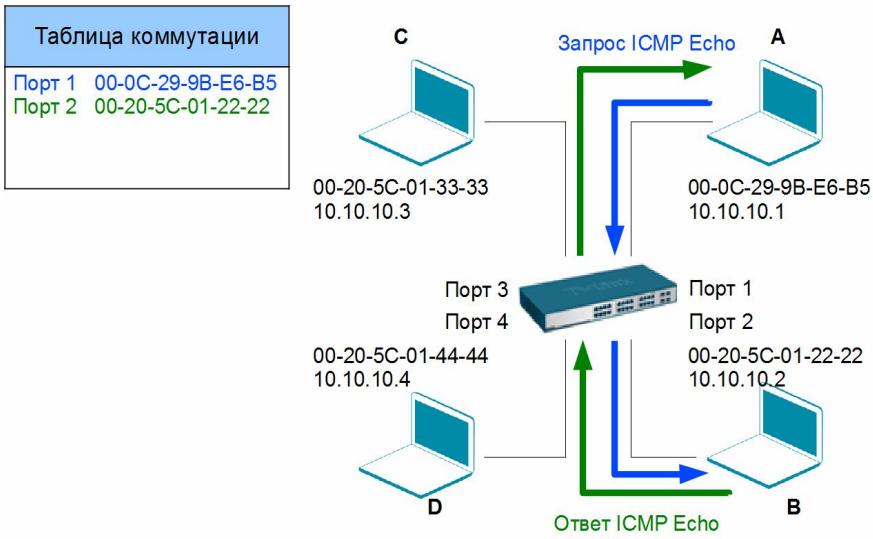


Рис. 1.4. Передача кадра с порта на порт коммутатора

1.3 Методы коммутации

Первым шагом, который выполняет коммутатор, прежде чем принять решение о передаче кадра, является его получение и анализ содержимого. В современных коммутаторах используются следующие методы коммутации, определяющие поведение устройства при получении кадра:

•коммутация с промежуточным хранением (store-and-forward);

•коммутация без буферизации (cut-through).

Оба метода коммутации принимают решение о продвижении кадров на основе MAC-адреса получателя, но отличаются последовательностью действий, которые коммутатор выполнит, прежде чем передать или отбросить поступивший на его порт кадр.

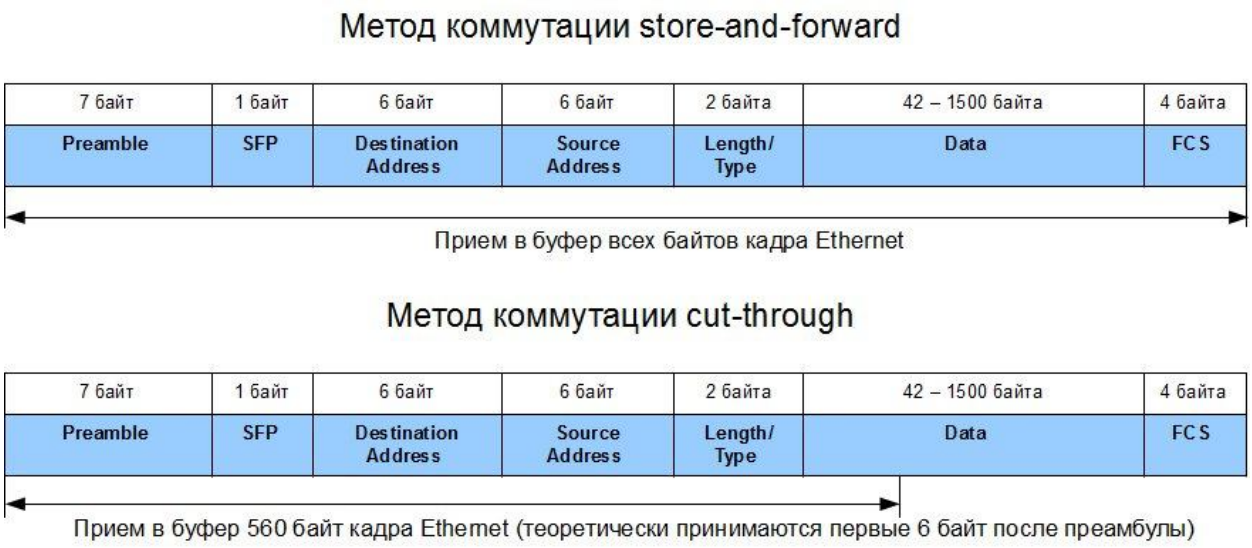


Рис. 1.5. Методы коммутации

Метод коммутации с промежуточным хранением (store-and-forward) исторически появился первым. Он характеризуется тем, что коммутатор, прежде чем передать кадр, полностью копирует его в буфер и производит проверку на наличие ошибок. Если кадр содержит ошибки (не совпадает контрольная сумма, или кадр меньше 64 байт или больше 1518 байт), то он отбрасывается. Если кадр не содержит ошибок, то коммутатор находит MAC-адрес приемника в своей таблице коммутации и определяет выходной порт. Затем, если не определены никакие фильтры, коммутатор передает кадр через соответствующий порт устройству назначения.

Несмотря на то, что этот способ передачи связан с задержками (чем больше размер кадра, тем больше времени требуется на его прием и проверку на наличие ошибок), он обладает двумя существенными преимуществами:

- коммутатор может быть оснащен портами, поддерживающими разные технологии и скорости передачи, например, 10/100 Мбит/с, 1000 Мбит/с и 10 Гбит/с;
- коммутатор может проверять целостность кадра, благодаря чему поврежденные кадры не будут передаваться в соответствующие сегменты.

В большинстве коммутаторах D-Link реализован этот метод коммутации. Благодаря использованию в устройствах высокопроизводительных процессоров и контроллеров ASIC (Application-Specific Integrated Circuit), задержка, вносимая коммутацией store-and-forward при передаче кадров, оказывается незначительной.

Коммутация без буферизации (cut-through) была реализована в первом коммутаторе Ethernet, разработанном фирмой Kalpana в 1990 г. При работе в этом режиме теоретически коммутатор копирует в буфер только MAC-адрес назначения (первые 6 байт после преамбулы) и сразу начинает передавать кадр, не дожидаясь его полного приема. Однако современные коммутаторы не всегда реализуют коммутацию без буферизации в классическом варианте. В зависимости от реализации коммутатор дожидается приема в буфер определенного количества байтов кадра и, если на порте не определены никакие фильтры, принимает решение о его передаче. Так как при работе в режиме cut-through коммутатор не дожидается приема всего кадра, то он не выполняет проверку кадров на наличие ошибок. Проверка кадра на наличие ошибок возлагается на принимающий узел. Однако, современная сетевая инфраструктура, включающая оборудование и кабельную систему позволяет свести вероятность возникновения ошибочных кадров к минимуму.

Основным преимуществом коммутация без буферизации по сравнению с коммутацией с промежуточным хранением является уменьшение времени передачи кадров большого размера. Например, если приложение использует Jumbo-фреймы (кадры размером

9200 байт), то коммутатор, работающий в режиме cut-through, будет передавать данные на несколько микро или миллисекунд (в зависимости от скорости портов коммутатора) быстрее коммутатора, использующего режим store-and-forward.

Помимо этого, коммутаторы с поддержкой режима cut-through хорошо подходят для использования в сетях, например в центрах обработки данных, с приложениями критичными к задержкам.

Однако в некоторых случаях, метод cut-through теряет свои преимущества в скорости передачи. Это может произойти при перегрузке сети, использовании функций фильтрации, требующих обработки на ЦПУ, или когда порты коммутатора поддерживают разную скорость (если коммутационная матрица плохо спроектирована).

Коммутаторы D-Link серии DXS-3600-xx обеспечивают гибкость в выборе метода коммутации, т.к. поддерживают selectable store-and-forward/cut-through mode. По умолчанию в коммутаторах этой серии используется режим store-and-forward, поэтому для получения преимуществ от использования режима cut-through, администратор сети должен сначала его активизировать. Коммутатор будет копировать в буфер и изучать первые 560 байт кадра. Если размер кадра окажется больше 560 байт, коммутатор автоматически переключится в режим cut-through и начнет процесс продвижения кадра, не дожидаясь его полного приема. Соответственно для кадров, чей размер меньше или равен 560 байт будет использоваться режим коммутации store-and-forward.

1.4 Конструктивное исполнение коммутаторов

В зависимости от конструктивного исполнения (габаритных размеров), можно выделить три группы коммутаторов:

- настольные коммутаторы (Desktop switch);
- автономные коммутаторы, монтируемые в телекоммуникационную стойку (Rack mounted switch);
- коммутаторы на основе шасси (Chassis switch).

Как следует из названия, *настольные коммутаторы* предназначены для размещения на столах, иногда они могут оснащаться, входящими в комплект поставки, скобами для крепления на стену. Обычно такие коммутаторы обладают корпусом обтекаемой формы с относительно небольшим количеством фиксированных портов (у коммутаторов D-Link количество портов варьируется от 5 до 16), внешним или внутренним блоком питания и небольшими ножками (обычно резиновыми) для обеспечения вентиляции нижней поверхности устройства. Чаще всего коммутаторы настольного форм-фактора используются в сетях класса *SOHO* (*Small Office, Home Office*), где не требуется высокая производительность и расширенные сетевые функции. В качестве примера коммутатора в настольном исполнении можно привести коммутатор D-Link модели DES-1008A.

Автономные коммутаторы в стоечном исполнении высотой 1U обладают корпусом для монтажа в 19” стойку, встроенным блоком питания и фиксированным количеством портов (у коммутаторов D-Link количество портов может достигать 52-х штук). По сравнению с настольными коммутаторами, коммутаторы, монтируемые в стойку, обеспечивают более высокую производительность и надежность, а также предлагают широкий набор сетевых функций и интерфейсов. Как правило, такие коммутаторы используются на уровнях доступа и распределения сетей малых и средних предприятий (*Small to Medium Business, SMB*), корпоративных сетей и сетей провайдеров услуг (*Internet Service Provider, ISP*).

Среди коммутаторов в стоечном исполнении с фиксированным количеством портов можно выделить отдельную группу устройств – *стековые коммутаторы*. Эти устройства представляют собой коммутаторы, которые могут работать как автономно, так как выполнены в отдельном корпусе, так и совместно, благодаря наличию специальных интерфейсов, позволяющих объединять коммутаторы в одно логическое устройство с целью

увеличения количества портов, удобства управления и мониторинга. Говорится, что в этом случае отдельные коммутаторы образуют *стек*.

Коммутаторы на основе шасси содержат слоты, которые могут быть использованы для установки интерфейсных модулей расширения, резервных источников питания и процессорных модулей. Модульное решение обеспечивает гибкость применения, высокую плотность портов и возможность резервирования критичных для функционирования коммутатора компонентов. Модули такого коммутатора поддерживают технологию «*hot swap*» (горячая замена), то есть допускают замену на ходу, без выключения питания коммутатора. Коммутаторы на основе шасси предназначены для применения на магистрали сети крупных корпоративных сетей, городских сетей или сетей операторов связи.

1.5 Физическое стекирование коммутаторов

Под **физическим стекированием** понимается объединение нескольких коммутаторов в одно логическое устройство с целью увеличения количества портов, удобства управления и мониторинга. Объединенные в стек коммутаторы имеют общие таблицы коммутации и маршрутизации (для коммутаторов 3 уровня).

В коммутаторах D-Link используются 2 топологии физического стекирования: «*кольцо*» (ring) и «*цепочка*» (chain).

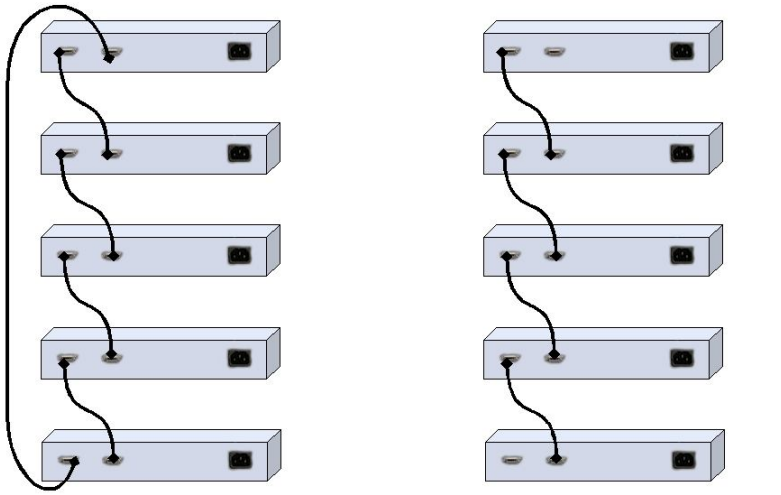


Рис. 1.6. Топологии стекирования «кольцо» и «цепочка»

Стек типа «кольцо» (*кольцевая топология*) строится по следующей схеме: каждое устройство в стеке подключается к вышележащему и нижележащему, при этом самый нижний и самый верхний коммутатор в стеке также соединяются. При передаче данных пакет последовательно передается от одного устройства стека к другому до тех пор, пока не достигнет порта назначения. Система автоматически определяет оптимальный путь передачи трафика, что позволяет достичь полного использования полосы пропускания. Преимуществом топологии «кольцо» является то, что при выходе одного устройства из строя или обрыве связи, остальные устройства стека будут продолжать функционировать в обычном режиме.

В *стеке типа «цепочка»* (*линейная топология*) каждое устройство соединено с вышележащим и нижележащим. Самый верхний и самый нижний коммутаторы не соединяются.

Физическое стекирование по линейной и кольцевой топологии реализовано в семи сериях коммутаторов D-Link. Коммутаторы серии DGS-3120-xx позволяют объединить в стек до 6 устройств, коммутаторы серии DGS-3610-xx – до 8 устройств, а коммутаторы серий

DGS-3420-xx , DGS-36xx, DGS-3620-xx – до 12 устройств, используя интерфейсы 10 Gigabit Ethernet (10GE). Коммутаторы серии DGS-31xx объединяются в стек через интерфейсы HDMI. Максимальное количество коммутаторов в стеке равно 6. Коммутаторы серии DES-3528/3552 поддерживают физическое стекирование через интерфейсы Gigabit Ethernet и позволяют объединить в стек до 8 устройств.

Все устройства стека управляются через один IP-адрес. Передача данных между ними ведется в полнодуплексном режиме.

1.6 Типы интерфейсов коммутаторов

В зависимости от выполняемых задач коммутаторы могут быть оборудованы различным количеством и типом портов. В таблице 2 приведены типы наиболее часто используемых интерфейсов и их основные характеристики в соответствии со стандартом IEEE 802.3-2008.

Наиболее распространенными интерфейсами, реализуемыми в коммутаторах, являются фиксированные интерфейсы с разъемом RJ-45 на основе витой пары, поддерживающие технологию Fast или Gigabit Ethernet, автосогласование скоростей, полудуплексного или дуплексного режима работы и автоматического определения полярности витой пары MDI/MDIX.

Для обеспечения большей гибкости в выборе типа подключения, многие коммутаторы оборудованы специальными слотами для установки компактных сменных интерфейсных модулей *GBIC* (Gigabit Interface Converter), *SFP* (Small Form Factor Pluggable), *SFP+* (Enhanced Small Form Factor Pluggable) и *XFP* (10 Gigabit Small Form Factor Pluggable), поддерживающих режим «горячей замены».

Таблица 2

Стандарт	Тип кабеля	Максимальное расстояние передачи, м
10BASE-T	Кабель на основе витой пары категории 3 или 5	100
100BASE-TX	Кабель на основе витой пары категории 5	100
100BASE-FX	Многомодовый оптический кабель	412 (полудуплекс) 2000 (дуплекс)
100BASE-BX10	Одноволоконный одномодовый оптический кабель (длина волны: 1310 нм восходящий поток, 1550 нм нисходящий)	10 000
100BASE-LX10	Одномодовый оптический кабель (длина волны 1310 нм)	10 000
1000BASE-T	Кабель на основе витой пары категории 5, 5e, 6 или 7	100
1000BASE-SX	Многомодовый оптический кабель 62.5/125 микрон/ 50/125 микрон	220/550
1000BASE-LX	Одномодовый оптический кабель Многомодовый оптический кабель	5 000 550
1000BASE-LX10	Одномодовый оптический кабель (длина волны 1310 нм) Многомодовый оптический кабель (длина волны 1310 нм)	10 000 550
1000BASE-BX10	Одноволоконный одномодовый оптический кабель (длина волны: 1310 нм восходящий поток, 1550 нм нисходящий)	10 000
1000BASE-ZX	Одномодовый оптический кабель (длина волны 1550 нм)	80 000
1000BASE-LH (Long Haul)	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

Самой первой спецификацией на компактные сменные интерфейсные модули была спецификация SFF-8053 комитета SFF, описывающая конвертеры гигабитного интерфейса

(Gigabit Interface Converter, GBIC). Модули GBIC поддерживают стандарты Gigabit Ethernet или Fibre Channel для передачи данных, голоса и видео по медным или оптическим кабелям, но преимущественно представляют собой оптические трансиверы для приема или передачи сигнала по многомодовому или одномодовому волокну. Компания D-Link выпускает большой перечень модулей GBIC с поддержкой технологии Gigabit Ethernet с оптическими и медными интерфейсами.

Рис. 1.7. Модуль GBIC DGS-703 с 1 портом 1000Base-LX для одномодового оптического кабеля

Спустя несколько лет после появления спецификации GBIC разработчики предложили усовершенствованную, компактную модификацию сменного интерфейса (Small Form Factor Pluggable, SFP). Модули SFP в два раза меньше своих предшественников по габаритным размерам. Посадочный размер SFP (форм-фактор) определяется величиной медного разъема RJ-45. Интерфейсы SFP поддерживают практически любые существующие протоколы: Ethernet (на 10, 100, 1000 Мбит/с), SONET/SDH (OC3/ 12/48 и STM 1/4/16), Fibre Channel (1 и 2 Гбит/с). Компания D-Link выпускает модули SFP, поддерживающие стандарты Fast и Gigabit Ethernet и предназначенные для работы с разнообразным оптическим кабелем — одномодовым, одноволоконным одномодовым для систем с технологией *WDM* (Wavelength Division Multiplexing) и многомодовым.

Рис. 1.8. Модуль SFP DEM-310GT с 1 портом 1000Base-LX для одномодового оптического кабеля

Технология сменных модулей оказалась очень эффективной в системах оптического уплотнения (Wavelength Division Multiplexing, WDM), широко применяемых в сетях передачи данных и в телекоммуникационной отрасли. Основной принцип, на котором базируется работа этих устройств — модуляция сигнала для смещения спектра несущего сигнала в другой диапазон. Сигналы каждого канала переносятся в собственном диапазоне частот. Далее они собираются в мультиплексоре и передаются уже по одному волокну, образуя широкополосный канал. Таким образом, по одному волокну параллельно передается несколько независимых каналов (каждый на своей длине волны), что позволяет повысить пропускную способность системы передачи в целом. Задача объединения или разделения частот решается на уровне приемника или передатчика.

Компания D-Link производит модули SFP с использованием технологии WDM, поддерживающие скорости передачи 100 Мбит/с и 1000 Мбит/с. Эти модули позволяют одновременно передавать и получать сигналы на длинах волн 1310 нм и 1550 нм по одному

оптическому волокну на расстояние до 40 км. Это достигается путем установки SFP-модуля передатчика и SFP-модуля приемника на разных концах линии связи.

Рис. 1.9. Модули SFP DEM-331R и DEM-331T с 1 портом 1000BASE-BX10 с поддержкой технологии WDM

Модули SFP могут поддерживать важные функции цифровой диагностики (описанные в спецификации SFF-8472), позволяющие в реальном времени осуществлять мониторинг таких параметров как мощность передатчика, чувствительность приемника, напряжение питания и температура каждого оптического компонента. Информация о поддержке этой функции обычно указывается в спецификации на устройство.

Каждый модуль SFP выпускается с собственной электронной меткой, где содержатся сведения об идентификационном номере устройства и спецификации внешнего порта. Информация о внешнем порте может включать данные о длине волны, характеристиках волокна, скорости передачи данных, поддерживаемых протоколах, а также о длине канала. Идентификация SFP полезна при инвентаризации: с ее помощью отслеживается установка и замена компонентов и определяется местонахождение того или иного модуля.

Следующей ступенью эволюции сменных интерфейсов стала разработка оптических трансиверов XFP (10 Gigabit Small Form Factor Pluggable) для волн 850, 1310 и 1550 нм. Они поддерживают 10GE, 10 Gigabit SONET/SDH, Fibre Channel и еще некоторые высокоскоростные протоколы. XFP имеют несколько большие размеры, чем трансиверы SFP. Модули могут поддерживать систему цифровой диагностики для мониторинга состояния оптических линий.

В настоящее время компания D-Link выпускает трансиверы XFP 10GE, предназначенные для работы с одномодовым и многомодовым оптическим кабелем разной дальности передачи.

Рис. 1.10. Модуль XFP DEM-423XT с 1 портом 10GE (10GBASE-ER) для одномодового оптического кабеля

Новым поколением оптических сменных интерфейсных модулей с поддержкой скоростей 10 Гбит/с стали трансиверы SFP+. Требования к модулям SFP+, которые являются расширенной версией SFP, определены в спецификации SFF-8431. Несмотря на то, что модули SFP+ имеют ряд усовершенствований по сравнению с классическими модулями SFP, в коммутаторах D-Link слоты SFP+ поддерживают установку модулей SFP.

По сравнению с трансиверами XFP, модули SFP+ обладают меньшими габаритными размерами и тепловыделением, что позволяет повысить плотность размещения портов 10 Гбит/с на корпусе телекоммуникационных устройств.

Модули SFP+, также как и модули SFP могут поддерживать систему цифровой диагностики в соответствии со спецификацией SFF-8472.

Компания D-Link производит широкий спектр трансиверов SFP+ с поддержкой и без поддержки функции цифровой диагностики. Различают модули, предназначенные для работы как с одномодовым или многомодовым оптическим кабелем на длинах волн 850, 1310 и 1550 нм, так и с одноволоконным с поддержкой технологии WDM, использующие длины волн 1330 и 1270 нм для приема/передачи сигналов.

Рис. 1.11. Модуль SFP+ DEM-432XT-DD с 1 портом 10GE (10GBASE-LR) для одномодового оптического кабеля и поддержкой функции цифровой диагностики

1.7 Архитектура коммутаторов

Одним из основных компонентов всего коммутационного оборудования является *коммутирующая матрица* (*switch fabric*). Коммутирующая матрица представляет собой чипсет, соединяющий множество входов с множеством выходов на основе фундаментальных технологий и принципов коммутации. Коммутирующая матрица выполняет три функции:

- переключает трафик с одного порта матрицы на другой, обеспечивая их равнозначность;
- предоставляет качество обслуживания (Quality of Service, QoS);
- обеспечивает отказоустойчивость.

Поскольку коммутирующая матрица является ядром аппаратной платформы, к ней предъявляются требования по масштабированию производительности и возможности быстрого развития системы QoS.

Производительность коммутирующей матрицы (*switch capacity*) определяется как общая полоса пропускания (bandwidth), обеспечивающая коммутацию без отбрасывания пакетов трафика любого типа (одноадресного, многоадресного, широковещательного).

«Неблокирующей» коммутирующей матрицей (*non-blocking switch fabric*) является такая матрица, у которой производительность и QoS не зависят от типа трафика, коммутируемого через матрицу и производительность равна сумме скоростей всех портов:

$$\sum_{1}^{N} Cp_i \times 2 \text{ Мбит/с}^{*},$$

где N – количество портов, Cp_i - максимальная производительность протокола, поддерживаемого *i*-м портом коммутатора.

* Умножение на 2 для дуплексного режима работы.

Например, производительность коммутатора с 24 портами 10/100 Мбит/с и 2 портами 1 Гбит/с вычисляется следующим образом:

((24 x 100 Мбит/с) + (2 x 1 Гбит/с)) x 2 = 8.8 Гбит/с

Коммутатор обеспечивает портам равноправный доступ к матрице, если в системе не установлено преимущество одних портов над другими.

Поскольку коммутирующая матрица располагается в ядре платформы коммутатора, то одним из наиболее важных вопросов остается ее отказоустойчивость. Этот вопрос решается за счет реализации отказоустойчивой архитектуры, предусматривающей резервирование критичных для работы коммутатора блоков.

Одним из ключевых компонентов архитектуры современных коммутаторов является контроллер *ASIC* (*Application Specific Integrated Circuit*). Контроллеры ASIC представляют собой быстродействующие и относительно недорогие кремниевые кристаллы, которые предназначены для выполнения определенных операций. Использование в архитектуре коммутаторов контроллеров ASIC повышает производительность системы, т.к. ASIC выполняет операции аппаратно, благодаря чему не возникают накладные расходы, связанные с выборкой и интерпретацией хранимых команд. Современные контроллеры ASIC часто содержат на одном кристалле 32-битные процессоры, блоки памяти, включая ROM, RAM, EEPROM, Flash, и встроенное программное обеспечение. Такие ASIC получили название System-on-a-Chip (SoC).

В настоящее время существует много типов архитектур коммутирующих матриц. Выбор архитектуры матрицы во многом определяется ролью коммутатора в сети и количеством трафика, которое ему придется обрабатывать. В действительности, матрица обычно реализуется на основе комбинации двух или более базовых архитектур. Рассмотрим самые распространенные типы архитектур коммутирующих матриц.

1.7.1 Архитектура с разделяемой шиной

Архитектура с разделяемой шиной (*Shared Bus*), как следует из ее названия, использует в качестве разделяемой среды шину, которая обеспечивает связь подключенных к ней устройств ввода-вывода (портов). Шина используется в режиме разделения времени, т.е. в каждый момент времени только одному источнику разрешено передавать по ней данные. Управление доступом к шине осуществляется через централизованный арбитр, который предоставляет источнику право передавать данные.

Применительно к системам с разделяемой шиной, под термином «неблокирующая» понимается то, что сумма скоростей портов матрицы меньше, чем скорость шины. Т.е. производительность системы ограничена производительностью шины. Даже если общая полоса пропускания ниже производительности шины, количество и производительность устройств ввода-вывода ограничены производительностью централизованного арбитра.

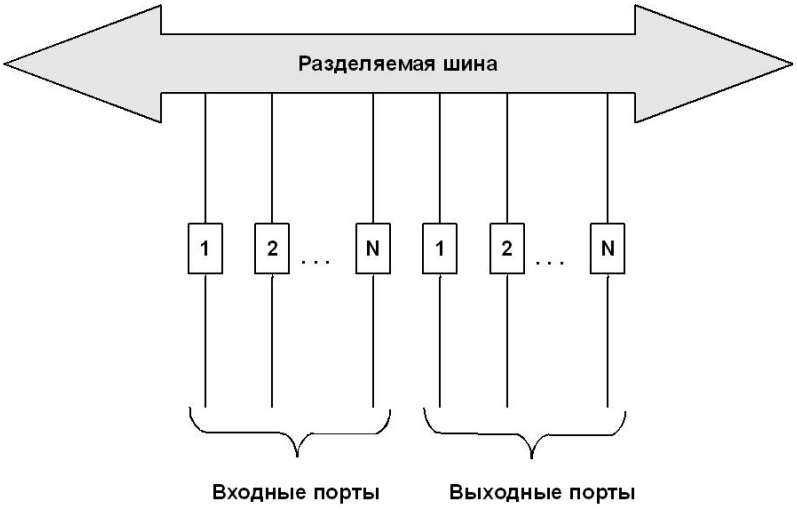


Рис. 1.12. Архитектура с разделяемой шиной

1.7.2 Архитектура с разделяемой памятью

Улучшения архитектуры с разделяемой шиной привели к появлению высокопроизводительной *архитектуры с разделяемой памятью* (*Shared Memory*).

Архитектура с разделяемой памятью обычно основана на использовании быстрой памяти RAM большой емкости в качестве общего буфера коммутационной системы, предназначенного для хранения входящих пакетов перед их передачей. Память обычно организуется в виде множества выходных очередей, ассоциирующихся с одним из устройств ввода-вывода или портом. Для обеспечения неблокирующей работы полоса пропускания памяти для операции «запись» и операции «чтение» должна быть равна максимальной суммарной полосе пропускания всех входных портов.

Типовая архитектура коммутаторов с разделяемой памятью показана на рис. 1.12. Входящие пакеты преобразуются из последовательного формата в параллельный и затем записываются в двухпортовую память. Запись в память осуществляется по принципу мультиплексирования с разделением по времени (Time Division Multiplexing, TDM), поэтому в каждый момент времени только один входной порт может поместить кадр в ячейку разделяемой памяти. Заголовки каждого кадра передаются в контроллер памяти. На основе этой информации он определяет выходной порт назначения и выходную очередь, в которую необходимо поместить кадр. Порядок, в котором выходные кадры будут считываться из памяти, определяется контроллером памяти с помощью механизма арбитража. Считанные кадры отправляются на соответствующие выходные порты (выходные кадры демультиплексируются с разделением по времени таким образом, что только один выходной порт может получить доступ к разделяемой памяти), где они вновь преобразуются из параллельного формата в последовательный.

Одним из преимуществ использования общего буфера для хранения пакетов является то, что он позволяет минимизировать количество выходных буферов, требуемых для поддержания скорости потери пакетов на низком уровне. С помощью централизованного буфера можно воспользоваться преимуществами статического разделения буферной памяти. При высокой скорости трафика на одном из портов он может захватить большее буферное пространство, если общий буферный пул не занят полностью.

Архитектура с разделяемой памятью обладает рядом недостатков. Так как пакеты записываются и считываются из памяти одновременно, она должна обладать суммарной пропускной способностью портов, т.е. операции записи и чтения из памяти должны выполняться в N (количество портов) раз выше скорости работы портов. Т.к. доступ к памяти физически ограничен, необходимость ускорения работы в N раз ограничивает масштабируемость архитектуры. Более того, контроллер памяти должен обрабатывать пакеты с той же скоростью, что и память. Такая задача может быть трудно выполнимой в случае управления множеством классов приоритетов и сложными операциями планирования. Коммутаторы с разделяемой памятью обладают единой точкой отказа, поскольку добавление еще одного общего буфера является сложным и дорогим. В результате этого в чистом виде архитектура с разделяемой памятью используется для построения коммутаторов с небольшим количеством портов.

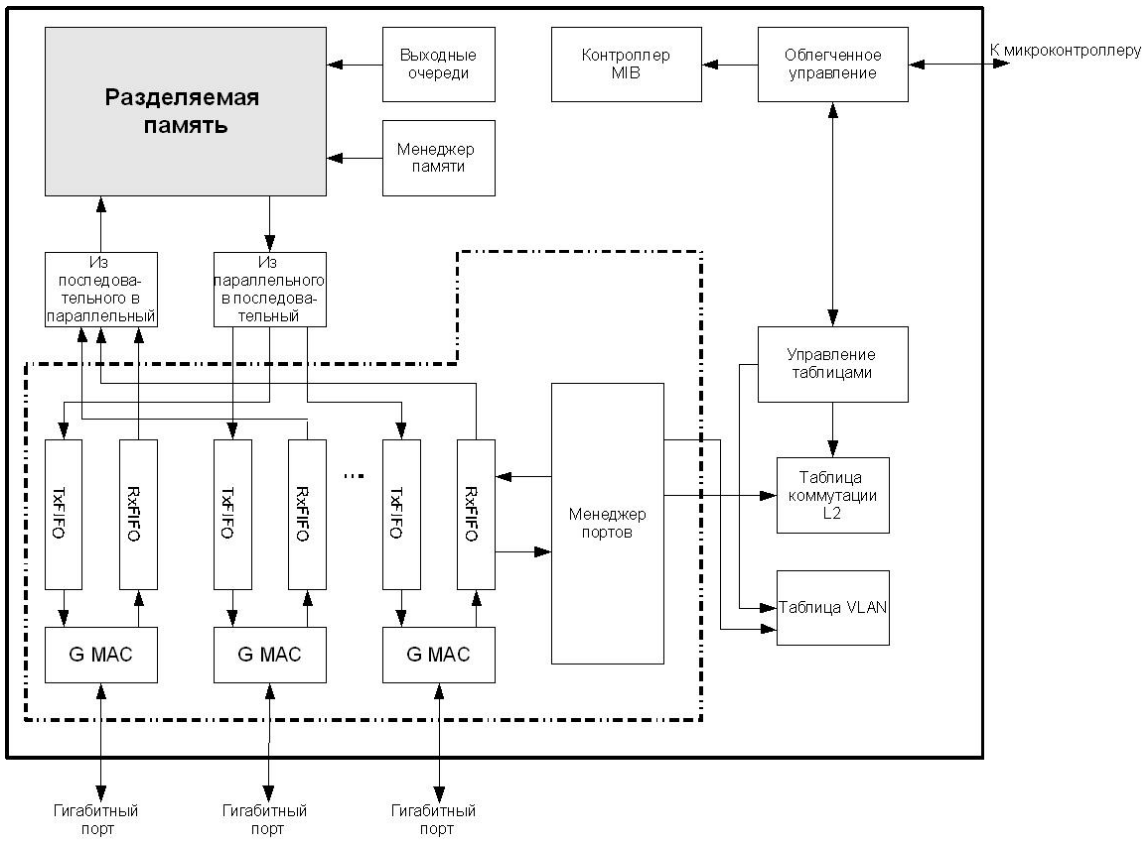


Рис. 1.13. Архитектура с разделяемой памятью

1.7.3 Архитектура на основе коммутационной матрицы

Параллельно с появлением архитектуры с разделяемой памятью (в середине 1990-х годов) была разработана *архитектура на основе коммутационной матрицы* (*Crossbar architecture*). Эта архитектура используется для построения коммутаторов различных типов.

Существует множество вариаций архитектуры этого типа. Базовая архитектура на основе коммутационной матрицы *N x N* непосредственно соединяет *N* входных портов с *N* выходными портами в виде матрицы. В местах пересечения проводников, соединяющих входы и выходы, находятся коммутирующие устройства, которыми управляет специальный контроллер. В каждый момент времени, анализируя адресную информацию, контроллер сообщает коммутирующим устройствам, какой выход должен быть подключен к какому входу. В том случае, если два входящих пакета от разных портов-источников будут переданы на один и тот же выходной порт, он будет заблокирован. Существуют различные подходы к решению этой проблемы: повышение производительности матрицы по сравнению с производительностью входных портов или использование буферов памяти и арбитров.

Несмотря на простой дизайн, одной из фундаментальных проблем архитектуры на основе коммутационной матрицы остается ее масштабируемость. При увеличении количества входов и выходов усложняется схемотехника матрицы и в особенности контроллера. Поэтому для построения многопортовых коммутационных матриц используется другой подход, который заключается в том, что простые коммутационные матрицы связываются между собой, образуя одну большую коммутационную матрицу.

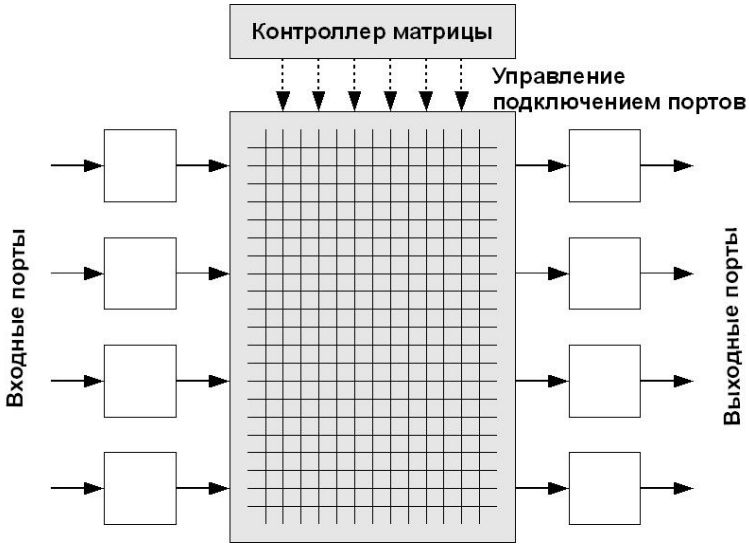


Рис. 1.14. Архитектура на основе коммутационной матрицы

Можно выделить два типа коммутаторов на основе коммутационной матрицы:
- коммутаторы на основе коммутационной матрицы с буферизацией (buffered crossbar);
- коммутаторы на основе коммутационной матрицы с арбитражем (arbitrated crossbar).

1.7.3.1 Коммутаторы на основе коммутационной матрицы с буферизацией

В *коммутаторах на основе коммутационной матрицы с буферизацией* буферы расположены на трех основных стадиях: на входе и выходе, и непосредственно на коммутационной матрице. Благодаря наличию очередей на трех стадиях, эта архитектура позволяет избежать сложностей, связанных с реализацией механизма централизованного арбитража. На выходе каждой из стадий осуществляется управление очередями с помощью одного из алгоритмов диспетчеризации.

Несмотря на то, что эта архитектура является простейшей архитектурой коммутаторов, из-за независимости стадий для нее существуют сложности с реализацией качества обслуживания (QoS) в пределах коммутатора.

1.7.3.2 Коммутаторы на основе коммутационной матрицы с арбитражем

Эта архитектура характеризуется наличием безбуферных коммутирующих элементов и арбитра, который управляет передачей трафика между входами и выходами матрицы. Отсутствие буферов у коммутирующих элементов компенсируется наличием буферов входных и выходных портов. Обычно разработчики используют один из трех методов буферизации: выходные буферы, входные буферы, комбинированные входные и выходные буферы.

В *коммутаторах с входными очередями* (*Input-Queued Switch*) память каждого входного порта организована в виде очередей типа FIFO (First Input First Output, «первым пришел, первым ушел»), которая используется для буферизации пакетов перед началом процесса коммутации. Одной из проблем этого типа коммутационной матрицы является *блокировка первым в очереди* (*Head-Of-Line blocking, HOL*). Она возникает в том случае, когда коммутатор пытается одновременно передать пакеты из нескольких входных очередей на один выходной порт. При этом пакеты, находящиеся в начале этих очередей блокируют все остальные пакеты, находящиеся за ними. Для принятия решения о том, какой пакет и из какой очереди может получить доступ к матрице, используется арбитр. Перед передачей пакета входные порты направляют арбитру запросы на подключение к разделяемому ресурсу (в данном случае пути матрицы) и получают от него право на подключение.

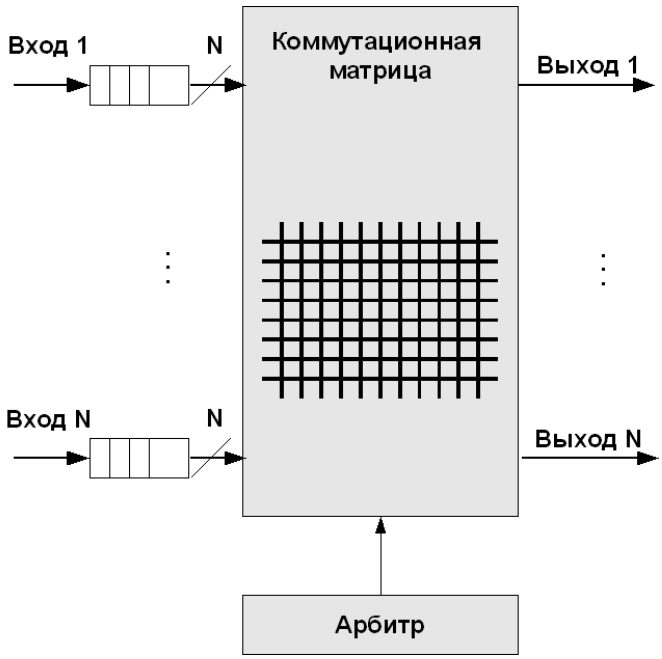


Рис. 1.15. Архитектура на основе коммутационной матрицы с входными очередями

Арбитр принимает решение о последовательности передачи пакетов из входных очередей на основе алгоритма диспетчеризации (scheduling algorithm).

В *коммутаторах с выходными очередями* (*output-queued switch*) пакеты буферизируются только на выходных портах после завершения процесса коммутации. В этом случае удается избежать проблемы, связанной с блокированием очередей HOL. Коммутаторы этой архитектуры используют арбитр для управления временем, за которое пакеты коммутируются через матрицу. При правильно разработанном арбитре, коммутаторы с выходными очередями могут обеспечивать качество обслуживания (QoS).

Следует отметить, что выходной буфер каждого порта требует большего объема памяти по сравнению с входным буфером. Это позволяет избежать блокирования на выходе, когда все входные порты пытаются подключиться к одному выходу. Еще одним важным фактором, является скорость выполнения операции «запись» коммутируемых пакетов в выходную очередь. По этим двум причинам архитектура с выходными очередями должна быть реализована на высокоскоростных элементах, что делает ее очень дорогостоящей.

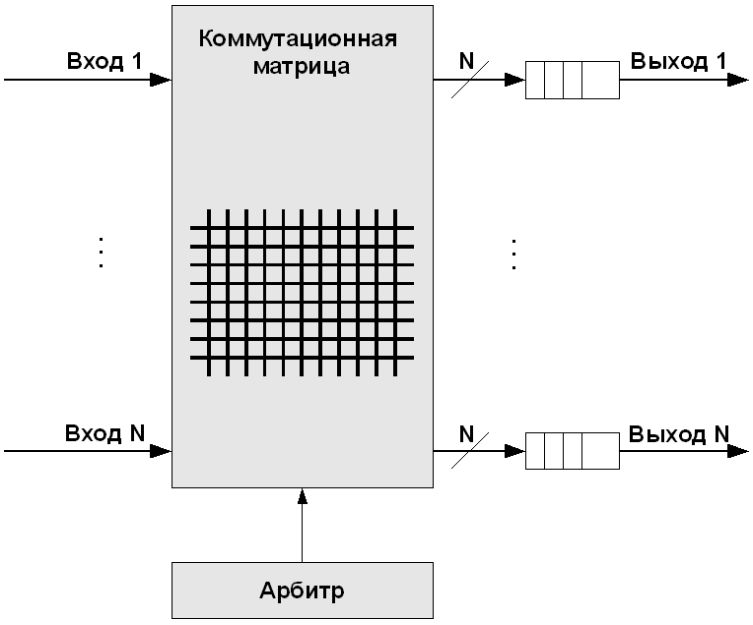


Рис. 1.16. Архитектура на основе коммутационной матрицы с выходными очередями

Коммутаторы с *виртуальными очередями* (*Virtual Output Queues, VOQ*) позволяют преодолеть проблему блокировки очередей HOL, не внося издержек по сравнению с коммутаторами с выходными очередями. В этой архитектуре память каждого входного порта организована в виде *N* (*N* –количество выходных портов) логических очередей типа FIFO, по одной для каждого выходного порта. Эти очереди используются для буферизации пакетов, поступающих на входной порт и предназначенных для выходного порта *j* (*j* = 1,….*N*).

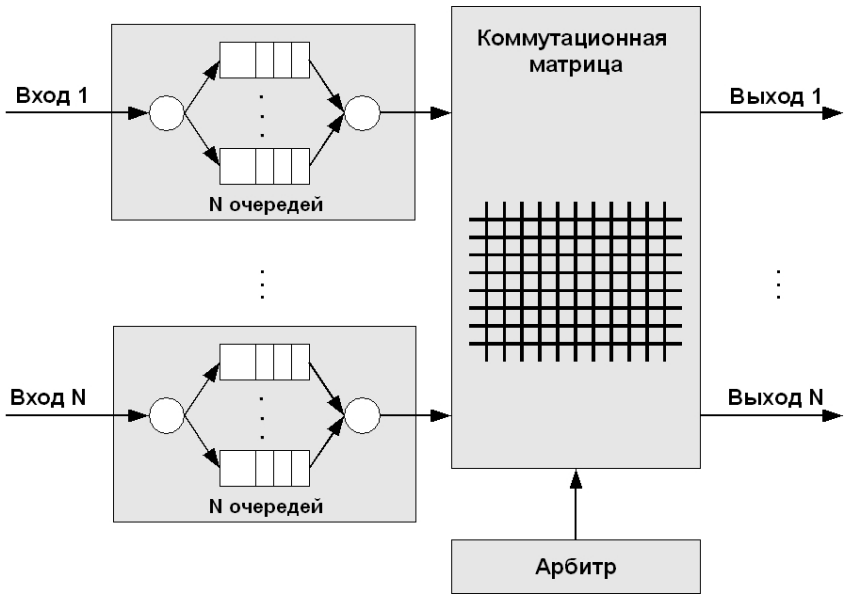


Рис. 1.17. Архитектура на основе коммутационной матрицы с виртуальными очередями

В том случае, если существует несколько виртуальных очередей, может возникнуть проблема, связанная с одновременным доступом к коммутационной матрице и блокировкой очередей. Для решения этой проблемы используется арбитр, который на основе алгоритма диспетчеризации выбирает пакеты из разных очередей.

В *коммутаторах с комбинированными входными и выходными очередями* (*Combined Input and Output Queued, CIOQ*) буферы памяти подключены как к входным, так и выходным портам. Память каждого из входных портов организована в виде *N* виртуальных выходных очередей типа FIFO, по одной для каждого выходного порта. Каждый из *N* выходных портов также содержит очередь типа FIFO, которая используется для буферизации пакетов, ожидающих передачи через него. Система коммутации работает по принципу конвейера, каждая стадия которого называется временным слотом (time slot). В течение временного слота 1, который называется стадией прибытия, пакеты поступают на входные порты. Для передачи внутри коммутатора все пакеты сегментируются на ячейки фиксированного размера. Размер такой ячейки данных определяется производителем коммутатора. Каждая ячейка снабжается меткой с указанием размера, номера входного порта и порта назначения, и помещается в виртуальную выходную очередь соответствующего выходного порта. Входные порты отправляют «запросы на подключение к выходам» централизованному арбитру, а все выходные порты отправляют ему «информацию о перегрузке» (переполнении выходных буферов).

Во временной слот 2, который называется стадией диспетчеризации, ячейки передаются из входных очередей в выходные. Последовательность передачи ячеек определяется централизованным арбитром с помощью алгоритма диспетчеризации. Для того чтобы выходные очереди быстро заполнялись пакетами из входных очередей (с целью уменьшения задержки передачи пакетов и обеспечения QoS), алгоритм диспетчеризации должен обеспечивать циклическое высокоскоростное сопоставление входных и выходных

очередей. Это сопоставление используется для настройки управляемых переключателей матрицы перед передачей пакетов с входов на выходы.

Во временной слот 3, который называется стадией передачи, осуществляется сборка пакетов и их передача с выходных портов.

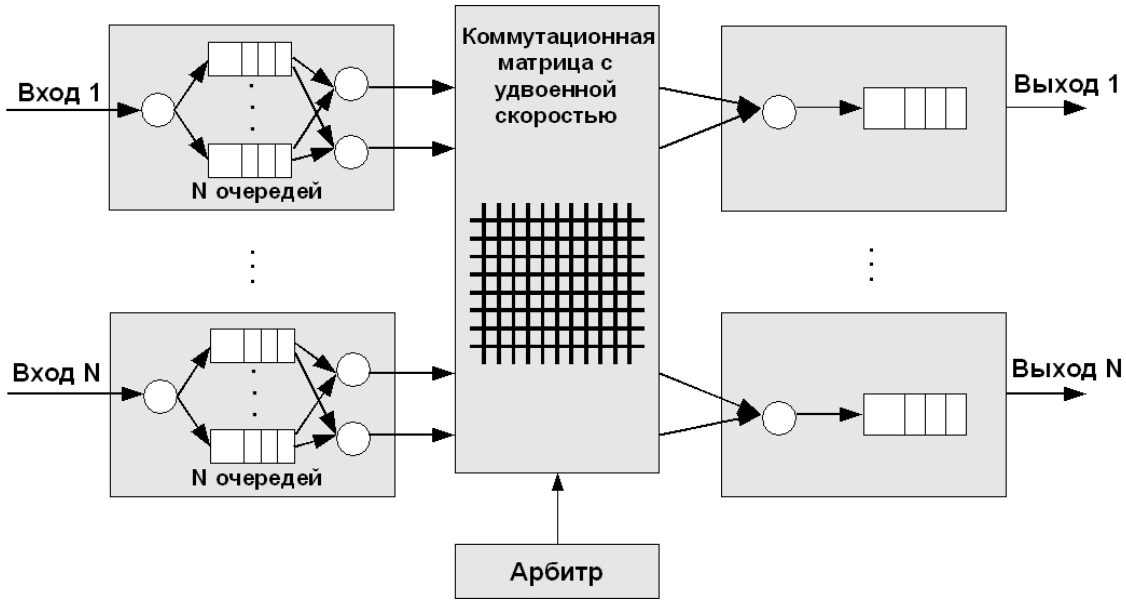


Рис. 1.18. Архитектура на основе коммутационной матрицы с CIOQ

1.8 Характеристики, влияющие на производительность коммутаторов

Производительность коммутатора – характеристика, на которую сетевые интеграторы и опытные администраторы обращают внимание в первую очередь при выборе устройства.

Основными показателями коммутатора, характеризующими его производительность, являются:

- скорость фильтрации кадров;
- скорость продвижения кадров;
- пропускная способность;
- задержка передачи кадра.

Кроме того, существует несколько характеристик коммутатора, которые в наибольшей степени влияют на указанные характеристики производительности. К ним относятся:

- тип коммутации;
- размер буфера (буферов) кадров;
- производительность коммутирующей матрицы;
- производительность процессора или процессоров;
- размер таблицы коммутации.

1.8.1 Скорость фильтрации и скорость продвижения кадров

Скорость фильтрации и продвижения кадров – это две основные характеристики производительности коммутатора. Эти характеристики являются интегральными показателями и не зависят от того, каким образом технически реализован коммутатор.

Скорость фильтрации (filtering) определяет скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

- прием кадра в свой буфер;
- отбрасывание кадра, в случае обнаружения в нем ошибки (не совпадает контрольная сумма, или кадр меньше 64 байт или больше 1518 байт);

- отбрасывание кадра для исключения петель в сети;
- отбрасывание кадра в соответствии с настроенными на порте фильтрами;
- просмотр таблицы коммутации с целью поиска порта назначения на основе MAC-адреса приемника кадра и отбрасывание кадра, если узел-отправитель и получатель кадра подключены к одному порту.

Скорость фильтрации практически у всех коммутаторов является неблокирующей - коммутатор успевает отбрасывать кадры в темпе их поступления.

Скорость продвижения (forwarding) определяет скорость, с которой коммутатор выполняет следующие этапы обработки кадров:

- прием кадра в свой буфер;
- просмотр таблицы коммутации с целью нахождения порта назначения на основе MAC-адреса получателя кадра;
- передача кадра в сеть через найденный по таблице коммутации порт назначения.

Как скорость фильтрации, так и скорость продвижения измеряется обычно в кадрах в секунду. Если в характеристиках коммутатора не уточняется, для какого протокола и для какого размера кадра приведены значения скоростей фильтрации и продвижения, то по умолчанию считается, что эти показатели даются для протокола Ethernet и кадров минимального размера, то есть кадров длиной 64 байт (без преамбулы) с полем данных в 46 байт. Применение в качестве основного показателя скорости обработки коммутатором кадров минимальной длины объясняется тем, что такие кадры всегда создают для коммутатора наиболее тяжелый режим работы по сравнению с кадрами другого формата при равной пропускной способности передаваемых пользовательских данных. Поэтому при проведении тестирования коммутатора режим передачи кадров минимальной длины используется как наиболее сложный тест, который должен проверить способность коммутатора работать при наихудшем сочетании параметров трафика.

Пропускная способность коммутатора (throughput) измеряется количеством пользовательских данных (в мегабитах или гигабитах в секунду), переданных в единицу времени через его порты. Так как коммутатор работает на канальном уровне, для него пользовательскими данными являются те данные, которые переносятся в поле данных кадров протоколов канального уровня – Ethernet, Fast Ethernet и т.д. Максимальное значение пропускной способности коммутатора всегда достигается на кадрах максимальной длины, так как при этом доля накладных расходов на служебную информацию кадра гораздо ниже, чем для кадров минимальной длины, а время выполнения коммутатором операций по обработке кадра, приходящееся на один байт пользовательской информации, существенно меньше. Поэтому коммутатор может быть блокирующим для кадров минимальной длины, но при этом иметь очень хорошие показатели пропускной способности.

Задержка передачи кадра (forward delay) измеряется как время, прошедшее с момента прихода первого байта кадра на входной порт коммутатора до момента появления этого байта на его выходном порту. Задержка складывается из времени, затрачиваемого на буферизацию байт кадра, а также времени, затрачиваемого на обработку кадра коммутатором, а именно просмотра таблицы коммутации, принятия решения о продвижении и получения доступа к среде выходного порта.

Величина вносимой коммутатором задержки зависит от используемого в нем метода коммутации. Если коммутация осуществляется без буферизации, то задержки обычно невелики и составляют от 5 до 40 мкс, а при полной буферизации кадров – от 50 до 200 мкс (для кадров минимальной длины).

1.8.2 Размер таблицы коммутации

Максимальная емкость таблицы коммутации определяет предельное количество MAC-адресов, которыми может одновременно оперировать коммутатор. В таблице

коммутации для каждого порта могут храниться как динамически изученные MAC-адреса, так и статические MAC-адреса, которые были созданы администратором сети.

Значение максимального числа MAC-адресов, которое может храниться в таблице коммутации, зависит от области применения коммутатора. Коммутаторы D-Link для рабочих групп и малых офисов обычно поддерживают таблицу MAC-адресов емкостью от 1К до 8К. Коммутаторы крупных рабочих групп поддерживают таблицу MAC-адресов емкостью от 8К до 16К, а коммутаторы магистралей сетей – как правило, от 16К до 64К адресов и более.

Недостаточная емкость таблицы коммутации может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если таблица коммутации полностью заполнена, и порт встречает новый MAC-адрес источника в поступившем кадре, коммутатор не сможет занести его в таблицу. В этом случае ответный кадр на этот MAC-адрес будет разослан через все порты (за исключением порта-источника), т.е. вызовет лавинную передачу.

1.8.3 Объем буфера кадров

Для обеспечения временного хранения кадров в тех случаях, когда их невозможно немедленно передать на выходной порт, коммутаторы, в зависимости от реализованной архитектуры, оснащаются буферами на входных, выходных портах или общим буфером для всех портов. Размер буфера влияет как на задержку передачи кадра, так и на скорость потери пакетов. Поэтому чем больше объем буферной памяти, тем менее вероятны потери кадров.

Обычно коммутаторы, предназначенные для работы в ответственных частях сети, обладают буферной памятью в несколько десятков или сотен килобайт на порт. Общий для всех портов буфер обычно имеет объем в несколько мегабайт.

1.9 Управление потоком в полудуплексном и дуплексном режимах

Механизм *управления потоком (Flow Control)* позволяет предотвратить потерю данных в случае переполнения буфера принимающего устройства.

Для управления потоком в *полудуплексном режиме* обычно используется *метод «Обратного давления»* (Backpressure). Принимающее устройство (порт коммутатора), в случае переполнения его буфера, посылает искусственно созданный сигнал обнаружения коллизии или обратно отправляет устройству-отправителю его кадры.

Для управления потоком в *дуплексном режиме* используется стандарт IEEE 802.3x. Согласно этому стандарту управление потоком осуществляется между MAC-уровнями с помощью специального кадра-паузы, который автоматически формируется MAC-уровнем принимающего устройства. В случае переполнения буфера принимающее устройство отправляет кадр-паузу с указанием периода времени, на который требуется остановить передачу данных, либо на уникальный MAC-адрес соответствующей станции, либо на специальный групповой адрес 01-80-C2-00-00-01. Если переполнение буфера будет ликвидировано до истечения периода ожидания, то для восстановления передачи, принимающая станция отправляет второй кадр-паузу с нулевым значением времени ожидания.

Общая схема управления потоком показана на рис. 1.19.

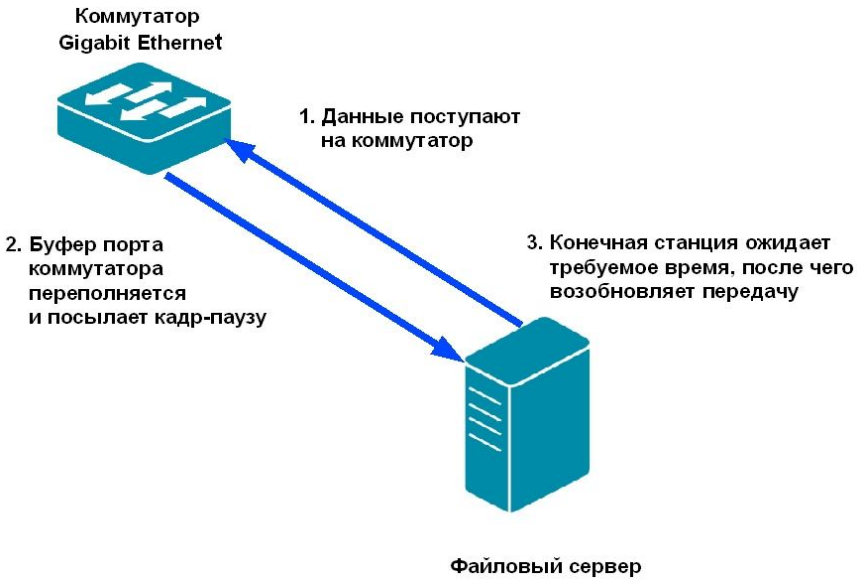


Рис. 1.19. Последовательность управления потоком IEEE 802.3x

Дуплексный режим работы и сопутствующее ему управление потоком являются дополнительными режимами для всех MAC-уровней Ethernet независимо от скорости передачи. Кадры-паузы идентифицируются как управляющие MAC-кадры по уникальным значениям полей «Длина/тип» (88-08) и «Код операции управления MAC» (00-01).

Преамбула	Начальный ограничитель кадра	Адрес назначения	Адрес источника	Длина/ тип	Код операции управления MAC (00-01)	Время паузы (от 00-00 до FF-FF)	Зарезер-вировано	Контрольная сумма кадра
7 байт	1 байт	6 байт	6 байт	2 байта	2 байта	2 байта	42 байта	4 байта

Рис. 1.20. Формат кадра-паузы

Правильно сконфигурированная функция управления потоком на устройствах позволяет повысить общую производительность сети за счет уменьшения потери данных и повторных передач. Управление потоком данных IEEE 802.3x большинства интерфейсных сетевых карт и встроенных сетевых карт включено по умолчанию. Коммутаторы D-Link имеют разные настройки функции IEEE 802.3x по умолчанию:

- неуправляемые коммутаторы – управление потоком IEEE 802.3x включено;
- коммутаторы серии Smart – управление потоком IEEE 802.3x отключено;
- управляемые коммутаторы – управление потоком IEEE 802.3x отключено.

Поэтому, для корректной работы функции IEEE 802.3x на порте коммутатора должна быть активизирована функция управления потоком.

1.10 Технологии коммутации и модель OSI

Коммутаторы локальных сетей можно классифицировать в соответствии с уровнями модели OSI, на которых они передают, фильтруют и коммутируют кадры. Различают коммутаторы уровня 2 (Layer 2 (L2) Switch) и коммутаторы уровня 3 (Layer 3 (L3) Switch).

Коммутаторы уровня 2 анализируют входящие кадры, принимают решение об их дальнейшей передаче и передают их пунктам назначения на основе MAC-адресов канального уровня модели OSI. Основное преимущество коммутаторов уровня 2 – прозрачность для протоколов верхнего уровня. Т.к. коммутатор функционирует на 2-м уровне, ему нет необходимости анализировать информацию верхних уровней модели OSI.

Коммутация 2-го уровня – аппаратная. Она обладает высокой производительностью. Передача кадра в коммутаторе может осуществляться специализированным контроллером ASIC. В основном коммутаторы 2-го уровня используются для сегментации сети и объединения рабочих групп.

Несмотря на преимущества коммутации 2-го уровня, она все же имеет некоторые ограничения. Наличие коммутаторов в сети не препятствует распространению широковещательных кадров по всем сегментам сети.

Коммутатор уровня 3 осуществляют коммутацию и фильтрацию на основе адресов канального (уровень 2) и сетевого (уровень 3) уровней модели OSI. Коммутаторы 3-го уровня выполняет коммутацию в пределах рабочей группы и маршрутизацию между различными подсетями или виртуальными локальными сетями (VLAN).

Коммутаторы уровня 3 осуществляют маршрутизацию пакетов аналогично традиционным маршрутизаторам. Они поддерживают протоколы маршрутизации RIP (Routing Information Protocol), OSPF (Open Shortest Path First), BGP (Border Gateway Protocol), для обеспечения связи с другими коммутаторами уровня 3 или маршрутизаторами и построения таблиц маршрутизации, осуществляют маршрутизацию на основе политик, управление многоадресным трафиком.

Существует две разновидности маршрутизации: аппаратная (коммутация 3 уровня) и программная. При аппаратной реализации пересылка пакетов осуществляется при помощи специализированных контроллеров ASIC. При программной реализации, для пересылки пакетов устройство использует центральный процессор. Обычно в коммутаторах 3 уровня и старших моделях маршрутизаторов маршрутизация пакетов аппаратная, что позволяет выполнять ее на скорости канала связи, а в маршрутизаторах общего назначения функция маршрутизации выполняется программно.

1.11 Программное обеспечение коммутаторов

Программное обеспечение коммутаторов D-Link предоставляет набор программных сервисов, предназначенных для выполнения различных функций, обеспечивающих безопасность, отказоустойчивость сети, управление многоадресной рассылкой, качество обслуживания (QoS), а также развитые средства настройки и управления. Помимо этого, программное обеспечение коммутаторов взаимодействует с приложениями D-Link D-View v.6, представляющими собой прикладные программы сетевого управления. Эти управляющие программы поддерживаются всей линейкой управляемых коммутаторов D-Link.

Системное программное обеспечение располагается во Flash-памяти коммутатора, размер которой, в зависимости от модели, может быть до 32 Мбайт. Компания D-Link предоставляет возможность бесплатного обновления программного обеспечения коммутаторов, по мере появления новых версий с обновленным функционалом.

1.12 Общие принципы сетевого дизайна

Грамотный сетевой проект основывается на многих принципах, базовыми из которых являются:

- *Изучение возможных точек отказа сети*. Для того чтобы единичный отказ не мог изолировать какой-либо из сегментов сети, в ней может быть предусмотрена избыточность. Под избыточностью понимается резервирование жизненно важных компонентов сети и распределение нагрузки. Так в случае отказа в сети, может существовать альтернативный или резервный путь к любому ее сегменту. Распределение нагрузки используется в том случае, если к пункту назначения имеется два или более путей, которые могут использоваться в зависимости от загруженности

сети. Требуемый уровень избыточности сети меняется в зависимости от ее конкретной реализации.

- *Определение типа трафика сети.* Например, если в сети используются клиент-серверные приложения, то поток вырабатываемого ими трафика является критичным для эффективного распределения ресурсов, таких как количество клиентов, использующих определенный сервер, или количество клиентских рабочих станций в сегменте.
- *Анализ доступной полосы пропускания*. Например, в сети не должно быть большого различия в доступной полосе пропускания между различными уровнями иерархической модели (описание иерархической модели сети находится в следующем разделе). Важно помнить, что иерархическая модель ссылается на концептуальные уровни, которые обеспечивают функциональность.
- *Создание сети на базе иерархической или модульной модели.* Иерархия позволяет объединить через межсетевые устройства отдельные сегменты, которые будут функционировать как единая сеть. Фактическая граница между уровнями не обязательно должна проходить по физическому каналу связи – ей может быть и внутренняя магистраль определенного устройства.

1.13 Трехуровневая иерархическая модель сети

Иерархическая модель определяет подход к проектированию сетей и включает в себя три логических уровня (рис. 1.21):

- уровень доступа (*access layer*);
- уровень распределения/агрегации (*distribution layer*);
- уровень ядра (*core layer*).

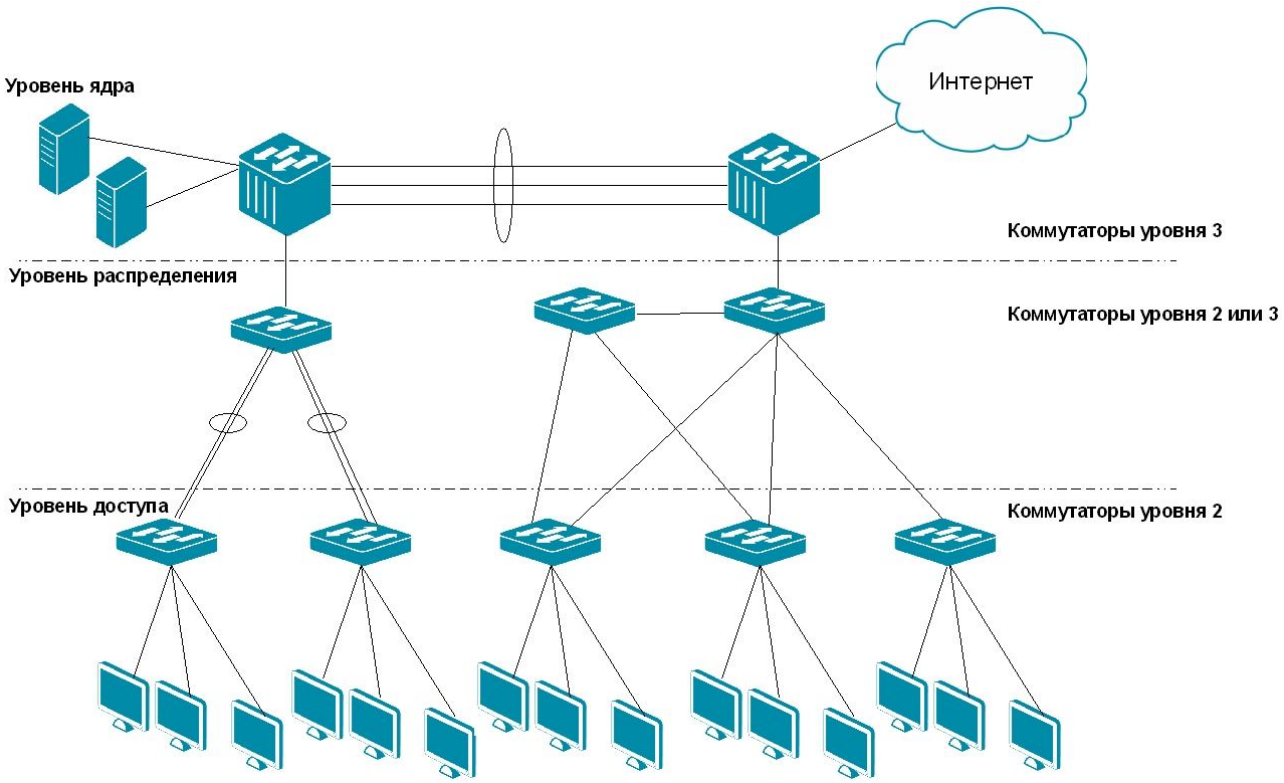


Рис. 1.21. Трехуровневая модель сети

Для каждого уровня определены свои функции. Три уровня не обязательно предполагают наличие трех различных устройств. Если провести аналогию с иерархической

моделью OSI, то в ней отдельный протокол не всегда соответствует одному из семи уровней. Иногда протокол соответствует более чем одному уровню модели OSI, а иногда несколько протоколов реализованы в рамках одного уровня. Так и при построении иерархических сетей, на одном уровне может быть как несколько устройств, так и одно устройство, выполняющее все функции, определенные на двух соседних уровнях.

Уровень ядра – находится на самом верху иерархии и отвечает за надежную и быструю передачу больших объемов данных. Трафик, передаваемый через ядро, является общим для большинства пользователей. Сами пользовательские данные обрабатываются на уровне распределения, который, при необходимости, пересылает запросы к ядру.

Для уровня ядра большое значение имеет его отказоустойчивость, поскольку сбой на этом уровне может привести к потере связности между уровнями распределения сети.

Уровень распределения, который иногда называют уровнем рабочих групп, является связующим звеном между уровнями доступа и ядра. В зависимости от способа реализации, уровень распределения может выполнять следующие функции:

- обеспечение маршрутизации, качества обслуживания и безопасности сети;
- агрегирование каналов;
- переход от одной технологии к другой (например, от 100Base-TX к 1000Base-T).

Уровень доступа управляет доступом пользователей и рабочих групп к ресурсам объединенной сети. Основной задачей уровня доступа является создание точек входа/выхода пользователей в сеть. Уровень выполняет следующие функции:

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

2. Начальная настройка коммутатора

2.1 Классификация коммутаторов по возможности управления

Коммутаторы локальной сети можно классифицировать по возможности управления. Существует три категории коммутаторов:

- неуправляемые коммутаторы;
- управляемые коммутаторы;
- настраиваемые коммутаторы.

Неуправляемые коммутаторы не поддерживают возможности управления и обновления программного обеспечения.

Управляемые коммутаторы являются сложными устройствами, позволяющими выполнять расширенный набор функций 2 и 3 уровня модели OSI. Управление коммутаторами может осуществляться посредством Web-интерфейса, командной строки (CLI), протокола SNMP, Telnet и т.д.

Настраиваемые коммутаторы занимают промежуточную позицию между ними. Они предоставляют пользователям возможность настраивать определенные параметры сети с помощью интуитивно понятных утилит управления, Web-интерфейса, упрощенного интерфейса командной строки, протокола SNMP.

2.2 Средства управления коммутаторами

Большинство современных коммутаторов поддерживают различные функции управления и мониторинга. К ним относятся дружественный пользователю Web-интерфейс управления, интерфейс командной строки (Command Line Interface, CLI), Telnet, SNMP-управление. В коммутаторах D-Link серии Smart также реализована поддержка начальной настройки и обновления программного обеспечения через утилиту D-Link SmartConsole Utility.

Web-интерфейс управления позволяет осуществлять настройку и мониторинг параметров коммутатора, используя любой компьютер, оснащенный стандартным Web-браузером. Браузер представляет собой универсальное средство доступа и может непосредственно подключаться к коммутатору по протоколу HTTP.

Главная страница Web-интерфейса обеспечивает доступ к различным настройкам коммутатора и отображает всю необходимую информацию об устройстве. Администратор может быстро посмотреть статус устройства, статистику по производительности и т.д., а также произвести необходимые настройки.

Доступ к интерфейсу командной строки коммутатора осуществляется путем подключения к его консольному порту терминала или персонального компьютера с установленной программой эмуляции терминала. Это метод доступа наиболее удобен при первоначальном подключении к коммутатору, когда значение IP-адреса не известно или не установлено, в случае необходимости восстановления пароля и при выполнении расширенных настроек коммутатора. Также доступ к интерфейсу командной строки может быть получен по сети с помощью протокола Telnet.

Пользователь может использовать для настройки коммутатора любой удобный ему интерфейс управления, т.к. набор доступных через разные интерфейсы управления функций одинаков для каждой конкретной модели.

Еще один способ управления коммутатором – использование протокола SNMP (Simple Network Management Protocol). Протокол SNMP является протоколом 7 уровня модели OSI и разработан специально для управления и мониторинга сетевыми устройствами и приложениями связи, путем обмена управляющей информацией между агентами,

располагающимися на сетевых устройствах, и менеджерами, расположенными на станциях управления. Коммутаторами D-Link поддерживается протокол SNMP версий 1, 2c и 3.

Также стоит отметить возможность обновления программного обеспечения коммутаторов (за исключением неуправляемых). Это обеспечивает более долгий срок эксплуатации устройств, т.к. позволяет добавлять новые функции либо устранять имеющиеся ошибки по мере выхода новых версий ПО, что существенно облегчает и удешевляет использование устройств. Компания D-Link распространяет новые версии ПО бесплатно. Сюда же можно включить возможность сохранения настроек коммутатора на случай сбоев с последующим восстановлением или тиражированием, что избавляет администратора от выполнения рутинной работы.

2.3 Подключение к коммутатору

Перед тем, как начать настройку коммутатора, необходимо установить физическое соединение между ним и рабочей станцией. Существуют два типа кабельного соединения, используемых для управления коммутатором. Первый тип – через консольный порт (если он имеется у устройства), второй – через порт Ethernet (по протоколу Telnet или через Web-интерфейс). Консольный порт используется для первоначальной конфигурации коммутатора и обычно не требует настройки. Для того чтобы получить доступ к коммутатору через порт Ethernet, в браузере необходимо ввести IP-адрес по умолчанию его интерфейса управления (обычно он указан в руководстве пользователя).

При подключении к медному (разъем RJ-45) порту Ethernet коммутатора Ethernet-совместимых серверов, маршрутизаторов или рабочих станций, используется четырехпарный кабель UTP категории 5, 5e или 6 для Gigabit Ethernet. Поскольку коммутаторы D-Link поддерживают функцию автоматического определения полярности (MDI/MDIX), можно использовать любой тип кабеля (прямой или кроссовый).

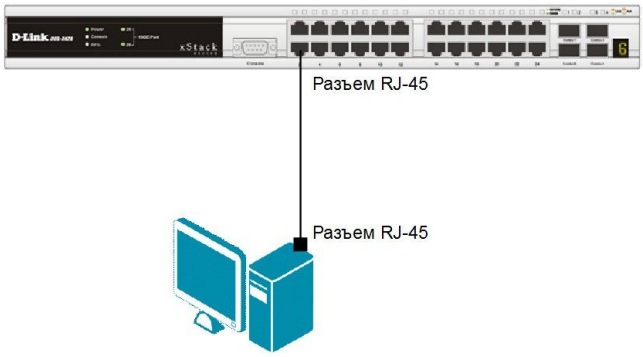


Рис. 2.1. Подключение компьютера к коммутатору

Для подключения к медному (разъем RJ-45) порту Ethernet другого коммутатора так же можно использовать любой четырехпарный кабель UTP категории 5, 5e, 6, при условии, что порты коммутатора поддерживают автоматическое определение полярности. В противном случае надо использовать кроссовый кабель.

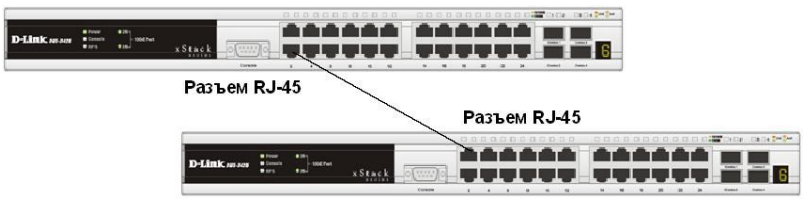


Рис. 2.2. Соединение коммутаторов с помощью прямого или кроссового кабеля

Правильность подключения поможет определить светодиодная индикация порта. Если соответствующий индикатор горит, то связь между коммутатором и подключенным

устройством установлена. Если индикатор не горит, возможно, что не включено питание одного из устройств или возникли проблемы с сетевым адаптером подключенного устройства, или имеются неполадки с кабелем. Если индикатор загорается и гаснет, возможно, есть проблемы с автоматическим определением скорости и режимом работы (дуплекс/полудуплекс). (За подробным описанием сигналов индикаторов необходимо обратиться к руководству пользователя коммутатора конкретной модели).

2.3.1 Подключение к консоли интерфейса командной строки коммутатора

Управляемые коммутаторы D-Link оснащены консольным портом. В зависимости от модели коммутатора, консольный порт может обладать разъемом DB-9 или RJ-45. С помощью консольного кабеля, входящего в комплект поставки, коммутатор подключается к последовательному порту компьютера. Подключение по консоли иногда называют «Out-of-Band»-подключением. Это означает, что консоль использует отличную от обычного сетевого подключения схему (не использует полосу пропускания портов Ethernet).

После подключения к консольному порту коммутатора, на персональном компьютере необходимо запустить программу эмуляции терминала VT100 (например, программу HyperTerminal в Windows). В программе следует установить следующие параметры подключения, которые, как правило, указаны в документации к устройству:

Скорость (бит/с):	9600 или 115200*
Биты данных:	8
Четность:	нет
Стоповые биты:	1
Управление потоком:	нет

* Этот параметр зависит от модели коммутатора и указывается в руководстве пользователя.

При соединении коммутатора с консолью появится следующее окно (только для коммутаторов, имеющих поддержку интерфейса командной строки CLI):

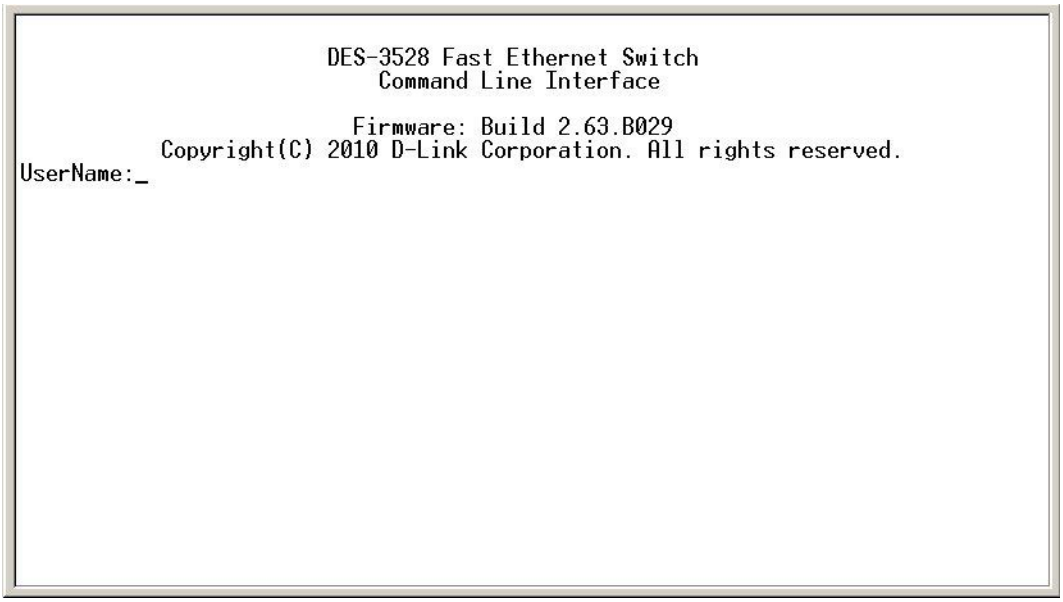


Рис. 2.3. Первоначальное окно консоли

Если окно не появилось, необходимо нажать Ctrl+r, чтобы его обновить.

Все управляемые коммутаторы обладают защитой от доступа неавторизованных пользователей, поэтому после загрузки устройства появится приглашение ввести имя пользователя и пароль. По умолчанию имя пользователя и пароль не определены, поэтому

необходимо дважды нажать клавишу Enter. После этого в командной строке появится следующее приглашение, например **DES-3528#**. Теперь можно вводить команды.

2.4 Начальная конфигурация коммутатора

2.4.1 Вызов помощи по командам

Существует большое количество команд CLI. Команды бывают сложные, многоуровневые, требующие ввода большого количества параметров, и простые, состоящие из одного параметра. Наберите в командной строке «?» и нажмите клавишу «Enter» для того, чтобы вывести на экран список всех команд данного уровня.

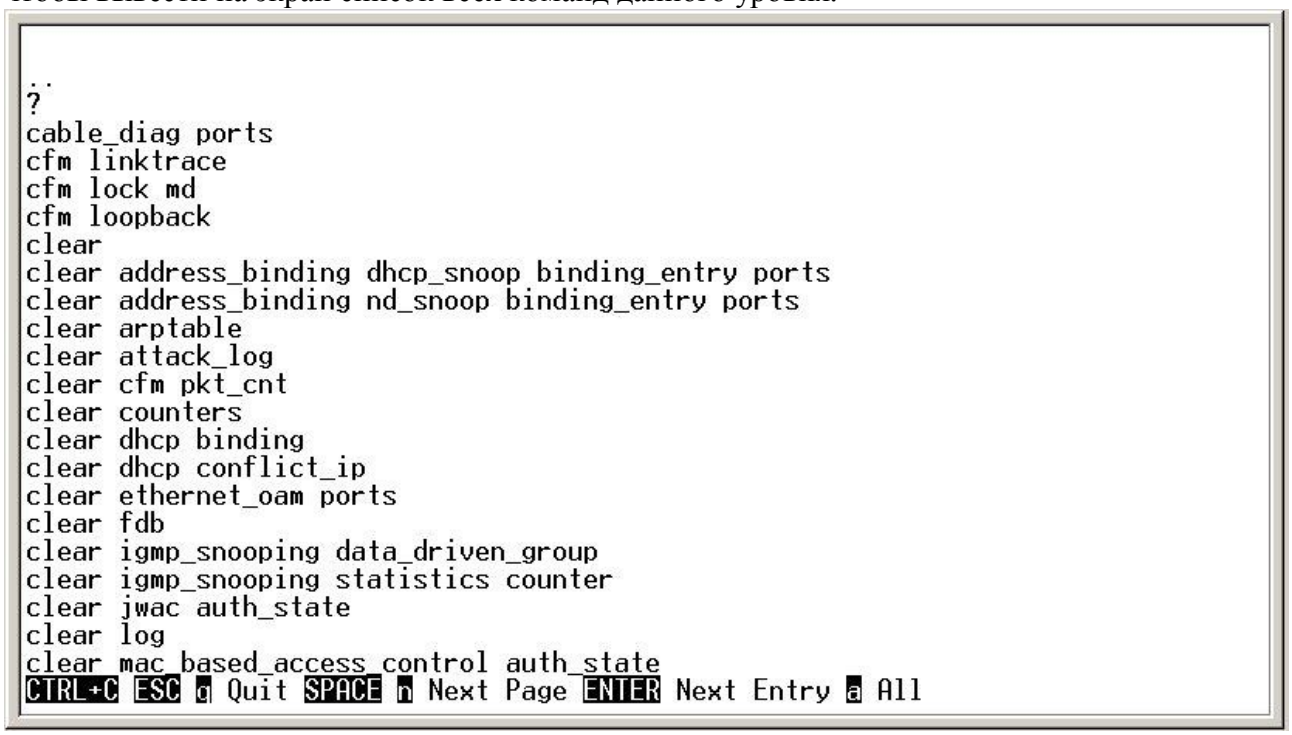


Рис. 2.4. Результат выполнения команды «?»

Используйте знак вопроса «?» так же в том случае, если вы не знаете параметров команды. Например, если надо узнать возможные варианты синтаксиса команды show, введите в командной строке:

DES-3528**#show** + пробел

Далее можно ввести «?» или нажать кнопку Enter. На экране появятся все возможные завершения команды. Также можно воспользоваться кнопкой TAB, которая будет последовательно выводить на экран все возможные завершения команды.

Внимание: при работе в CLI можно вводить сокращенный вариант команды. Например, если ввести команду «sh sw», то коммутатор интерпретирует эту команду как «show switch».

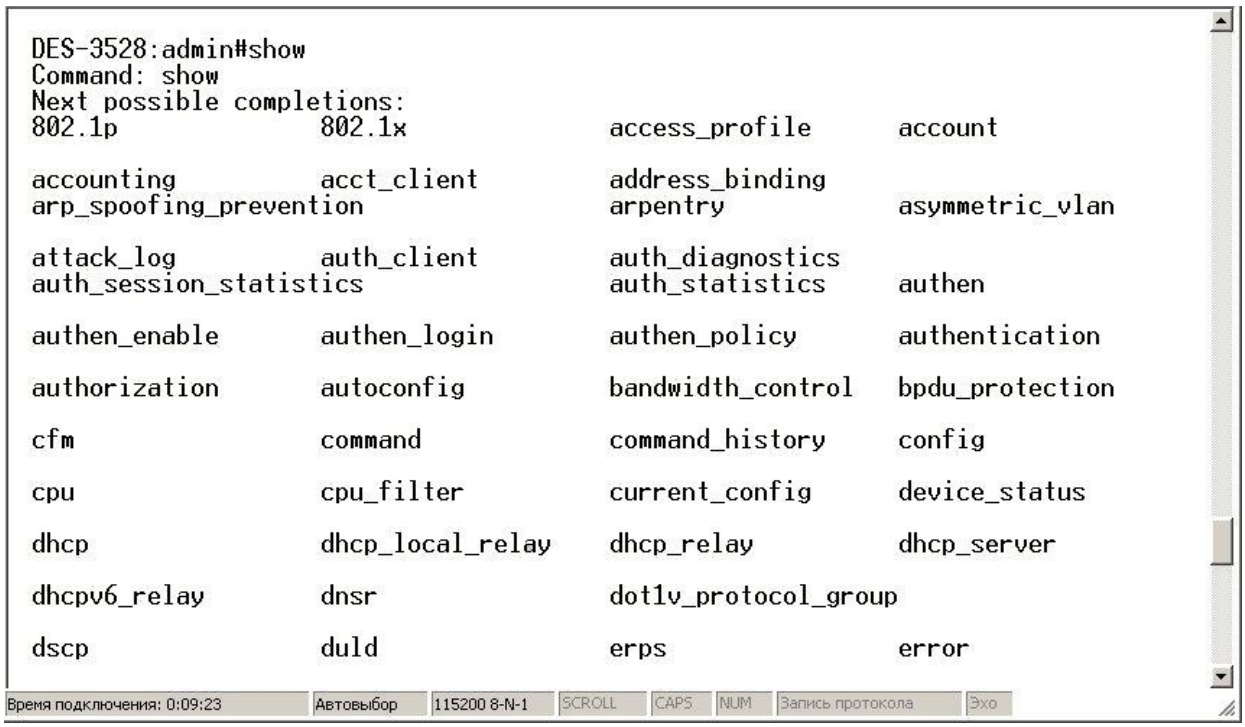


Рис. 2.5. Результат вызова помощи о возможных параметрах команды show

Внимание: далее в книге примеры настроек приведены для коммутаторов серии DES-3528, если не указано иное.

2.4.2 Базовая конфигурация коммутатора

Шаг 1. Обеспечение защиты коммутатора от доступа неавторизованных пользователей.

Самым первым шагом при создании конфигурации коммутатора является обеспечение его защиты от доступа неавторизованных пользователей. Самая простая форма безопасности – создание учетных записей для пользователей с соответствующими правами. Создавая учетную запись для пользователя, можно задать один из следующих уровней привилегий: *Admin*, *Operator* или *User*. Учетная запись *Admin* имеет наивысший уровень привилегий.

Создать учетную запись пользователя можно с помощью следующих команд CLI:

create account [admin | operator | user] <*username 15***>**

Далее появится приглашение для ввода пароля и подтверждения ввода:

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

Максимальная длина имени пользователя и пароля – 15 символов. На коммутаторе можно создать до 8 учетных записей пользователей. После успешного создания учетной записи на экране появится слово *Success*.

Внимание: все команды чувствительны к регистру. Перед вводом команды удостоверьтесь, что отключен Caps Lock или другие нежелательные функции, которые изменят регистр текста.

Ниже приведен пример создания учетной записи с уровнем привилегий «admin» и именем пользователя (Username) «dlink» на коммутаторе DES-3528:

```
DES-3528#create account admin dlink
Command: create account admin dlink
Enter a case-sensitive new password:****
Enter the new password again for confirmation:****
Success.
```

Изменить пароль для пользователя с существующей учетной записью, можно с помощью команды:

config account <*username***> {encrypt [plain_text | sha_1] <***password***>}**

Ниже приведен пример создания на коммутаторе DES-3528 нового пароля для учетной записи dlink:

```
DES-3528#config account dlink
Command: config account dlink
Enter a old password:****
Enter a case-sensitive new password:****
Enter the new password again for confirmation:****
Success
```

Проверить созданную учетную запись можно с помощью команды:

show account

Ниже приведен пример выполнения этой команды на коммутаторе DES-3528.

```
DES-3528#show account
Command: show account

Current Accounts:
Username            Access Level
---------------     ------------
dlink               Admin

Total Entries: 1
```

Удалить учетную запись можно, выполнив команду:

delete account <*username*>

Ниже приведен пример удаления учетной записи dlink на коммутаторе DES-3528.

```
DES-3528#delete account dlink
Command: delete account dlink
Are you sure to delete the last administrator account?(y/n)
```

```
Success.
```

Шаг 2. Настройка IP-адреса.

Для того чтобы коммутатором можно было удаленно управлять через Web-интерфейс или Telnet, ему необходимо назначить IP-адрес из адресного пространства сети, в которой планируется его использовать. IP-адрес может быть задан автоматически с помощью протоколов DHCP или BOOTP или статически, с помощью следующих команд CLI:

config ipif System dhcp

config ipif System ipaddress xxx.xxx.xxx.xxx/yyy.yyy.yyy.yyy

где xxx.xxx.xxx.xxx – IP-адрес, yyy.yyy.yyy.yyy – маска подсети, System – имя управляющего интерфейса коммутатора.

Ниже приведен пример использования команды присвоения IP-адреса управляющему интерфейсу на коммутаторе DES-3528:

```
DES-3528#config ipif System ipaddress
192.168.100.240/255.255.255.0
Command: config ipif System ipaddress 192.168.100.240/24
Success.
```

Шаг 3. Настройка параметров портов коммутатора.

По умолчанию порты всех коммутаторов D-Link поддерживают автоматическое определение скорости и режима работы (дуплекса). Но может возникнуть ситуация, в которой автоопределение будет действовать некорректно и потребуется ручная установка скорости и режима. В этом случае ручную настройку параметров необходимо выполнить на обоих концах канала связи.

Для установки параметров портов, таких как скорость передачи, дуплексный/полудуплексный режим работы, активизация/отключение управления потоком, изучение MAC-адресов, автоматическое определение полярности и т.д., на коммутаторах D-Link можно воспользоваться командой **config ports**.

Ниже приведен пример настройки параметров портов на коммутаторе DES-3528. Для портов 1, 2, 3 производятся следующие настройки: скорость передачи устанавливается равной 10 Мбит/с, активизируются дуплексный режим работы, изучение MAC-адресов, управление потоком, порты переводятся в состояние «включен».

```
DES-3528#config ports 1-3 speed 10_full learning enable state
enable flow_control enable
Command: config ports 1-3 speed 10_full learning enable state
enable flow_control enable
Success
```

Команда **show ports** <*список портов*> выведет на экран информацию о настройках портов коммутатора. Ниже показан результат выполнения команды show ports на коммутаторе DES-3528.

```
DES-3528#show ports 1-3
Command: show ports 1-3
Port State/    Settings                Connection              Address
```

```
     MDIX      Speed/Duplex/FlowCtrl    Speed/Duplex/FlowCtrl    Learning
---  ------    ---------------------    ---------------------    -----------
1   Enabled    10M/Full/Enabled         Link Down                Enabled
     Auto
2    Enabled   10M/Full/Enabled         Link Down                Enabled
     Auto
3    Enabled   10M/Full/Enabled         Link Down                Enabled
     Auto
```

Шаг 4. Сохранение текущей конфигурации коммутатора в энергонезависимую память NVRAM. Для этого необходимо выполнить команду **save**.

```
DES-3528#save

Command: save

Saving all settings to NV-RAM……Done
```

Внимание: активная конфигурация хранится в оперативной памяти SDRAM. При отключении питания, конфигурация, хранимая в этой памяти, будет потеряна. Для того чтобы сохранить конфигурацию в энергонезависимой памяти NVRAM, необходимо выполнить команду «save».

Шаг 5. Перезагрузка коммутатора с помощью команды reboot.

```
DES-3528#reboot

Command: reboot

Are you sure you want to proceed with the system reboot? (y/n)

Please wait, the switch is rebooting...
```

Сброс настроек коммутатора к заводским установкам выполняется с помощью команды:

reset {[config | system]} {force_agree}

Если в команде не будет указано никаких ключевых слов, то все параметры, за исключением IP-адреса, учетных записей пользователей и Log-файла, будут возвращены к заводским параметрам по умолчанию. Коммутатор не сохранит настройки в энергонезависимой памяти NVRAM и не перегрузится.

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

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.

Параметр **force_agree** позволяет произвести безусловное выполнение команды reset. Не нужно вводить «Y/N». На коммутаторе восстановятся все заводские настройки по умолчанию, исключая IP-адрес, учетные записи пользователей и журнал историй.

```
DES-3528#reset

Command: reset

Success
```

Шаг 6. Просмотр конфигурации коммутатора.

Получить информацию о коммутаторе (посмотреть его общую конфигурацию) можно с помощью команды **show switch**.

```
DES-3528#show switch
Command: show switch
Device Type        : DES-3528 Fast Ethernet Switch
MAC Address        : 00-1E-58-50-15-10
IP Address         : 192.168.100.241 (Manual)
VLAN Name          : default
Subnet Mask        : 255.255.255.0
Default Gateway    : 0.0.0.0
Boot PROM Version  : Build 1.00.B007
Firmware Version   : Build 2.20.B028
Hardware Version   : A1
Serial Number      : P1UM186000004
System Name        :
System Location    :
System Contact     :
Spanning Tree      : Disabled
GVRP               : Disabled
IGMP Snooping      : Disabled
MLD Snooping       : Disabled
VLAN Trunk         : Disabled
TELNET             : Enabled (TCP 23)
WEB                : Enabled (TCP 80)
SNMP               : Disabled
SSL Status         : Disabled
```

Команды «**Show**» являются удобным средством проверки состояния и параметров коммутатора, предоставляя информацию, требуемую для мониторинга и поиска неисправностей в работе коммутаторов. Ниже приведен список наиболее общих команд «Show».

show config	эта команда используется для отображения конфигурации, сохраненной в NV RAM или созданной в текущий момент
show fdb	эта команда используется для отображения текущей таблицы коммутации
show switch	эта команда используется для отображения общей информации о коммутаторе
show device_status	эта команда используется для отображения состояния внутреннего и внешнего питания коммутатора
show error ports	эта команда используется для отображения статистики об ошибках для заданного диапазона портов
show packet ports	эта команда используется для отображения статистики о переданных и полученных портом пакетах
show firmware information	эта команда используется для отображения информации о программном обеспечении коммутатора (прошивке)
show ipif	эта команда используется для отображения информации о настройках IP-интерфейса на коммутаторе
show log	эта команда используется для просмотра Log-файла коммутатора

2.5 Подключение к Web-интерфейсу управления коммутатора

Коммутаторы D-Link позволяют выполнять настройки через Web-интерфейс управления, который состоит из дружественного пользовательского графического интерфейса (GUI), запускающегося на клиенте, и HTTP-сервера, запускающегося на коммутаторе.

Web-интерфейс является альтернативой командной строки, обеспечивает графическое представление интерфейса управления коммутатора в режиме реального времени и предоставляет подробную информацию о состоянии портов, модулей, их типе и т.д.

Связь между клиентом и сервером обычно осуществляется через TCP/IP соединение с номером порта HTTP равным 80.

При первом подключении к HTTP-серверу на коммутаторе, необходимо выполнить следующие шаги:

1. Проверить, что IP-адрес компьютера, с которого осуществляется управление, принадлежит той же подсети, что и IP-адрес коммутатора, если в сети не настроена маршрутизация. На компьютере запустить Web-браузер, в адресной строке которого ввести IP-адрес интерфейса управления коммутатора по умолчанию (обычно он указывается в руководстве пользователя).
2. В появившемся окне аутентификации, поля User name и Password необходимо оставить пустыми и нажать кнопку OK. После этого появится окно Web-интерфейса управления коммутатора.

Если требуется изменить IP-адрес интерфейса управления на новый, то в случае использования управляемого коммутатора необходимо подключиться к его консольному порту и, используя интерфейс командной строки, выполнить следующую команду:

config ipif System ipaddress xxx.xxx.xxx.xxx/yyy.yyy.yyy.yyy,

где xxx.xxx.xxx.xxx – IP-адрес, yyy.yyy.yyy.yyy – маска подсети

Проверить правильность настройки IP-адреса коммутатора можно с помощью команды: **show ipif**

```
DES-3528#show ipif
Command: show ipif

IP Interface                  : System
VLAN Name                     : default
Interface Admin State         : Enabled
DHCPv6 Client State           : Disabled
Link Status                   : LinkUp
IPv4 Address                  : 192.168.100.241/24 (Manual)  Primary
Proxy ARP                     : Disabled   (Local : Disabled)
IPv4 State                    : Enabled
IPv6 State                    : Enabled

Total Entries: 1
```

При использовании настраиваемых коммутаторов, изменить IP-адрес на новый можно с помощью программы SmartConsole Utility, установленной на рабочую станцию, с которой осуществляется управление.

Условно пользовательский интерфейс можно разделить на 3 области, как показано на рис. 2.6. Область 1 содержит список папок, объединяющих семейство функций, предназначенных для выполнения той или иной задачи. Например, в папке Configuration находятся функции, предназначенные для выполнения базовой конфигурации коммутатора,

включая настройку IP-адреса, учетных записей пользователей, конфигурации портов и т.д. В папке L2 Features находятся функции 2 уровня, включая Jumbo Frame, 802.1Q VLAN, QinQ, 802.1v Protocol VLAN и т.д.

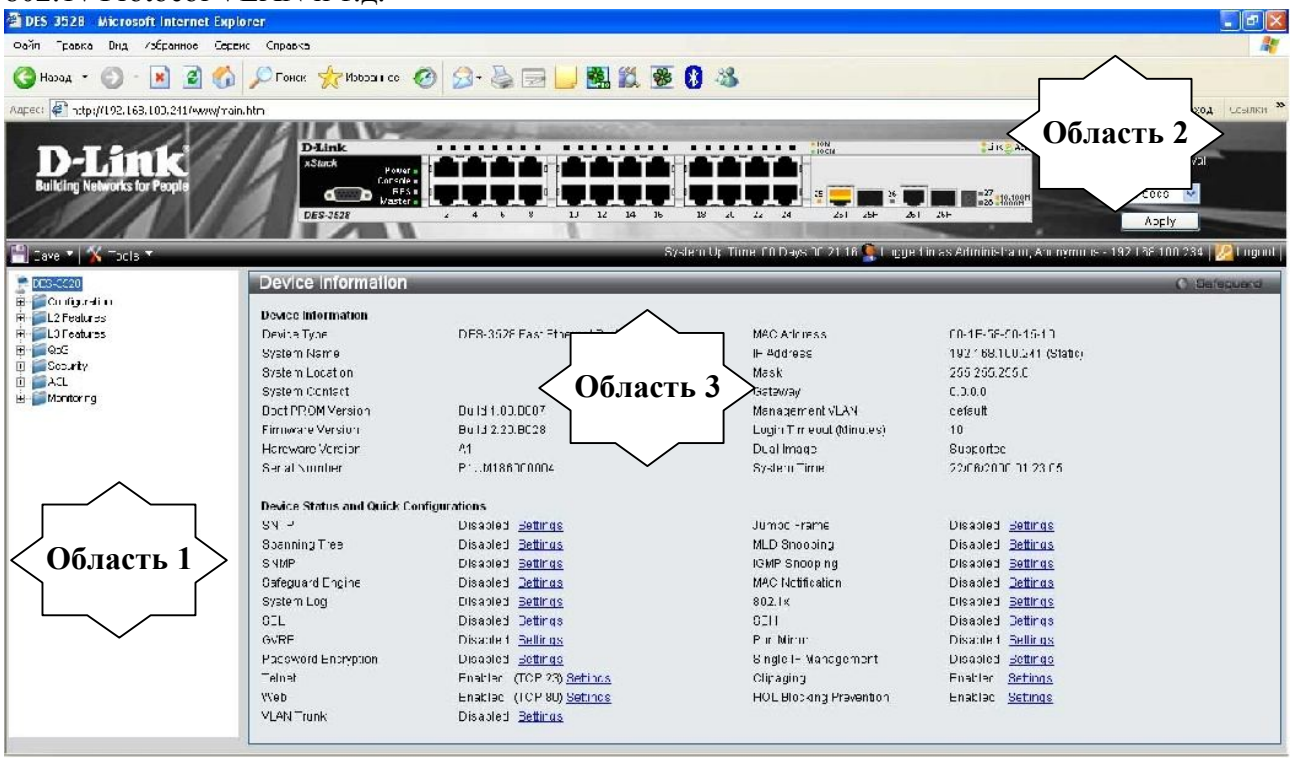


Рис. 2.6. Web-интерфейс управления

Если щелкнуть кнопкой мыши по одной из папок и выбрать необходимую функцию, то в области 3 Web-интерфейса появится окно, предназначенное для ввода и/или выбора данных.

Область 2 представляет собой графическое изображение передней панели коммутатора в режиме реального времени. Эта область отражает порты и модули расширения коммутатора, и их состояние.

2.6 Загрузка нового программного обеспечения в коммутатор

В результате добавления нового функционала или исправления ошибок, появляются новые версии программного обеспечения для коммутаторов D-Link, которые можно бесплатно загрузить с FTP-сервера компании ftp.dlink.ru.

Новое программное обеспечение загружается в коммутатор с помощью протокола TFTP (Trivial File Transfer Protocol). В рабочую папку, установленного на рабочую станцию сервера TFTP, необходимо поместить новое программное обеспечение. Сервер TFTP должен быть включен и находиться в той же IP-подсети, что и коммутатор, если в сети не настроена маршрутизация. В процессе обновления ПО нельзя выключать питание коммутатора.

Некоторые модели управляемых коммутаторов D-Link могут хранить в памяти две версии прошивки, что позволяет обеспечить работоспособность устройства в случае проблем с одной из них. Пользователи могут указать, какая из прошивок будет загружаться при старте коммутатора.

Для загрузки прошивки в коммутатор используется следующая команда (здесь приводится синтаксис коммутатора модели DES-3528; синтаксис команды в других моделях коммутаторов может отличаться):

download firmware_fromTFTP <*ipaddr***> <***path_filename 64***>**

{image_id <*int 1-2*>}

В качестве параметров команды надо указать IP-адрес сервера TFTP, путь к загружаемому файлу и его имя, например, C:\3528.had (можно не указывать полный путь к файлу, если он находится в рабочей директории TFTP-сервера), а также идентификатор загружаемой при старте прошивки. Например:

```
DES-3528#download firmware_fromTFTP 10.48.74.121 3528.had image_id 1
```

Посмотреть информацию о хранимых в памяти коммутатора прошивках можно с помощью команды:

show firmware information

```
DES-3528#show firmware information
Command: show firmware information

ID  Version    Size(B)  Update Time         From              User
--  --------   -------- ------------------ -------------  --------
1   1.00-T003  2103164  2000/01/02 01:21:21 10.90.90.11(R)   Anonymous
*2  1.03.B008  2317149  days 00:00:00        Serial Port(Prom) Unknown

'*' means boot up firmware
(R) means firmware update through Serial Port(RS232)
(T) means firmware update through TELNET
(S) means firmware update through SNMP
(W) means firmware update through WEB
(SSH) means firmware update through SSH
(SIM) means firmware update through Single IP Management
```

Для того чтобы изменить номер, загружаемой при старте коммутатора прошивки, необходимо выполнить команду:

config firmware image_id <*int 1-2*> boot_up

Удалить файл ПО коммутатора можно с помощью команды:

config firmware image_id <*int 1-2*> delete

2.7 Загрузка и резервное копирование конфигурации коммутатора

Управляемые коммутаторы позволяют осуществлять загрузку и резервное копирование конфигурации на TFTP-сервер.

Также как и в случае загрузки ПО, сервер TFTP должен быть включен и находиться в той же IP-подсети, что и коммутатор, если в сети не настроена маршрутизация.

Для загрузки конфигурации в коммутатор используется следующая команда:

download cfg_fromTFTP <*ipaddr*> <*path_filename 64*>

В качестве параметров команды надо указать IP-адрес сервера TFTP, путь к загружаемому файлу конфигурации, его имя.

Например:

```
DES-3528#download cfg_fromTFTP 10.48.74.121 /cfg/setting.txt
```

Для сохранения текущей конфигурации на сервере TFTP, необходимо выполнить команду:

upload cfg_toTFTP <*ipaddr***> <***path_filename 64***>**

Например:

```
DES-3528#upload cfg_toTFTP 10.48.74.121 /cfg/setting.txt
```

3. Обзор функциональных возможностей коммутаторов

Так как коммутатор представляет собой довольно сложное вычислительное устройство, имеющее несколько процессорных модулей, то, помимо выполнения основной функции передачи кадров с порта на порт по алгоритму моста, в нем реализованы дополнительные функции, полезные при построении современных, расширяемых, надежных и гибких сетей. Большинство современных коммутаторов, независимо от производителя, поддерживают множество дополнительных возможностей, отвечающих общепринятым стандартам. Среди них самые распространенные и наиболее используемые сегодня это:

- виртуальные локальные сети (VLAN);
- семейство протоколов Spanning Tree – IEEE 802.1D, 802.1w, 802.1s;
- статическое и динамическое по протоколу IEEE 802.3ad агрегирование каналов Ethernet;
- сегментация трафика;
- обеспечение качества обслуживания QoS;
- функции обеспечения безопасности, включая аутентификацию 802.1X, функции Port Security, IP-MAC-Port Binding и т.д.;
- протоколы поддержки Multicast-вещания;
- SNMP-управление и др.

4. Виртуальные локальные сети (VLAN)

Поскольку коммутатор Ethernet является устройством канального уровня, то в соответствии с логикой работы он будет рассылать широковещательные кадры через все порты (за исключением порта-приемника такого кадра). Хотя трафик с конкретными адресами (соединения «точка – точка») изолирован парой портов, широковещательные кадры передаются во всю сеть (на каждый порт). *Широковещательные кадры* – это кадры, передаваемые на все узлы сети. Они необходимы для работы многих сетевых протоколов, таких как ARP, BOOTP или DHCP. С их помощью рабочая станция оповещает другие компьютеры о своем появлении в сети. Так же рассылка широковещательных кадров может возникать из-за некорректно работающего сетевого адаптера. Широковещательные кадры могут привести к нерациональному использованию полосы пропускания, особенно в крупных сетях. Для того чтобы этого не происходило, важно ограничить область распространения широковещательного трафика (эта область называется *широковещательным доменом*) - организовать небольшие **широковещательные домены** или **виртуальные локальные сети (Virtual LAN, VLAN)**.

Виртуальной локальной сетью называется логическая группа узлов сети, трафик которой, в том числе и широковещательный, на канальном уровне полностью изолирован от других узлов сети. Это означает, что передача кадров между разными виртуальными сетями на основании MAC-адреса невозможна независимо от типа адреса – уникального, группового или широковещательного. В то же время, внутри виртуальной сети кадры передаются по технологии коммутации, то есть только на тот порт, который связан с адресом назначения кадра. Таким образом, с помощью виртуальных сетей решается проблема распространения широковещательных кадров и вызываемых ими следствий, которые могут развиться в широковещательные штормы и существенно снизить производительность сети.

VLAN обладают следующими преимуществами:

- Гибкость внедрения. VLAN являются эффективным способом группировки сетевых пользователей в виртуальные рабочие группы, несмотря на их физическое размещение в сети.
- VLAN обеспечивают возможность контроля широковещательных сообщений, что увеличивает полосу пропускания, доступную для пользователя.
- VLAN позволяют повысить безопасность сети, определив с помощью фильтров, настроенных на коммутаторе или маршрутизаторе, политику взаимодействия пользователей из разных виртуальных сетей.

Рассмотрим пример, показывающий эффективность использования логической сегментации сетей с помощью технологии VLAN при решении типовой задачи организации доступа в Интернет сотрудникам офиса, при условии изоляции трафика разных отделов.

Предположим, что в офисе имеется несколько кабинетов, в каждом из которых располагается небольшое количество сотрудников. Каждый кабинет представляет собой отдельную рабочую группу.

При стандартном подходе к решению задачи с помощью физической сегментации трафика каждого отдела, потребовалось бы в каждый кабинет устанавливать отдельный коммутатор, который бы подключался к маршрутизатору, предоставляющему подключение в Интернет. При этом маршрутизатор должен обладать достаточным количеством портов, обеспечивающим возможность подключения всех физических сегментов (кабинетов) сети. Данное решение плохо масштабируемо и дорогостоящее, т.к. при увеличении количества отделов, увеличивается количество необходимых коммутаторов, интерфейсов маршрутизатора и магистральных кабелей.

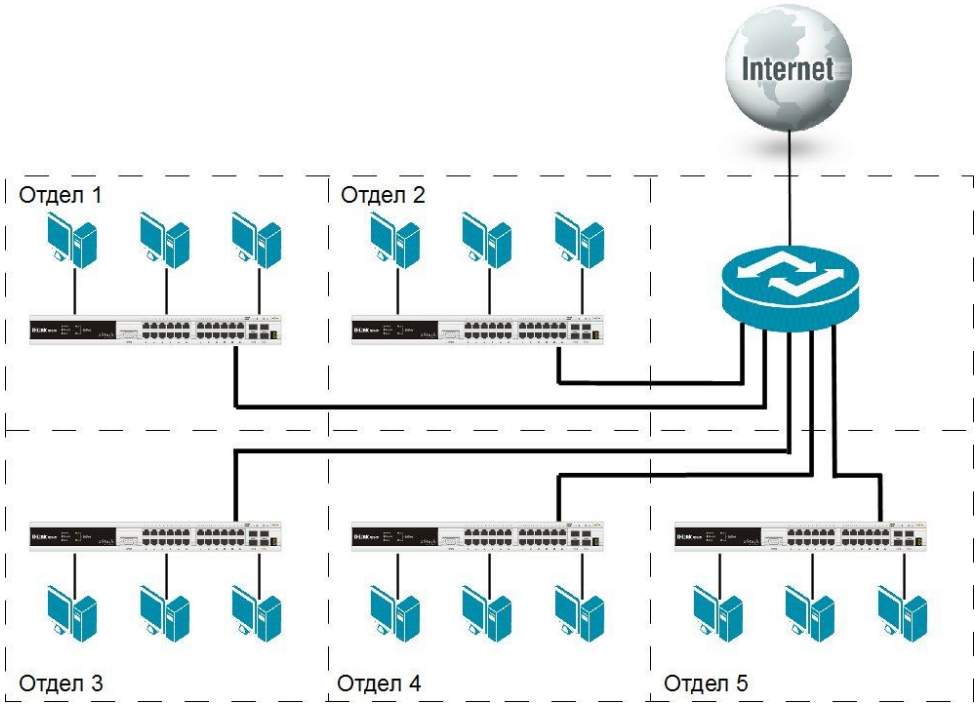


Рис. 4.1. Физическая сегментация сети

При использовании виртуальных локальных сетей уже не требуется подключать пользователей одного отдела к отдельному коммутатору, что позволяет сократить количество используемых устройств и магистральных кабелей. Коммутатор, программное обеспечение которого поддерживает функцию виртуальных локальных сетей, позволяет выполнять логическую сегментацию сети путем соответствующей программной настройки. Это позволяет подключать пользователей, находящихся в разных сегментах к одному коммутатору, а также сокращает количество необходимых физических интерфейсов на маршрутизаторе.

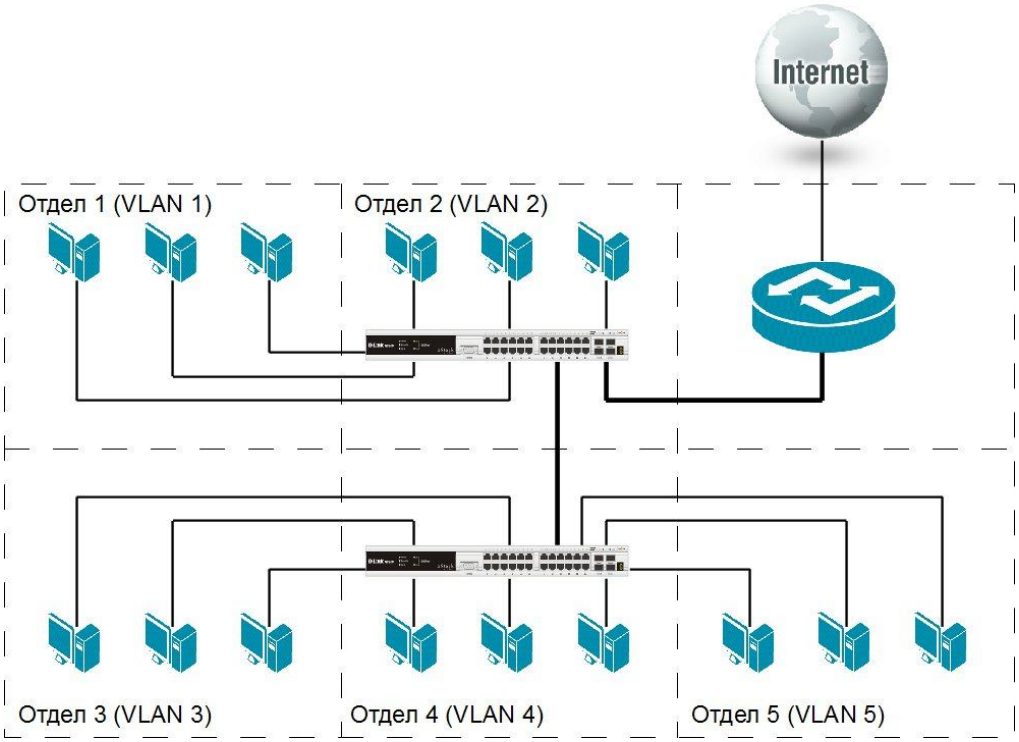


Рис. 4.2. Логическая группировка сетевых пользователей в VLAN

4.1 Типы VLAN

В коммутаторах могут быть реализованы следующие типы VLAN:

- на основе портов;
- на основе стандарта IEEE 802.1Q;
- на основе стандарта IEEE 802.1ad (Q-in-Q VLAN);
- на основе портов и протоколов IEEE 802.1v;
- на основе MAC-адресов;
- асимметричные.

Также для сегментирования сети на канальном уровне модели OSI в коммутаторах могут использоваться другие функции, например функция *Traffic Segmentation*.

4.2 VLAN на основе портов

При использовании VLAN на основе портов (Port-based VLAN), каждый порт назначается в определенную VLAN, независимо от того, какой пользователь или компьютер подключен к этому порту. Это означает, что все пользователи, подключенные к этому порту, будут членами одной VLAN. Конфигурация портов статическая и может быть изменена только вручную.

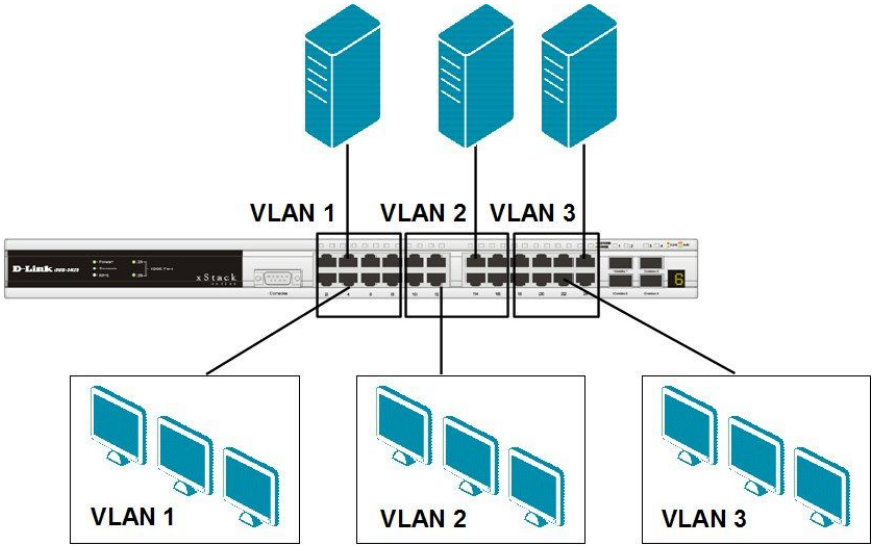


Рис. 4.3. VLAN на основе портов

Основные характеристики VLAN на основе портов:

1. Применяются в пределах одного коммутатора. Если необходимо организовать несколько рабочих групп в пределах небольшой сети на основе одного коммутатора, например, необходимо разнести технический отдел и отдел продаж, то решение VLAN на базе портов оптимально подходит для данной задачи.
2. Простота настройки. Создание виртуальных сетей на основе группирования портов не требует от администратора большого объема ручной работы – достаточно всем портам, помещаемым в одну VLAN, присвоить одинаковый идентификатор VLAN (VLAN ID) .
3. Возможность изменения логической топологии сети без физического перемещения станций. Достаточно всего лишь изменить настройки порта, с одной VLAN (например, VLAN технического отдела) на другую (VLAN отдела продаж) и рабочая станция сразу же получает возможность совместно использовать ресурсы с членами

новой VLAN. Таким образом, VLAN обеспечивают гибкость при перемещениях, изменениях и наращивании сети.

4. Каждый порт может входить только в одну VLAN. Для объединения виртуальных подсетей как внутри одного коммутатора, так и между двумя коммутаторами, нужно использовать сетевой уровень OSI-модели. Один из портов каждой VLAN подключается к интерфейсу маршрутизатора, который создает таблицу маршрутизации для пересылки кадров из одной подсети (VLAN) в другую (IP-адреса подсетей должны быть разными).

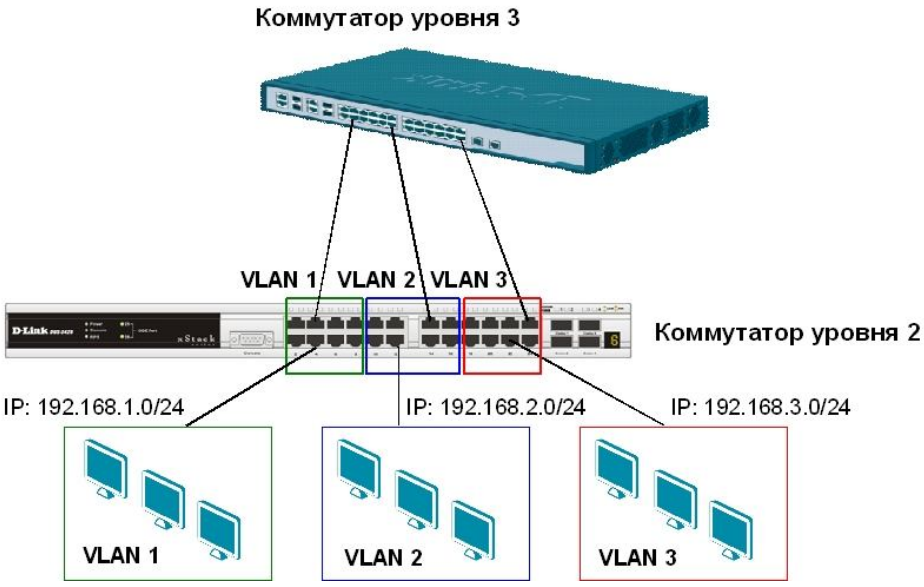


Рис. 4.4. Объединение VLAN с помощью маршрутизирующего устройства

Недостатком такого решения является то, что один порт каждой VLAN необходимо подключать к маршрутизатору. Это приводит к дополнительным расходам на покупку кабелей и маршрутизатор, а также порты коммутатора используются очень расточительно. Решить данную проблему можно двумя способами: использовать коммутаторы, которые на основе фирменного решения позволяют включать порт в несколько VLAN, или использовать коммутаторы уровня 3.

4.3 VLAN на основе стандарта IEEE 802.1Q

Построение VLAN на основе портов основано только на добавлении дополнительной информации к адресным таблицам коммутатора и не использует возможности встраивания информации о принадлежности к виртуальной сети в передаваемый кадр. Виртуальные локальные сети, построенные на основе стандарта IEEE 802.1Q, используют дополнительные поля кадра для хранения информации о принадлежности к VLAN при его перемещении по сети. С точки зрения удобства и гибкости настроек, VLAN стандарта IEEE 802.1Q является лучшим решением, по сравнению с VLAN на основе портов. Его основные преимущества:

1. Гибкость и удобство в настройке и изменении – можно создавать необходимые комбинации VLAN как в пределах одного коммутатора, так и во всей сети, построенной на коммутаторах с поддержкой стандарта IEEE 802.1Q. Способность добавления тегов позволяет информации о VLAN распространяться через множество 802.1Q-совместимых коммутаторов по одному физическому соединению (*магистральному каналу, Trunk Link*).
2. Позволяет активизировать алгоритм связующего дерева (Spanning Tree) на всех портах и работать в обычном режиме. Протокол Spanning Tree оказывается весьма

полезным для применения в крупных сетях, построенных на нескольких коммутаторах, и позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой. Для нормальной работы коммутатора требуется отсутствие замкнутых маршрутов в сети. Эти маршруты могут создаваться администратором специально для образования резервных связей или же возникать случайным образом, что вполне возможно, если сеть имеет многочисленные связи, а кабельная система плохо структурирована или документирована. С помощью протокола Spanning Tree коммутаторы после построения схемы сети блокируют избыточные маршруты. Таким образом, автоматически предотвращается возникновение петель в сети.

3. Способность VLAN IEEE 802.1Q добавлять и извлекать теги из заголовков кадров позволяет использовать в сети коммутаторы и сетевые устройства, которые не поддерживают стандарт IEEE 802.1Q.
4. Устройства разных производителей, поддерживающие стандарт, могут работать вместе, не зависимо от какого-либо фирменного решения.
5. Чтобы связать подсети на сетевом уровне, необходим маршрутизатор или коммутатор L3. Однако для более простых случаев, например, для организации доступа к серверу из различных VLAN, маршрутизатор не потребуется. Нужно включить порт коммутатора, к которому подключен сервер, во все подсети, а сетевой адаптер сервера должен поддерживать стандарт IEEE 802.1Q.

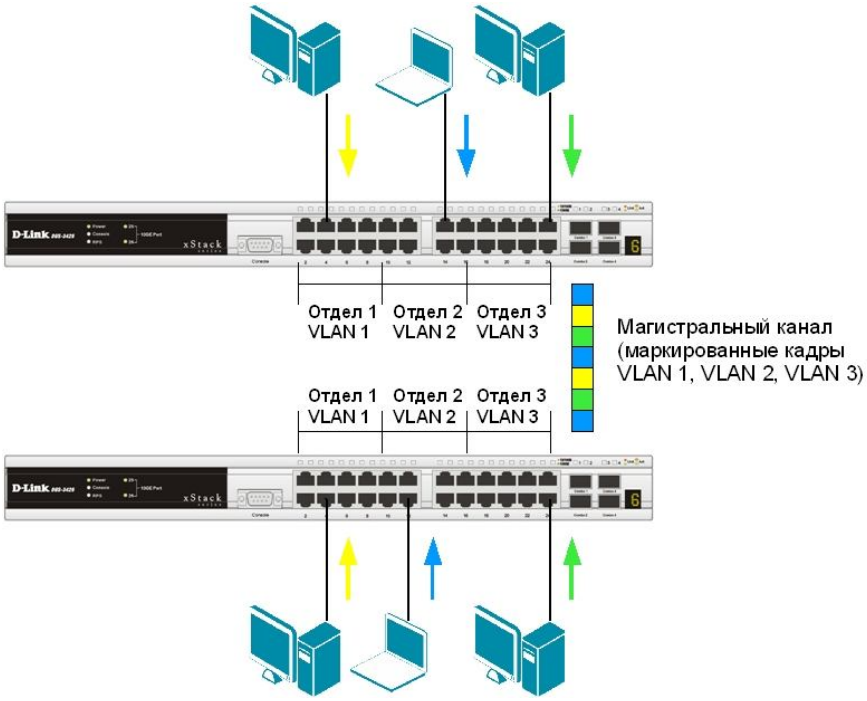


Рис. 4.5. Передача кадров разных VLAN по магистральному каналу связи

4.3.1 Некоторые определения IEEE 802.1Q

- **Tagging (Маркировка кадра)** – процесс добавления информации о принадлежности к 802.1Q VLAN в заголовок кадра.
- **Untagging (Извлечение тега из кадра)** – процесс извлечения информации о принадлежности к 802.1Q VLAN из заголовка кадра.
- **VLAN ID (VID)** – идентификатор VLAN.
- **Port VLAN ID (PVID)** – идентификатор порта VLAN.
- **Ingress port (Входной порт)** – порт коммутатора, на который поступают кадры, и при этом принимается решение о принадлежности к VLAN.

- **Egress port (Выходной порт)** – порт коммутатора, с которого кадры передаются на другие сетевые устройства – коммутаторы или рабочие станции, и, соответственно, на нем должно приниматься решение о маркировке.

Любой порт коммутатора может быть настроен как *tagged* (маркированный) или как *untagged* (немаркированный). Функция *untagging* позволяет работать с теми сетевыми устройствами виртуальной сети, которые не понимают тегов в заголовке кадра Ethernet. Функция *tagging* позволяет настраивать VLAN между несколькими коммутаторами, поддерживающими стандарт IEEE 802.1Q.

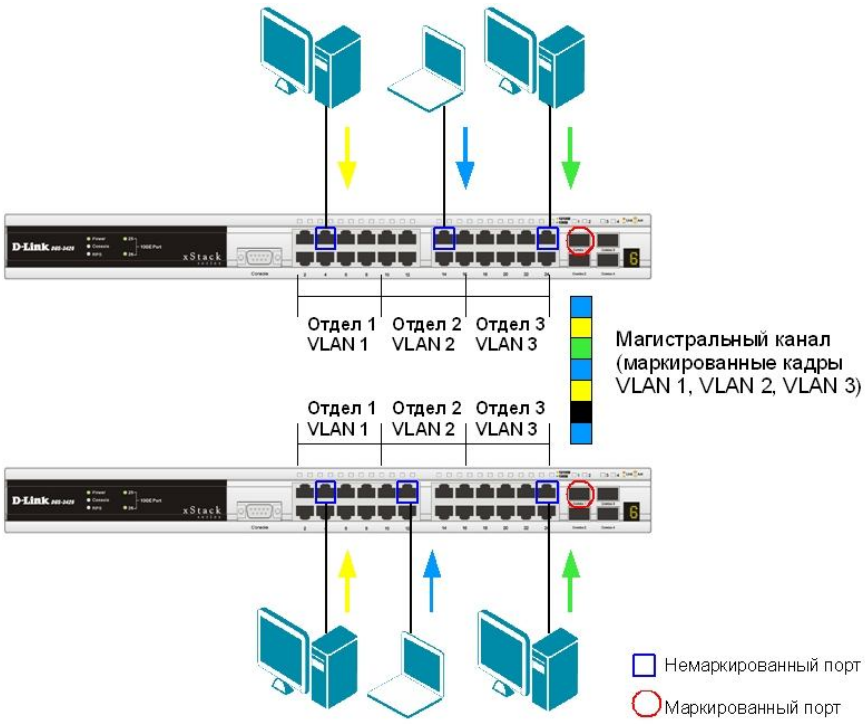


Рис. 4.6. Маркированные и немаркированные порты VLAN

4.3.2 Тег VLAN IEEE 802.1Q

Стандарт IEEE 802.1Q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о VLAN по сети. На рис. 4.7 изображен формат тега 802.1Q VLAN. К кадру Ethernet добавлены 32 бита (4 байта), которые увеличивают его размер до 1522 байт. Первые 2 байта (поле Tag Protocol Identifier, TPID) с фиксированным значение 0x8100 определяют, что кадр содержит тег протокола 802.1Q. Остальные 2 байта содержат следующую информацию:

- Priority (Приоритет) – 3 бита поля приоритета передачи кодируют до восьми уровней приоритета (от 0 до 7, где 7 – наивысший приоритет), которые используются в стандарте 802.1p;
- Canonical Format Indicator (CFI) – 1 бит индикатора канонического формата зарезервирован для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet;
- VID (VLAN ID) – 12-ти битный идентификатор VLAN определяет какой VLAN принадлежит трафик. Поскольку под поле VID отведено 12 бит, то можно задать 4094 уникальных VLAN (VID 0 и VID 4095 зарезервированы).

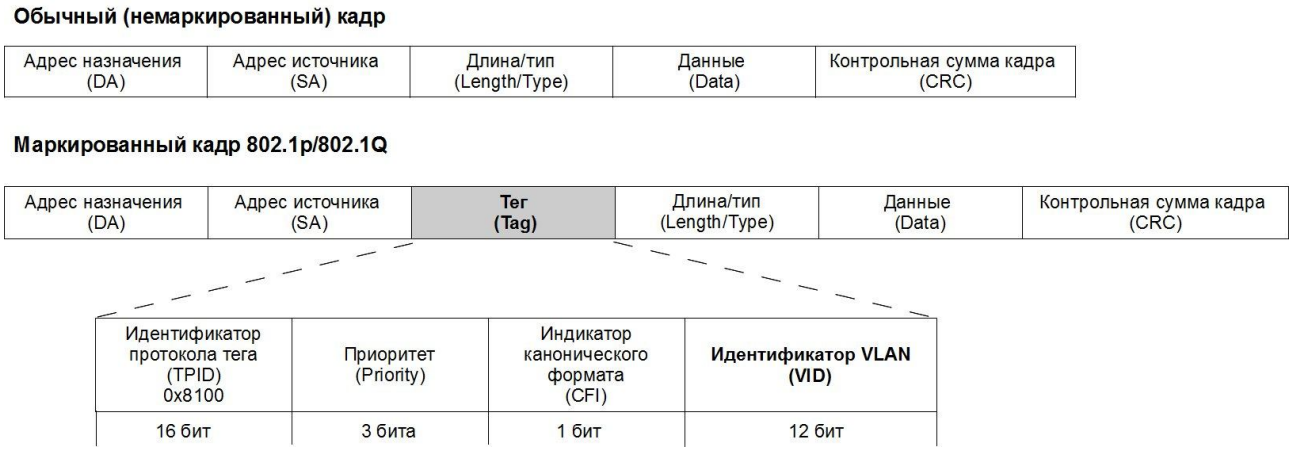


Рис. 4.7. Маркированный кадр Ethernet

4.3.3 Port VLAN ID

Каждый физический порт коммутатора имеет параметр, называемый *идентификатор порта VLAN* (*PVID*). Этот параметр используется для того, чтобы определить, в какую VLAN коммутатор направит входящий немаркированный кадр с подключенного к порту сегмента, когда кадр нужно передать на другой порт (внутри коммутатора в заголовки всех *немаркированных кадров* добавляется идентификатор VID равный *PVID* порта, на который они были приняты). Этот механизм позволяет одновременно существовать в одной сети устройствам с поддержкой и без поддержки стандарта IEEE 802.1Q.

Коммутаторы, поддерживающие протокол IEEE 802.1Q, должны хранить таблицу, связывающую идентификаторы портов PVID с идентификаторами VID сети. При этом каждый порт такого коммутатора может иметь только один PVID и столько идентификаторов VID, сколько поддерживает данная модель коммутатора.

Если на коммутаторе не настроены VLAN, то все порты по умолчанию входят в одну VLAN с PVID = 1.

4.3.4 Продвижение кадров VLAN IEEE 802.1Q

Решение о продвижении кадра внутри виртуальной локальной сети принимается на основе трех следующих видов правил:

- Правила входящего трафика (*ingress rules*) – классификация получаемых кадров относительно принадлежности к VLAN.
- Правила продвижения между портами (*forwarding rules*) – принятие решения о продвижении или отбрасывании кадра.
- Правила исходящего трафика (*egress rules*) –принятие решения о сохранении или удалении в заголовке кадра тега 802.1Q перед его передачей.

Правила входящего трафика выполняют классификацию каждого получаемого кадра относительно принадлежности к определенной VLAN, а также могут служить для принятия решения о приеме кадра для дальнейшей обработки или его отбрасывании на основе классификации и формата принятого кадра.

Классификация кадра по принадлежности VLAN осуществляется следующим образом:

а) Если кадр не содержит информацию о VLAN (*немаркированный кадр*), то в его заголовок коммутатор добавляет тег с идентификатором VID, равным идентификатору *PVID* порта, через который этот кадр был принят.

б) Если кадр содержит информацию о VLAN (*маркированный кадр*), то его принадлежность к конкретной VLAN определяется по идентификатору VID в заголовке кадра. Значение тега в нем не изменяется.

Активизировав функцию проверки формата кадра на входе, администратор сети может указать, кадры каких форматов будут приниматься коммутатором для дальнейшей обработки. Управляемые коммутаторы D-Link позволяют настраивать прием портами либо только маркированных кадров (*tagged_only*), либо обоих типов кадров – маркированных и немаркированных (*admit_all*).

Внимание: внутри коммутатора все кадры являются маркированными.

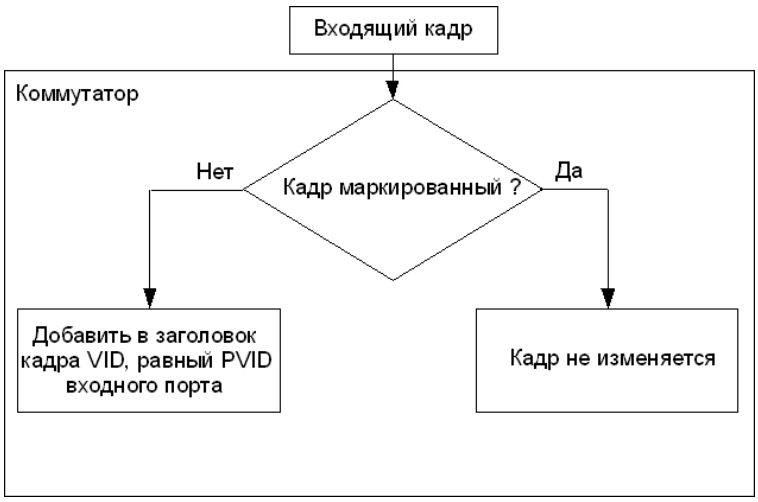


Рис. 4.8. Правила входящего трафика

Правила продвижения между портами осуществляют принятие решения об отбрасывании или передаче кадра на порт назначения на основе его информации о принадлежности конкретной VLAN и MAC-адреса узла-приемника.

Если входящий кадр маркированный, то коммутатор определяет, является ли входной порт членом той же VLAN путем сравнения идентификатора VID в заголовке кадра и набора идентификаторов VID, ассоциированных с портом, включая его PVID. Если нет, то кадр отбрасывается. Этот процесс называется *ingress filtering* (входной фильтрацией) и используется для сохранения пропускной способности внутри коммутатора путем отбрасывания кадров, не принадлежащих той же VLAN, что и входной порт, на стадии их приема.

Если кадр немаркированный, входная фильтрация не выполняется.

Далее определяется, является ли порт назначения членом той же VLAN. Если нет, то кадр отбрасывается. Если же выходной порт входит в данную VLAN, то коммутатор передает кадр в подключенный к нему сегмент сети.

Правила исходящего трафика определяют формат исходящего кадра – маркированный или немаркированный. Если выходной порт является немаркированным (*untagged*), то он будет извлекать тег 802.1Q из заголовков всех выходящих через него маркированных кадров. Если выходной порт настроен как маркированный (*tagged*), то он будет сохранять тег 802.1Q в заголовках всех выходящих через него маркированных кадров.

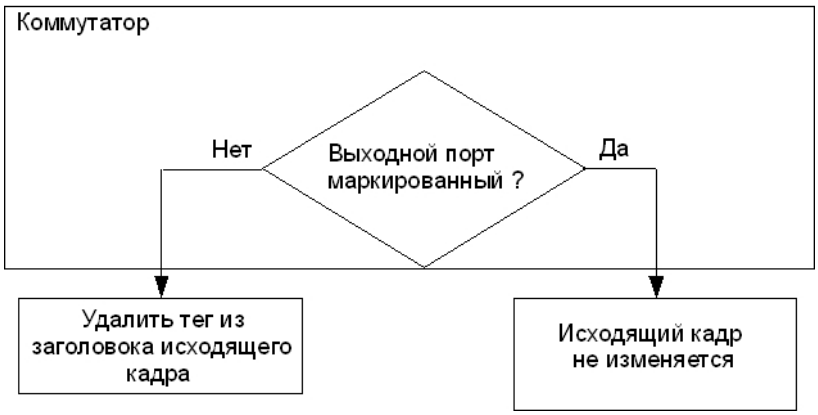


Рис. 4.9 Правила исходящего трафика

На рис. 4.10–4.13 приведен пример передачи немаркированного и маркированного кадра через маркированный и немаркированный порты коммутатора.

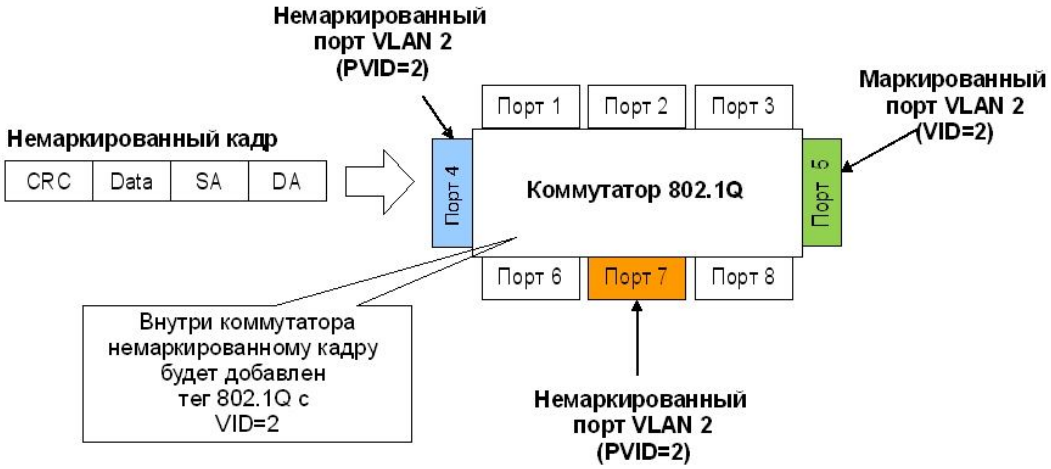


Рис. 4.10. Входящий немаркированный кадр

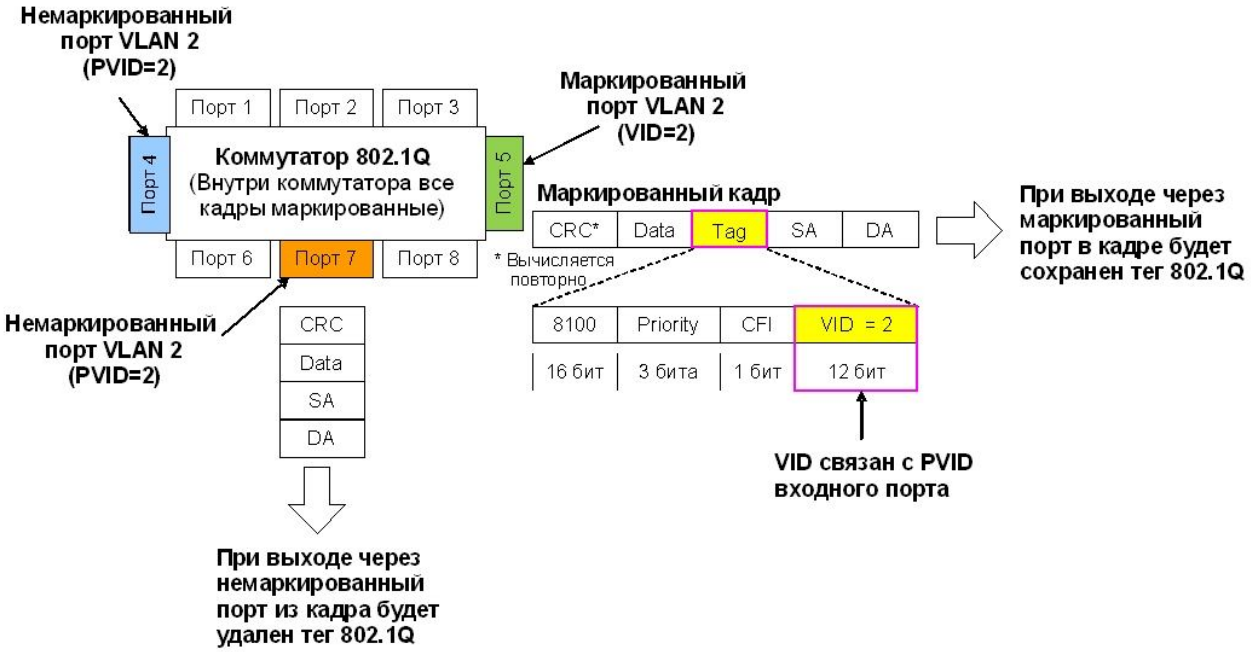


Рис. 4.11. Немаркированный кадр, передаваемый через маркированный и немаркированный порты

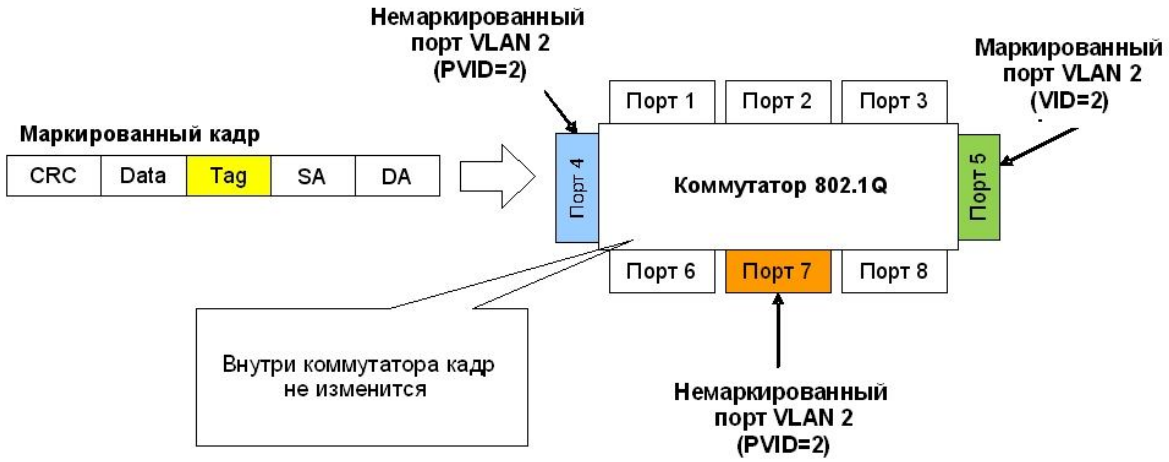


Рис. 4.12. Входящий маркированный кадр

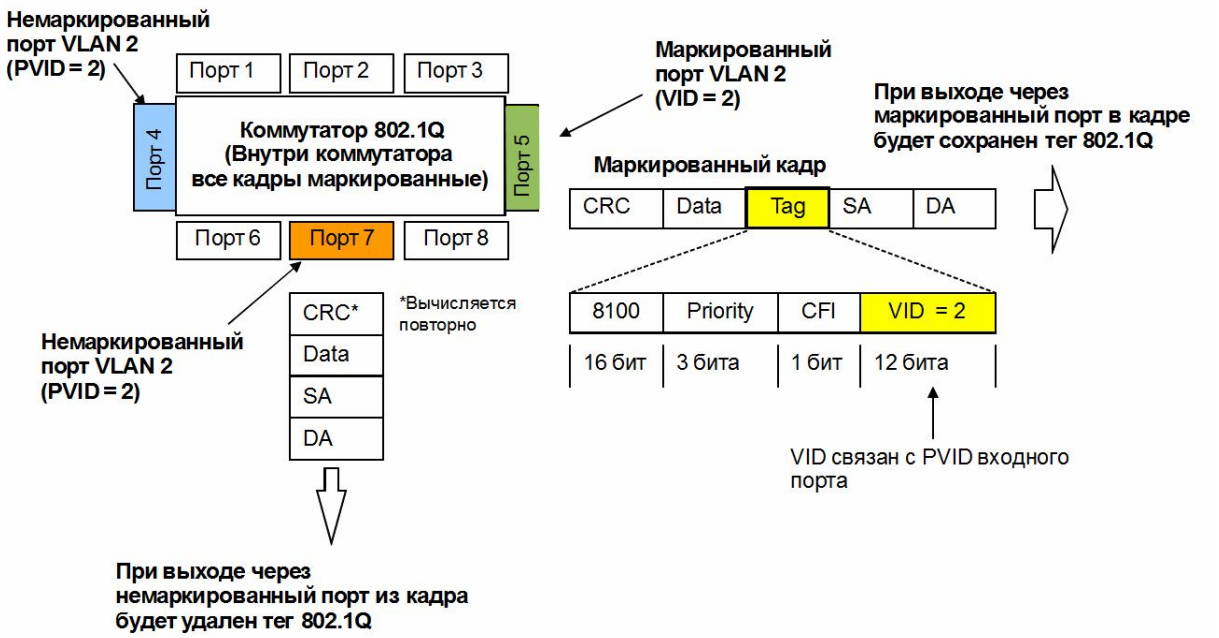


Рис. 4.13. Маркированный кадр, передаваемый через маркированный и немаркированный порты

4.3.5 Пример настройки VLAN IEEE 802.1Q

На рис. 4.14 показана схема сети, состоящая из двух групп VLAN. В качестве примера передачи данных между устройствами одной VLAN, построенной на нескольких коммутаторах, рассмотрим пересылку кадра с порта 5 коммутатора 1 на порт 6 коммутатора 3.

- Порт 5 коммутатора 1 является немаркированным портом VLAN v2 (PVID=2). Поэтому, когда любой немаркированный кадр поступает на порт 5, коммутатор снабжает его тегом 802.1Q со значением VID равным 2.
- Далее коммутатор 1 проверяет в своей таблице коммутации, через какой порт необходимо передать кадр и принадлежит ли этот порт VLAN v2. Кадр может быть передан через порт 1, т.к. он является маркированным членом VLAN v2. После передачи кадра через порт 1 тег 802.1Q в нем будет сохранен.
- После этого маркированный кадр поступит на порт 1 коммутатора 2. Прежде чем передать кадр дальше, порт 1 проверит, является ли он сам членом VLAN v2. Поскольку порт 1 коммутатора 2 является маркированным членом VLAN v2, он примет кадр и передаст его на порт 2, согласно таблице коммутации. После передачи кадра через порт 2 коммутатора 2 тег 802.1Q в нем будет сохранен, т.к. порт 2 является маркированным портом VLAN v2.
- Порт 1 коммутатора 3 примет поступивший кадр. После проверки на принадлежность к VLAN, порт 1 передаст кадр на порт 6, найденный обычным образом в таблице коммутации коммутатора 3. Порт 6 является немаркированным портом VLAN v2, поэтому при выходе кадра через этот порт, тег 802.1Q из него будет удален.

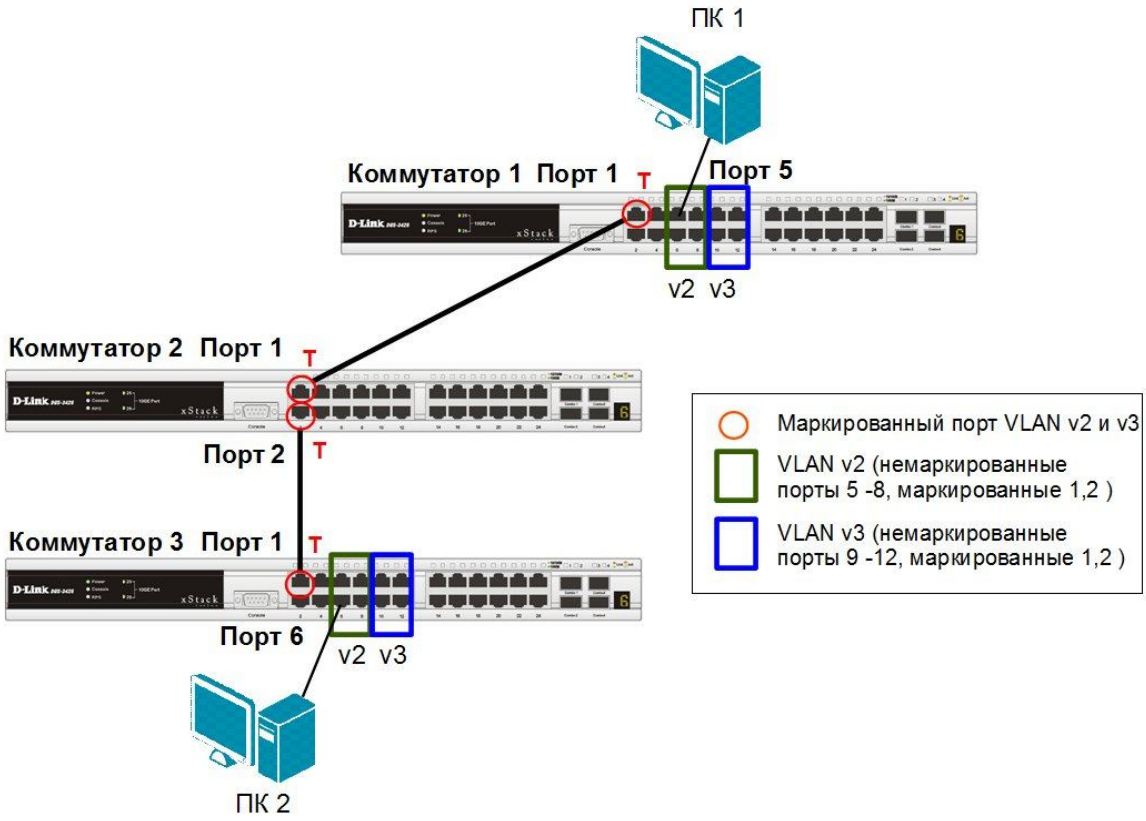


Рис. 4.14. Схема сети VLAN

Ниже приведен пример настройки коммутаторов, позволяющий реализовать заданную схему сети VLAN.

Настройка коммутатора 1

- Удалить соответствующие порты из VLAN по умолчанию (default VLAN) и создать новые VLAN.

config vlan default delete 1-12
create vlan v2 tag 2
create vlan v3 tag 3

- В созданные VLAN добавить порты, для которых необходимо указать, какие из них являются маркированными и немаркированными.

config vlan v2 add untagged 5-8
config vlan v2 add tagged 1-2
config vlan v3 add untagged 9-12
config vlan v3 add tagged 1-2

Настройка коммутатора 2

config vlan default delete 1-2
create vlan v2 tag 2
create vlan v3 tag 3
config vlan v2 add tagged 1-2
config vlan v3 add tagged 1-2

Настройка коммутаторов 3

config vlan default delete 1-12
create vlan v2 tag 2
create vlan v3 tag 3

config vlan v2 add untagged 5-8
config vlan v2 add tagged 1
config vlan v3 add untagged 9-12
config vlan v3 add tagged 1

Внимание: заводские установки по умолчанию назначают все порты коммутатора в default VLAN с VID = 1. **Перед созданием новой VLAN необходимо удалить из default VLAN все порты, которые требуется сделать немаркированными членами новой VLAN.**

4.4 Статические и динамические VLAN

Для корректной работы виртуальной локальной сети требуется, чтобы в базе данных фильтрации (*Filtering Database*) содержалась информация о членстве в VLAN. Эта информация необходима для принятия правильного решения (переслать или отбросить) при передаче кадров между портами коммутатора.

Существуют два основных способа, позволяющих устанавливать членство в VLAN:

- статические VLAN;
- динамические VLAN.

В статических VLAN установление членства осуществляется вручную администратором сети. При изменении топологии сети или перемещении пользователя на другое рабочее место, администратору требуется вручную выполнять привязку порт-VLAN для каждого нового соединения.

Членство в динамических VLAN может устанавливаться динамически на магистральных интерфейсах коммутаторов на основе протокола GVRP (GARP VLAN Registration Protocol). Протокол GARP (Generic Attribute Registration Protocol) используется для регистрации и отмены регистрации атрибутов, таких как VID.

Статические записи о регистрации в VLAN (*Static VLAN Registration Entries*) используются для представления информации о статических VLAN в базе данных фильтрации. Эти записи позволяют задавать точные настройки для каждого порта VLAN: идентификатор VLAN, тип порта (маркированный или немаркированный), один из управляющих элементов протокола GVRP:

– Fixed (порт всегда является членом данной VLAN);
– Forbidden (порту запрещено регистрироваться как члену данной VLAN);
– Normal (обычная регистрация с помощью протокола GVRP).

Управляющие элементы GVRP используются для активизации работы протокола на портах коммутатора, а также для указания того, может ли данная VLAN быть зарегистрирована на порте.

Динамические записи о регистрации в VLAN (*Dynamic VLAN Registration Entries*) используются для представления в базе данных фильтрации информации о портах, членство в VLAN которых установлено динамически. Эти записи создаются, обновляются и удаляются в процессе работы протокола GVRP.

4.5 Протокол GVRP

Протокол GVRP определяет способ, посредством которого коммутаторы обмениваются информацией о сети VLAN, чтобы автоматически зарегистрировать членов VLAN на портах во всей сети. Он позволяет динамически создавать и удалять VLAN стандарта IEEE 802.1Q на магистральных портах, автоматически регистрировать и исключать атрибуты VLAN (под регистрацией VLAN подразумевается включение порта в VLAN, под исключением – удаление порта из VLAN).

Протокол GVRP использует сообщения GVRP BPDU (GVRP Bridge Protocol Data Units), рассылаемые на многоадресный MAC-адрес 01-80-C2-00-00-21 для оповещения

устройств-подписчиков о различных событиях. Оповещения (*advertisement*) могут содержать информацию о выполнении следующих действий:

- **Join message** – регистрация порта в VLAN.
  JoinEmpty: VLAN на локальном подписчике не настроена;
  JoinIn: VLAN на локальном подписчике зарегистрирована.
- **Leave message** – удаление VLAN с конкретного порта.
  LeaveEmpty: VLAN на локальном подписчике не настроена;
  LeaveIn: VLAN на локальном подписчике удалена.
- **Leave message** – удаление всех, зарегистрированных на порте VLAN. Это сообщение отправляется после того, как истечет время, заданное таймером LeaveAll Timer.
- **Empty message** – требование повторного динамического оповещения и статической настройки VLAN.

4.5.1 Таймеры GVRP

- **Join Timer** – время в миллисекундах (100-100000), через которое отправляются сообщения JoinIn или JoinEmpty. Определяет промежуток времени между моментом получения коммутатором информации о вступлении в VLAN и фактическим моментом вступления в VLAN. По умолчанию установлено значение 200 миллисекунд.
- **Leave Timer** – когда коммутатор получает сообщение об исключении порта из VLAN (Leave message) от другого подписчика GVRP, он ожидает заданный период времени (от 100 до 100000 миллисекунд), определяемый таймером Leave Timer, чтобы убедиться, что информация о данной VLAN больше не существует в сети. Например, когда коммутатор получает сообщение Leave, он не удаляет мгновенно информацию о соответствующей VLAN, а запускает Leave Timer и ждет, когда его время истечет. Если за это время не будет получено сообщение JoinIn с информацией об удаляемой VLAN, то она будет коммутатором удалена. Обычно, значение таймера Leave Timer устанавливают в два раза больше значения таймера Join Timer. По умолчанию значение таймера равно 600 миллисекунд.
- **LeaveAll Timer** – интервал времени в миллисекундах (100-100000), через который отправляется сообщение LeaveAll. Когда коммутатор-подписчик GVRP получает это сообщение, он перезапускает все таймеры, включая LeaveAll Timer. Обычно значение таймера LeaveAll устанавливают в два раза больше значения таймера Leave Timer. По умолчанию значение таймера равно 10000 миллисекунд.

На рис. 4.15 показан процесс распространения информации о VLAN по сети с использованием протокола GVRP. На коммутаторе 1 созданы статические виртуальные сети VLAN v10, v20 и v30. Порт 25 является маркированным членом всех VLAN. Коммутатор 1 отправляет оповещение о VLAN v30 через порт 25 коммутатору 2 (сообщение JoinEmpty). Коммутатор 2 получает это оповещение, динамически создает VLAN v30 и включает в нее порт 25. Порт 26 коммутатора 2 отправляет оповещение о VLAN v30 коммутатору 3 (сообщение JoinEmpty), но сам не становится членом этой VLAN.

Коммутатор 3 получает оповещение, динамически создает VLAN v30 и включает в нее порт 26. Далее коммутатор 3 изменяет состояние VLAN v30 с динамического на статическое и отправляет через порт 26 сообщение JoinIn о регистрации виртуальной сети. Коммутатор 2 получает это оповещение и регистрирует порт 26 в VLAN v30, которая уже была создана ранее. Сообщение о регистрации VLAN v30 отправляется через порт 25 коммутатору 1. Получив это сообщение, коммутатор 1 перестает рассылать оповещения о VLAN v30.

Внимание: порт с поддержкой протокола GVRP подключается к сети VLAN только в том случае, если он непосредственно получает оповещение о ней. Если порт с поддержкой протокола GVPR передает оповещение, полученное от другого порта коммутатора, он не подключается к этой сети VLAN.

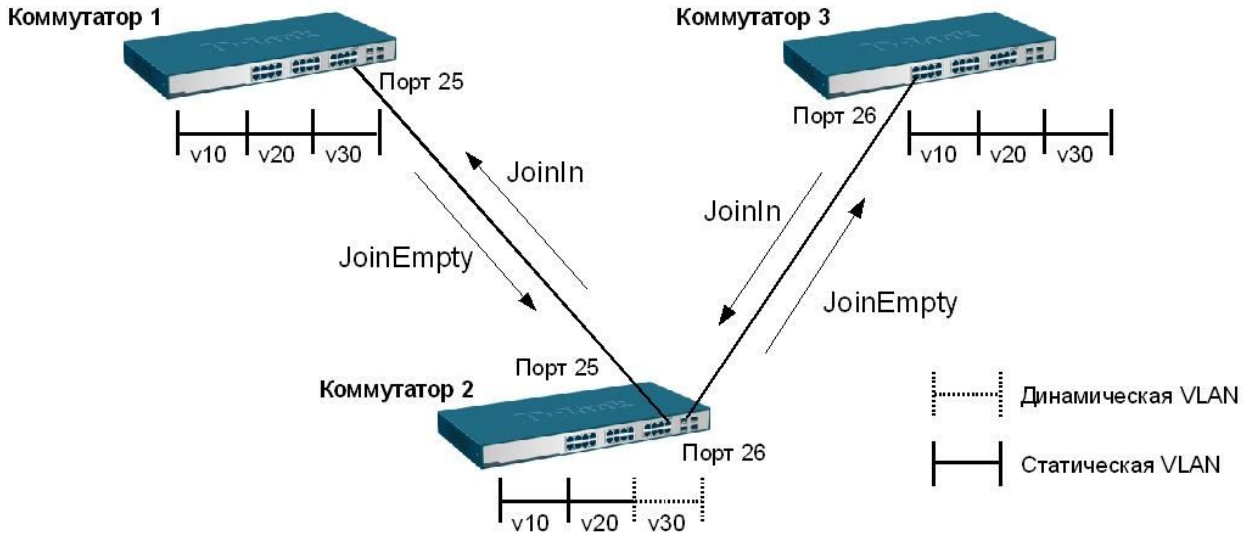


Рис. 4.15. Процесс распространения информации о регистрации VLAN по сети

Рис. 4.16 показывает процесс распространения информации об удалении VLAN по сети. На коммутаторе 1 удалена статическая VLAN v30 и он отправляет сообщение LeaveIn через порт 25 коммутатору 2. Когда коммутатор 2 получит оповещение об удалении VLAN v30, он исключит порт 25 из этой VLAN и отправит сообщение LeaveIn коммутатору 3 через порт 26. Коммутатор 3 получит оповещение об удалении VLAN v30, но удалит ее не сразу, а по истечении периода, установленного таймером Leave Timer. После удаления VLAN v30, коммутатор 3 отправит через порт 26 сообщение LeaveEmpty. После получения этого сообщения коммутатор 2 исключит порт 26 из VLAN v30 и удалит ее по истечении периода, установленного таймером Leave Timer. Через порт 25 будет передано сообщение LeaveEmpty коммутатору 1. Коммутатор 1 исключит свой порт 25 из динамической VLAN v30.

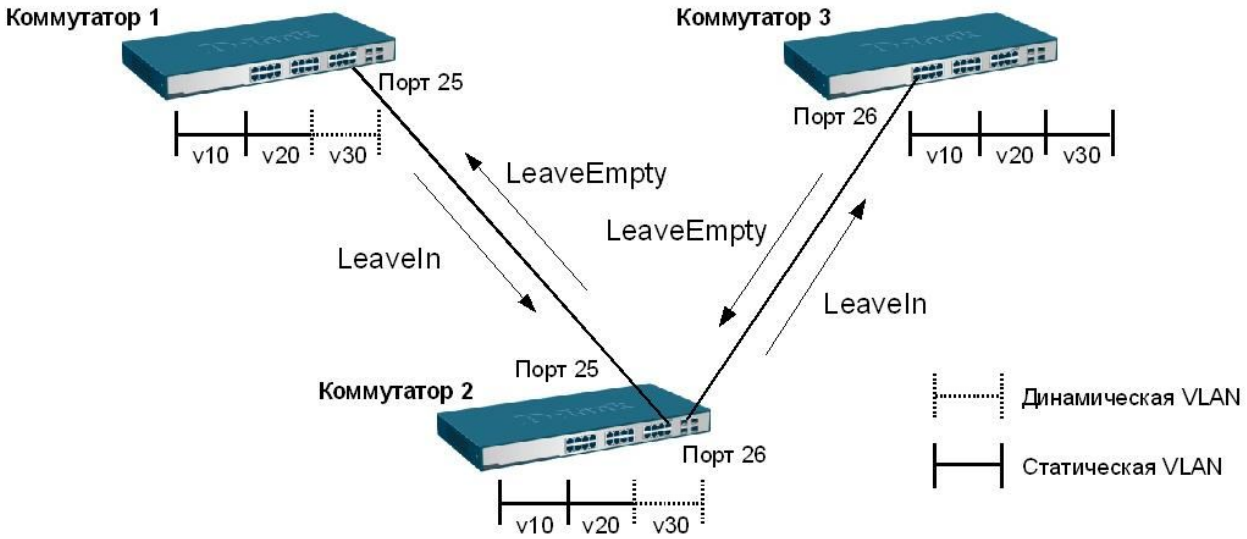


Рис. 4.16. Процесс распространения информации об удалении VLAN по сети

4.5.2 Пример настройки протокола GVRP

В примере, показанном на рис. 4.17, требуется настроить возможность динамического распространения по сети информации о VLAN v30 с использованием протокола GVRP. Ниже приведены настройки коммутаторов.

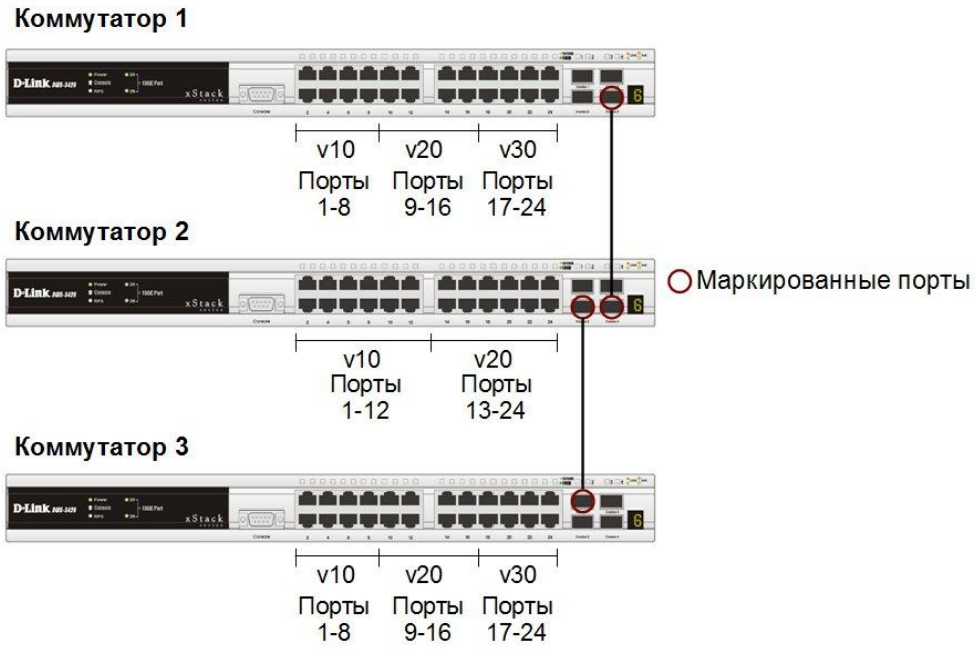


Рис. 4.17. Схема сети VLAN

Настройка коммутаторов 1, 3

- Удалить соответствующие порты из VLAN по умолчанию (default VLAN) и создать новые VLAN.

config vlan default delete 1-24
create vlan v10 tag 10
create vlan v20 tag 20
create vlan v30 tag 30

- В созданные VLAN добавить порты, для которых необходимо указать, какие из них являются маркированными и немаркированными.

config vlan v10 add untagged 1-8
config vlan v20 add untagged 9-16
config vlan v30 add untagged 17-24
config vlan v10 add tagged 25-26
config vlan v20 add taged 25-26

- Активизировать протокол GVRP и функцию оповещения о соответствующей VLAN (в данном примере VLAN v30) по сети.

config vlan v30 advertisement enable
enable gvrp
config port_vlan 25-26 gvrp_state enable

Настройка коммутатора 2

config vlan default delete 1-24
create vlan v10 tag 10
create vlan v20 tag 20

config vlan v10 add untagged 1-12
config vlan v20 add untagged 13-24
config vlan v10 add tagged 25-26
config vlan v20 add tagged 25-26
enable gvrp
config port_vlan 25-26 gvrp_state enable

4.6 Q-in-Q VLAN

Функция *Q-in-Q*, также известная как *Double VLAN*, соответствует стандарту IEEE 802.1ad, который является расширением стандарта IEEE 802.1Q. Она позволяет добавлять в маркированные кадры Ethernet второй тег IEEE 802.1Q.

Благодаря функции Q-in-Q провайдеры могут использовать их собственные уникальные идентификаторы VLAN (называемые Service Provider VLAN ID или *SP-VLAN ID*) при оказании услуг пользователям, в сетях которых настроено несколько VLAN. Это позволяет сохранить используемые пользователями идентификаторы VLAN (Customer VLAN ID или *CVLAN ID*), избежать их совпадения и изолировать трафик разных клиентов во внутренней сети провайдера.

4.6.1 Формат кадра Q-in-Q

На рис. 4.18 изображены форматы обычного кадра Ethernet, кадра Ethernet с тегом 802.1Q, кадра Ethernet с двумя тегами 802.1Q.

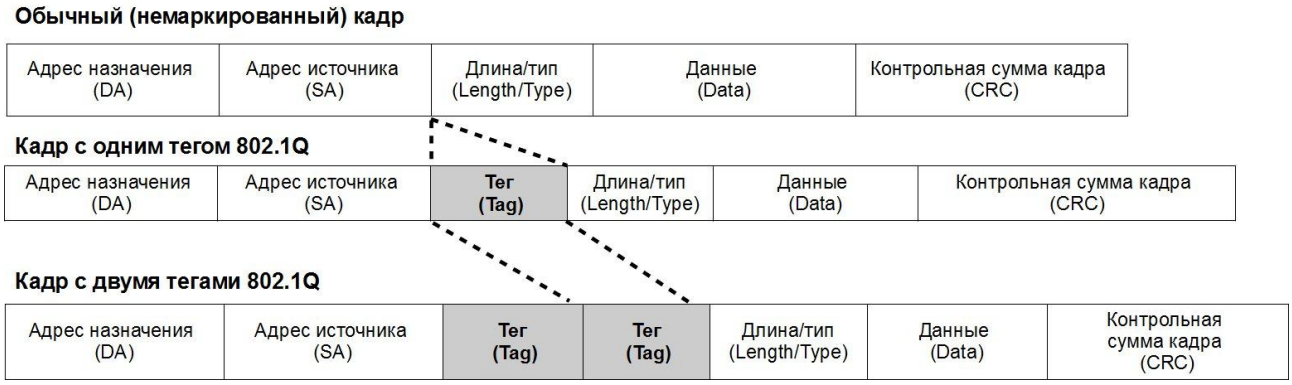


Рис. 4.18. Формата кадра Ethernet с двумя тегами 802.1Q.

Инкапсуляция кадра Ethernet вторым тегом происходит следующим образом: тег, содержащий идентификатор VLAN сети провайдера (*внешний тег*) вставляется перед *внутренним тегом*, содержащим клиентский идентификатор VLAN. Передача кадров в сети провайдера осуществляется только на основе внешнего тега SP-VLAN ID, внутренний тег пользовательской сети CVLAN ID при этом скрыт.

Функция Q-in-Q позволяет расширить доступное пространство идентификаторов и использовать до 4094 x 4094 = 16 760 836 уникальных виртуальных локальных сетей.

4.6.2 Реализации Q-in-Q

Существует две реализации функции Q-in-Q: *Port-based Q-in-Q* и *Selective Q-in-Q*. Функция *Port-based Q-in-Q* по умолчанию присваивает любому кадру, поступившему на порт доступа граничного коммутатора провайдера, идентификатор *SP-VLAN* равный идентификатору PVID порта. Порт маркирует кадр независимо от того, является он маркированным или немаркированным. При поступлении маркированного кадра, в него

добавляется второй тег с идентификатором равным *SP-VLAN*. Если на порт пришел немаркированный кадр, в него добавляется только тег с *SP-VLAN* порта.

Функция *Selective Q-in-Q* является более гибкой по сравнению с *Port-based Q-in-Q*. Она позволяет:

- Маркировать кадры внешними тегами с различными идентификаторами SP-VLAN в зависимости от значений внутренних идентификаторов CVLAN.
- Задавать приоритеты обработки кадров внешних SP-VLAN на основе значений приоритетов внутренних пользовательских CVLAN.
- Добавлять к немаркированным пользовательским кадрам помимо внешнего тега SP-VLAN внутренний тег CVLAN.

4.6.3 Значения TPID в кадрах Q-in-Q

В теге VLAN имеется поле идентификатора протокола тега (TPID, Tag Protocol IDentifier), который определяет тип протокола тега. По умолчанию значение этого поля для стандарта IEEE 802.1Q равно 0x8100.

На устройствах разных производителей TPID внешнего тега VLAN кадров Q-in-Q может иметь разные значения по умолчанию. Для того чтобы кадры Q-in-Q могли передаваться по общедоступным сетям через устройства разных производителей, рекомендуется использовать значение TPID внешнего тега равное 0x88A8, согласно стандарту IEEE 802.1ad.

4.6.4 Роли портов в Port-based Q-in-Q и Selective Q-in-Q

Все порты граничного коммутатора, на котором используются функции Port-based Q-in-Q или Selective Q-in-Q, должны быть настроены как **порты доступа (UNI)** или **Uplink-порты (NNI)**:

- **UNI (User-to-Network Interface)** – эта роль назначается портам, через которые будет осуществляться взаимодействие граничного коммутатора провайдера с клиентскими сетями.
- **NNI (Network-to-Network Interface)** – эта роль назначается портам, которые подключаются к внутренней сети провайдера или другим граничным коммутаторам.

4.6.5 Политики назначения внешнего тега и приоритета в Q-in-Q

Функция Selective Q-in-Q позволяет добавлять в кадры различные внешние теги VLAN, основываясь на значениях внутренних тегов. Для этого на портах UNI граничного коммутатора необходимо задать правила соответствия идентификаторов CVLAN идентификаторам SP-VLAN (*vlan translation*).

Помимо этого, на коммутаторах D-Link с поддержкой функции Q-in-Q, можно активизировать режим Missdrop. При настройке Selective Q-in-Q, включение этого режима позволит отбрасывать кадры, не подходящие ни под одно из правил соответствия идентификаторов. При настройке Port-based Q-in-Q, режим Missdrop надо отключать, чтобы порт коммутатора мог принимать кадры не подходящие ни под одно из правил vlan translation. В этом случае входящим кадрам будет присваиваться внешний тег равный PVID соответствующего порта UNI.

Значение приоритета внешнего тега по умолчанию равно значению приоритета внутреннего тега, если кадр является маркированным, или не сделаны соответствующие настройки. Если приоритет в полученном кадре отсутствует, то в качестве приоритета вешнего тега будет использоваться приоритет соответствующего входного порта UNI.

4.6.6 Базовая архитектура сети с функцией Port-based Q-in-Q

На рис. 4.19 показана базовая архитектура сети провайдера услуг с функцией Port-based Q-in-Q. Граничные коммутаторы сети провайдера услуг PE-1 и PE-2 позволяют

обрабатывать трафик виртуальных локальных сетей двух подключенных к ним клиентских сетей. Каждому клиенту провайдером присвоен уникальный идентификатор VLAN: SP-VLAN 50 для клиента A и SP-VLAN 100 для клиента B. При передаче кадра из клиентской сети в сеть провайдера, в его заголовок будет добавляться второй тег 802.1Q: для сети A – SP-VLAN 50, для сети B – SP-VLAN 100. При передаче кадра из сети провайдера в клиентскую сеть, второй тег будет удаляться граничным коммутатором.

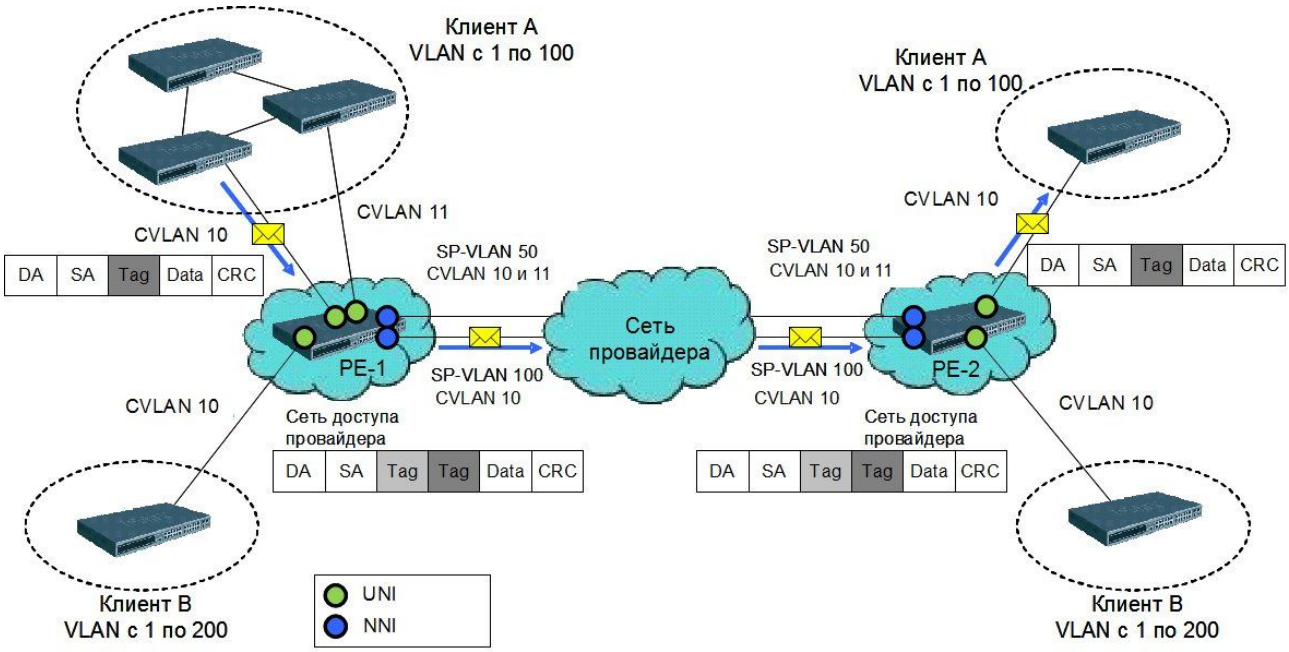


Рис. 4.19. Базовая архитектура сети провайдера с применением функции Port-based Q-in-Q

4.6.7 Пример настройки функции Port-based Q-in-Q

Рассмотрим пример настройки функции Port-based Q-in-Q на коммутаторах D-Link. На рис. 4.20 приведена схема подключения двух клиентских VLAN к сети провайдера услуг. Граничными коммутаторами являются коммутаторы Gigabit Ethernet 3 уровня. В сети клиента используются коммутаторы Fast Ethernet 2 уровня.

Внимание: функцию Q-in-Q VLAN необходимо настраивать только на устройствах сети провайдера услуг.

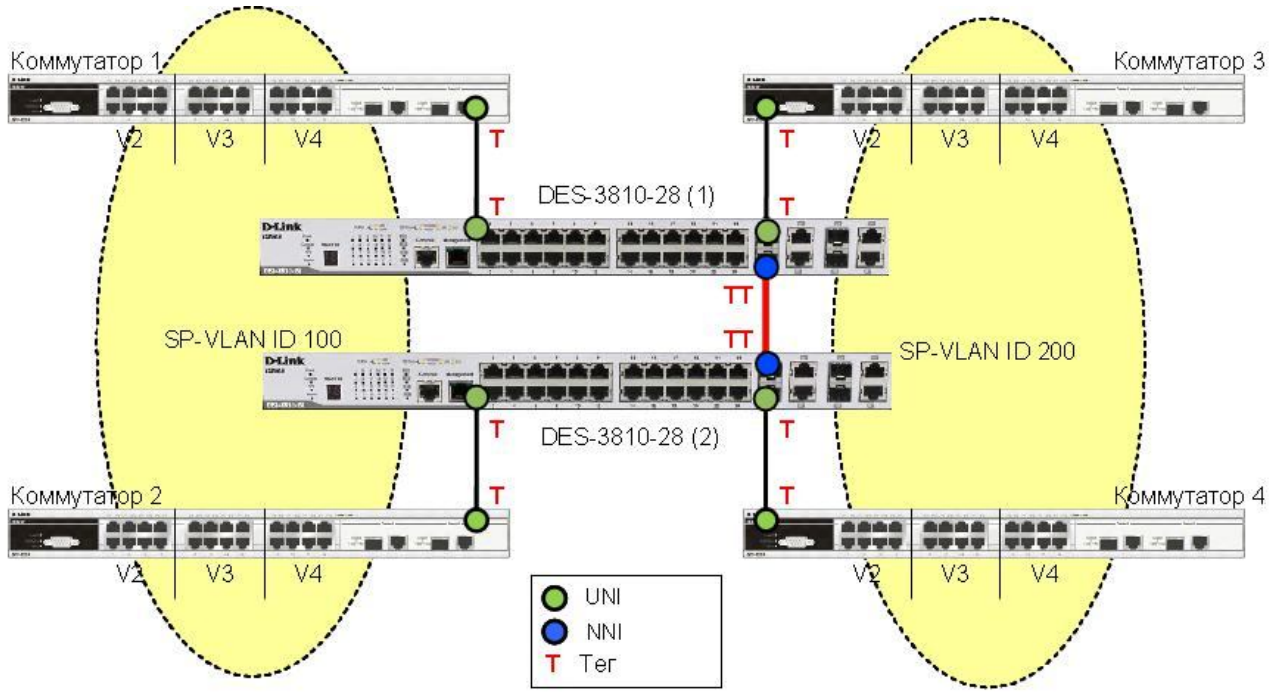


Рис. 4.20. Схема подключения клиентских VLAN к сети провайдера услуг

Настройка коммутатора DES-3810-28

- Активизировать функцию Q-in-Q VLAN на коммутаторе.

enable qinq

- Удалить соответствующие порты из Q-in-Q VLAN по умолчанию и создать новые VLAN.

config vlan default delete 1-24
create vlan d100 tag 100
create vlan d200 tag 200

- Назначить порты доступа в созданных Q-in-Q VLAN.

config vlan d100 add untagged 1-12
config vlan d200 add untagged 13-24

- Назначить Uplink-порты в созданных Q-in-Q VLAN.

config vlan d100 add tagged 25-27
config vlan d200 add tagged 25-27

- Настроить роли портов доступа в Q-in-Q и отключить режим Missdrop на них.

config qinq ports 1-24 role uni missdrop disable

Настройка коммутаторов 1, 2, 3, 4

- Удалить соответствующие порты из VLAN по умолчанию (default VLAN) и создать новые VLAN.

config vlan default delete 1-26
create vlan v2 tag 2
create vlan v3 tag 3
create vlan v4 tag 4

- В созданные VLAN добавить порты, для которых необходимо указать, какие из них являются маркированными и немаркированными.

config vlan v2 add untagged 1-8
config vlan v2 add tagged 25-26
config vlan v3 add untagged 9-16
config vlan v3 add tagged 25-26
config vlan v4 add untagged 17-24
config vlan v4 add tagged 25-26

4.6.8 Пример настройки функции Selective Q-in-Q

На рис. 4.21 показана схема подключения двух клиентских VLAN к граничным коммутаторам провайдера. Каждому клиенту провайдером назначен уникальный идентификатор: SP-VLAN 1000 для клиента CVLAN 200 и SP-VLAN 1001 для клиента CVLAN 300. В качестве граничных коммутаторов используются коммутаторы Fast Ethernet 2 уровня. Порты 9 обоих граничных коммутаторов служат для подключения к пользовательским сетям (UNI-порты), передача данных в сеть провайдера осуществляется через порты 11 (NNI-порты).

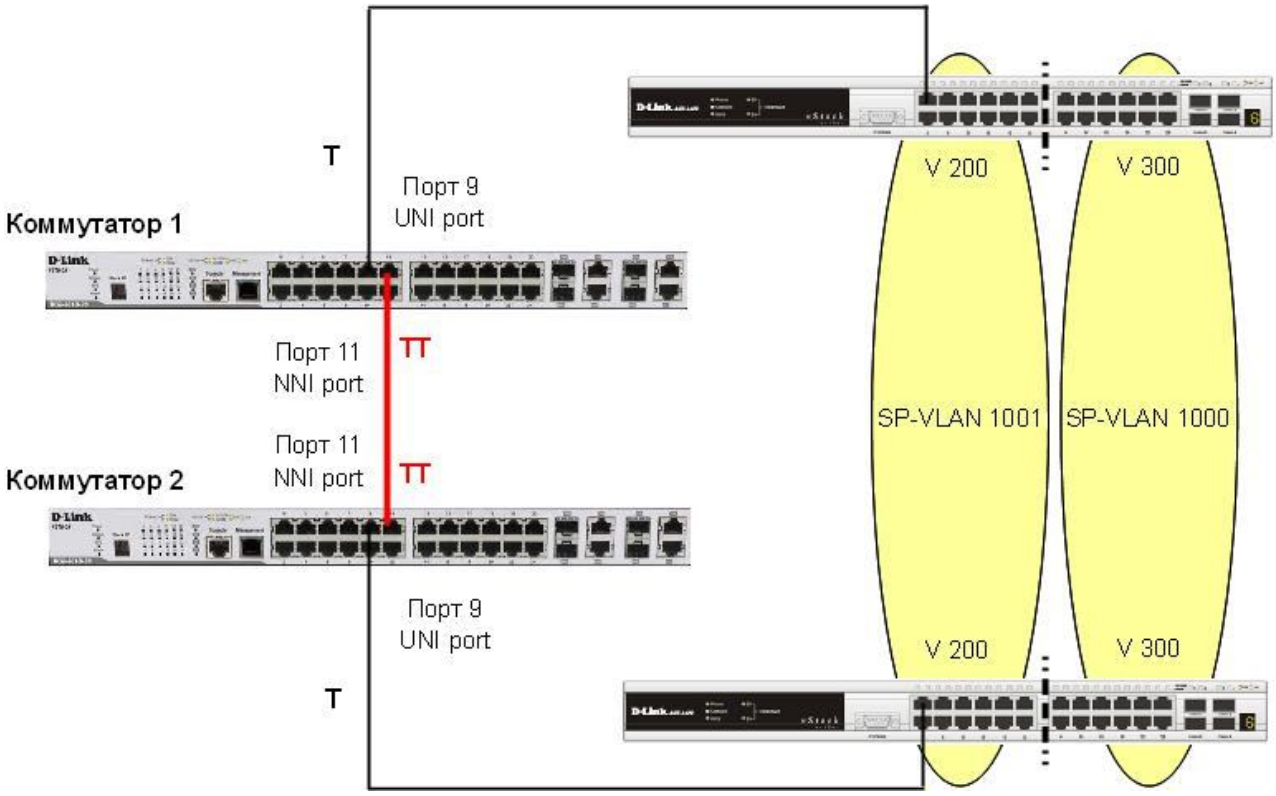


Рис. 4.21. Схема подключения клиентских VLAN к пограничным коммутаторам сети провайдера услуг

Для того чтобы граничные коммутаторы могли осуществлять передачу пользовательских кадров с использованием функции Selective Q-in-Q, на них необходимо выполнить следующие настройки.

Настройка коммутаторов 1, 2

- Создать требуемые VLAN и добавить порты, для которых необходимо указать, какие из них являются маркированными и немаркированными.

create vlan v1000 tag 1000

create vlan v1001 tag 1001
config vlan v1000 add tag 9,11
config vlan v1001 add tag 9,11

- Активизировать функцию Q-in-Q VLAN, указать значения TPID внутреннего и внешнего тега, роли портов и задать правила соответствия идентификаторов CVLAN идентификаторам SP-VLAN.

enable qinq
config qinq ports 9 role uni
create vlan_translation ports 9 cvid 200 add svid 1000
create vlan_translation ports 9 cvid 300 add svid 1001

4.7 VLAN на основе портов и протоколов – стандарт IEEE 802.1v

Стандарт IEEE 802.1v является расширением стандарта IEEE 802.1Q. Он позволяет объединять узлы сети в виртуальные локальные сети на основе поддерживаемых ими протоколов. При определении членства в VLAN стандарт классифицирует *немаркированные* кадры по типу протокола и порту. Формат тега 802.1v аналогичен формату тега 802.1Q.

В стандарте IEEE 802.1v определены следующие правила классификации входящих кадров:

- При поступлении на порт *немаркированного* кадра, коммутатором осуществляется проверка заголовка канального уровня и типа протокола вышележащего уровня. Если тип протокола соответствует типу VLAN 802.1v на этом порте, то в заголовок кадра добавляется тег с идентификатором VID, равным идентификатору соответствующей VLAN 802.1v. Если совпадения не найдены, то в заголовок кадра добавляется тег с идентификатором VID, равным идентификатору входного порта *PVID*.
- При поступлении на порт *маркированного* кадра значение тега VLAN в нем не изменяется.

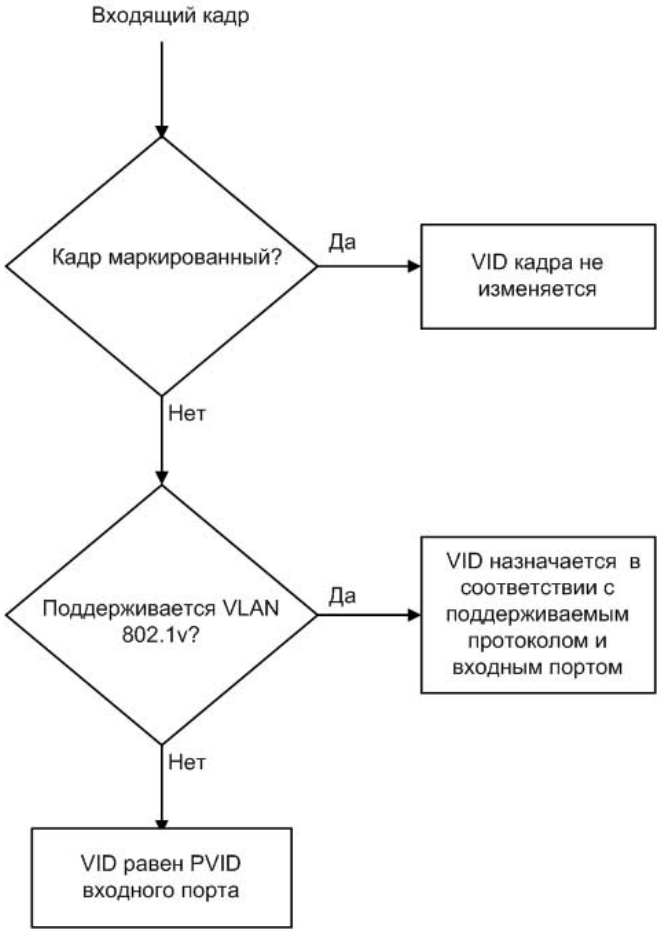


Рис. 4.22. Правила классификации входящих кадров

Внутри коммутатора все кадры являются маркированными. Передача кадров осуществляется на основе таблицы VLAN, путем сравнения значений идентификаторов VID. Если порт назначения является членом той же VLAN, что и входной порт, то он передает кадр в подключенный к нему сегмент сети. В противном случае, кадр отбрасывается.

Для выходных портов действуют такие же правила, как для стандарта IEEE 802.1Q.

Механизм классификации 802.1v требует, чтобы на коммутаторе были настроены группы протоколов. Каждый протокол в группе определяется типом кадра (Ethernet II, IEEE 802.3 SNAP или IEEE 802.3 LLC) и значением поля идентификации протокола в нем. Порт может быть ассоциирован с несколькими группами протоколов, что позволяет классифицировать поступающие немаркированные кадры по принадлежности к разным VLAN в зависимости от их содержимого. Одна и та же группа протоколов может быть ассоциирована с разными портами коммутатора, при этом на каждом входном порте ей должны быть присвоены уникальные идентификаторы VLAN.

4.7.1 Пример настройки IEEE 802.1v VLAN

На рис. 4.23 показано типовое подключение клиентов к сети провайдера услуг. Пользователи локальной сети находятся в выделенной VLAN (VLAN 20). Их подключение в Интернет осуществляется через PPPoE-сервер (VLAN 10). Для того чтобы трафик локальной сети был отделен от трафика PPPoE, на коммутаторе для протокола PPPoE создана VLAN 802.1v с идентификатором VID=10.

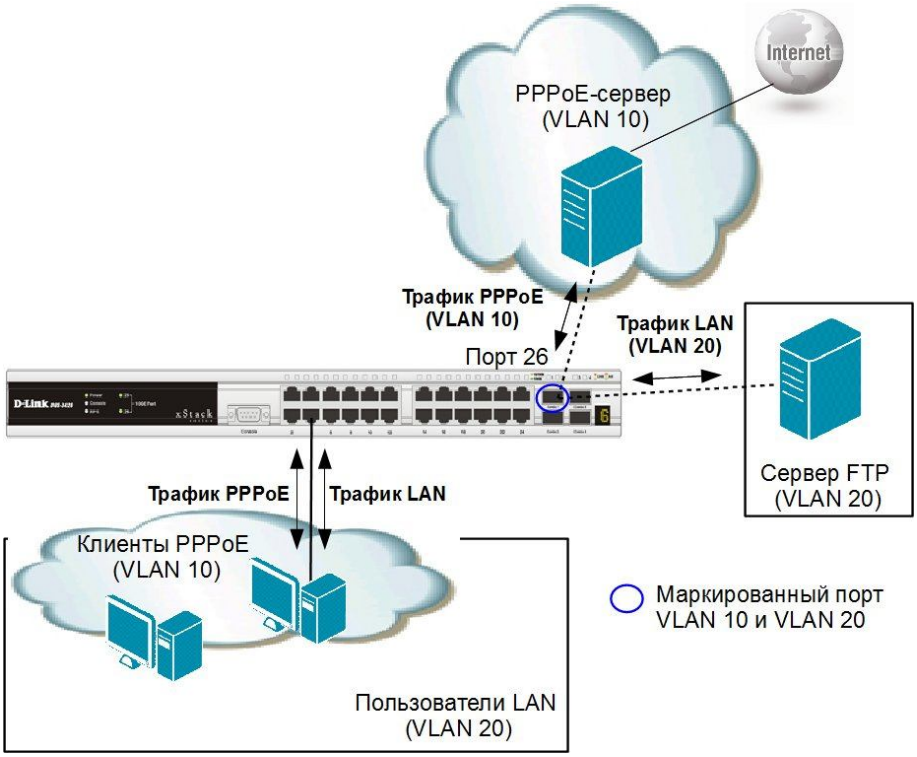


Рис. 4.23. Схема сети VLAN

Настройка коммутатора

• Создание новых VLAN 802.1Q.

config vlan default delete 1-28
create vlan pppoe tag 10
config vlan pppoe add untagged 1-24
config vlan pppoe add tagged 26
create vlan base tag 20
config vlan base add tagged 26
config vlan base add untagged 1-24

• Настройка PVID портов, к которым подключены пользователи.

config port_vlan 1-24 pvid 20

• Создание VLAN 802.1v для протокола PPPoE (первая группа протоколов настроена для кадров PPPoE, передаваемых на стадии исследования, вторая – для кадров PPPoE установленной сессии).

create dot1v_protocol_group group_id 1 group_name pppoe_disc
config dot1v_protocol_group group_id 1 add protocol ethernet_2 8863
create dot1v_protocol_group group_id 2 group_name pppoe_session
config dot1v_protocol_group group_id 2 add protocol ethernet_2 8864
config port dot1v ports 1-24 add protocol_group group_id 1 vlan pppoe
config port dot1v ports 1-24 add protocol_group group_id 2 vlan pppoe

4.8 Асимметричные VLAN

Для обеспечения возможности использования разделяемых ресурсов (серверов, Интернет-шлюзов и т.д.) пользователями из разных сетей VLAN, в программном обеспечении коммутаторов **2-го уровня** D-Link реализована поддержка функции *Asymmetric VLAN* (ассиметричные VLAN). Эта функция позволяет клиентам из разных VLAN взаимодействовать с разделяемыми устройствами (например, серверами), *не поддерживающим* тегирование *802.1Q,* через один физический канал связи с коммутатором, не требуя использования внешнего маршрутизатора. Активизация функции Asymmetric VLAN на коммутаторе 2-го уровня позволяет сделать его *немаркированные* порты *членами нескольких виртуальных локальных сетей*. При этом рабочие станции остаются полностью изолированными друг от друга. Например, асимметричные VLAN могут быть настроены так, чтобы обеспечить доступ к почтовому серверу всем почтовым клиентам. Клиенты смогут отправлять и получать данные через порт коммутатора, подключенный к почтовому серверу, но прием и передача данных через остальные порты будет для них запрещена.

При активизации асимметричных VLAN, каждому порту коммутатора назначается уникальный PVID в соответствии с идентификатором VLAN, членом которой он является. При этом каждый порт, может получать кадры от VLAN по умолчанию.

Внимание: функция Asymmetric VLAN не поддерживается коммутаторами 3-го уровня. Организация обмена данными между устройствами различных VLAN не поддерживающих тегирование реализуется в таких коммутаторах с помощью маршрутизации и списков управления доступом (ACL), ограничивающих доступ устройств к сети.

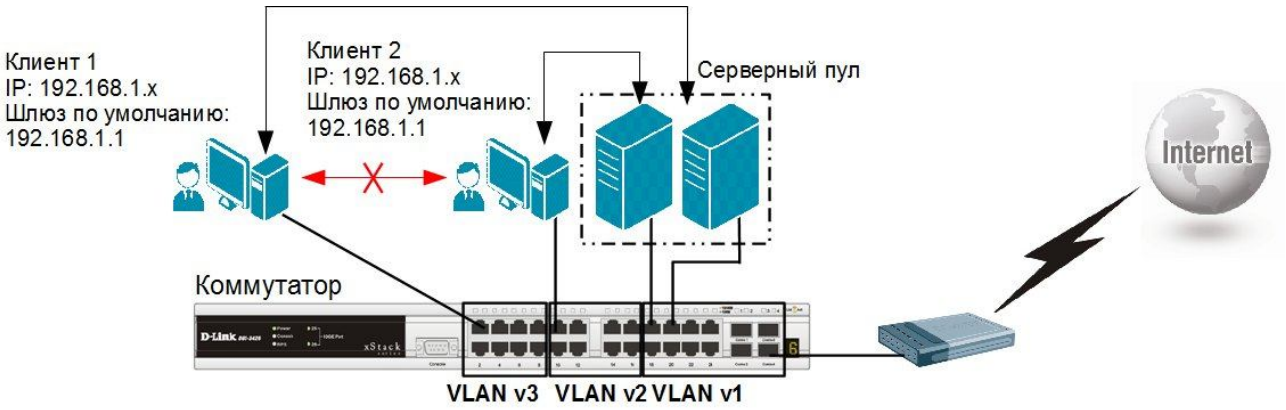
Основное различие между базовым стандартом 802.1Q VLAN (или симметричными VLAN) и асимметричными VLAN заключается в том, как выполняется отображение MAC-адресов. Симметричные VLAN используют отдельные адресные таблицы, и, таким образом, не происходит пересечения MAC-адресов между виртуальными локальными сетями. Асимметричные VLAN используют одну общую таблицу MAC-адресов.

При использовании асимметричных VLAN существует следующее ограничение: не функционирует механизм IGMP Snooping.

По умолчанию асимметричные VLAN на коммутаторах D-Link отключены.

4.8.1 Пример настройки асимметричных VLAN

На рис. 4.24 показана схема реализации ассиметричных VLAN в пределах одного коммутатора. Пользователи VLAN v2 и v3 могут получать доступ к разделяемым серверам и Интернет-шлюзу, находящимся в VLAN v1. Виртуальные локальные сети VLAN v2 и v3 изолированы друг от друга.



Коммутатор

	VLAN v1 (разделяемая VLAN)	VLAN v2 (пользовательская VLAN)	VLAN v3 (пользовательская VLAN)
Немаркированные порты	17-24	9-16	1-8
Маркированные порты	-	-	-

Рис. 4.24. Асимметричные VLAN в пределах одного коммутатора

Для реализации этой схемы на коммутаторе D-Link необходимо выполнить следующие настройки:

Настройка коммутатора
enable asymmetric_vlan
create vlan v2 tag 2
create vlan v3 tag 3
config vlan v2 add untagged 9-24
config vlan v3 add untagged 1-8,17-24
config gvrp 1-8 pvid 3
config gvrp 9-16 pvid 2
config gvrp 17-24 pvid 1

4.9 Функция Traffic Segmentation

Функция *Traffic Segmentation* (сегментация трафика) служит для разграничения доменов на канальном уровне. Она позволяет настраивать порты или группы портов коммутатора таким образом, чтобы они были полностью изолированы друг от друга, но в то же время имели доступ к разделяемым портам, используемым для подключения серверов или магистрали сети. Этот метод изоляции трафика аналогичен функции Asymmetric VLAN, но его применение ограничено пределами одного коммутатора или нескольких коммутаторов в стеке, т.к. членство в группе портов не может распространяться по сети.

Можно выделить следующие преимущества функции Traffic Segmentation по сравнению с Asymmetric VLAN:

- простота настройки;
- поддерживается работа IGMP Snooping;
- функция Traffic Segmentation может быть представлена в виде иерархического дерева (при иерархическом подходе разделяемые ресурсы должны быть на «вершине» дерева);
- нет ограничений на создание количества групп портов.

Функция сегментации трафика может использоваться с целью сокращения трафика внутри сетей VLAN 802.1Q, позволяя разбивать их на более маленькие группы. При этом правила VLAN имеют более высокий приоритет при передаче трафика. Правила Traffic Segmentation применяются после них.

4.9.1 Примеры использования и настройки функции Traffic Segmentation

В качестве примера рассмотрим решение задачи совместного использования ресурсов сети разными группами пользователей с использованием функции Traffic Segmentation (рис. 4.25).

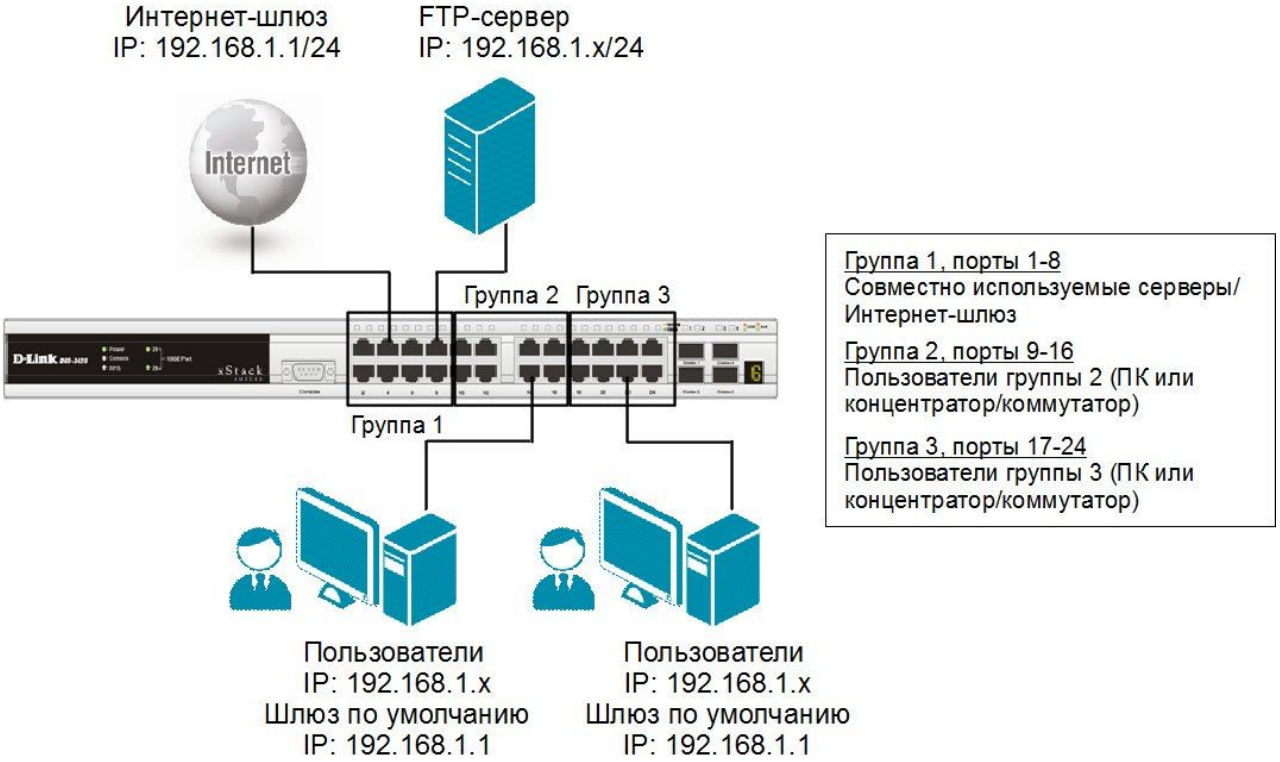


Рис. 4.25. Пример использования функции Traffic Segmentation

Пользователи групп 2 и 3 имеют доступ к совместно-используемому FTP-серверу и Интернет-шлюзу, но обмен данными между группами 2 и 3 запрещен.

Настройка коммутатора
config traffic_segmentation 1-8 forward_list 1-24
config traffic_segmentation 9-16 forward_list 1-16
config traffic_segmentation 17-24 forward_list 1-8,17-24

Используя возможности построения иерархического дерева функции Traffic Segmentation можно решать типовые задачи изоляции портов в сетях с многоуровневой структурой.

В примере, показанном на рис. 4.26 все компьютеры от A до Q, находящиеся в одной IP-подсети, не могут принимать/отправлять пакеты данных друг другу, но при этом имеют доступ к серверам и Интернет. Все коммутаторы сети поддерживают иерархию Traffic Segmentation.

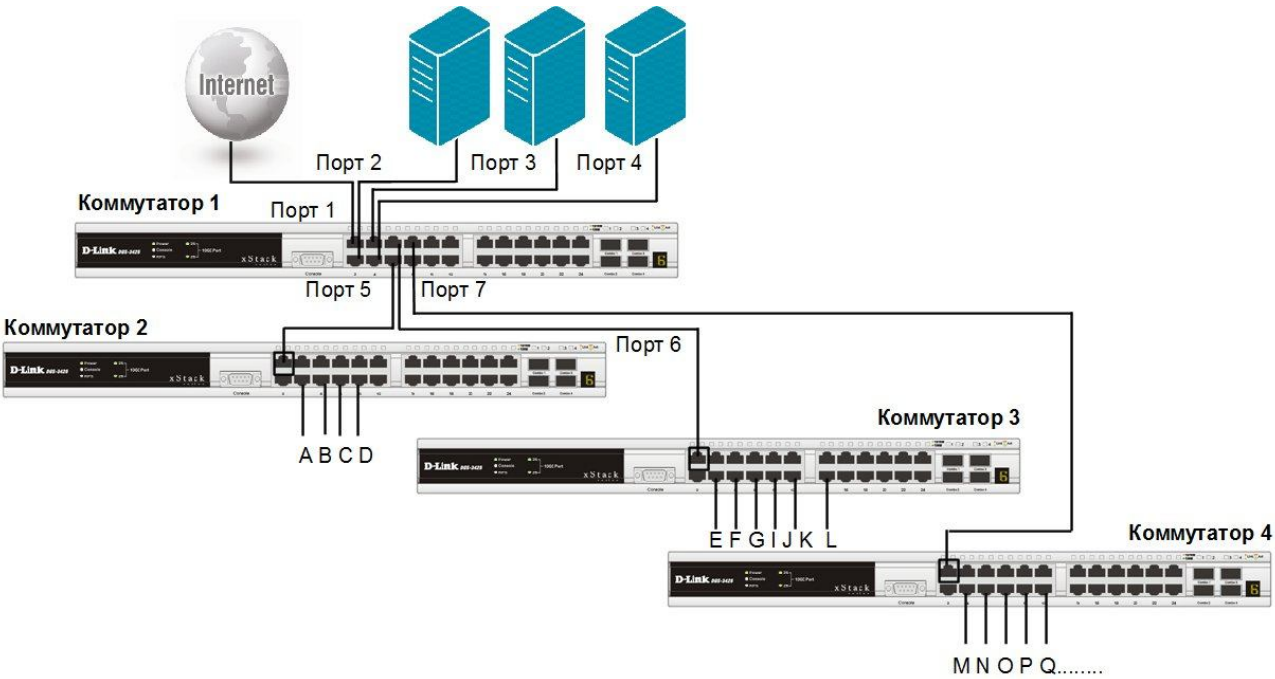


Рис. 4.26. Иерархическая структура Traffic Segmentation

Настройка коммутатора 1
config traffic_segmentation 1-4 forward_list 1-26
config traffic_segmentation 5 forward_list 1-5
config traffic_segmentation 6 forward_list 1-4, 6
config traffic_segmentation 7 forward_list 1-4, 7

Настройка коммутаторов 2, 3, 4
config traffic_segmentation 1 forward_list 1-26
config traffic_segmentation 2-26 forward_list 1

5. Функции повышения надежности и производительности

В настоящее время для повышения надежности и производительности каналов связи в распоряжении интеграторов и сетевых администраторов имеется целый набор протоколов и функций. Наиболее распространенным является создание резервных связей между коммутаторами на основе двух технологий:

1. Резервирование соединений с помощью протоколов семейства Spanning Tree.
2. Балансировка нагрузки, обеспечивающая параллельную передачу данных по всем альтернативным соединениям с помощью механизма агрегирования портов.

5.1 Протоколы Spanning Tree

Протокол связующего дерева **Spanning Tree Protocol (STP)** является протоколом 2 уровня модели OSI, который позволяет строить древовидные, свободные от петель, конфигурации связей между коммутаторами локальной сети. Помимо этого, алгоритм обеспечивает возможность автоматического резервирования альтернативных каналов связи между коммутаторами на случай выхода активных каналов из строя.

В настоящее время существуют следующие версии протоколов связующего дерева:

- IEEE 802.1D Spanning Tree Protocol (STP);
- IEEE 802.1w Rapid Spanning Tree Protocol (RSTP);
- IEEE 802.1s Multiple Spanning Tree Protocol (MSTP).

5.2 Spanning Tree Protocol (STP)

5.2.1 Понятие петель

Если для обеспечения избыточности между коммутаторами создается несколько соединений, то могут возникать коммутационные петли. Петля предполагает существование нескольких маршрутов по промежуточным сетям, а сеть с несколькими маршрутами между источником и приемником отличается повышенной отказоустойчивостью. Хотя наличие избыточных каналов связи очень полезно, петли, тем не менее, создают проблемы, самые актуальные из которых:

- широковещательные штормы;
- множественные копии кадров;
- множественные петли.

Широковещательный шторм.

Распространение широковещательных сообщений в сетях с петлями представляет серьезную проблему. Предположим, что первый кадр, поступивший от одного из узлов, является широковещательным. Тогда все коммутаторы будут пересылать кадры бесконечно, как показано на рис. 5.1 (Пример 1), используя всю доступную полосу пропускания сети и блокируя передачу других кадров во всех сегментах.

Множественные копии кадров.

Еще одна проблема заключается в том, что коммутатор нередко получает несколько копий одного кадра, одновременно приходящих из нескольких участков сети. В этом случае таблица коммутации не сможет определить расположение устройства, потому что коммутатор будет получать кадр из нескольких каналов. Может случиться так, что коммутатор вообще не сможет переслать кадр, т.к. будет постоянно обновлять таблицу коммутации.

Множественные петли.

Одна из самых сложных проблем – это множественные петли, образующиеся в объединенной сети. Возможно появление петли внутри других петель. Если за этим последует широковещательный шторм, то сеть не сможет выполнять коммутацию кадров.

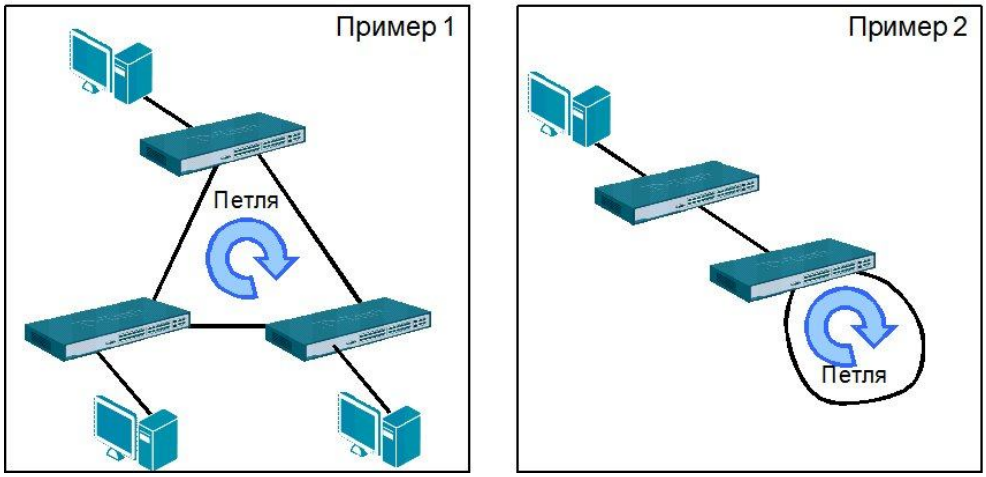


Рис. 5.1. Примеры петель между коммутаторами

Для решения этих проблем и был разработан протокол связующего дерева, который был определен в стандарте IEEE 802.1D-1998.

Коммутаторы, поддерживающие протокол STP, автоматически создают древовидную конфигурацию связей без петель в компьютерной сети. Такая конфигурация называется связующим деревом – Spanning Tree (иногда ее называют остовым или покрывающим деревом). Конфигурация связующего дерева строится коммутаторами автоматически с использованием обмена служебными кадрами, называемыми *Bridge Protocol Data Units (BPDU)*.

5.2.2 Построение активной топологии связующего дерева

Для построения устойчивой активной топологии с помощью протокола STP необходимо с каждым коммутатором сети ассоциировать уникальный *идентификатор моста (Bridge ID)*, а с каждым портом коммутатора ассоциировать *стоимость пути (Path Cost)* и *идентификатор порта* (*Port ID*).

Процесс вычисления связующего дерева начинается с выбора **корневого моста** *(Root Bridge),* от которого будет строиться дерево. В качестве корня дерева выбирается коммутатор с наименьшим значением идентификатора моста. Идентификатор моста – это 8-байтное поле, которое состоит из 2-х частей: приоритета моста (2 байта), назначаемого администратором сети, и MAC-адреса блока управления коммутатора (6 байт). При сравнении идентификаторов двух коммутаторов, сначала сравниваются значения приоритетов. Корневым мостом становится коммутатор с наименьшим значением приоритета. Если они одинаковы (по умолчанию приоритет равен 32768), то корневой мост определяется по наименьшему MAC-адресу.

Для того чтобы в качестве корневого моста было выбрано определенное устройство (исходя из структуры сети), администратор может вручную назначить соответствующему коммутатору наименьшее значение приоритета.

Второй этап работы STP – выбор **корневых портов** *(Root Port)*.

Когда процесс выбора корневого моста завершен, оставшиеся коммутаторы сети определяют стоимость каждого возможного пути от себя до корня дерева. Стоимость пути рассчитывается как *суммарное условное время* на передачу данных от порта данного коммутатора до порта корневого моста. Условное время сегмента рассчитывается, как время передачи одного бита информации через канал с определенной полосой пропускания.

Стоимости пути по умолчанию для каждого канала определены в стандарте IEEE 802.1D-2004.

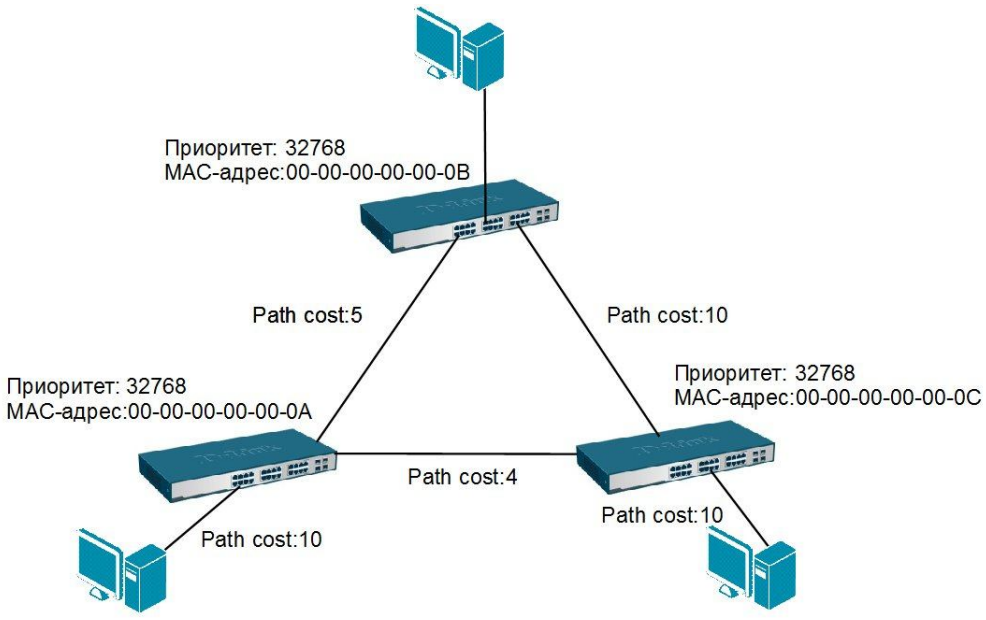
Сравнив стоимости всех возможных маршрутов до корня, каждый коммутатор выбирает среди них один с наименьшим значением стоимости. Порт, соединяющий коммутатор с этим маршрутом, становится корневым портом. В случае если минимальные стоимости пути нескольких маршрутов окажутся одинаковыми, корневым портом станет порт, имеющий наименьшее значение идентификатора порта.

Третий шаг работы STP – определение **назначенных портов** *(Designated Port)*.

Каждый сегмент в коммутируемой сети имеет один назначенный порт. Этот порт функционирует как единственный порт моста, т.е. принимает кадры от сегмента и передает их в направлении корневого моста через корневой порт данного коммутатора. Коммутатор, содержащий назначенный порт для данного сегмента, называется **назначенным мостом** *(Designated Bridge)* этого сегмента. Назначенный порт сегмента определяется путем сравнения значений стоимости пути всех маршрутов от данного сегмента до корневого моста. Им становится порт, имеющий наименьшее значение стоимости, среди всех портов, подключенных к данному сегменту. Если минимальные значения стоимости пути окажутся одинаковыми у двух или нескольких портов, то для выбора назначенного порта сегмента STP принимает решение на основе последовательного сравнения идентификаторов мостов и идентификаторов портов.

У корневого моста все порты являются назначенными, а их расстояние до корня полагается равным нулю. Корневого порта у корневого моста нет.

После выбора корневых и назначенных портов все остальные порты коммутаторов сети переводятся в состояние Blocking (Блокировка), то есть такое, при котором они принимают и передают только кадры BPDU. При таком выборе активных портов в сети исключаются петли, и оставшиеся связи образуют связующее дерево.



Перед применением протокола STP

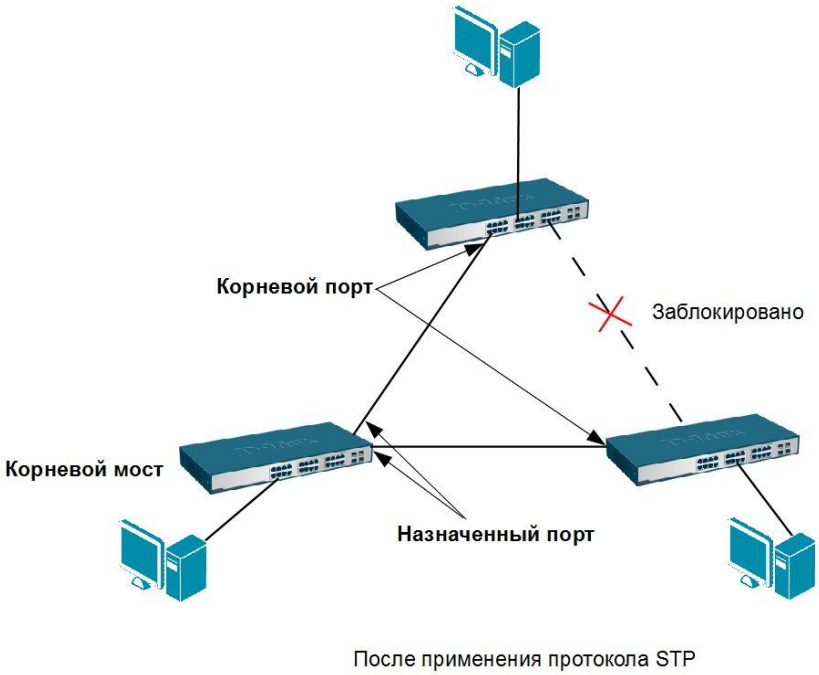


Рис. 5.2. Пример функционирования протокола STP

5.2.3 Bridge Protocol Data Unit (BPDU)

Вычисление связующего дерева происходит при включении коммутатора и при изменении топологии. Эти вычисления требуют периодического обмена информацией между коммутаторами связующего дерева, что достигается при помощи специальных кадров, называемых блоками данных протокола моста – BPDU (Bridge Protocol Data Unit).

Коммутатор отправляет BPDU, используя уникальный MAC-адрес порта в качестве адреса-источника и многоадресный MAC-адрес протокола STP 01-80-C2-00-00-00 в качестве адреса-приемника. Кадры BPDU помещаются в поле данных кадров канального уровня, например, кадров Ethernet.

Внимание: иногда, с целью повышения безопасности, сетевым администраторам необходимо отключать возможность передачи кадров BPDU на граничные коммутаторы сети, чтобы избежать получения случайных кадров BPDU клиентскими портами, которые могут распространить вычисления STP по клиентским сетям. Управляемые коммутаторы D-Link поддерживают возможность включения и отключения передачи кадров BPDU для каждого порта.

Существует три типа кадров BPDU:

- Configuration BPDU (CBPDU) – конфигурационный кадр BPDU, который используется для вычисления связующего дерева (тип сообщения: 0x00).
- Topology Change Notification (TCN) BPDU – уведомление об изменении топологии сети (тип сообщения: 0x80).
- Topology Change Notification Acknowledgement (TCA) – подтверждение о получении уведомления об изменении топологии сети.

Коммутаторы обмениваются BPDU через равные интервалы времени (по умолчанию 2 сек.), что позволяет им отслеживать состояние топологии сети.

	Байты
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

Рис. 5.3. Формат кадра BPDU

Кадр BPDU состоит из следующих полей:

- Идентификатор протокола (Protocol Identifier) – 2 байта. Значение всегда равно 0.
- Версия протокола STP (Protocol Version Identifier) – 1 байт. Значение всегда равно 0.
- Тип BPDU (BPDU Type) – 1 байт. Значение «00» – конфигурационный BPDU, «01» – изменение топологии.
- Флаги (Flags) – 1 байт. Бит 1 – флаг изменения топологии, бит 8 – флаг подтверждения изменения топологии.
- Идентификатор корневого моста (Root Identifier) – 8 байтов. Идентификатор текущего корневого моста.
- Расстояние до корневого моста (Root Path Cost) – 2 байта. Суммарная стоимость пути до корневого моста.
- Идентификатор моста (Bridge Identifier) – 8 байтов. Идентификатор текущего моста.
- Идентификатор порта (Port Identifier) – 2 байта. Уникальный идентификатор порта, который отправил этот BPDU.
- Время жизни сообщения (Message Age) – 2 байта. Нефиксированный временной интервал в секундах, прошедший с момента отправки BPDU корневым мостом. Служит для выявления устаревших сообщений BPDU. Первоначальное значение равно нулю. По мере передачи кадра BPDU по сети, каждый коммутатор, добавляет ко времени жизни сообщения время его задержки данным коммутатором. По умолчанию оно равно 1 сек. Значение параметра Message Age должно быть меньше значения таймера Max Age.
- Максимальное время жизни сообщения (Max Age) – 2 байта. Временной интервал в секундах, определяющий максимальное время хранения конфигурации STP, прежде чем коммутатор ее отбросит.
- Время приветствия (Hello Time) – 2 байта. Временной интервал в секундах, через который посылаются кадры BPDU.
- Задержка смены состояний (Forward Delay) – 2 байта. Временной интервал в секундах, в течение которого порт коммутатора находится в состояниях «Прослушивание» и «Обучение».

5.2.4 Состояния портов

В процессе построения топологии сети каждый порт коммутатора проходит несколько стадий:

- **Blocking** («Блокировка») – при инициализации коммутатора все порты (за исключением отключенных) автоматически переводятся в состояние «Блокировка». В этом случае порт принимает и обрабатывает только кадры BPDU. Все остальные кадры отбрасываются.
- **Listening** («Прослушивание») – в этом состоянии порт продолжает принимать, обрабатывать и ретранслировать только кадры BPDU. Из этого состояния порт может перейти в состояние «Блокировка», если получит BPDU с лучшими параметрами, чем его собственные (стоимость пути, идентификатор моста или порта). В противном случае, при истечении периода, установленного таймером задержки смены состояний (Forward Delay), порт перейдет в следующее состояние «Обучение».
- **Learning** («Обучение») – порт начинает принимать все кадры и на основе MAC-адресов источника строить таблицу коммутации. Порт в этом состоянии все еще не продвигает кадры. Порт продолжает участвовать в работе алгоритма STP и, при поступлении BPDU с лучшими параметрами, переходит в состояние «Блокировка». В противном случае, при истечении периода, установленного таймером задержки смены состояний, порт перейдет в следующее состояние «Продвижение».
- **Forwarding** («Продвижение») – в этом состоянии порт может обрабатывать кадры данных в соответствии с построенной таблицей коммутации. Также продолжают приниматься, передаваться и обрабатываться кадры BPDU.
- **Disable** («Отключен») – в это состояние порт переводит администратор. Отключенный порт не участвует ни в работе протокола STP, ни в продвижении кадров данных. Порт можно также вручную включить, и первоначально он перейдет в состояние «Блокировка».

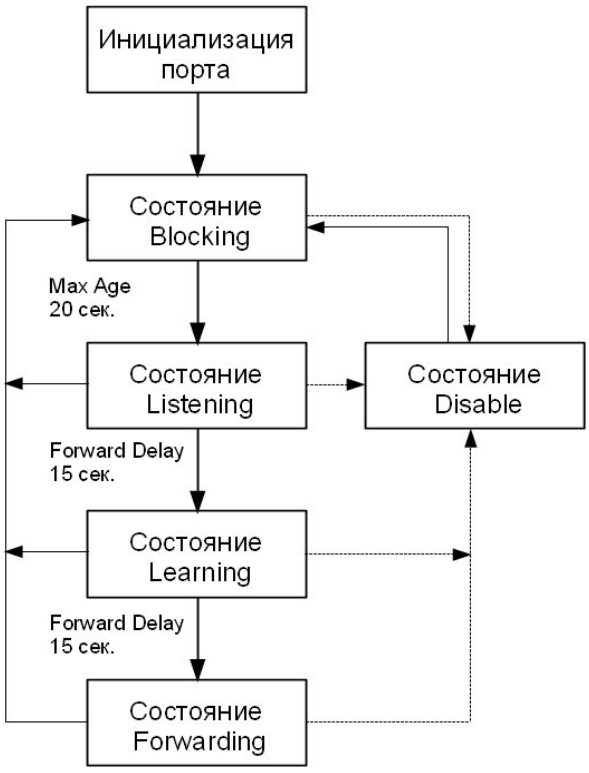


Рис. 5.4. Состояния портов

В процессе нормальной работы корневой мост продолжает генерировать служебные кадры BPDU, а остальные коммутаторы продолжают их принимать своими корневыми портами и ретранслировать назначенными. Если по истечении максимального времени жизни сообщения (по умолчанию – 20 секунд) корневой порт любого коммутатора сети не получит служебный кадр BPDU, то он инициализирует новую процедуру построения связующего дерева.

5.2.5 Таймеры STP

Для того чтобы все коммутаторы сети имели возможность получить точную информацию о конфигурации связующего дерева, в протоколе STP используются следующие таймеры:

- **Hello Time** – это интервал времени, через который корневой мост отправляет конфигурационные BPDU. Значение таймера Hello Time, настроенное на корневом мосте, будет определять значения таймеров Hello Time на всех некорневых коммутаторах, т.к. они просто пересылают конфигурационные BPDU, когда получают их от корня. Значение таймера Hello Time по умолчанию 2 секунды, диапазон возможных значений от 1 до 10 секунд.
- **Forward Delay** – это интервал времени, в течение которого порт коммутатора находится в состояниях «Прослушивание» и «Обучение». Такая задержка смены состояний необходима, чтобы исключить возможность временного возникновения альтернативных маршрутов при неодновременной смене состояний портов во время реконфигурации. Значение таймера Forward Delay по умолчанию 15 секунд, диапазон возможных значений от 4 до 30 секунд.
- **Max Age** – это интервал времени, в течение которого коммутатор хранит параметры текущей конфигурации связующего дерева. Значение таймера Max Age устанавливается корневым мостом и позволяет гарантировать, что все коммутаторы сети обладают одинаковой информацией о времени хранения конфигурации STP. Если период времени, определенный таймером истек, а коммутатор за это время не получил кадр BPDU от корневого моста, то он начинает считать себя корневым мостом и рассылает свои собственные BPDU всем коммутаторам сети, инициируя новую процедуру построения связующего дерева. Значение таймера Max Age по умолчанию 20 секунд, диапазон возможных значений от 6 до 40 секунд.

Значения таймеров Hello Time, Forward Delay и Max Age могут быть вручную настроены администратором сети на коммутаторе. Обычно эти настройки выполняются только на коммутаторе, являющемся корневым для данной топологии связующего дерева. При настройке важно помнить, что неправильно подобранные значения таймеров могут значительно увеличить время сходимости топологии STP и снизить производительность сети, поэтому рекомендуется использовать значения таймеров по умолчанию.

5.2.6 Изменение топологии

Коммутатор отправляет BPDU с уведомлением об изменении топологии (Topology Change Notification BPDU, TCN BPDU) в случае возникновения одного из следующих событий:

- некорневой мост получает сообщение TCN BPDU на свой назначенный порт;
- после истечения времени, определенного таймером Forward Delay, порт переходит в состояние Forwarding, но коммутатор уже имеет назначенный порт для данного сегмента;
- порт, находившийся в состоянии Forwarding или Listening, переходит в состояние Blocking (в случае проблем с каналом связи);
- когда коммутатор становится корневым мостом.

TCN BPDU отправляется коммутатором в тот сегмент сети, к которому подключен его корневой порт. Эти BPDU будут передаваться через интервал Hello до тех пор, пока коммутатор не получит подтверждение Topology Change Notification Acknowledgement (TCN-ACK) от вышестоящего коммутатора. Соседний коммутатор продолжит трансляцию TCN BPDU через свой корневой порт в направлении корневого моста сети, используя такую же процедуру. Этот процесс будет продолжаться до тех пор, пока TCN BPDU не достигнет корневого моста.

Когда корневой мост получает TCN BPDU или сам изменяет топологию, он устанавливает во всех передаваемых конфигурационных BPDU флаг изменения топологии (Topology Change, TC) на период времени, равный сумме значений таймеров Forward Delay и Max Age. Когда нижележащие коммутаторы получат конфигурационные BPDU с флагом Topology Change, они установят значения таймеров старения записей адресных таблиц (Aging Timer) равными длительности таймера задержки передачи Forward Delay.

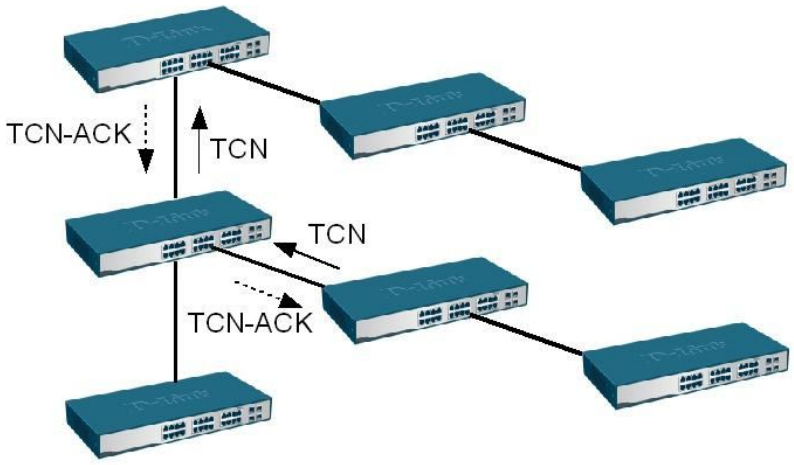


Рис. 5.5. Процесс уведомления об изменении топологии

Управляемые коммутаторы D-Link при настройке функции STP позволяют включать и отключать на каждом порте возможность приема TCN BPDU с помощью параметра *restricted_tcn*. По умолчанию параметр restricted_tcn отключен. Использование данного параметра позволяет избежать сетевых атак, связанных с отправкой ложных кадров TCN BPDU.

5.2.7 Настройка STP

Рассмотрим пример настройки STP на коммутаторах D-Link в сети, показанной на рис. 5.6.

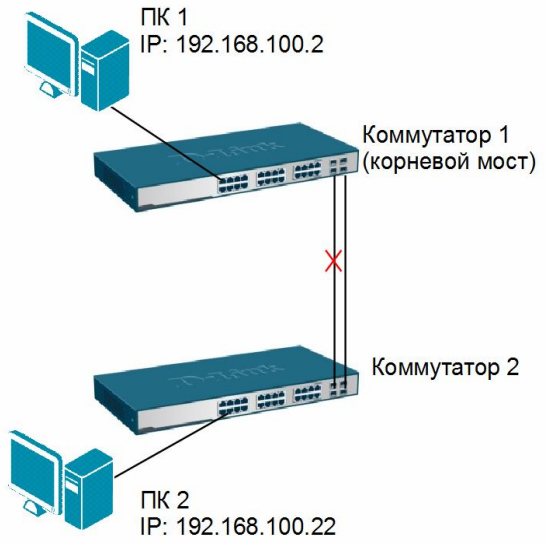


Рис. 5.6. Схема сети

Внимание: по умолчанию протокол STP на коммутаторах D-Link отключен.

Настройка коммутатора 1

- Активизировать STP

enable stp
config stp version stp

- Установить коммутатору 1 наименьшее значение приоритета, чтобы он был выбран корневым мостом (приоритет по умолчанию =32768)

config stp priority 4096 instance_id 0

- Настроить порты STP

config stp ports 1-24 state enable

Настройка коммутатора 2

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

5.3 Rapid Spanning Tree Protocol

Протокол Rapid Spanning Tree Protocol (RSTP) является развитием протокола STP и в настоящее время определен в стандарте IEEE 802.1D-2004 (ранее был определен в стандарте IEEE 802.1w-2001). Он был разработан для преодоления отдельных ограничений протокола STP, связанных с его производительностью. Протокол RSTP значительно ускоряет время сходимости коммутируемой сети за счет мгновенного перехода корневых и назначенных портов в состояние продвижения.

RSTP может работать с оборудованием, поддерживающим STP, однако все преимущества от его использования будут потеряны.

Основные понятия и терминология протоколов STP и RSTP одинаковы. Существенным их отличием является способ перехода портов в состояние продвижения и то, каким образом этот переход влияет на роль порта в топологии. RSTP объединяет состояния

Disabled, Blocking и Listening, используемые в STP, и создает единственное состояние *Discarding* («Отбрасывание»), при котором порт не активен.

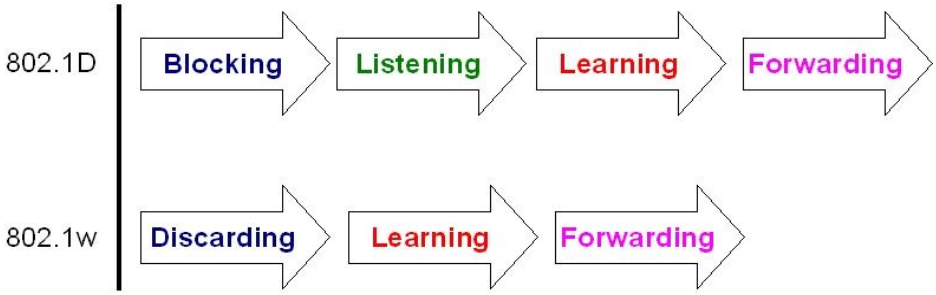


Рис. 5.7. Состояния портов протоколов STP и RSTP

Таблица 3 **Различия между состояниями портов в STP и RSTP**

Состояние порта STP	Административное состояние порта коммутатора	Порт изучает MAC-адреса?	Состояние порта RSTP	Роль порта в активной топологии
Disable	Disabled	Нет	Discarding	Исключен (Disabled)
Disable	Enabled	Нет	Discarding	Исключен (Disabled)
Blocking	Enabled	Да	Discarding	Исключен (Alternate, Backup)
Listening	Enabled	Да	Discarding	Включен (Root, Designated)
Learning	Enabled	Да	Learning	Включен (Root, Designated)
Forwarding	Enabled	Да	Forwarding	Включен (Root, Designated)

5.3.1 Роли портов

Выбор активной топологии завершается присвоением протоколом RSTP определенной роли каждому порту. Эти роли следующие:

- корневой порт (Root Port);
- назначенный порт (Designated Port);
- альтернативный порт (Alternate Port);
- резервный порт (Backup Port).

Корневой порт – это порт коммутатора, который имеет по сети кратчайшее расстояние (в терминах стоимости пути) до корневого коммутатора.

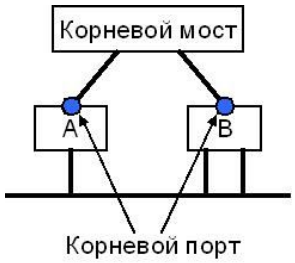


Рис. 5.8. Корневой порт

Порт является **назначенным**, если он посылает BPDU с наилучшими параметрами в тот сегмент, к которому подключен.

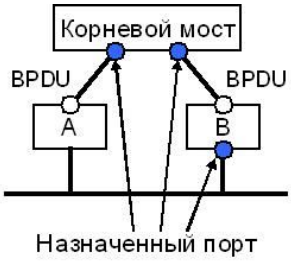


Рис. 5.9. Назначенный порт

Роли «корневой порт» и «назначенный порт» включают порт в активную топологию.

В RSTP существуют две дополнительные роли – альтернативный порт (*Alternate*) и резервный порт (*Backup*), соответствующие состоянию «Заблокирован» в STP и исключающие порт из активной топологии.

Альтернативный порт предлагает альтернативный основному маршруту путь в направлении корневого моста и может заменить корневой порт в случае выхода его из строя.

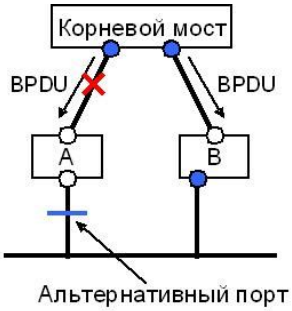


Рис. 5.10. Альтернативный порт

Резервный порт предназначен для резервирования пути, предоставляемого назначенным портом в направлении сегментов сети, и не может гарантировать альтернативное подключение к корневому мосту. Резервные порты существуют только в конфигурациях, где есть два или более соединения данного моста с данной сетью (сегментом сети).

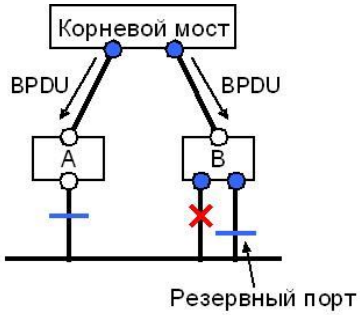


Рис. 5.11. Резервный порт

5.3.2 Формат BPDU

Формат кадра BPDU протокола RSTP аналогичен формату BPDU протокола STP за исключением следующего:

- Поля версии протокола и типа BPDU RSTP содержат значение 2.
- В поле Flag BPDU протокола STP используются только два бита, которые определяют флаги изменения топологии TC и подтверждения TC (TCA). В поле Flag протокола RSTP используются все 8 бит. Бит 1 – флаг изменения топологии (*Topology Change*),

бит 2 – флаг предложения (*Proposal*), биты 3 и 4 предназначены для кодирования роли порта (*Port Role*), бит 5 – флаг изучения (*Learning*), бит 6 – флаг продвижения (*Forwarding*), бит 7 – флаг соглашения (*Agreement*), бит 8 – флаг подтверждения ТС (*Topology Change Acknowledgment*).

- Кадр BPDU протокола RSTP имеет дополнительное поле *Version 1 Length* длиной 1 байт. Это поле содержит значение 0000 0000 и показывает, что BPDU не содержит никакой информации протокола STP версии 1.

	Байты
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

Рис. 5.12. Формат кадра BPDU протокола RSTP

5.3.3 Быстрый переход в состояние продвижения

Процесс построения связующего дерева у протоколов STP и RSTP одинаков. Однако, при работе RSTP, порт может перейти в состояние продвижения значительно быстрее, т.к. он больше не зависит от настроек таймеров. Протокол RSTP предоставляет механизм предложений и соглашений, который обеспечивает быстрый переход корневых и назначенных портов в состояние Forwarding, а альтернативных и резервных портов в состояние Discarding. Для этого протокол RSTP вводит два новых понятия: **граничный порт** и **тип соединения**.

Граничным портом (*Edge Port*) объявляется порт, непосредственно подключенный к сегменту сети, в котором не могут быть созданы петли. Например, порт подключен к рабочей станции, которая может периодически включаться или выключаться и активизировать механизм уведомления об изменении топологии или чтобы избежать распространения вычислений STP по клиентским сетям, с целью повышения безопасности. Граничный порт мгновенно переходит в состояние продвижения, минуя состояния прослушивания и обучения. Граничный порт теряет свой статус и становится обычным портом связующего дерева в том случае, если получит кадр BPDU.

При работе протокола RSTP назначенный порт может выполнять быстрый переход в состояние продвижения в соединениях типа «точка – точка» (*Point-to-Point, P2P*), т.е. если он подключен только к одному коммутатору.

Порты, удовлетворяющие, по крайней мере, одному из следующих условий, автоматически рассматриваются протоколом RSTP как порты P2P:

- порт принадлежит агрегированному каналу связи;

- на порте включена функция автосогласования, и она определила работу в полнодуплексном режиме;

- работа в полнодуплексном режиме на порте была настроена вручную администратором сети.

Администратор сети может вручную включать или выключать статусы Edge и P2P, либо устанавливать их работу в автоматическом режиме, выполнив соответствующие настройки порта коммутатора.

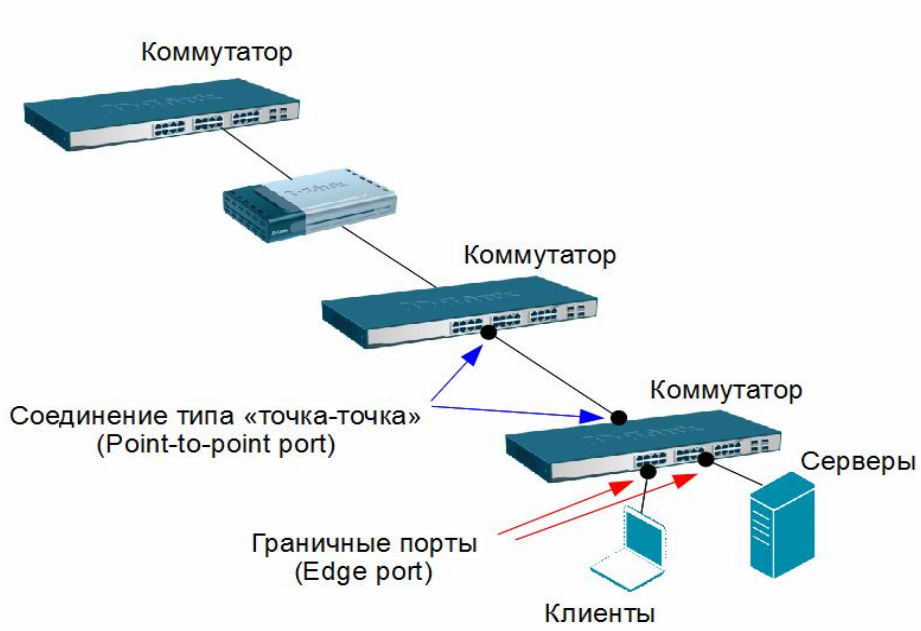


Рис. 5.13. Граничные порты и порты «точка-точка»

5.3.4 Механизм предложений и соглашений

На рис. 5.14 показан процесс работы механизма предложений и соглашений. Коммутаторы A и B соединены между собой каналом типа «точка – точка». Предположим, что коммутатор A является корневым мостом сети. Коммутатор A посылает коммутатору B кадр BPDU с установленным флагом Proposal (шаг 1 на рис. 5.14), предлагая себя в качестве назначенного моста этого сегмента (BPDU-предложение будет передаваться только в том случае, если порт находится в состоянии Discarding или Learning). После получения предложения, коммутатор B выберет в качестве нового корневого порта тот порт, через который этот BPDU был получен (порт p2) и переведет все неграничные порты в заблокированное состояние. Все остальные порты будут синхронизированы с новой информацией, чтобы иметь непротиворечивую информацию о топологии сети.

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

- он находится в заблокированном состоянии (это состояние Discarding в стабильной топологии);
- он является граничным портом.

Чтобы продемонстрировать действие метода синхронизации на различные типы портов, предположим, что в коммутаторе B имеются граничные порты p3 и p5, и назначенный порт p4. Порты p3 и p5 уже удовлетворяют одному из условий синхронизации. Чтобы находиться в режиме синхронизации (шаг 2 на рис. 5.14), коммутатору B необходимо заблокировать порт p4, переведя его в состояние Discarding.

После того, как коммутатор B убедится, что все порты синхронизированы, он разблокирует свой новый корневой порт (шаг 3 на рис. 5.14) и отправит через него коммутатору A согласие на предложение. Это сообщение является копией BPDU-предложения, в котором вместо бита Proposal установлен бит Agreement. Благодаря этому

порт p1 коммутатора A точно знает, какому предложению соответствует полученное согласие. После этого коммутатор A мгновенно переведет свой назначенный порт p1 в состояние продвижения.

Находясь в заблокированном состоянии порт p4 коммутатора B начнет отсылать предложения нижележащему коммутатору и пытаться быстро перейти в состояние продвижения (шаг 4 на рис. 5.14).

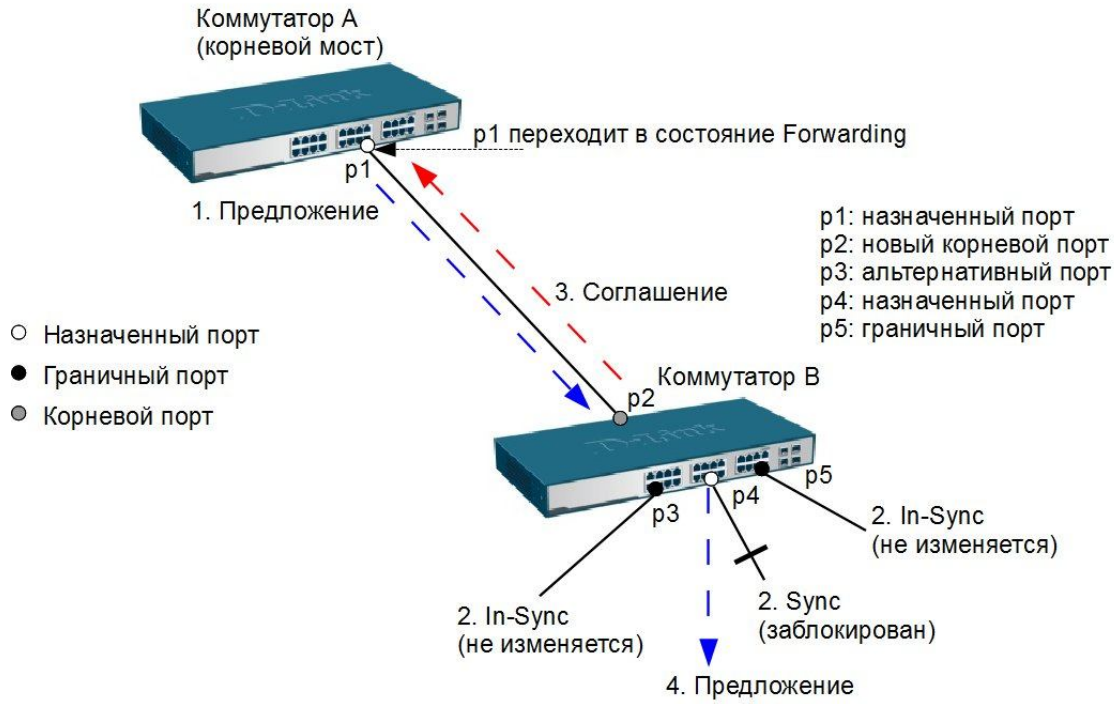


Рис. 5.14. Механизм предложений и соглашений

5.3.5 Новый механизм изменения топологии

1. Определение изменений топологии.

В протоколе RSTP только негранничные порты, переходя в состояние продвижения, могут вызвать процесс изменения топологии. Это означает, что разрыв соединения больше не рассматривается как изменение в топологии, в отличие от протокола STP, т.е. при переходе порта в заблокированное состояние, соответствующий коммутатор не генерирует TCN BPDU. Когда мост RSTP обнаруживает изменение топологии, происходит следующее:

- Коммутатор устанавливает начальное значение таймера TC While равным удвоенному интервалу Hello для всех негранничных назначенных портов и корневого порта. While Timer – это интервал времени, в течение которого мост RSTP активно информирует остальные мосты в сети об изменении топологии.
- Удаляет MAC-адреса, ассоциированные со всеми негранничными назначенными портами и корневым портом.
- До тех пор, пока не истечет время, установленное таймером TC While, запущенным на порте, в BPDU, отправляемых через него, будет установлен бит TC.

2. Распространение информации об изменении топологии.

Когда коммутатор получает от соседа BPDU с установленным битом TC, происходит следующее:

- Коммутатор удаляет все MAC-адреса, изученные его негранничными назначенными портами и корневым портом, за исключением того порта, который получил информацию об изменении топологии.

- Коммутатор запускает таймер TC While и отправляет BPDU с установленным битом TC через все неграничные порты (RSTP не использует специальные TCN BPDU, за исключением случаев, когда требуется уведомить коммутатор, поддерживающий только протокол STP).

Коммутатор-отправитель BPDU с битом TC непосредственно распространяет информацию об изменении топологии через всю сеть (в отличие от STP, где это может выполнить только корневой мост). Этот механизм распространения информации об изменении топологии быстрее, чем его аналог в протоколе STP, т.к. нет необходимости ждать, когда будет уведомлен корневой мост, и потом поддерживать состояние изменения топологии для всей сети в течение периода времени, равного сумме значений таймеров Forward Delay и Max Age.

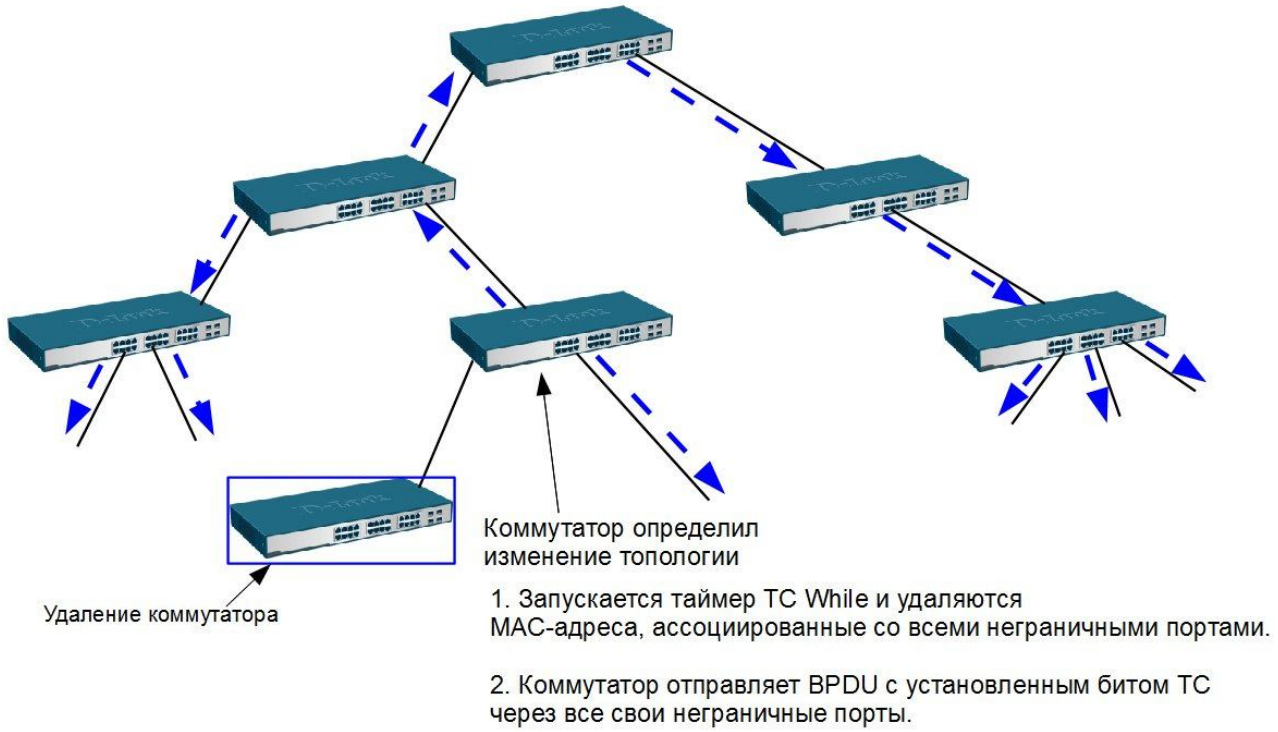


Рис. 5.15. Новый механизм изменения топологии

5.3.6 Стоимость пути RSTP

Протокол RSTP определяет следующие рекомендованные значения стоимости пути по умолчанию для портов коммутаторов. Эти значения вычисляются в соответствии со скоростью канала связи, к которому подключен порт.

Таблица 4 **Стоимость пути RSTP**

Параметр	Скорость канала	Рекомендованное значение	Рекомендованный диапазон	Диапазон значений
Стоимость пути	10 Мбит/с	2 000 000	200 000–20 000 000	1–200 000 000
Стоимость пути	100 Мбит/с	200 000	20 000–2 000 000	1–200 000 000
Стоимость пути	1 Гбит/с	20 000	2 000–200 000	1–200 000 000
Стоимость пути	10 Гбит/с	2 000	200–20 000	1–200 000 000

5.3.7 Совместимость с STP

Протокол RSTP может взаимодействовать с оборудованием, поддерживающим STP и, если необходимо, автоматически преобразовывать кадры BPDU в формат 802.1D. Однако, преимущество быстрой сходимости RSTP (когда все коммутаторы быстро переходят в состояние пересылки или блокировки и обладают тождественной информацией) теряется.

Каждый порт хранит переменную, определяющую тип протокола, используемого в соответствующем сегменте. При включении порта активизируется таймер задержки миграции (*Migration delay timer*), длительностью 3 секунды. При запуске этого таймера, текущий режим (STP или RSTP) ассоциированный с портом, блокируется. Как только истечет время задержки миграции, порт начнет работать в режиме, соответствующем типу следующего полученного им BPDU. Если в результате получения BPDU порт изменил свой режим работы, таймер задержки миграции запускается вновь, что позволяет ограничить частоту возможной смены режимов.

Предположим, что коммутаторы A и B (Рис. 5.16) работают в режиме RSTP. Коммутатор A является выделенным мостом этого сегмента. К существующему каналу связи подключается коммутатор C, который является коммутатором с поддержкой протокола STP. Так как коммутаторы STP игнорируют BPDU протокола RSTP и отбрасывают их, то коммутатор C считает, что в этом сегменте сети больше коммутаторов нет и начинает отправлять BPDU формата 802.1D.

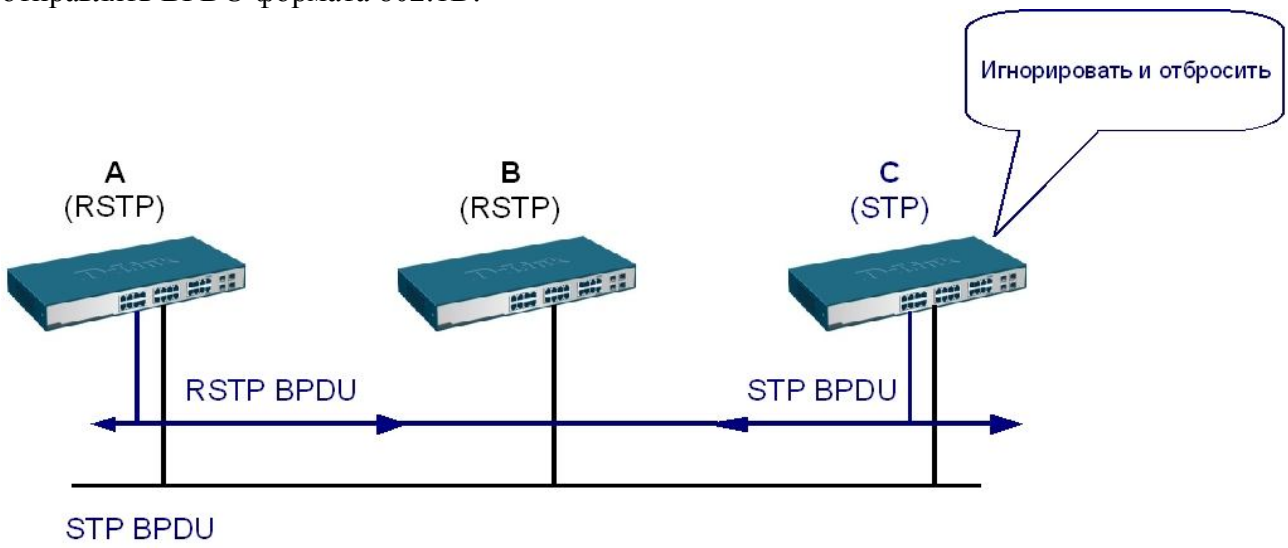


Рис. 5.16. Пример совместной работы коммутаторов STP и RSTP

Коммутатор A получает эти BPDU и, после истечения периода времени, установленного таймером задержки миграции переходит на этом порте в режим работы STP. В результате, коммутатор C начинает понимать BPDU коммутатора A и признает его назначенным коммутатором этого сегмента.

Следует отметить, что если бы в этом частном случае, коммутатор C был удален из сегмента, то коммутатор A остался бы работать в режиме STP на этом порте, хотя он мог бы эффективно работать в режиме RSTP со своим единственным соседом коммутатором B. Т.е. у коммутатора A нет возможности узнать, что коммутатор C удален из этого сегмента. В этом частном случае для перезагрузки протокола, используемого на порте коммутатора, требуется вмешательство администратора сети. Когда порт находится в режиме совместимом с 802.1D, он также может обрабатывать уведомления об изменении топологии TCN BPDU с установленными битами TC и TCA.

5.3.8 Настройка RSTP

Настройка протокола RSTP на коммутаторах D-Link аналогична настройке протокола STP. Рассмотрим пример настройки RSTP, в сети показанной на рис. 5.17.

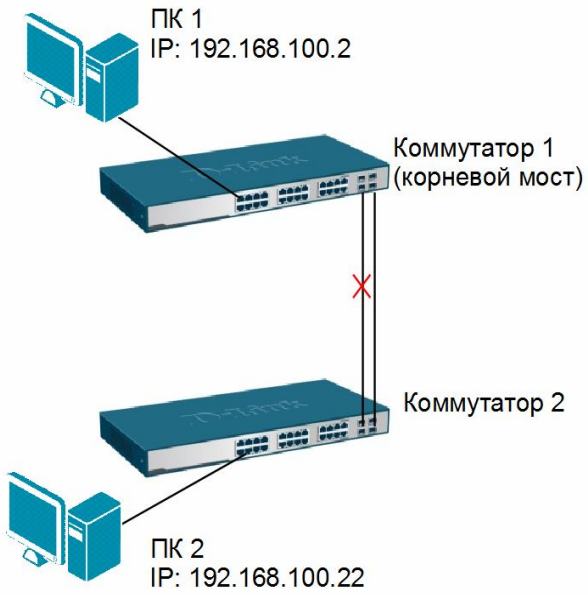


Рис. 5.17. Схема сети

Настройка коммутатора 1

- Активизировать RSTP

enable stp
config stp version rstp

- Установить коммутатору 1 наименьшее значение приоритета, чтобы он был выбран корневым мостом (приоритет по умолчанию =32768)

config stp priority 4096 instance_id 0

- Настроить граничные порты RSTP

config stp ports 1-24 edge true state enable

Настройка коммутатора 2

enable stp
config stp version rstp
config stp ports 1-24 edge true state enable

5.4 Multiple Spanning Tree Protocol

Несмотря на то, что протокол RSTP обеспечивает быструю сходимость сети, он, также как и протокол STP, обладает следующим недостатком – не поддерживает возможность создания отдельного связующего дерева для каждой VLAN настроенной в сети. Это означает, что резервные каналы связи не могут блокироваться на основе VLAN, и все VLAN образуют одну логическую топологию, не обладающую достаточной гибкостью.

Протокол Multiple Spanning Tree Protocol (MSTP), являющий расширением протокола RSTP преодолевает это ограничение. В дополнение к обеспечению быстрой сходимости сети, он позволяет настраивать отдельное связующее дерево для любой VLAN или группы VLAN, создавая множество маршрутов передачи трафика, и позволяя осуществлять балансировку нагрузки. Первоначально протокол MSTP был определен в стандарте IEEE

802.1s, но позднее был добавлен в стандарт IEEE 802.1Q-2003. Протокол MSTP обратно совместим с протоколами STP и RSTP.

5.4.1 Логическая структура MSTP

Протокол MSTP делит коммутируемую сеть на **регионы MST** (*Multiple Spanning Tree (MST) Region*), каждый из которых может содержать множество **копий связующих деревьев** (*Multiple Spanning Tree Instance, MSTI*) с независимой друг от друга топологией. Другими словами регион MST, представляющий собой набор физически подключенных друг к другу коммутаторов, делит данную физическую топологию на множество логических.

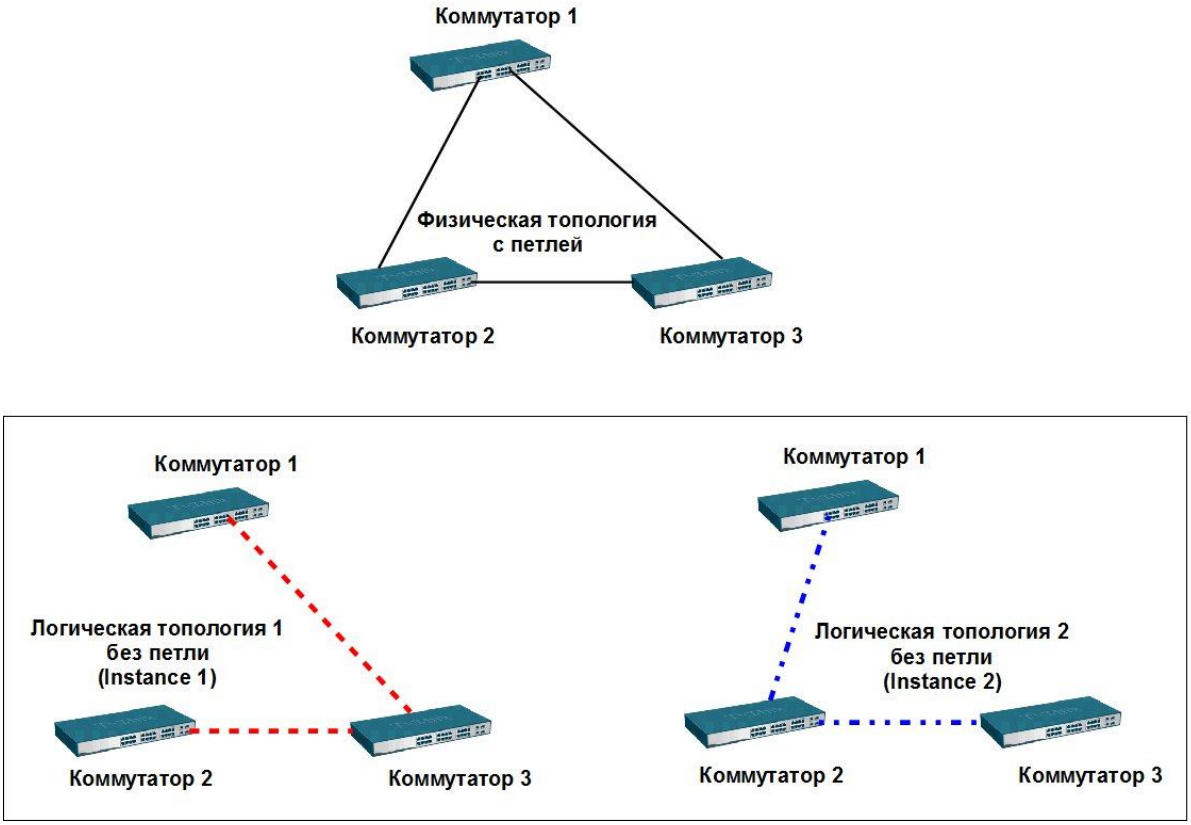


Рис. 5.18. Физическая и логическая топология региона MST

Для того чтобы два и более коммутатора принадлежали одному региону MST, они должны обладать одинаковой конфигурацией MST.

Конфигурация MST включает такие параметры как номер ревизии MSTP (*MSTP revision level number*), имя региона (*Region name*), карту привязки VLAN к копии связующего дерева (*VLAN-to-instance mapping*).

Внутри коммутируемой сети может быть создано множество MST-регионов.

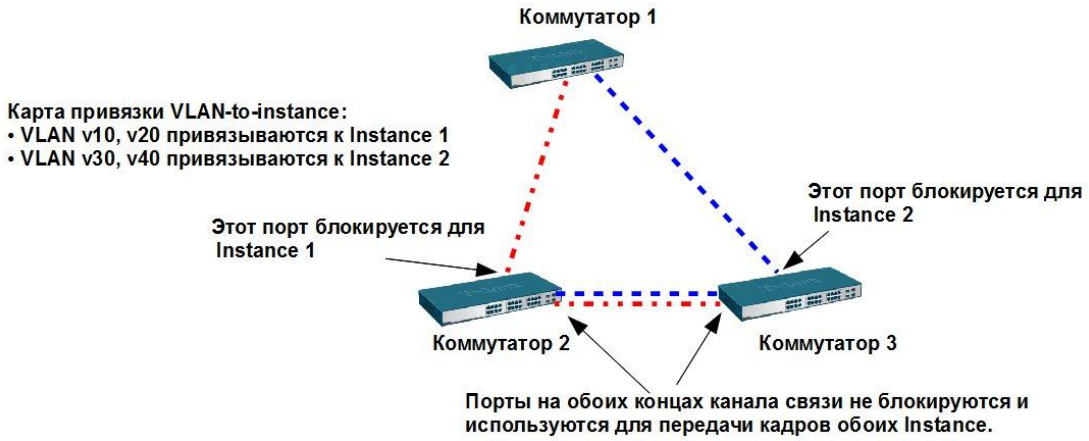


Рис. 5.19. Регион MST

Протокол MSTP определяет следующие типы связующих деревьев:

- **Internal Spanning Tree (IST)** – специальная копия связующего дерева, которая по умолчанию существует в каждом MST-регионе. IST присвоен номер 0 (Instance 0). Она может отправлять и получать кадры BPDU и служит для управления топологией внутри региона. По умолчанию все VLAN одного региона привязаны к IST. Если в регионе создано несколько MSTI, то VLAN не ассоциированные с ними, остаются привязанными к IST. Динамические VLAN, созданные с помощью протокола GVRP также ассоциируются с IST.
- **Common Spanning Tree (CST)** – единое связующее дерево, вычисленное с использованием протоколов STP, RSTP, MSTP и объединяющее все регионы MST и мосты SST (Single Spanning Tree (SST) Bridge).
- **Common and Internal Spanning Tree (CIST)** – единое связующее дерево, объединяющее CST и IST каждого MST-региона.
- **Single Spanning Tree (SST) Bridge** – это мост, поддерживающий только единственное связующее дерево, CST. Это единственное связующее дерево может поддерживать протокол STP или протокол RSTP.

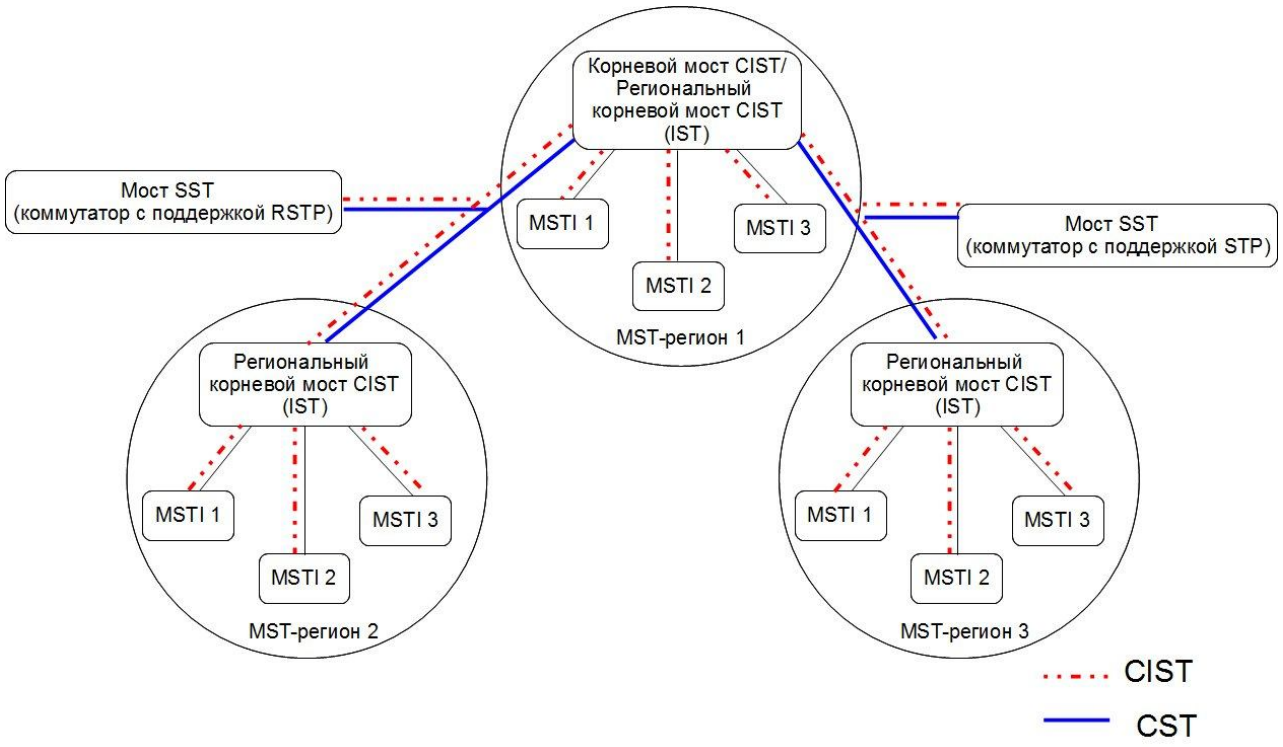


Рис. 5.20. Логическая структура MSTP

5.4.2 Multiple Spanning Tree Instance (MSTI)

По умолчанию все VLAN данного MST-региона назначены в IST. Помимо IST, в каждом MST-регионе может быть дополнительно создано множество связующих деревьев с независимой друг от друга архитектурой - MSTI. К каждой MSTI администратор сети может вручную привязать соответствующие сети VLAN.

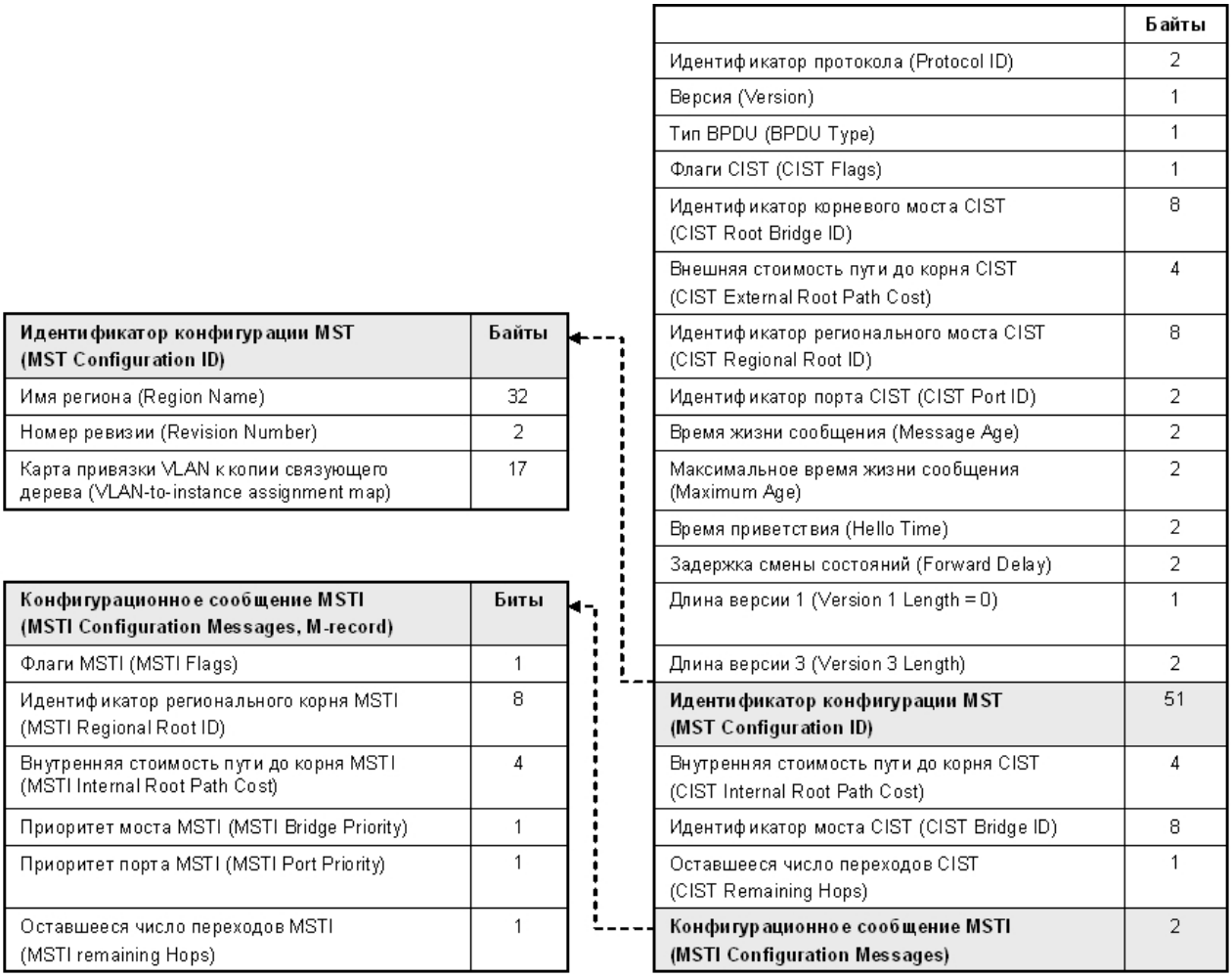
MSTI обладают следующими характеристиками:

- MSTI является копией связующего дерева, существующей только внутри региона;
- MSTI не может отправлять BPDU за пределы своего региона (отправлять и получать BDPU может только IST);
- все MSTI внутри региона могут нумероваться 1, 2, 3, 4 и т.д. (максимальное количество MSTI зависит от модели коммутатора и версии программного обеспечения);
- MSTI не отправляет индивидуальные BPDU. Вся информация о данной MSTI помещается в конфигурационное сообщение MSTI (MSTI Configuration Message, M-record), которое инкапсулируется в кадры MSTP BPDU, рассылаемые IST.

Для того чтобы каждая MSTI представляла собой отдельную от IST логическую топологию, администратор сети может присвоить коммутаторам и портам внутри MSTI собственные значения приоритетов и стоимости пути.

5.4.3 Формат MSTP BPDU

Формат MSTP BPDU аналогичен формату RSTP BPDU за исключением полей, предназначенный для передачи информации об IST, каждой MSTI (если они созданы в регионе) и конфигурации MST.



	Байты
Идентификатор протокола (Protocol ID)	2
Версия (Version)	1
Тип BPDU (BPDU Type)	1
Флаги CIST (CIST Flags)	1
Идентификатор корневого моста CIST (CIST Root Bridge ID)	8
Внешняя стоимость пути до корня CIST (CIST External Root Path Cost)	4
Идентификатор регионального моста CIST (CIST Regional Root ID)	8
Идентификатор порта CIST (CIST Port ID)	2
Время жизни сообщения (Message Age)	2
Максимальное время жизни сообщения (Maximum Age)	2
Время приветствия (Hello Time)	2
Задержка смены состояний (Forward Delay)	2
Длина версии 1 (Version 1 Length = 0)	1
Длина версии 3 (Version 3 Length)	2
Идентификатор конфигурации MST (MST Configuration ID)	51
Внутренняя стоимость пути до корня CIST (CIST Internal Root Path Cost)	4
Идентификатор моста CIST (CIST Bridge ID)	8
Оставшееся число переходов CIST (CIST Remaining Hops)	1
Конфигурационное сообщение MSTI (MSTI Configuration Messages)	2

Идентификатор конфигурации MST (MST Configuration ID)	**Байты**
Имя региона (Region Name)	32
Номер ревизии (Revision Number)	2
Карта привязки VLAN к копии связующего дерева (VLAN-to-instance assignment map)	17

Конфигурационное сообщение MSTI (MSTI Configuration Messages, M-record)	**Биты**
Флаги MSTI (MSTI Flags)	1
Идентификатор регионального корня MSTI (MSTI Regional Root ID)	8
Внутренняя стоимость пути до корня MSTI (MSTI Internal Root Path Cost)	4
Приоритет моста MSTI (MSTI Bridge Priority)	1
Приоритет порта MSTI (MSTI Port Priority)	1
Оставшееся число переходов MSTI (MSTI remaining Hops)	1

Рис. 5.21. Формат MSTP BPDU

5.4.4 Вычисления в MSTP

Прежде чем начать рассматривать вопрос вычисления активной топологии MSTP, следует отметить, что спецификация MSTP рассматривает MST-регион как один виртуальный мост RSTP и вычисление активной топологии внутри региона отделено от вычисления топологии всей коммутируемой сети. Другими словами, соединения между мостами внутри региона не зависят от внешних соединений между MST-регионами.

Процесс вычисления в MSTP начинается с выбора **корневого моста CIST** (*CIST Root*) сети. При выборе CIST Root используется тот же фундаментальный алгоритм, который описан в стандарте IEEE 802.1D-2004.

Первоначально каждый коммутатор сети считает себя корневым мостом CIST и рассылает BPDU, в поле CIST Root Bridge ID которых указано значение его идентификатора, а **внешняя стоимость пути до корня** (*CIST External Root Path Cost*) равна нулю. Эти два параметра являются основными для определения активной топологии всей коммутируемой сети.

Коммутатор перестанет заявлять себя в качестве корневого моста CIST, как только получит BPDU с меньшим значением идентификатора моста. В качестве CIST Root будет выбран коммутатор, обладающий наименьшим значением идентификатора моста среди всех коммутаторов сети.

Одновременно с выбором корневого моста CIST в каждом регионе выбирается **региональный корневой мост CIST** (*CIST Regional Root*). Им становится коммутатор, обладающий наименьшей внешней стоимостью пути к корню CIST среди всех коммутаторов принадлежащих данному региону. Внешняя стоимость пути до корня CIST представляет

собой суммарное условное время пути от граничного коммутатора MST-региона или моста SST до порта корневого моста CIST. Следует отметить, что значение CIST External Root Path Cost не изменяется при передаче конфигурационного BPDU между коммутаторами внутри региона. Это значение увеличивается на условное время передачи только портами граничных коммутаторов, подключающих данный регион к другим регионам. В протоколе MSTP рекомендованные значения стоимости пути аналогичны значениям, определенным в RSTP.

В случае наличия в регионе коммутаторов с одинаковой стоимостью пути, в качестве CIST Regional Root будет выбран коммутатор с наименьшим значением идентификатора моста. При этом маршрут от этого коммутатора до CIST Root не должен проходить через другие коммутаторы этого региона. Т.е. в качестве CIST Regional Root выбирается коммутатор, находящийся на границе региона. Следует отметить, что регион, содержащий CIST Root, использует его также в качестве CIST Regional Root.

Протокол MSTP, используя механизм предложений и соглашений RSTP, блокирует все избыточные каналы связи от всех CIST Regional Root к CIST Root, делая их резервными или альтернативными.

При наличии в регионе отдельных связующих деревьев MSTI, для каждой MSTI, независимо от остальных, выбирается **региональный корневой мост MSTI** (*MSTI Regional Root*). Им становится коммутатор, обладающий наименьшим значением идентификатора моста среди всех коммутаторов данной MSTI этого MST-региона. Определение роли портов и блокирование избыточных связей происходит аналогично процедурам, описанным в RSTP.

5.4.5 Роли портов MSTP

Протокол MSTP определяет роли портов, которые участвую в процессе вычисления активной топологии CIST и MSTI аналогичные протоколам STP и RSTP:

- корневой порт (Root Port);
- назначенный порт (Designated Port);
- альтернативный/резервный порт (Alternate/Backup Port).

Дополнительно в MSTI используется еще одна роль, которая может быть присвоена порту **- мастер-порт** (*Master Port*).

Роли портов CIST определяют роли каждого порта коммутатора, участвующего в построении активной топологии CIST:

Корневой порт (*Root Port*) – это порт, который обладает минимальной стоимостью пути от коммутатора до корневого моста CIST (в случае, если мост не является CIST Root) через региональный мост (в том случае, если коммутатор не является региональным корнем CIST).

Назначенный порт (*Designated Port*) – это порт, обладающий наименьшей стоимостью пути от подключенного сегмента сети до корневого моста CIST.

Альтернативный/резервный порт (*Alternate/Backup Port*) – это порт, который обеспечивает подключение, если происходит потеря соединения с какими-либо коммутаторами или сегментами сети.

Роли портов MSTI определяют роли каждого порта коммутатора, участвующего в построении активной топологии MSTI внутри границы региона:

Корневой порт (*Root Port*) – это порт, который обладает минимальной стоимостью пути от коммутатора до регионального корневого моста MSTI (в случае, если мост не является региональным корнем для этой MSTI).

Назначенный порт (*Designated Port*) – это порт, обладающий наименьшей стоимостью пути от подключенного сегмента сети до регионального корневого моста MSTI.

Альтернативный/резервный порт (*Alternate/Backup Port*) – это порт, который обеспечивает подключение, если происходит потеря соединения с какими-либо коммутаторами или сегментами сети.

Мастер-порт (*Master Port*) – это порт, который обеспечивает подключение региона к корневому мосту CIST, находящемуся за пределами данного региона. Корневой порт CIST регионального корневого моста CIST является мастером-портом для всех MSTI.

Протокол MSTP вводит еще одну роль, которая может быть присвоена порту – **пограничный порт** (*Boundary Port*). Пограничным портом является порт, который подключает MST-регион к другому региону или SST-мосту.

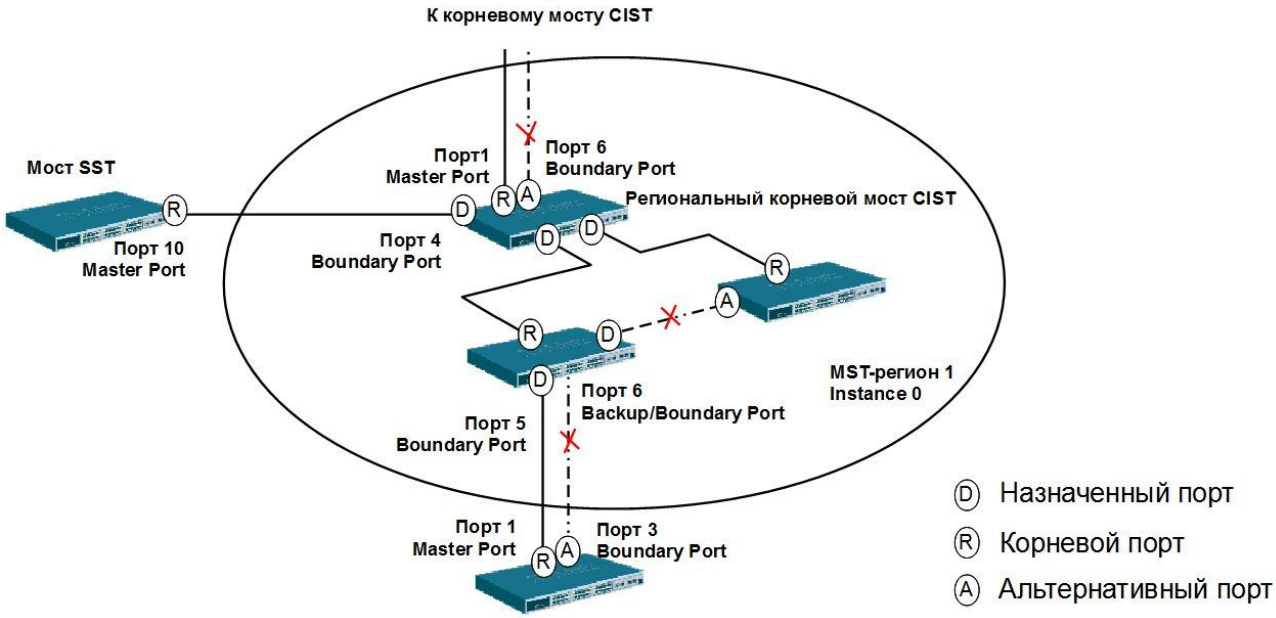


Рис. 5.22. Роли портов

5.4.6 Пример топологии MSTP

Рассмотрим пример топологии MSTP, приведенный на рис. 5.23. Сеть разбита на 3 MST-региона, в каждом регионе все коммутаторы ассоциированы с Instance 0.

1) Коммутатор 1 (SW-1) выбран в качестве корневого моста CIST, т.к. он обладает наименьшим среди всех коммутаторов сети значением идентификатора моста.
2) Коммутаторы 1, 2 и 3 (SW-1, SW-2, SW-3) находятся в одном MST-регионе с номером 1, т.к. обладают одинаковым идентификатором MST-конфигурации. Коммутаторы 2 и 3 находятся в одном регионе с корневым мостом CIST (коммутатор 1), поэтому их внешняя стоимость пути равна 0 и их региональный мост CIST совпадает с корневым мостом CIST.
3) Коммутаторы 4-10 (SW-4-SW-10) принадлежат одному региону, т.к. имеют одинаковые идентификаторы MST-конфигурации. Коммутатор 4 (SW-4) является региональным корневым мостом CIST для MST-региона 2, т.к. обладает наименьшей внешней стоимостью пути к CIST Root.
4) Коммутаторы 11, 12 и 13 (SW-11-SW-13) принадлежат к MST-региону 3, т.к. обладают одинаковыми идентификаторами MST-конфигурации. Коммутатор 11 (SW-11) выбран в качестве регионального корневого моста CIST для MST-региона 3, т.к. обладает наименьшей внешней стоимостью пути к CIST Root.

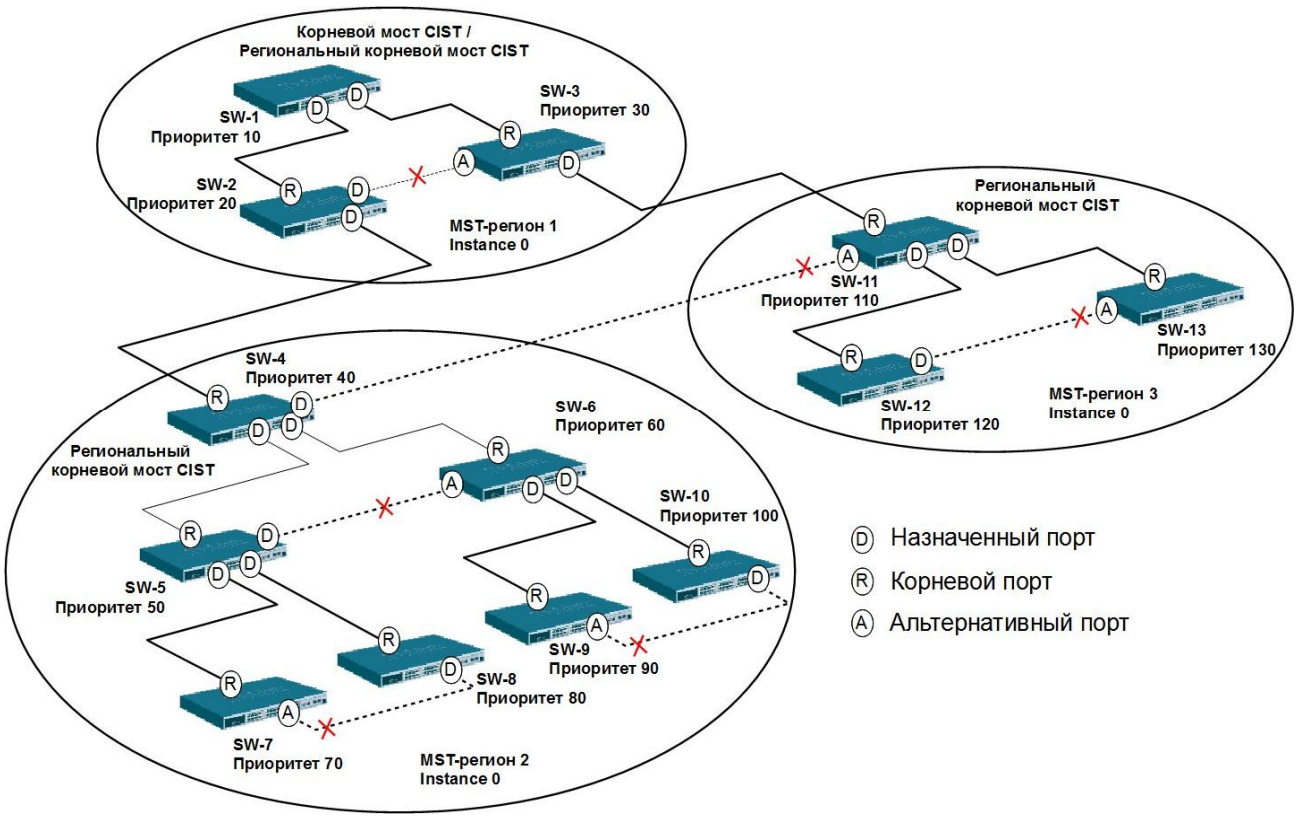


Рис. 5.23. Пример топологии MSTP

5.4.7 Состояние портов MSTP

В протоколе MSTP определены состояния, в которых могут находиться порты, аналогичные протоколу RSTP:

- **Learning** (Обучение) – порт может принимать/отправлять кадры BPDU, изучать MAC-адреса и строить таблицу коммутации. Порт в этом состоянии не передает пользовательские кадры.
- **Forwarding** (Продвижение) – в этом состоянии порт может передавать пользовательские кадры, изучать новые MAC-адреса и принимать/отправлять кадры BPDU.
- **Discarding** (Отбрасывание) – в этом состояние порт может только принимать кадры BPDU, передача пользовательского трафика и изучение MAC-адресов не выполняется.

5.4.8 Счетчик переходов MSTP

При вычислении активной топологии связующего дерева, IST и MSTI не используют значения полей Max Age и Message Age конфигурационного BPDU для отбрасывания устаревших сообщений. Вместо этого, используется механизм счетчика переходов (Hop count).

С помощью команды **config stp maxhops** на коммутаторах D-Link можно настроить максимальное число переходов между устройствами внутри региона, прежде чем кадр BPDU будет отброшен. Значение счетчика переходов устанавливается региональным корневым мостом MSTI или CIST и уменьшается на 1 каждым портом коммутатора, получившим кадр BPDU.

Внимание: значение счетчика переходов зависит от модели коммутатора. По умолчанию используется значение счетчика переходов равное 20.

После того как значение счетчика станет равным 0, кадр BPDU будет отброшен, и информация, хранимая портом, будет помечена как устаревшая.

Следует отметить, что коммутаторы не изменяют данные, хранимые в полях Max Age и Message Age конфигурационных BPDU при их передаче через коммутаторы MST-региона. Значение Message Age изменяется только коммутаторами, расположенными на границе региона, чтобы обеспечить совместимость с мостами STP и RSTP, которые могут использоваться в сети.

5.4.9 Настройка протокола MSTP на коммутаторах

Ниже приведены основные шаги, которые позволяют настроить протокол MSTP на коммутаторах D-Link:

1. Активизировать STP на всех устройствах.
2. Изменить версию STP на MSTP (по умолчанию используется RSTP).
3. Настроить имя MST-региона и ревизию.
4. Создать MSTI и карту привязки VLAN к MSTI.
5. Задать приоритет STP для выбора корневого моста. По умолчанию используется приоритет 32768.
6. Настроить приоритеты портов.
7. Настроить граничные порты.

Рассмотрим пример, показанный на рис. 5.24. В сети созданы две виртуальные локальных сети – VLAN v2 и VLAN v3. Каждая VLAN привязывается к одной копии связующего дерева.

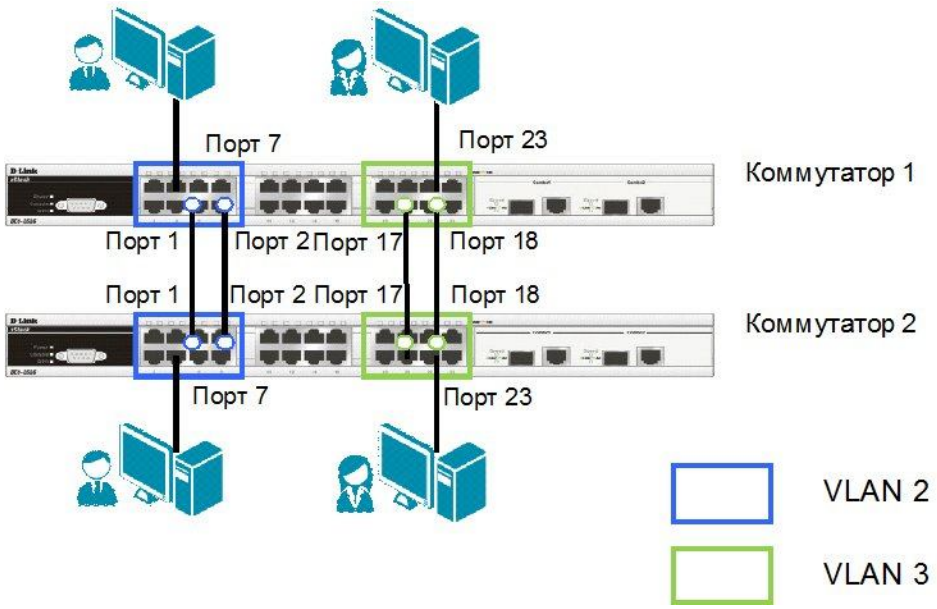


Рис. 5.24. Схема сети

Настройка коммутатора 1

- Создание VLAN

config vlan default delete 1-8,17-24
create vlan v2 tag 2
config vlan v2 add untagged 1-8
create vlan v3 tag 3
config vlan v3 add untagged 17-24

- Настройка MSTP

enable stp
config stp version mstp
config stp mst_config_id name dlink revision_level 1
create stp instance_id 2
config stp instance_id 2 add_vlan 2
create stp instance_id 3
config stp instance_id 3 add_vlan 3
config stp priority 4096 instance_id 0
config stp priority 4096 instance_id 2
config stp priority 4096 instance_id 3
config stp ports 7,23 edge true

Настройка коммутатора 2

- Создание VLAN

config vlan default delete 1-8,17-24
create vlan v2 tag 2
config vlan v2 add untagged 1-8
create vlan v3 tag 3
config vlan v3 add untagged 17-24

- Настройка MSTP

enable stp
config stp version mstp
config stp mst_config_id name dlink revision_level 1
create stp instance_id 2
config stp instance_id 2 add_vlan 2
create stp instance_id 3
config stp instance_id 3 add_vlan 3
config stp ports 7,23 edge true

Рассмотрим второй пример настройки протокола MSTP, позволяющий осуществлять балансировку нагрузки между каналами связи.

В примере, показанном на рис. 5.25, каждая VLAN привязывается к одной копии связующего дерева. Порты 25 и 26 являются маркированными портами обеих VLAN. Порт 25 используется в качестве активного канала связи для VLAN v2, порт 26 используется в качестве активного канала связи для VLAN v3. Т.к. для каждой VLAN будет построена своя собственная активная топология связующего дерева, то кадры VLAN v2 и VLAN v3 будут передаваться по разным маршрутам (через порты 25 и 26 соответственно), благодаря чему будет обеспечена балансировка нагрузки. В случае если один из каналов связи выйдет из строя, трафик VLAN v2 и VLAN v3 будет передаваться по одному оставшемуся каналу.

Настройка коммутатора 1

• Создание VLAN

config vlan default delete 1-8,17-24
create vlan v2 tag 2
config vlan v2 add tagged 25-26
config vlan v2 add untagged 1-8
create vlan v3 tag 3
config vlan v3 add tagged 25-26
config vlan v3 add untagged 17-24

• Настройка MSTP
enable stp
config stp version mstp
config stp mst_config_id name dlink revision_level 1
create stp instance_id 2
config stp instance_id 2 add_vlan 2
create stp instance_id 3
config stp instance_id 3 add_vlan 3
config stp ports 7,23 edge true

Настройка коммутатора 2
• Создание VLAN
config vlan default delete 1-8,17-24
create vlan v2 tag 2
config vlan v2 add tagged 25-26
config vlan v2 add untagged 1-8
create vlan v3 tag 3
config vlan v3 add tagged 25-26
config vlan v3 add untagged 17-24

• Настройка MSTP
enable stp
config stp version mstp
config stp mst_config_id name dlink revision_level 1
create stp instance_id 2
config stp instance_id 2 add_vlan 2
create stp instance_id 3
config stp instance_id 3 add_vlan 3
config stp mst_ports 25 instance_id 2 priority 96
config stp mst_ports 26 instance_id 2 priority 128
config stp mst_ports 25 instance_id 3 priority 128
config stp mst_ports 26 instance_id 3 priority 96
config stp ports 7,23 edge true

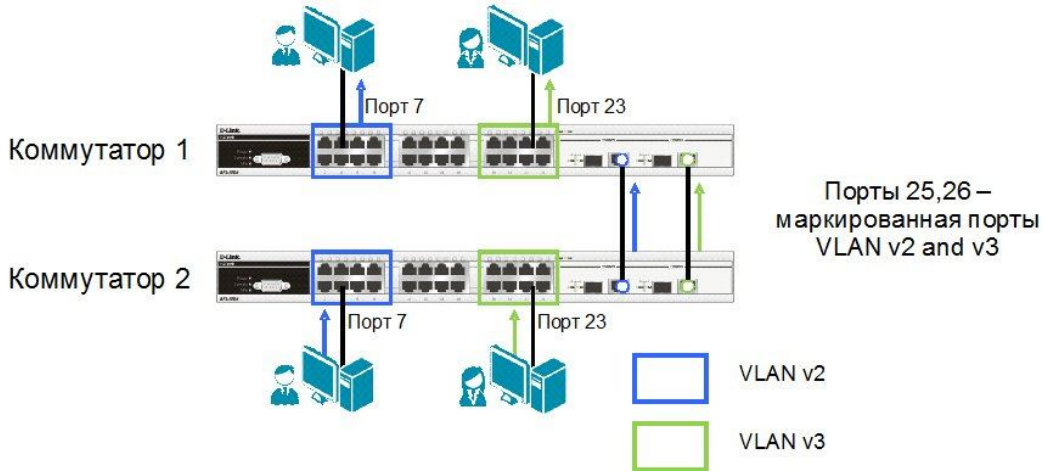


Рис. 5.25. Балансировка нагрузки с помощью MSTP

5.5 Дополнительные функции защиты от петель

Функция LoopBack Detection (LBD) обеспечивает дополнительную защиту от образования петель на уровне 2 модели OSI. Существует две реализации этой функции:

- STP LoopBack Detection;
- LoopBack Detection Independent STP.

На рис. 5.26 показана ситуация, когда к порту управляемого коммутатора подключен неуправляемый коммутатор, порты которого соединены с образованием петли. В этом случае в сети может возникнуть широковещательный шторм и ее работоспособность будет нарушена.

Функция STP LoopBack Detection предназначена для отслеживания таких ситуаций и временного блокирования тех портов коммутатора, на которых обнаружены петли, тем самым, предотвращая проблемы в сети. Коммутатор определяет наличие петли, когда отправленный им кадр BPDU вернулся назад на другой его порт. В этом случае порт-источник кадра BPDU и порт-приемник будут автоматически заблокированы, и администратору сети будет отправлен служебный пакет-уведомление. Порты будут находиться в заблокированном состоянии до истечения периода времени, установленного таймером LBD Recover Timer. По умолчанию на коммутаторах D-Link эта функция отключена. Следует отметить, что функция STP LoopBack Detection не определяет петлю, когда отправленный кадр BPDU вернулся назад на этот же порт.

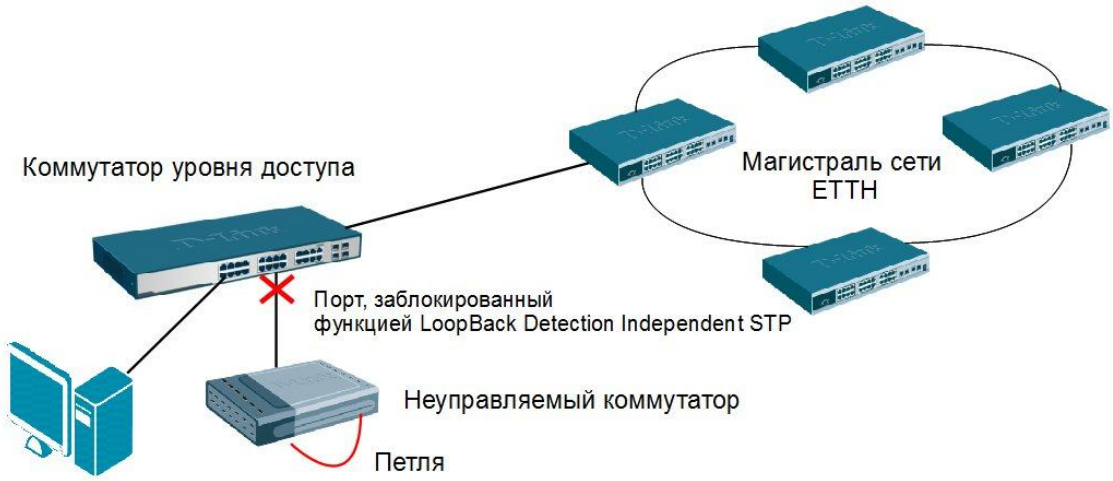


Рис. 5.26. Пример работы функции LoopBack Detection Independent STP

В отличие от STP LoopBack Detection, функция LoopBack Detection Independent STP не требует настройки протокола STP на портах, на которых необходимо определять наличие петли. В этом случае наличие петли обнаруживается путем отправки портом специального служебного кадра ECTP (Ethernet Configuration Testing Protocol). При получении кадра ECTP этим же портом, он блокируется на указанное в таймере время.

Следует отметить, что функция LoopBack Detection Independent STP версии 4.03 также может определять петли, возникающие между портами одного коммутатора.

Внимание: чтобы получить информацию о поддержке коммутатором функции LBD v.4.03 необходимо обратиться в службу технической поддержки D-Link.

Существуют два режима работы этой функции: Port-Based и VLAN-Based (начиная с LBD версии v.4.00).

В режиме Port-Based при обнаружении петли происходит автоматическая блокировка порта, и никакой трафик через него не передается.

В режиме VLAN-Based порт будет заблокирован для передачи трафика только той VLAN, в которой обнаружена петля. Остальной трафик через этот порт будет передаваться.

5.5.1 Настройка функции LoopBack Detection

В качестве примера приведем настройку функций STP LoopBack Detection и LoopBack Detection Independent STP в режимах Port-Based и VLAN-Based для ситуации, показанной на рис. 5.26.

Настройка функции STP LoopBack Detection
enable stp
config stp ports 1-24 state enable edge true lbd enable
config stp lbd_recover_timer 60

Настройка функции LoopBack Detection Independent STP (Port-Based)
enable loopdetect
config loopdetect recover_timer 60
config loopdetect interval 10
config loopdetect mode port-based
config loopdetect ports 1-24 state enabled

Настройка функции LoopBack Detection Independent STP (VLAN-Based)
enable loopdetect
config loopdetect recover_timer 60
config loopdetect interval 10
config loopdetect mode vlan-based
config loopdetect ports 1-24 state enabled

Внимание: таймер recover_timer – интервал времени в секундах, через который будет проверяться статус заблокированного функцией LBD порта. Если установить значение таймера равным 0, заблокированный порт не сможет быть автоматически разблокирован, и для его восстановления потребуется вмешательство администратора. Значение таймера задается глобально на коммутаторе.

loopdetect interval – временной интервал в секундах между отсылаемыми кадрами ECTP (Ethernet Configuration Testing Protocol).

5.6 Функции безопасности STP

Из-за ошибок в конфигурации или вредоносных атак в сети может возникнуть ситуация, когда корневой мост получит кадр BPDU, содержащий меньшее значение приоритета, и потеряет свою позицию. При настройке протоколов RSTP или MSTP на управляемых коммутаторах, расположенных на границе сети, с помощью параметра *restricted_role* можно ограничить роли выполняемые портом в активной топологии. При активизации этого параметра порт не будет выбран в качестве корневого порта даже в том случае, если получит BPDU с наименьшим значением приоритета. После выбора корневого порта, этот порт будет выбран в качестве альтернативного. По умолчанию функция restricted_role отключена.

Настройка коммутатора
enable stp
config stp version rstp
config stp priority 32768 instance_id 0
config stp ports 1-24 edge true restricted_role true restricted_tcn true state enable

config stp ports 25-28 edge false state enable fbpdu enable

5.7 Агрегирование каналов связи

Агрегирование каналов связи (Link Aggregation) – это объединение нескольких физических портов в одну логическую магистраль на канальном уровне модели OSI с целью образования высокоскоростного канала передачи данных и повышения отказоустойчивости.

В отличие от протокола STP, все избыточные связи в одном агрегированном канале остаются в рабочем состоянии, а имеющийся трафик распределяется между ними для достижения балансировки нагрузки. При отказе одной из линий, входящих в такой логический канал, трафик распределяется между оставшимися линиями.

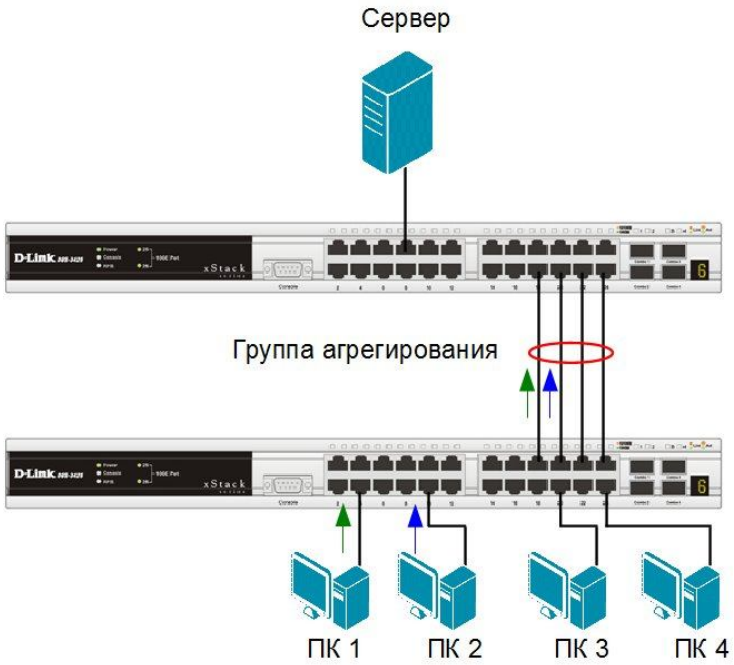


Рис. 5.27. Пример агрегированного канала связи между коммутаторами

Включенные в агрегированный канал порты называются членами **группы агрегирования** (*Link Aggregation Group*).

Внимание: количество портов в группе зависит от модели коммутатора. В управляемых коммутаторах в группу можно объединить до 8 портов.

Один из портов в группе выступает в качестве **мастера-порта** (*master port*). Так как все порты агрегированной группы должны работать в одном режиме, конфигурация мастера-порта распространяется на все порты в группе. Таким образом, при конфигурировании портов в группе агрегирования достаточно настроить мастер-порт.

Важным моментом при реализации объединения портов в агрегированный канал является распределение трафика по ним. Если пакеты одного сеанса будут передаваться по разным портам агрегированного канала, то может возникнуть проблема на более высоком уровне модели OSI. Например, если два или более смежных кадров одного сеанса станут передаваться через разные порты агрегированного канала, то из-за неодинаковой длины очередей в их буферах может возникнуть ситуация, когда из-за неравномерной задержки передачи кадра, более поздний кадр обгонит своего предшественника. Поэтому в большинстве реализаций механизмов агрегирования используются методы статического, а не динамического распределения кадров по портам, т.е. закрепление за определенным портом

агрегированного канала потока кадров определенного сеанса между двумя узлами. В этом случае все кадры будут проходить через одну и ту же очередь, и их последовательность не изменится. Обычно при статическом распределении выбор порта для конкретного сеанса выполняется на основе выбранного алгоритма агрегирования портов, т.е. на основании некоторых признаков поступающих пакетов. В коммутаторах D-Link поддерживается 9 алгоритмов агрегирования портов:

1. mac_source –MAC-адрес источника;
2. mac_destination – MAC-адрес назначения;
3. mac_source_dest – MAC-адрес источника и назначения;
4. ip_source – IP-адрес источника;
5. ip_destination – IP-адрес назначения;
6. ip_source_dest – IP-адрес источника и назначения;
7. l4_src_port – TCP/UDP-порт источника;
8. l4_dest_port – TCP/UDP-порт назначения;
9. l4_src_dest_port – TCP/UDP-порт источника и назначения.

В коммутаторах D-Link по умолчанию используется алгоритм **mac_source** (MAC-адрес источника).

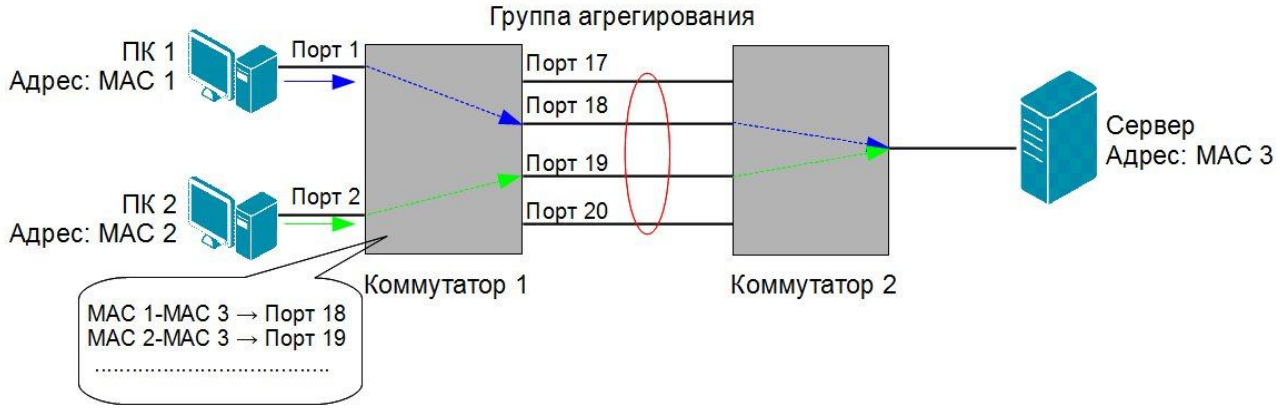


Рис. 5.28. Распределение потоков данных по каналам агрегированной линии связи для алгоритма mac_source_dest

Объединение каналов следует рассматривать как вариант настройки сети, используемый преимущественно для соединений «коммутатор – коммутатор» или «коммутатор – файл-сервер», требующих более высокой скорости передачи, чем может обеспечить одиночная линия связи. Также эту функцию можно применять для повышения надежности важных каналов связи. В случае повреждения линии связи объединенный канал быстро перенастраивается (не более чем за 1 сек.), а риск дублирования и изменения порядка кадров незначителен.

Программное обеспечение коммутаторов D-Link поддерживает два типа агрегирования каналов связи:

- статическое;
- динамическое, на основе стандарта IEEE 802.3ad (LACP).

При статическом агрегировании каналов (установлено по умолчанию), все настройки на коммутаторах выполняются вручную, и они не допускают динамических изменений в агрегированной группе.

Для организации динамического агрегирования каналов между коммутаторами и другими сетевыми устройствами используется протокол управления агрегированным каналом – Link Aggregation Control Protocol (LACP). Протокол LACP определяет метод управления объединением нескольких физических портов в одну логическую группу и предоставляет сетевым устройствам возможность автосогласования каналов (их добавления

или удаления), путем отправки управляющих кадров протокола LACP непосредственно подключенным устройствам с поддержкой LACP. Кадры LACP отправляются устройством через все порты, на которых активизирован протокол. Порты, на которых активизирован протокол LACP, могут быть настроены для работы в одном из двух режимов: **активном** (*active*) или **пассивном** (*passive*). При работе в активном режиме порты выполняют обработку и рассылку управляющих кадров протокола LACP. При работе в пассивном режиме порты выполняют только обработку управляющих кадров LACP.

Для того чтобы динамический канал обладал функцией автосогласования, рекомендуется порты, входящие в агрегированную группу, с одной стороны канала настраивать как активные, а с другой – как пассивные.

Следует отметить, что у портов, объединяемых в агрегированный канал, нижеперечисленные характеристики должны обладать одинаковыми настройками:

- тип среды передачи;
- скорость;
- режим работы – полный дуплекс;
- метод управления потоком (Flow Control) .

При объединении портов в агрегированный канал на них не должны быть настроены функции аутентификации 802.1X, зеркалирования трафика и блокировки портов.

5.7.1 Настройка статических и динамических агрегированных каналов

Рассмотрим пример, показанный на рис. 5.29. Для повышения пропускной способности канала связи между коммутатором 1, к которому подключен сервер, и коммутатором 2, к которому подключены пользователи, требуется объединить порты коммутаторов в статический агрегированный канал.

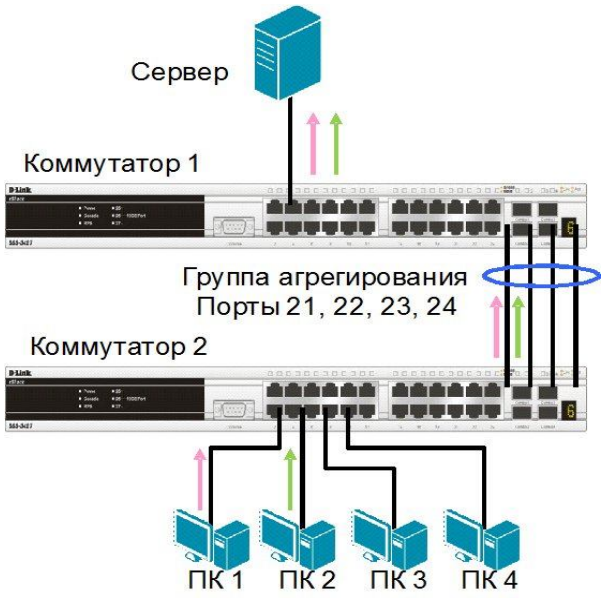


Рис. 5.29. Схема сети

На коммутаторах необходимо выполнить следующую настройку:

Настройка коммутатора 1

- Создать группу агрегирования (тип канала Static) и задать алгоритм агрегирования.

create link_aggregation group_id 1 type static
config link_aggregation algorithm mac_destination

- Включить порты 21, 22, 23, 24 в группу и выбрать порт 21 в качестве мастера-порта.

config link_aggregation group_id 1 master_port 21 ports 21,22,23,24 state enabled

Настройка коммутатора 2

- Создать группу агрегирования и задать алгоритм агрегирования.

create link_aggregation group_id 1 type static
config link_aggregation algorithm mac_source

- Включить порты 21, 22, 23, 24 в группу и выбрать порт 21 в качестве мастера-порта.

config link_aggregation group_id 1 master_port 21 ports 21,22,23,24 state enabled

Рассмотрим пример настройки коммутаторов при создании динамического агрегированного канала связи.

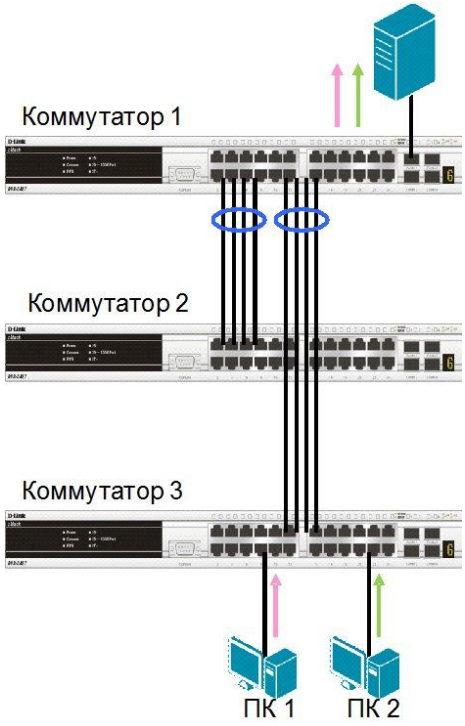


Рис. 5.30. Схема сети

Настройка коммутатора 1

- Создать группы агрегирования (тип канала LACP) и задать алгоритм агрегирования.

create link_aggregation group_id 1 type lacp
create link_aggregation group_id 2 type lacp
config link_aggregation algorithm mac_destination

- Включить порты 1, 2, 3, 4 в группу 1 и выбрать порт 1 в качестве мастера-порта.

config link_aggregation group_id 1 master_port 1 ports 1-4 state enabled

- Включить порты 5, 6, 7, 8 в группу 2 и выбрать порт 5 в качестве мастера-порта.

config link_aggregation group_id 2 master_port 5 port 5-8 state enabled

- Настроить для портов 1-8 активный режим работы.

config lacp_port 1-8 mode active

Настройка коммутаторов 2 и 3 (на портах 1-4 этих коммутаторов включено автосогласование)
create link_aggregation group_id 1 type lacp

config link_aggregation algorithm mac_source
config link_aggregation group_id master_port 1 ports 1-4 state enabled

Внимание:

1. Если один конец агрегированного канала настроен как LACP, другой конец должен также иметь тип LACP. Если один конец имеет тип LACP, а другой Static, то соединение установлено не будет.

2. Если коммутатор с поддержкой LACP требуется подключить к коммутатору, поддерживающему только статическое агрегирование, то тип агрегированного канала на коммутаторе LACP необходимо установить в Static.

3. Не соединяйте физически соответствующие порты устройств до тех пор, пока не настроено агрегирование каналов, т.к. в коммутируемой сети может возникнуть петля.

6. Адресация сетевого уровня и маршрутизация

6.1 Сетевой уровень

При построении сетей передачи данных часто возникает задача организации связи между различными сетями или подсетями, которые образуют *составную сеть (internetwork)*. Так, например, в локальных сетях, логически сегментированных с использованием VLAN, администраторам часто требуется организовать передачу данных между ними. Это выполняется с помощью функций *сетевого уровня* (*network layer)*.

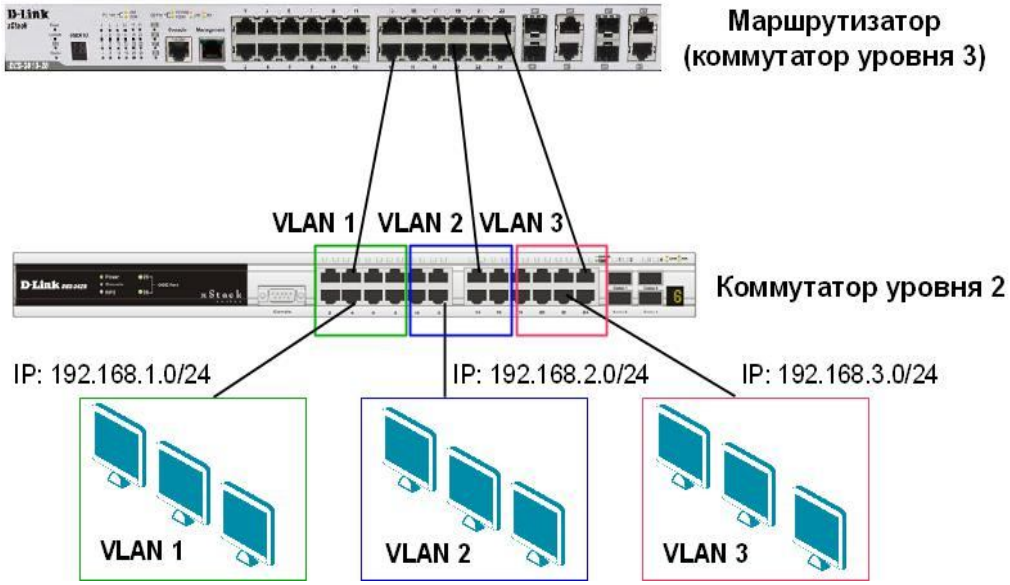


Рис. 6.1. Составная сеть

Основным протоколом сетевого уровня является протокол *IP (Internet Protocol)*, который позволяет доставлять данные в сетях TCP/IP между любыми узлами составной сети и выполняет две основные функции:

- адресация узлов (IP-адресация);
- маршрутизация.

Маршрутизация – это выбор наилучшего маршрута передачи пакета от источника к получателю.

Протокол IP не гарантирует надёжной доставки пакета до адресата, эта функция выполняется протоколами более высокого уровня. Такой тип доставки данных называют *best-effort*. В настоящее время существует две версии протокола IP:

- IP версии 4 (IPv4), который использует 32-битные адреса;
- IP версии 6 (IPv6), который использует 128-битные адреса.

6.2 Обзор адресации сетевого уровня

Основной задачей IP-протокола является передача данных между устройствами составной сети, для чего необходима информация о расположении адресата. Каждое устройство, которое выполняет передачу данных, имеет связанный с ним *физический адрес* (MAC-адрес) на канальном уровне, и *логический адрес* (IP-адрес) на сетевом уровне, который иногда называют *адресом третьего уровня*. Логические адреса не привязываются к конкретной аппаратуре и назначаются администратором сети независимо от физических адресов.

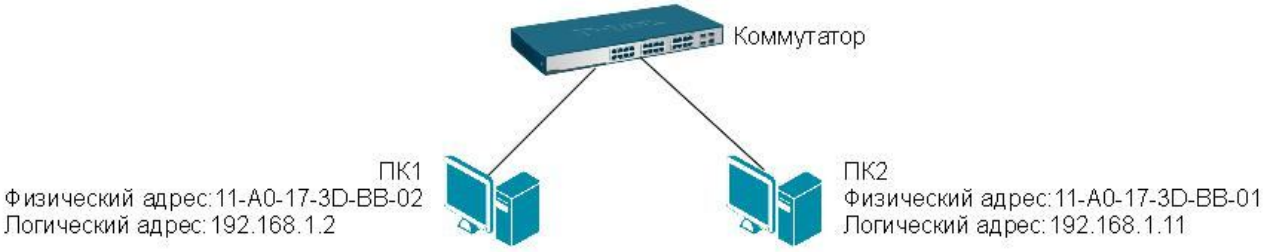


Рис. 6.2. Физические и логические адреса

Для того чтобы устройство могло участвовать в сетевом взаимодействии с помощью протокола IP, ему должен быть присвоен уникальный IP-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. Такие устройства, как коммутаторы L3, могут иметь несколько сетевых подключений и, соответственно, несколько IP-адресов.

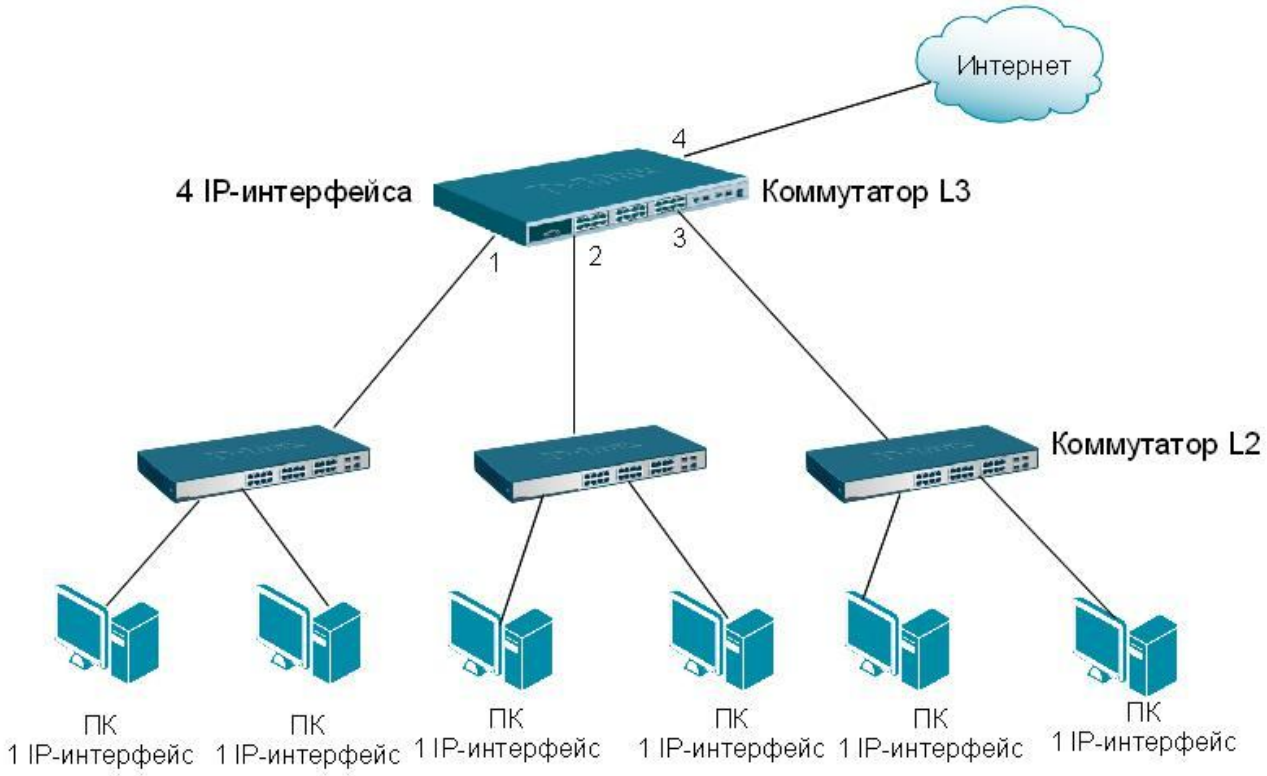


Рис. 6.3. IP-интерфейсы устройств

6.2.1 Формат пакета IPv4

Данные, передаваемые с использованием протокола IPv4, помещаются в сообщения, называемые *пакетами*. Протокол IPv4 использует пакет, который условно можно разделить на заголовок длиной, как правило, 20 байт и данные. Заголовок содержит адресные и управляющие поля, а в поле *Данные* находится непосредственно информация, которая передается через составную сеть. В отличие от формата некоторых других протоколов, например Ethernet, IPv4-пакет не содержит следующей за полем *Данные* контрольной суммы всего IPv4-пакета.

Рис. 6.4. Формат пакета IPv4

IPv4-пакет состоит из следующих полей:

- *Версия (Version)* – для IPv4 значение поля равно 4;
- *Длина заголовка (IHL, Internet Header Length)* – указывает на начало блока данных в пакете. Обычно значение для этого поля равно 5;
- *Тип сервиса (Type of Service)* – указывает приоритет пакета;
- *Общая длина (Total Length)* – общая длина пакета с учетом заголовка и поля данных;
- *Идентификатор пакета (Identification)* – используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета;
- *Флаги (Flag)* – содержит признаки, связанные с фрагментацией пакета;
- *Смещение фрагмента (Fragment Offset)* – значение, определяющее позицию фрагмента в потоке данных;
- *Время жизни (Time to Live)* – временной интервал, в течение которого пакет может перемещаться по сети маршрутизаторами;
- *Протокол (Protocol)* – указывает, какому протоколу верхнего уровня принадлежит информация, размещенная в поле данных пакета;
- *Контрольная сумма (Header Checksum)* – рассчитывается по заголовку и позволяет определить целостность заголовка пакета;
- *Адрес источника (Source IP Address) и адрес назначения (Destination IP Address)* – указывают отправителя и получателя пакета;
- *Опции (Options)* – необязательное поле, может использоваться при отладке работы сети.

Заголовок IPv4, как правило, имеет длину 20 байт. При использовании необязательного поля *Опции (Options),* длина заголовка может быть увеличена в зависимости от количества опций, но всегда остается кратной 32 битам.

6.2.2 Представление и структура адреса IPv4

Адрес IPv4 представляет собой 32-разрядную (4 байта) двоичное поле. Для удобства восприятия и запоминания этот адрес разделяют на 4 части по 8 бит (октеты), каждый октет переводят в десятичное число и при записи октеты разделяют точками. Это представление адреса называется *десятично-точечной нотацией*. Преобразование IP-адреса из двоичного (бинарного) представления в десятичное показано на рис. 6.5.

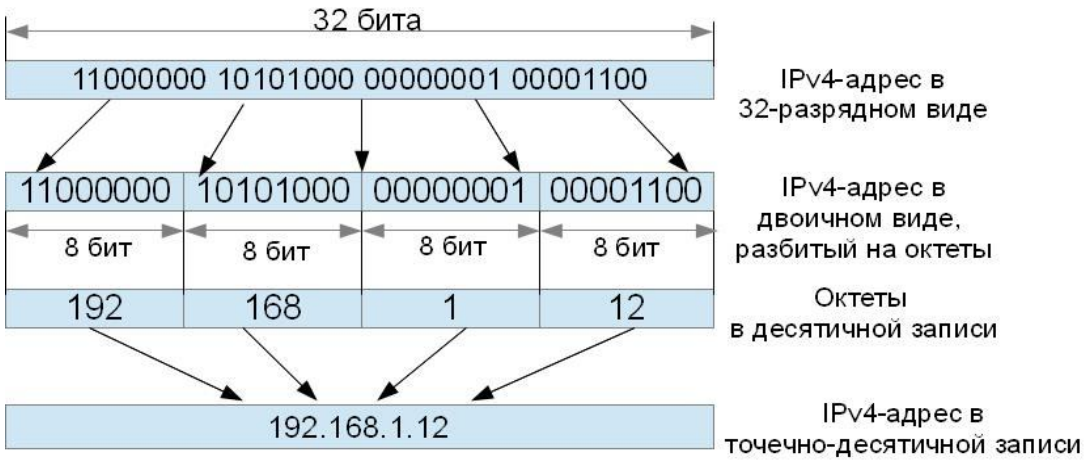


Рис. 6.5. Представление IPv4-адреса

Следует отметить, что максимальное значение октета равно 11111111 в двоичной системе счисления, что соответствует 255 в десятичной системе счисления, поэтому IP-адреса, в которых хотя бы один октет превышает максимальное значение, считаются недействительными.

Чтобы быстро в уме выполнить преобразование из двоичного вида в десятичный, полезно запомнить таблицу, приведенную ниже. Десятичное число легко вычисляется как сумма цифр, соответствующих ненулевым битам в октете (таблица 5).

Таблица 5 **Преобразование из двоичного вида в десятичный**

Двоичное значение октета	Значение битов октета	Десятичное значение октета
00000000	0	0
10000000	128	128
11000000	128+64	192
11100000	128+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

Маршрутизация пакетов в сетях передачи данных возможна благодаря тому, что IP-адрес структурирован и состоит из двух логических частей: *идентификатора сети (Net ID)* – сетевая часть адреса и *идентификатора узла (Host ID),* который однозначно определяет устройство в сетевом сегменте. Такая структура IP-адреса представляет собой двухуровневую иерархическую модель и позволяет устройству при передаче данных в составную сеть указывать не только удаленную сеть, но и узел в этой сети.

Рис. 6.6. Структура IPv4-адреса

Идентификатор сети определяет конкретную сеть или сегмент сети, в которой находится узел и используется для передачи данных на нужный сетевой интерфейс маршрутизатора или коммутатора 3-го уровня.

После того как данные достигают нужной сети, они передаются уникальному узлу в соответствии с *идентификатором узла*. Все узлы, использующие один и тот же идентификатор сети, должны быть расположены в одной сети или подсети (логическом сегменте сети).

6.2.3 Классовая адресация IPv4

При разработке базовых стандартов и протоколов, положенных в основу будущей глобальной сети (интернета), невозможно было представить, какое количество адресов потребуется для работы всех узлов сети. Размер IPv4-адреса был выбран длиной в 32 бита (при этом можно адресовать 2^{32} = 4,3 млрд. устройств). Как показала практика, этой длины адреса для современного интернета недостаточно. В связи с этим при использовании IPv4 очень важным вопросом является оптимизация выдаваемых адресов с точки зрения максимально эффективного использования IPv4-адресного пространства.

Хронологически первым методом разделения IP-адресов является так называемая *классовая модель* IP-адресации, которая частично решила проблему нерационального использования адресного пространства. Согласно этой модели, все пространство IP-адресов делится на 5 классов в зависимости от значения первых четырех бит IPv4-адреса. Классам присвоены имена от A до E.

Первые 3 класса A, B и C используются для индивидуальной (unicast) адресации сетей и узлов, класс D – для многоадресной или групповой (multicast) рассылки, а класс E зарезервирован для экспериментов. Классы A, B и C имеют различную длину сетевой части адреса.

Для сетей класса A под идентификатор сети отводится 1 байт (первый октет), а 3 оставшихся байта (3 октета) используются для идентификатора узла, причем старший (левый) бит идентификатора сети всегда равен 0.

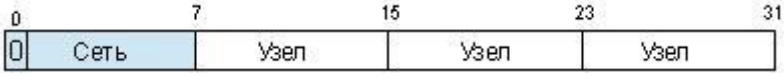


Рис. 6.7. Формат IPv4-адреса класса A

Поскольку первый бит идентификатора сети всегда равен нулю, то оставшиеся 7 бит позволяют адресовать 128 (2^7) различных сетей. Однако ввиду того, что адреса 0.0.0.0 и 127.0.0.0 являются специальными IPv4-адресами, количество доступных сетей класса A равно 126 (2^7-2). В каждой сети класса A можно адресовать до 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).

Сети класса B определяются значениями 10 в двух старших битах адреса. Первые 2 байта в адресе используются для идентификатора сети, а оставшиеся 2 байта – для

идентификатора узла. В результате количество доступных сетей класса B составляет 16 384 (2^{14}) с количеством узлов в каждой сети равным 65 534 (2^{16}-2).

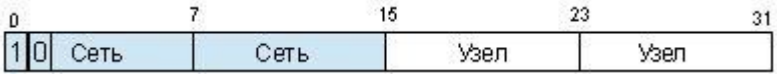


Рис. 6.8. Формат IPv4-адреса класса B

Для сетей класса C под идентификатор сети отводится 3 байта в то время как под идентификатор узла только 1 байт. Три старших бита первого октета всегда равны 110, позволяя определить, что адрес относится именно к классу C. Таким образом, получаем 2 097 152 (2^{21}) сетей, в каждой из которых находится 254 (2^8-2) узла.

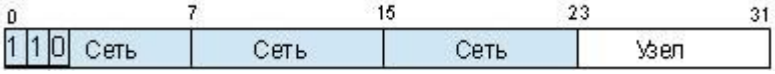


Рис. 6.9. Формат IPv4-адреса класса C

Сети класса D определяются значениями 1110 в первых четырех битах адреса, остальные биты используются для адресации многоадресной группы. Адресное пространство класса D зарезервировано для групповой рассылки и используется для адресации группы узлов. Идентификаторов сетей и узлов в IPv4-адресе класса D не выделяют.

Рис. 6.10. Формат IPv4-адреса класса D

Сети класса E являются экспериментальными и в настоящее время не используются. Адреса в этом классе определяются значениями 1111 в первых четырех битах.

Рис. 6.11. Формат IPv4-адреса класса E

6.2.4 Частные и публичные адреса IPv4

В интернете идентификация устройств осуществляется уникальными IPv4-адресами, которые не должны повторяться в глобальной сети. Такие IPv4-адреса называются *публичными* адресами. Однако число публичных адресов ограничено, поэтому в каждом из классов IP-сетей определено так называемое *частное пространство IP-адресов*. Частные IPv4-адреса предназначены для использования в локальных компьютерных сетях и не маршрутизируются в интернет. Для локальных сетей, не подключенных к интернету, можно использовать любые возможные адреса, уникальные в пределах данной сети.

Публичные адреса находятся в пределах от 1.0.0.1 до 223.255.255.254 за исключением частных адресов IPv4.

Адресное пространство частных IPv4-адресов состоит из 3 блоков:

- 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).

Помимо этого определены IPv4-адреса (таблица 6), которые имеют специальное назначение (специальные адреса).

Таблица 6 **Специальные IP-адреса**

Идентификатор сети	Идентификатор узла	Описание
Все «0»	Все «0»	0.0.0.0 - адрес узла, сгенерировавшего пакет. Используется устройством для ссылки на самого себя, если оно не знает свой IPv4-адрес. Используется, например, когда устройство пытается получить IPv4-адрес с помощью протокола DHCP
Все «0»	Идентификатор узла	Узел назначения принадлежит той же сети, что и узел-отправитель, например, 0.0.0.25
Идентификатор сети	Все «0»	Адрес IPv4-сети, например, 175.11.0.0
Идентификатор сети	Все «1»	Ограниченный широковещательный адрес (в пределах данной IP-сети), например, 192.168.100.255
Все «1»	Все «1»	255.255.255.255 – «глобальный» широковещательный адрес
127.0.0.0		Адрес интерфейса обратной петли (loopback), предназначен для тестирования оборудования без реального отправления пакета

6.3 Формирование подсетей

Изначально IPv4-адрес имел два уровня иерархии: идентификатор сети и идентификатор узла. Каждой организации выдавался IPv4-адрес из нужного диапазона (A, B или C) в зависимости от текущего числа компьютеров и его планируемого увеличения.

Для более эффективного использования адресного пространства были внесены изменения в существующую классовую систему адресации. В RFC 950 была описана процедура разбиения сетей на подсети, и в структуру IPv4-адреса был добавлен еще один уровень иерархии – *подсеть (subnetwork)*. Появление еще одного уровня иерархии не изменило самого IPv4-адреса, он остался 32-разрядным, а часть адреса, отведенная ранее под идентификатор узла, была разделена на 2 части – идентификатор подсети и идентификатор узла (рис. 6.12).

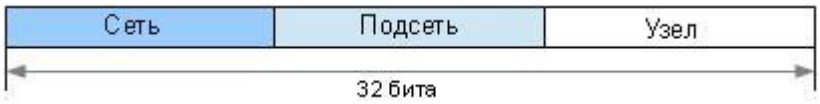


Рис. 6.12. Трехуровневая иерархия IP-адреса

Разбиение одной крупной сети на несколько более мелких позволяет:

- рационально использовать адресное пространство (т.е. выделить для сегмента сети блок адресов не целиком класса A, B или C, а только часть классовой сети);
- повысить безопасность и управляемость сети (за счет уменьшения размеров сегментов и изоляции трафика сегментов друг друга).

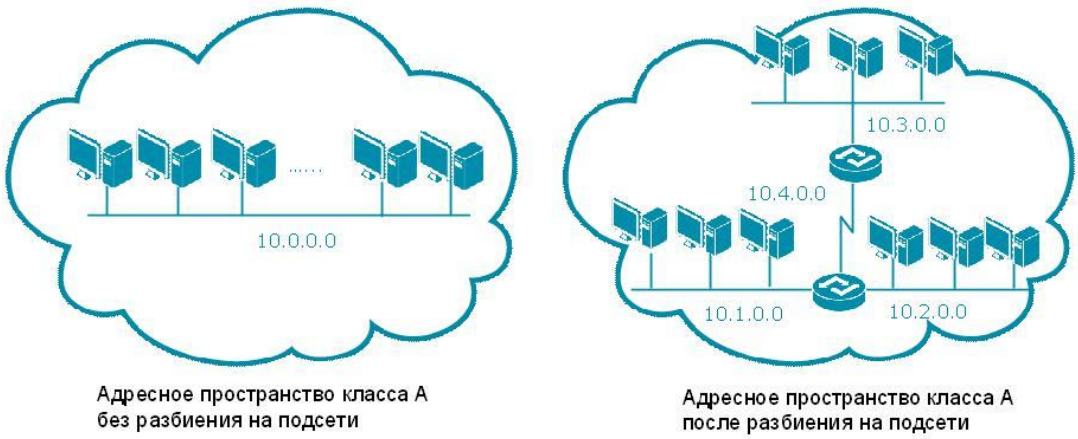


Рис. 6.13. Пример разбиения на подсети

С появлением трехуровневой иерархии IPv4-адреса потребовались дополнительные методы, которые позволяли бы определить, какая часть IPv4-адреса указывает на идентификатор подсети, а какая – на идентификатор узла. Было предложено использовать битовую маску (bit mask), которая отделяла бы часть адресного пространства идентификаторов узлов от адресного пространства идентификаторов подсети. Такая битовая маска названа *маской подсети (subnet mask)*.

Маска подсети – это 32-битное число, двоичная запись которого содержит единицы в тех разрядах, которые должны определяться как идентификатор сети. Поскольку идентификатор сети является цельной частью IPv4-адреса, последовательность единиц в маске подсети должна быть также непрерывной.

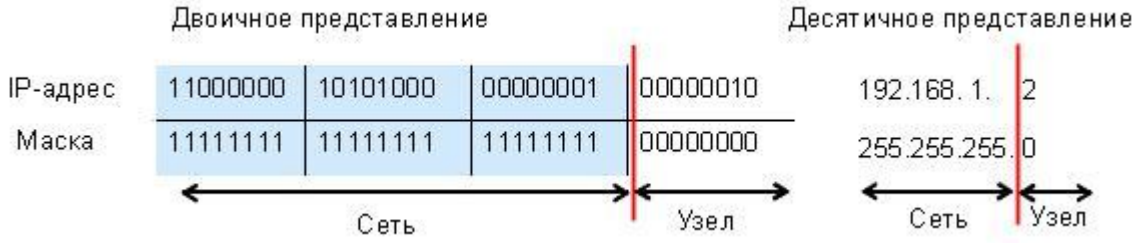


Рис. 6.14. Формирование маски подсети

Чтобы получить адрес сети, зная IPv4-адрес и маску подсети, необходимо применить к ним операцию *логическое «И»*. Другими словами, в тех позициях IPv4-адреса, в которых в маске подсети стоят двоичные единицы, находится идентификатор сети, а где двоичные 0 – идентификатор узла.

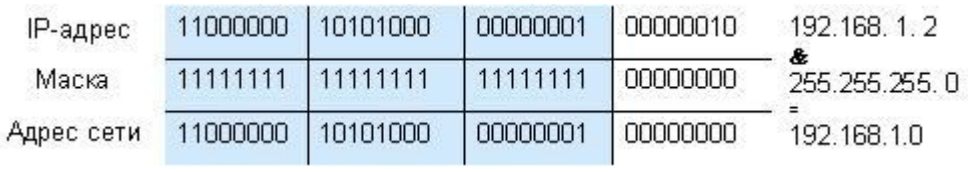


Рис. 6.15. Получение адреса сети из IP-адреса и маски подсети

Для сетей класса A, B и C определены фиксированные маски подсети, которые жестко определяют количество возможных IPv4-адресов и механизм маршрутизации (таблица 7).

Таблица 7 **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

При применении масок подсети сети можно разделять на меньшие по размеру подсети путем расширения сетевой части адреса и уменьшения узловой части. Технология разделения сети дает возможность создавать большее число сетей с меньшим количеством узлов в них, что позволяет эффективно использовать адресное пространство.

Для вычисления количества подсетей используется формула 2^s, где s – количество бит, занятых под идентификатор сети из части, отведенной под идентификатор узла. Количество узлов в каждой подсети вычисляется по формуле $2^n - 2$, где n – количество бит, оставшихся в части, идентифицирующей узел, а два адреса – адрес подсети и широковещательный адрес – в каждой полученной подсети зарезервированы.

Например, организации необходимо разбить сеть 192.168.1.0 на 20 подсетей по 6 компьютеров в каждой. Для начала необходимо определить, к какому классу относится адрес. 192.168.1.0 – это класс C, соответственно, стандартная маска подсети для класса C равна 255.255.255.0 и под идентификатор узла отведен 4-й октет. Затем определяется количество бит 4-го октета, занимаемых для формирования 20 подсетей. Поскольку найти число, при котором степень 2 будет равна 20 невозможно, выбираем ближайшее большее число $2^5 = 32$. Таким образом, 5 первых бит 4-го октета будут использованы для идентификации подсети, а оставшиеся 3 бита – для идентификации узлов в них (рис. 6.16).

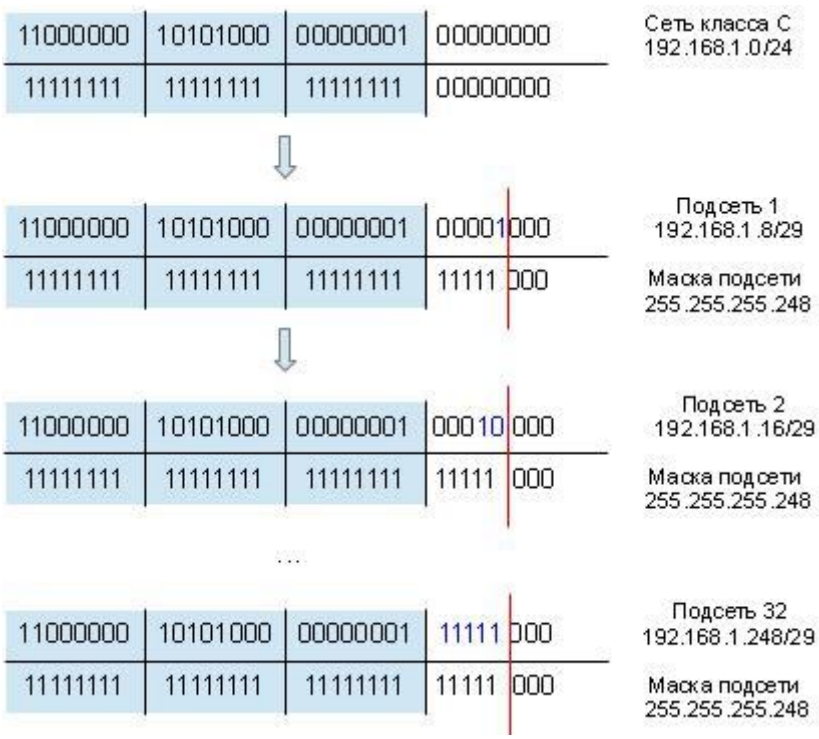


Рис. 6.16. Пример разбиения сети 192.168.1.0/24 на подсети

Во избежание проблем с адресацией и маршрутизацией все сетевые устройства TCP/IP в одном сегменте сети должны использовать одну и ту же маску подсети.

6.4 Бесклассовая адресация IPv4

Классовая модель IPv4-адресации оказалась нерациональной с точки зрения эффективного использования адресного пространства. Например, для сети из 1 000 устройств назначается диапазон адресов класса B, в котором 65 534 адресов. При этом 1 000 адресов используются, а оставшиеся 64 534 – не используются.

В случае классовой адресации сеть можно было разбить только на подсети одинакового размера. При этом если выбранная маска подсети обеспечивает нужное количество подсетей, возможно, что допустимого количества узлов для каждой подсети будет недостаточно или, наоборт, большая часть адресов не будет использована. Например, большое количество узлов является избыточным для подсети, которая связывает два маршрутизатора по схеме «точка-точка». В этом случае необходимо всего два IPv4-адреса для адресации интерфейсов соседних маршрутизаторов. Таким образом, разбиение сети на подсети разного размера позволило бы рационально использовать адресное пространство.

Постепенно с ростом интернета произошел отказ от классовой схемы, и была принята *бесклассовая модель IPv4-адресации*, в которой отсутствует привязка к классу сети и маске подсети по умолчанию. Бесклассовая адресация использует *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. Основная идея применения VLSM заключается в том, что можно разбить сеть на подсеть, потом подсеть разбить еще на подсети точно таким же образом, как была разбита первоначальная сеть. То есть сеть может быть разбита на подсети разных размеров, с разными масками. Маски подсети являются основой метода бесклассовой маршрутизации и 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 в маске подсети 255.255.255.248 29 единичных бит.

Деление сети на подсети с использованием масок переменной длины аналогично традиционному делению на подсети. Рассмотрим пример, показанный на рис.6.17.

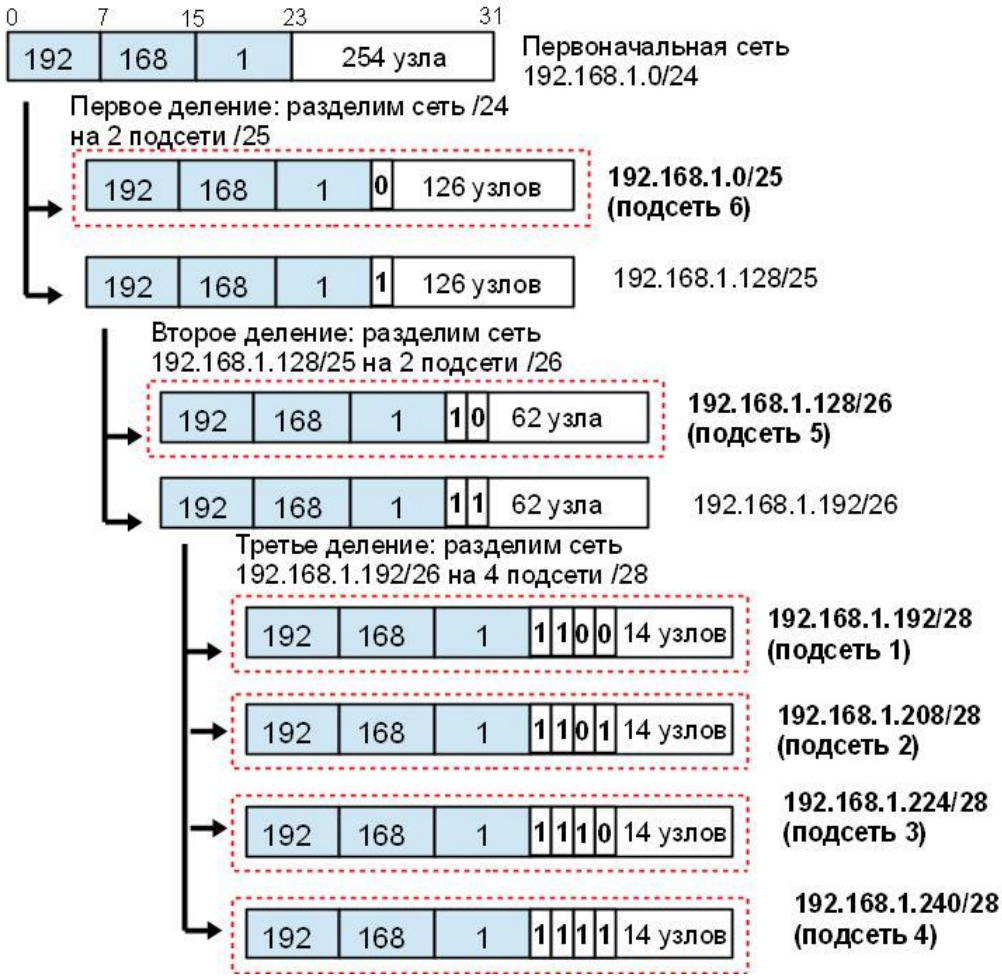


Рис. 6.17. Пример разбиения сети 192.168.1.0/24 на подсети при помощи VLSM

Допустим организации выделена сеть класса С 192.168.1.0/24. Требуется разделить ее на 6 подсетей. В подсетях 1, 2, 3 и 4 должно быть 10 узлов, в 5-й подсети – 50 узлов, в 6-й подсети – 100. Теоретически для сети 192.168.1.0/24 допустимое количество узлов равно 254, и разбить такую сеть на подсети с требуемым количеством узлов без использования VLSM невозможно.

Сначала необходимо разделить сеть 192.168.1.0/24 на две подсети. Для этого из 4-го октета необходимо занять 1 бит для идентификатора подсети, таким образом, для идентификации узлов останется 7 бит. В итоге получается две подсети 192.168.1.0/25 и 192.168.1.128/25, в каждой из которых может быть по 126 (2^7 – 2) узлов. Первую из них оставим, так как требуется, чтобы в 6-й подсети было 100 узлов, а вторую разделим еще на две подсети. Для этого возьмем 1 бит из оставшихся 7 бит, отведенных под идентификатор узла. Таким образом, получается две подсети 192.168.1.128/26 и 192.168.1.192/26, в каждой из которых допустимое количество узлов равно 62 (2^6 – 2). Первую подсеть необходимо оставить для 5-й подсети, в которой должно быть 50 узлов, а из второй подсети сформировать еще четыре подсети. Для этого займем еще 2 бита из оставшихся 6 бит, отведенных под идентификатор узла. В результате получим четыре подсети с 14 (2^4 – 2) узлами в каждой, что позволит адресовать требуемое количество узлов, необходимых для подсетей 1, 2, 3 и 4.

6.5 Способы конфигурации IPv4-адреса

IPv4-адрес может быть задан *статически* или присвоен сетевому интерфейсу *динамически*. Статические адреса назначаются вручную администратором. Динамические

адреса назначаются автоматически при подключении устройства к сети и используются в течение ограниченного промежутка времени или до его выключения. При новом назначении динамический IPv4-адрес клиента может быть изменен. Наиболее широко используемым протоколом динамического назначения адресов является *DHCP* (*Dynamic Host Configuration Protocol*), который описан в RFC 2131.

6.6 Протокол IPv6

Протокол IPv6 — это новая версия протокола IP, которая разработана в качестве преемника IPv4 и призвана окончательно решить проблему исчерпания адресного пространства. В отличие от адреса IPv4, который имеет длину 32 бита, размер адреса IPv6 составляет 128 бит, что позволяет адресовать примерно $3{,}4 \times 10^{38}$ интерфейсов устройств. Адрес IPv6 отображается как восемь групп по четыре шестнадцатеричные цифры, разделенные двоеточием.

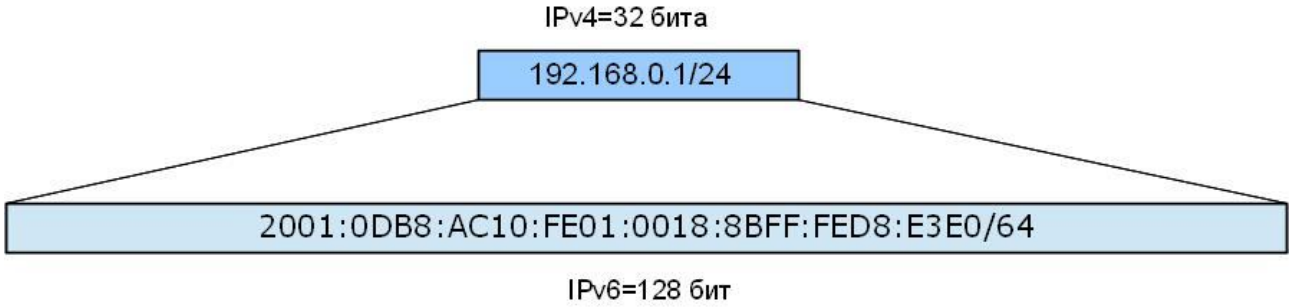


Рис. 6.18. Адреса IPv4 и IPv6

Одной из причин перехода на использование протокола IPv6 в сетях является потребность в большом количестве адресов, при этом технология IPv6 содержит ряд дополнительных преимуществ по сравнению с IPv4:

- улучшенные механизмы автоматического назначения адресов узлов;
- упрощение маршрутизации;
- улучшенные механизмы обеспечения качества обслуживания (QoS) и безопасности (IPSec);
- упрощенный заголовок пакета.

6.6.1 Формат заголовка IPv6

При разработке протокола IPv6 были внесены изменения в формат IP-пакета. Увеличение размера IPv6-адреса с 32 бит до 128 бит добавило 24 байта к заголовку пакета, что, в свою очередь, привело к попытке уменьшить его размер за счет исключения полей, связанных с фрагментацией, и поля контрольной суммы. В результате заголовок пакета IPv6 увеличился всего в два раза.

Пакет протокола IPv6 состоит из фиксированного заголовка и произвольного числа расширенных заголовков. Такой порядок способствует эффективной обработке пакетов на всем пути их следования. Фиксированный заголовок состоит из 40 байт и имеет формат, показанный на рисунке 6.19.

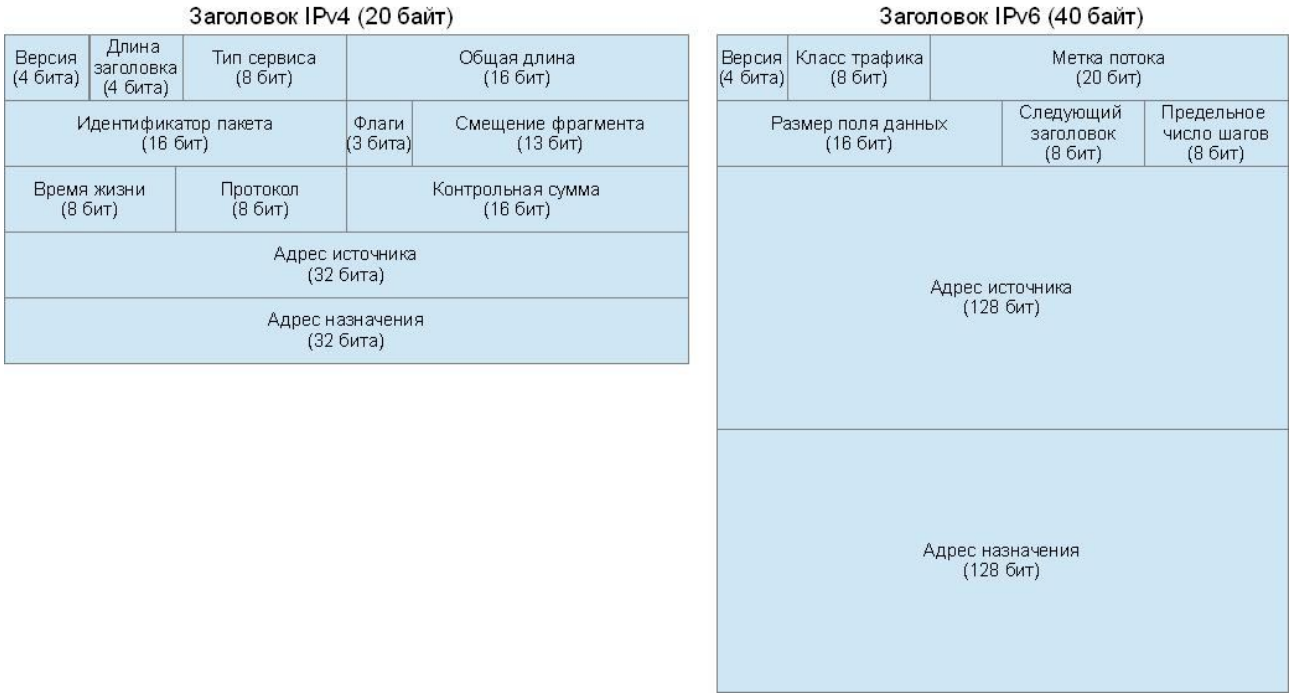


Рис. 6.19. Сравнение форматов заголовка IPv4 и IPv6

Заголовок IPv6-пакета состоит из следующих полей:

- *Версия (Version)* – для IPv6 значение поля должно быть равно 6;
- *Класс трафика (Traffic Class)* – поле приоритета пакета;
- *Метка потока (Flow Label)* – используется отправителем для обозначения последовательности пакетов, которые должны быть подвергнуты определенной обработке маршрутизаторами;
- *Размер поля данных (Payload Length)* – число, указывающее длину поля данных, идущего за заголовком пакета (с учетом расширенного заголовка);
- *Следующий заголовок (Next Header)* – задает тип расширенного заголовка IPv6, который следует за фиксированным;
- *Предельное число шагов (Hop Limit)* – уменьшается на 1 каждым маршрутизатором, через который передается пакет; при значении, равном 0, пакет отбрасывается;
- *Адрес источника (Source Address)* – 128-битный адрес отправителя пакета;
- *Адрес назначения (Destination Address)* – 128-битный адре получателя пакета.

Сравнение заголовка пакета IPv4 с заголовком IPv6 показывает что:

поле *Длина заголовка (Internet Header Length)* исчезло, так как фиксированный заголовок IPv6 имеет определенную длину (40 байт);

- поле *Тип сервиса (Type of Service)* трансформировалось в заголовке IPv6 в поля *Класс трафика (Traffic Class)* и *Метка потока (Flow Label)*;
- поля *Время жизни (Time to Live)* и *Протокол (Protocol)* в заголовке IPv6 изменили названия, соответственно, на *Предельное число шагов (Hop Limit)* и *Следующий заголовок (Next Header)* с некоторым уточнением трактовки;
- поле *Контрольная сумма (Header Checksum)* было ликвидировано, так как её подсчёт занимает некоторое время, что существенно снижает производительность узлов;
- поля в заголовке IPv4, связанные с фрагментацией были перенесены в расширенные заголовки IPv6;
- минимальный размер пакета, который должен передаваться в сетях IPv6 без фрагментации, увеличен с 576 до 1 280 байт.

Расширенные заголовки IPv6 используются для поддержки механизмов безопасности, фрагментации, сетевого управления и расположены между фиксированным заголовком и заголовком протокола более высокого уровня. Пакет IPv6 может содержать 0, 1 или несколько расширенных заголовков, каждый из которых идентифицируется значением поля Next Header предшествующего заголовка. Все существующие типы расширенных заголовков описаны в таблице 8.

Таблица 8 **Типы расширенных заголовков IPv6**

Расширенный заголовок	Тип	Описание
Hop-by-Hop Options	0	Параметры, которые должны быть обработаны каждым транзитным узлом на пути от отправителя до получателя пакета.
Routing	43	Позволяет отправителю определять список узлов, которые пакет должен пройти
Fragment	44	Содержит информацию о фрагментации пакета
Authentication Header (AH)	51	Содержит информацию для проверки подлинности зашифрованных данных при использовании IPSec
Encapsulating Security Payload (ESP)	50	Обеспечивает шифрование данных с помощью IPSec
Destination Options	60	Определяет произвольный набор опций, которые должны быть обработаны получателем пакета

Поле Next Header используется для логической связи всех заголовков пакета IPv6, например, Next Header в фиксированном заголовке указывает тип первого расширенного заголовка, поле Next Header в первом расширенном заголовке содержит тип следующего расширенного заголовка и т.д. Поле Next Header последнего расширенного заголовка содержит номер протокола транспортного уровня (TCP или UDP) (рис. 6.20).

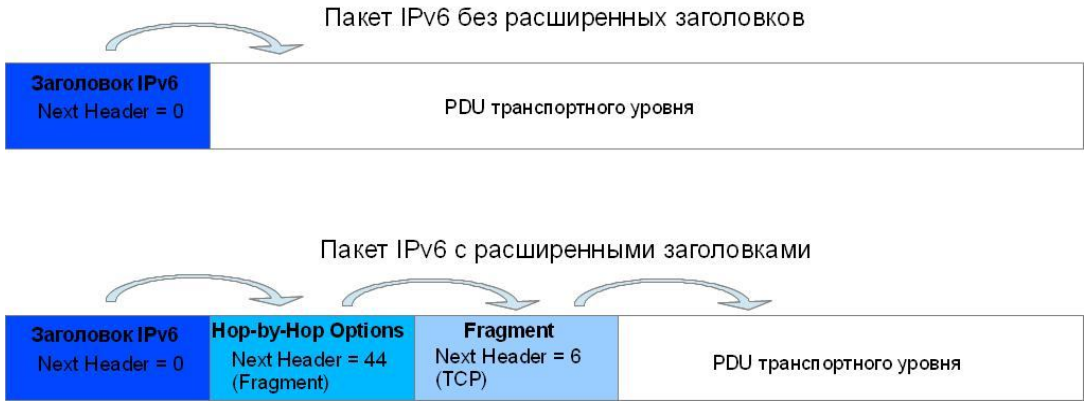


Рис. 6.20. Расширенные заголовки IPv6

Расширенные заголовки обрабатываются только узлом-получателем, за исключением заголовка *Hop-By-Hop Options*, который обрабатывается каждым промежуточным узлом на пути пакета, включая отправителя и получателя.

6.6.2 Представление и структура адреса IPv6

Адрес IPv6 имеет длину 128 бит и записывается как восемь групп по четыре шестнадцатеричные цифры, разделенные двоеточием. Например,

2001:0DB8:AC10:FE01:0018:8BFF:FED8:E3E0

Существует несколько способов, которые позволяют сократить запись IPv6-адреса:

- нули в начале группы можно заменить одним;
- одна или несколько идущих подряд групп, состоящих из нулей, может быть заменена знаком «**::**»;
- конечные нули в группе должны присутствовать.

Рассмотрим приведенный ниже адрес. Цифры, выделенные жирным шрифтом, представляют позиции, в которых адрес может быть сокращен.

2001:1000:**0000**:**0000**:**0000**:ABCD:**0000**:**000**1

Варианты возможных сокращений:

2001:1000::ABCD:0:0001
2001:1000::ABCD:0:1

Внимание: знак «::» не может использоваться дважды, поскольку такая запись воспринимается неоднозначно. Поэтому, например, адрес 2001:1000::ABCD::1 является недействительным.

Альтернативной формой записи адреса, которая более удобна для использования в смешанной среде с узлами IPv4 и IPv6 является запись вида x:x:x:x:d.d.d.d, где x - шестнадцатеричное значение 6 первых групп адреса; d - десятичное значение 4 последних групп адреса (стандартное представление адреса IPv4). Например:

0:0:0:0:0:0:13.1.68.3 или в сокращенном виде **::13.1.68.3**
0:0:0:0:0:FFFF:129.144.52.38 или в сокращенном виде **::FFFF:129.144.52.38**

IPv6-адрес состоит из двух логических частей – *префикса (Prefix)* и *идентификатора интерфейса (Interface ID)*.

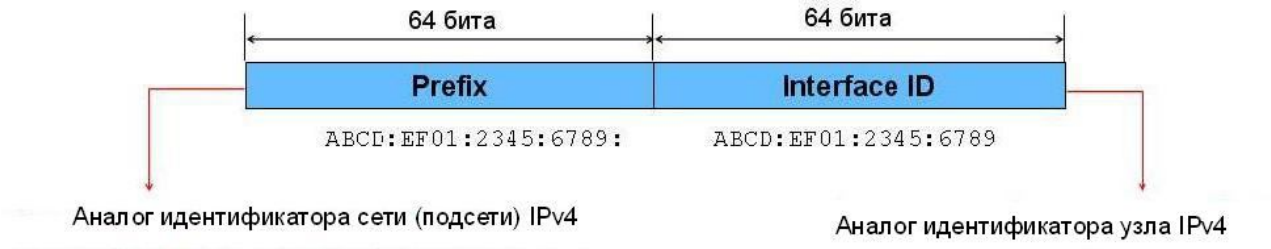


Рис. 6.21. Структура IPv6-адреса

Префикс (Prefix) – первые 64 бита адреса – часть адреса, отведенная под идентификатор сети/подсети (аналог идентификатора сети в IPv4). Представление префикса идентификатора для сети и подсети IPv6 аналогично записи префикса адреса IPv4 в нотации CIDR. Префикс адреса IPv6 записывается в виде адрес IPv6/длина префикса**.** Например:

21DA:D3::/48 – префикс сети;

21DA:D3:0:2F3B::/64 – префикс подсети.

Идентификатор интерфейса (Interface ID) – последние 64 бита IPv6-адреса, используемые для идентификации интерфейса в сегменте сети (аналог идентификатора узла в IPv4); он должен быть уникальным внутри сети/подсети.

6.7 Типы адресов IPv6

Адресное пространство протокола IPv6 разделено на три типа адресов:

- индивидуальные (unicast) адреса;
- многоадресные (multicast) адреса;
- альтернативные (anycast) адреса.

Индивидуальные адреса идентифицируют один интерфейс устройства. Пакеты, отправленные на этот адрес, доставляются только на этот интерфейс.

Многоадресные адреса IPv6, подобно одноименным адресам IPv4, идентифицируют группу интерфейсов. Пакеты, посылаемые на это адрес, доставляются всем интерфейсам – участникам группы рассылки.

Альтернативные адреса позволяют адресовать группу интерфейсов (обычно принадлежащих разным узлам). Однако, в отличие от многоадресных адресов, пакеты, передаваемые на альтернативный адрес, доставляются на один из интерфейсов (обычно «ближайший» интерфейс, согласно метрике маршрутизации), определяемых этим адресом.

Широковещательные адреса (Broadcast), которые используются в IPv4, в IPv6 отсутствуют, что способствует уменьшению сетевого трафика и снижению нагрузки на большинство систем. Широковещательные адреса заменены многоадресными.

Внимание: альтернативные адреса назначаются только маршрутизирующим устройствам.

6.7.1 Индивидуальные адреса

Существует несколько типов индивидуальных IPv6-адресов:

- Global Unicast-адреса;
- Unique-Local Unicast-адреса;
- Link-Local Unicast-адреса.

Для каждого типа индивидуального адреса определен свой диапазон.

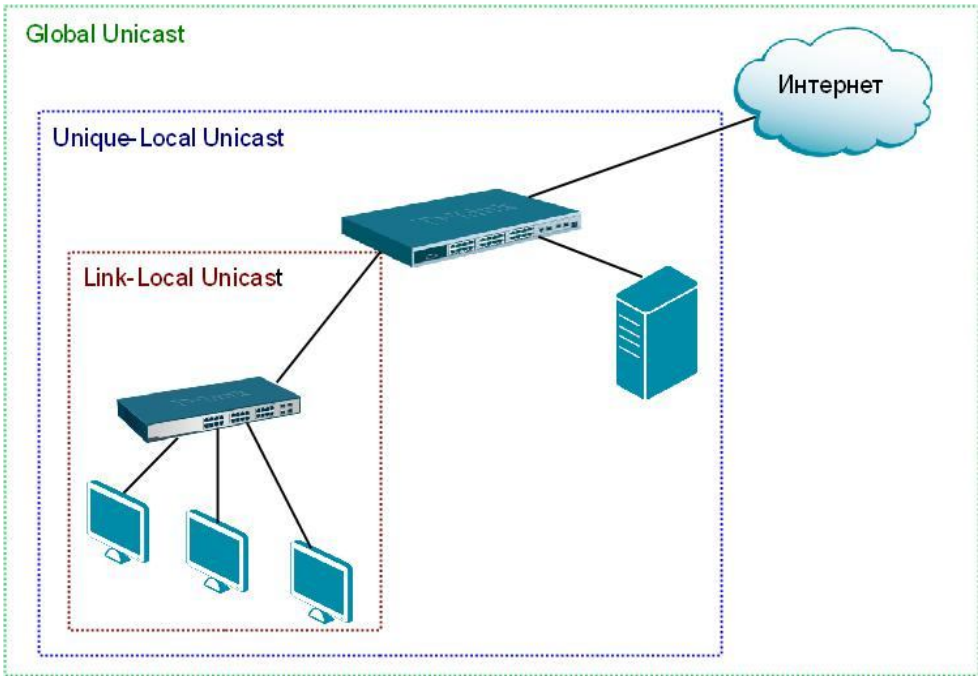


Рис. 6.22. Диапазон индивидуальных адресов IPv6

Global Unicast-адреса используются для идентификации устройств в глобальной сети и являются аналогом публичных IPv4-адресов. Эти адреса назначаются локальными интернет-регистраторами и имеют общий формат, показанный на рис. 6.23. В настоящее время Global Unicast IPv6-адреса назначаются с префиксом 2000::/3.

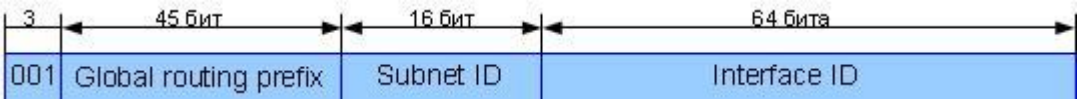


Рис. 6.23. Формат Global Unicast IPv6-адресов

Global Unicast IPv6-адрес разделен на три логические части: *глобальный префикс (Global routing prefix)*, *идентификатор подсети (Subnet ID)* и *идентификатор интерфейса*

(Interface ID). Три старших бита адреса равны 001. Следующие 45 бит формируют Global routing prefix – глобальный адрес, назначенный сети. Далее идет 16-битное поле Subnet ID, определяющее подсеть внутри сети, а последние 64 бита являются Interface ID.

Unique-Local Unicast-адреса (ULA) используются для идентификации устройств внутри организации, поэтому пакеты, которые в качестве источника или назначения имеют этот адрес, не будут передаваться через интернет. Такие адреса используются только внутри сетей организаций. Если провести аналогию с адресами IPv4, то Unique-Local Unicast-адреса эквивалентны частным IPv4-адресам, только в отличие от них являются уникальными в рамках глобальной сети.

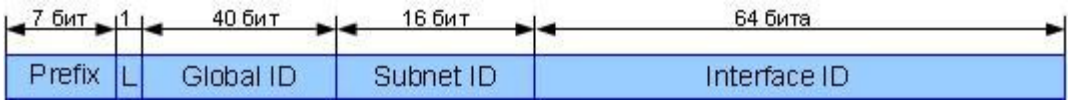


Рис. 6.24. Формат Unique-Local Unicast IPv6-адресов

Все Unique-Local Unicast-адреса начинаются с префикса FC00::/7. Бит *L* показывает, что префикс назначен локально (L=1), или адрес зарезервирован для будущих применений (L=0). Таким образом, бит L разбивает префикс FC00::/7 на два поддиапазона:

- FC00::/8 – зарезервирован для будущих применений;
- FD00::/8 – локально назначенный уникальный адрес.

Следующие 40 бит отведены под *глобальный идентификатор* (*Global ID)*, который определяет организацию. Он должен быть уникальным для того, чтобы минимизировать возможность совпадения с идентификаторами других организаций, поэтому назначается с помощью псевдослучайного алгоритма, который обеспечивает высокую вероятность его уникальности. Алгоритм для генерации Unique-Local Unicast-адреса можно найти в интернете (https://www.ultratools.com/tools/rangeGenerator). Далее в адресе следует 16-битное поле *идентификатор подсети* (*Subnet ID)*, которое определяет подсеть внутри сети организации и 64-битный *идентификатор интерфейса* (*Interface ID*).

Link-Local Unicast-адреса предназначены для взаимодействия внутри сегмента сети или по каналу связи «точка-точка» и используются только в пределах данного канала. Маршрутизаторы (коммутаторы 3-го уровня) не передают пакеты с Link-Local Unicast-адресами, указанными в качестве источника или назначения, через другие линии связи. Эти адреса автоматически назначаются узлу, независимо от наличия в сети маршрутизатора или DHCPv6-сервера.

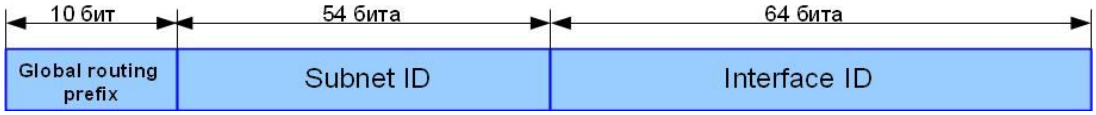


Рис. 6.25. Формат Link-Local Unicast IPv6-адресов

Link-Local Unicast-адреса имеют достаточно простой формат: адрес начинается с *глобального префикса маршрутизации* (*Global routing prefix*) FE80::/10. По сравнению с Global Unicast-адресом, префикс стал значительно короче, поэтому пространство, отведенное под *идентификатор подсети* (*Subnet ID*) увеличилось с 16 до 54 бит. В связи с тем, что Link-Local Unicast-адреса используются только в пределах линии связи, поле Subnet ID заполняется нулями. Последние 64 бита адреса отведены под *идентификатор интерфейса (Interface ID)*.

В IPv6, так же как и в IPv4, адрес идентифицирует не конкретное устройство, а его интерфейс. Главное отличие заключается в том, что протокол IPv6 позволяет назначить интерфейсу любое количество IPv6-адресов.

Существует несколько блоков специальных уникальных адресов IPv6:

- 0:0:0:0:0:0:0:0 (::/0) – маршрут по умолчанию. Аналогичен адресу 0.0.0.0 в IPv4;
- 0:0:0:0:0:0:0:0 (::/128) – никогда не назначается узлу, обозначает ситуацию отсутствия адреса;
- 0:0:0:0:0:0:0:1 (::1/128) - используется узлом для отправки самому себе пакетов IPv6. Аналогичен IPv4-адресу 127.0.0.1;
- 2002::/16 – служит для автоматического туннелирования трафика IPv6 через IPv4-сети;
- 2001::/32 – используется для организации Teredo-туннелей.

6.7.2 Многоадресные адреса

Многоадресные адреса IPv6 идентифицируют группу интерфейсов, участвующую в получении одного и того же контента (например, видео). Узел может входить более чем в одну группу, но не может использовать многоадресный адрес в качестве адреса источника в IPv6-пакетах. Многоадресные адреса имеют формат, показанный на рис. 6.26.

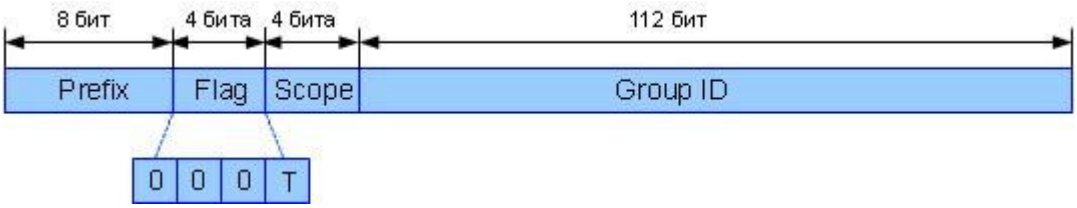


Рис. 6.26. Формат многоадресных адресов IPv6

Все многоадресные адреса начинаются с префикса FF00::/8. Следующие 4 бита – флаги (*Flag*). Первые 3 бита этого поля в настоящее время не используются и зарезервированы для будущего применения. Последний бит T определяет тип адреса:

- T=0 – адрес является постоянным, официально выделенным для использования в интернете;
- T=1 – адрес является временным.

Следующее поле *Scope (область)* занимает 4 бита и определяет область действия данного многоадресного адреса, т. е. показывает, как далеко друг от друга могут находиться члены одной многоадресной группы. На данный момент определено шесть значений этого поля, остальные зарезервированы для будущего применения:

- 1 – *Interface-Local* – многоадресная группа является локальной и определена в рамках одного узла;
- 2 – *Link-Local* – многоадресная группа определена в пределах линии связи;
- 4 – *Admin-Local* – многоадресная группа определена внутри области, задаваемой администратором сети.
- 5 – *Site-Local* – многоадресная группа определена в рамках локальной сети;
- 8 – *Organization* – многоадресная группа определена в рамках распределенной сети организации;
- E – *Global* – глобальная многоадресная группа.

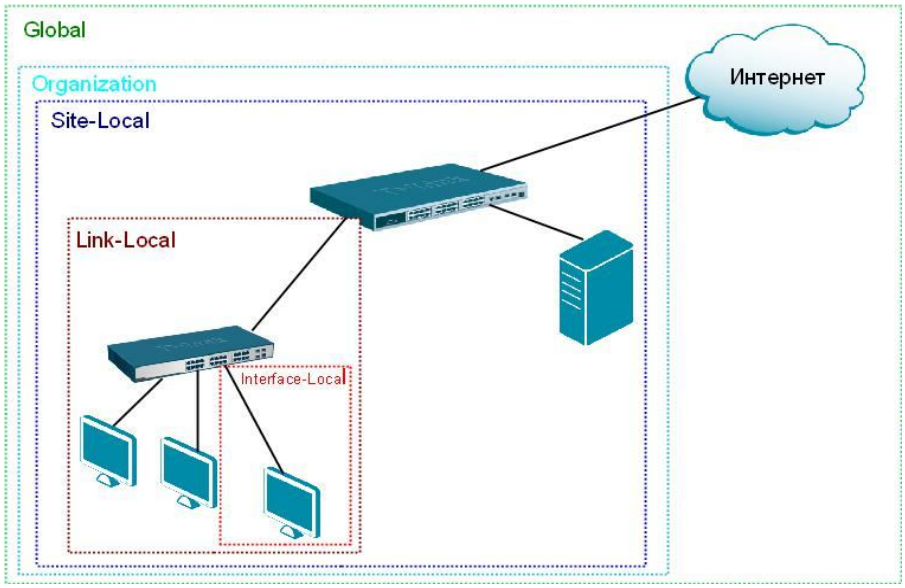


Рис. 6.27. Диапазон действия многоадресных адресов IPv6

Временные многоадресные адреса используются в пределах данной области (Scope). Последние 112 бит группового адреса определяют идентификатор группы (*Group ID*) в пределах области действия адреса.

Функцию широковещательных адресов в протоколе IPv6 выполняют специальные многоадресные адреса, которые не назначаются многоадресной группе:

- FF01::1 – идентифицирует группу, включающую в себя все IPv6-узлы в пределах диапазона Interface-Local;
- FF02::1 – идентифицирует группу, включающую в себя все IPv6-узлы в пределах диапазона Link-Local;
- FF01::2 – идентифицирует группу всех IPv6-маршрутизаторов в пределах диапазона Interface-Local;
- FF02::2 – идентифицирует группу всех IPv6-маршрутизаторов в пределах диапазона Link-Local;
- FF02::5 – идентифицирует группу всех IPv6-маршрутизаторов в пределах диапазона Site-Local.

В протоколе IPv6 многоадресные адреса используются также в процессе разрешения адресов для сегмента сети, т. е. получения адресов канального уровня (MAC-адресов) на основе известных IPv6-адресов. Адрес, который используется в процессе разрешения адресов, называется *Solicited-Node* (адрес запрашивающего узла). Он должен присваиваться каждому интерфейсу вместе с индивидуальными адресами. Этот адрес исользуется только внутри линии связи или в сегментах сети.

Solicited-Node-адрес формируется из младших 24 бит поля Interface ID индивидуального или альтернативного адреса путем прибавления префикса FF02:0:0:0:0:1:FF00::/104 (рис. 6.28).

Рис. 6.28. Формирование адреса Solicited-Node

Устройство, которому необходимо получить адрес канального уровня, отправляет запрос всем узлам на многоадресный адрес Solicited-Node. В результате на запрос ответят только те устройства, у которых совпадают последние 24 бита поля Interface ID. Такой механизм в отличие от IPv4, где запрос отправляется широковещательно, позволяет сократить число узлов, обрабатывающих запрос.

6.7.3 Альтернативные адреса

В протоколе IPv6 появился новый тип адреса – альтернативный адрес. Этот IPv6-адрес назначается нескольким интерфейсам. При этом пакет, отправленный на этот адрес, направляется на «ближайший» (имеющий минимальную метрику маршутизации) интерфейс. В соответствии с RFC 4291 альтернативный адрес не может использоваться в качестве адреса источника в пакетах IPv6 и назначается только маршрутизаторам (коммутаторам L3), а не конечным узлам. Пакеты, отправленные на альтернативный адрес, будут доставлены всем маршрутизаторам сети, но данные будут передаваться только через интерфейс «ближайшего» маршрутизатора, как показано на рис. 6.29.

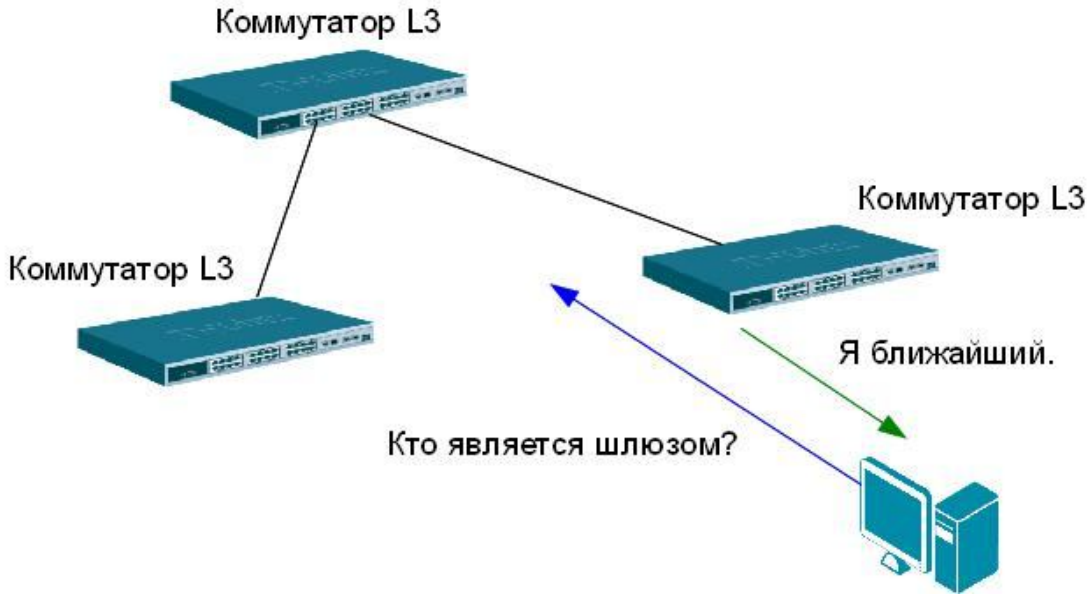


Рис. 6.29. Использование альтернативного адреса IPv6

Альтернативным адресам не выделен специальный блок адресов, они входят в адресное пространство индивидуальных адресов. Он состоит из префикса подсети (Subnet prefix), за которым следуют все 0.

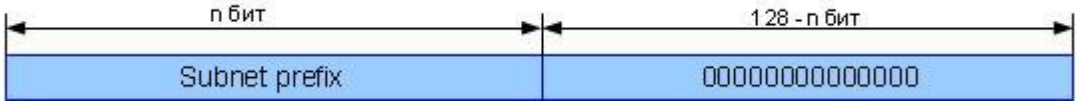


Рис. 6.30. Формат альтернативных адресов IPv6

Префикс подсети может занимать столько бит, сколько необходимо для уникальной идентификации подсети, которую обслуживают маршрутизаторы (коммутаторы L3).

Одним из применений альтернативных адресов является идентификация группы маршрутизаторов, принадлежащих интернет-провайдеру. Такие адреса в маршрутном заголовке IPv6 могут использоваться в качестве промежуточных, чтобы обеспечить доставку пакета через определенного провайдера или последовательность провайдеров. В документе RFC 4291 описывается схема применения альтернативных адресов.

6.8 Формирование идентификатора интерфейса

Как говорилось ранее, идентификатор интерфейса (Interface ID) представляет собой 64-битное поле IPv6-адреса, используемое для идентификации интерфейса в сегменте сети. Уникальный идентификатор интерфейса может быть получен несколькими способами:

- настроен вручную;
- назначен с помощью протокола DHCPv6;
- сгенерирован автоматически случайным образом;
- сформирован из 48-битного MAC-адреса путем его преобразования в формат Modified EUI-64.

Для всех индивидуальных адресов, начинающихся с битов 001, идентификатор интерфейса должен быть сформирован в соответствии с форматом Modified EUI-64.

Рассмотрим получение идентификатора интерфейса путем преобоазования его MAC-адреса. Так как MAC-адрес состоит из 48 бит, а для идентификатора интерфейса необходимо 64 бита, требуется расширение MAC-адреса преобразованием его в адрес EUI-64.

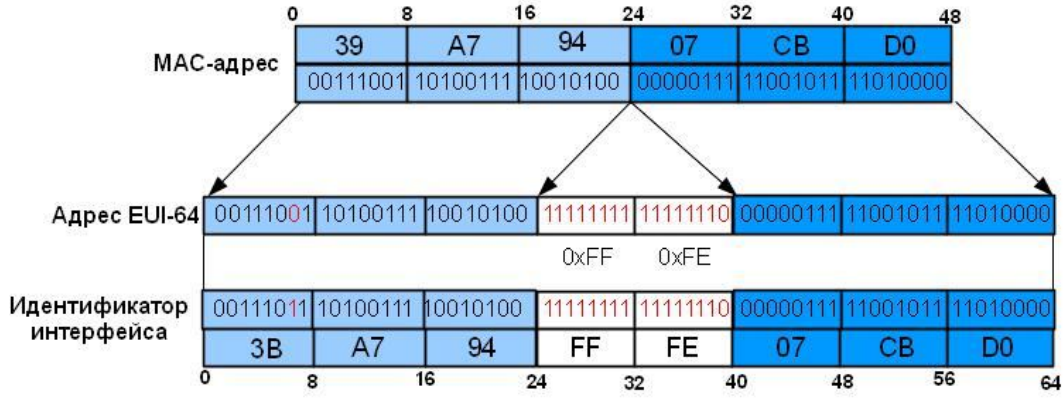


Рис. 6.31. Преобразование MAC-адреса в идентификатор интерфейса

MAC-адрес делится на две части по 24 бита, между которыми вставляется блок битов FFFE. Идентификатор интерфейса формируется путем установления в адресе EUI-64 значения 1 в бите U (7 бит слева), определяющего, является ли MAC-адрес универсальным или локально администрируемым.

В случаях, когда идентификатор интерфейса формируется из MAC-адреса, существует возможность определения и отслеживания трафика конкретного узла независимо от префикса IPv6-адреса. С учетом этого в RFC 3041 описан метод генерации узлом псевдослучайного идентификатора интерфейса, изменяемого с течением времени. Итоговый IPv6-адрес, основанный на таком псевдослучайном идентификаторе интерфейса, называют *временным адресом*, который рекомендуется для использования в интернете.

6.9 Способы конфигурации IPv6-адреса

В отличие от протокола IPv4, где настройка параметров узла проводится либо вручную, либо с помощью протокола DHCP, в протоколе IPv6 узел может практически самостоятельно сконфигурировать параметры своих интерфейсов.

В протоколе IPv6 определены два механизма автоконфигурации: *Stateless autoconfiguration* (описан в RFC 4862) и *Stateful autoconfiguration* (описан RFC 3315).

Stateful autoconfiguration позволяет узлам получать адрес интерфейса и/или конфигурационные параметры с помощью протокола DHCPv6.

Stateless autoconfiguration позволяет узлам генерировать свой собственный адрес на основе комбинации локально доступной информации и информации, объявляемой маршрутизаторами (коммутаторами 3-го уровня). Маршрутизаторы объявляют префиксы,

идентифицирующие подсеть (или подсети), а узлы самостоятельно генерируют идентификаторы интерфейсов. В отсутствии маршрутизирующего устройства узлы могут автоматически генерировать Link-Local Unicast IPv6-адрес.

Механизмы автоконфигурации stateless и stateful могут дополнять друг друга и использоваться совместно.

Рассмотрим последовательность действий, которые выполняются в процессе автоконфигурации узла.

Шаг 1. Генерация Link-Local Unicast IPv6-адрес с префиксом FE80::/10.

Шаг 2. Тестирование адреса на уникальность.

Узел проверяет, используется ли уже такой адрес в локальном сетевом сегменте. Для этого он отправляет сообщение *Neighbor Solicitation (NS)* протокола *Neighbor Discovery Protocol (NDP)*. Если в ответ на него получено сообщение *Neighbor Advertisement (NA)*, значит, этот адрес уже используется другим узлом (подробнее о протоколе NDP см. в разделе 6.11). В этом случае процесс автоконфигурации завершается и требуется ручная настройка интерфейса.

Шаг 3. Присвоение адреса Link-Local Unicast.

Если тест на уникальность успешно пройден, узел присваивает полученный на шаге 1 IPv6-адрес своему интерфейсу. Этот адрес может использоваться только для связи с устройствами внутри сегмента сети.

Шаг 4. Обнаружение маршрутизатора (коммутатора 3-го уровня).

После присвоения интерфейсу Link-Local Unicast-адреса узел отправляет сообщение *Router Solicitation (RS)* протокола NDP, используя в качестве адреса источника свой Link-Local Unicast IPv6-адрес, а в качестве адреса получателя – адрес группы всех маршрутизаторов в сегменте сети FF02::2. Если в сети имеются маршрутизаторы (коммутаторы L3), они отвечают сообщением *Router Advertisement (RA)* и сообщают узлам, каким образом продолжать процесс автоконфигурации. Адресом источника в сообщении *Router Advertisement (RA)* является локальный адрес маршрутизатора, а адресом получателя – FF02::1 группы всех IPv6-узлов в пределах области Link-Local.

Шаг 5. Генерация Global Unicast-адреса.

1. В случае Stateless autoconfiguration Global Unicast-адрес состоит из префикса, предоставленного маршрутизатором (коммутатором 3-го уровня) и идентификатора интерфейса, созданного на шаге 1.

Рис. 6.32. Генерация Global Unicast-адреса при использовании механизма Stateless autoconfiguration

2. В случае Stateful autoconfiguration, узел отправляет запрос к DHCPv6-серверу об аренде IPv6-адреса/длины префикса и других сетевых параметров. Главное отличие

протокола DHCPv6 от DHCPv4 заключается в том, что DHCPv6-сервер не рассылает DHCPv6-клиентам информацию о шлюзе по умолчанию.

В протоколе IPv6, так же как и в протоколе IPv4, существует возможность ручной настройки на интерфейсе IPv6-адреса, шлюза по умолчанию, длины префикса. Ручная настройка обычно используется для конфигурации интерфейсов маршрутизаторов (коммутаторов L3) или других сетевых устройств. Если в сети нет маршрутизирующих устройств, которые рассылают объявления с информацией, требуемой для автоматической конфигурации узла, интерфейс узла может быть настроен вручную.

6.9.1 Пример настройки автоматической конфигурации (Stateless autoconfiguration) адреса IPv6

Рассмотрим пример реализации автоматической настройки (Stateless autoconfiguration) Unique-Local Unicast-адресов узлов локальной сети с помощью коммутатора 3-го уровня DES-3810-28. Коммутатор отправляет узлам локальной сети информацию о префиксе после получения от них сообщений RS. Узлы автоматически формируют свои Unique-Local Unicast-адреса на основе данных, полученных от коммутатора (рис. 6.33).

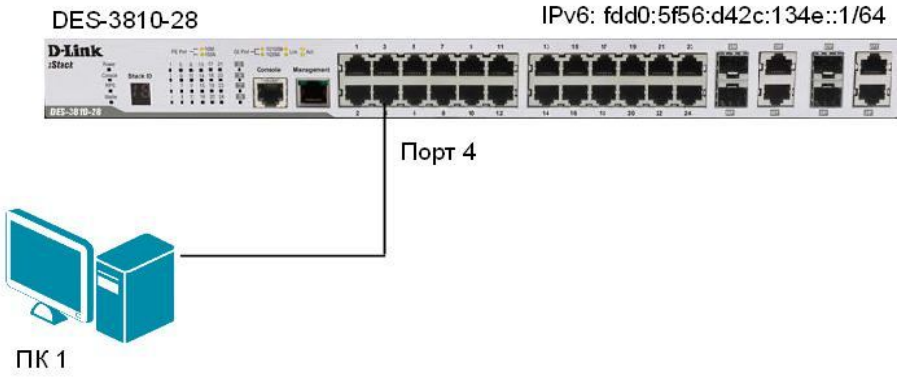


Рис. 6.33. Схема сети

Настройка коммутатора DES-3810-28

- Настроить Unique-Local Unicast-адрес на интерфейсе System.

config ipif System ipv6 ipv6address fdd0:5f56:d42c:134e::1/64

- Активизировать автоматическую конфигурацию адреса на интерфейсе System.

config ipv6 nd ra ipif System state enable

Следует отметить, что в качестве префикса, рассылаемого узлам, будет использоваться префикс адреса интерфейса System (в данном случае – fdd0:5f56:d42c:134e::/64).

6.10 Планирование подсетей IPv6

При использовании протокола IPv4 каждой организации выделялась сеть класса A, B или C, и организация самостоятельно формировала подсети из полученной сети. Этот же принцип справедлив и для сетей IPv6. Адресное пространство IPv6 позволяет гибко планировать схему адресации сети. Рассмотрим пример планирования IPv6-подсетей.

Предположим, что организация планирует использовать в своей сети Unique-Local Unicast-адреса и хочет разбить сеть на 5 подсетей. Сначала формируют 64-битовый префикс сети. Unique-Local Unicast-адреса начинаются с префикса FD00::/8, далее с помощью генератора получают Global ID (40 бит), например 895a473947. Затем назначают 5 номеров

подсети (Subnet ID) длиной 16 бит. Для получения номера подсети можно также воспользоваться генератором (таблица 9).

Таблица 9 **Формирование подсетей IPv6**

Номер подсети	Префикс сети	Диапазон адресов
0710	fd89:5a47:3947:0710::/64	fd89:5a47:3947:710:0:0:0:0 – fd89:5a47:3947:710:ffff:ffff:ffff:ffff
0711	fd89:5a47:3947:0711::/64	fd89:5a47:3947:711:0:0:0:0 – fd89:5a47:3947:711:ffff:ffff:ffff:ffff
0712	fd89:5a47:3947:0712::/64	fd89:5a47:3947:712:0:0:0:0 – fd89:5a47:3947:712:ffff:ffff:ffff:ffff
0713	fd89:5a47:3947:0713::/64	fd89:5a47:3947:713:0:0:0:0 – fd89:5a47:3947:713:ffff:ffff:ffff:ffff
0714	fd89:5a47:3947:0714::/64	fd89:5a47:3947:714:0:0:0:0 – fd89:5a47:3947:714:ffff:ffff:ffff:ffff

Идентификаторы узлов в каждой подсети могут быть сформированы динамически одним из описанных ранее способов или вручную.

6.11 Протокол NDP

Изменения, которые были сделаны в IPv6, коснулись не только самого протокола IP, но и служебных протоколов сетевого уровня. В частности, в стеке TCP/IPv4 для разрешения адресов канального уровня используется протокол ARP. В стеке TCP/IPv6 функция разрешения адресов и ряд функций, относящихся к взаимодействию устройств в локальной сети, реализованы протоколом *NDP* (*Neighbor Discovery Protocol – протокол обнаружения соседей)*. Понятие «сосед» используется в различных сетевых стандартах и технологиях для обозначения устройств, способных отправлять сообщения непосредственно друг другу.

В RFC 4861 определены девять функций, выполняемых протоколом NDP. Для ясности эти функции можно разбить на три группы, как показано на рис. 6.34.

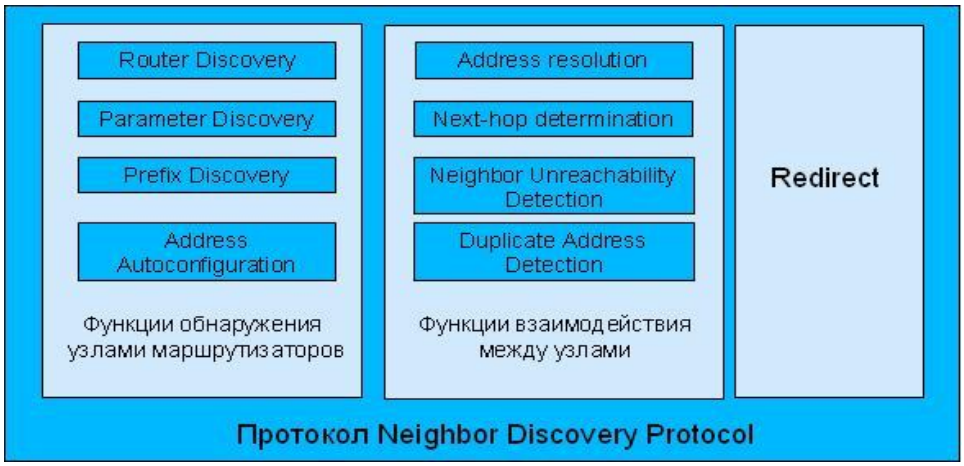


Рис. 6.34. Функции, выполняемые протоколом NDP

Функции обнаружения маршрутизаторов (коммутаторов 3-го уровня) узлами:

- *Router Discovery* – позволяет узлам локальной сети обнаруживать маршрутизаторы и получать от них сетевые параметры, необходимые для автоконфигурации;
- *Parameter Discovery* – позволяет узлам получать параметры локальной сети и/или маршрутизаторов, например MTU локального канала связи;
- *Prefix Discovery* – используется для определения префикса сети;
- *Address Autoconfiguration* – необходима для автоконфигурации узлов и взаимодействия между ними.

Функции взаимодействия между узлами:

- *Address Resolution* – функция разрешение IPv6-адресов канального уровня;
- *Next-Hop Determination* – позволяет определить IPv6-адрес назначения пакета и путь до следующего маршрутизатора;

- *Neighbor Unreachability Detection* – позволяет отслеживать состояние каналов связи между соседними узлами локальной сети;
- *Duplicate Address Detection* – позволяет определить дублирование адресов узлов локальной сети.

Последняя группа функций – *Redirect* – используется маршрутизаторами для уведомления узлов о наилучшем маршруте к пункту назначения.

Большинство функций протокола NDP выполняется с использованием пяти сообщений протокола ICMPv6:

1. *Router Solicitation* – отправляется узлами для того, чтобы запросить любой локальный маршрутизатор отправить сообщение Router Advertisement, не дожидаясь следующего периодического объявления. Используется при автоконфигурации узла;

2. *Router Advertisement* – регулярно отправляется маршрутизаторами для того, чтобы объявить о своем существовании в сети и предоставить узлам информацию о префиксе и/или дополнительных параметрах. Это сообщение также может быть отправлено в ответ на сообщение Router Solicitation;

3. *Neighbor Solicitation* – отправляется узлом для того, чтобы определить адрес канального уровня соседнего устройства или проверить доступность соседа с помощью адреса канального уровня, хранимого в NDP-таблице. Также используется для определения дублирования адресов *(Duplicate Address Detection)*;

4. *Neighbor Advertisement* – отправляется в ответ на сообщение Neighbor Solicitation. Это сообщение также может быть отправлено узлом при изменении адреса канального уровня;

5. *Redirect* – используется маршрутизирующими устройствами для уведомления узлов о наилучшем маршруте к пункту назначения.

6.11.1 Разрешение адресов IPv6 с помощью протокола NDP и определение недоступности соседа

Базовая концепция разрешения IPv6-адресов осталась такой же, как и в IPv4. При необходимости отправки IPv6-пакета соседнему устройству в локальной сети и отсутствии информации о физическом адресе получателя устройство инициирует процесс разрешения адресов, но использует не широковещательный ARP-запрос как в IPv4, а сообщение Neighbor Solicitation (NS), отправляемое на групповой Solicited-Node-адрес. При получении сообщения Neighbor Solicitation устройство назначения отправляет в ответ устройству-отправителю сообщение Neighbor Advertisement (аналогично ARP Reply в IPv4). Поскольку адрес Solicited-Node не является уникальным, то устройство-получатель должно убедиться, что оно является тем устройством, чей адрес пытается разрешить устройство-отправитель.

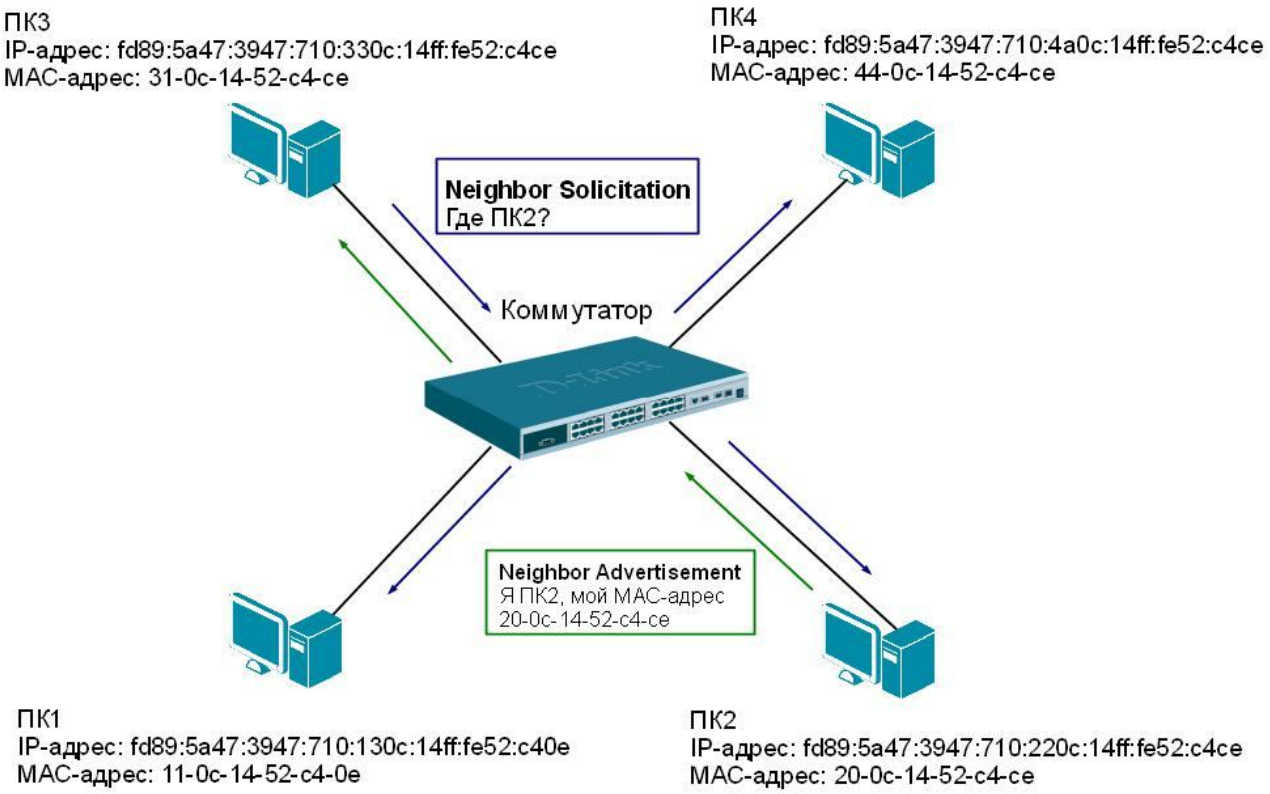


Рис. 6.35. Разрешение адресов с помощью протокола NDP

На основании полученного сообщения Neighbor Advertisement, устройство добавляет в NDP-таблицу *(neighbor cache)* новую запись, связывающую IPv6-адрес с соответствующим MAC-адресом соседнего устройства, от которого это сообщение получено. Так же, как и в таблице ARP, в NDP-таблице могут храниться и статические и динамические записи (рис. 6.36).

```
DES-3810-28:admin#show ipv6 neighbor_cache ipif System all
Command: show ipv6 neighbor_cache ipif System all

Neighbor                  Link Layer Address  Interface    State
------------------------  ------------------  -----------  -----
FE80::20B:6AFF:FECF:7EC6  00-0B-6A-CF-7E-C6   System        T

Total Entries: 1

State:
(I) means Incomplete state. (R) means Reachable state.
(S) means Stale state.      (D) means Delay state.
(P) means Probe state.      (T) means Static state.
```

Рис. 6.36. Пример NDP-таблицы на коммутаторе DES-3810-28

Динамическая запись в NDP-таблице может находиться в одном из пяти состояний:

1. *Incomplete* – состояние, когда сообщение Neighbor Solicitation отправлено на групповой адрес Solicited-Node, но ответное сообщение Neighbor Advertisement еще не получено;

2. *Reachable* – состояние, когда сообщение Neighbor Advertisement получено. Продолжительность этого состояния записи в NDP-таблице ограничено таймером *ReachableTime* (по умолчанию 30 секунд);

3. *Stale* – состояние, в которое переходит запись по истечении времени таймера ReachableTime с момента последнего получения сообщения Neighbor Advertisement;

4. *Delay* – состояние, в которое переходит запись при передаче данных соседнему устройству. При этом устанавливается таймер *Delay_First_Probe_Time* (по умолчанию 5 секунд). Если по истечении времени таймера запись все еще остается в состоянии Delay, статус записи меняется на Probe. Если же подтверждение достижимости было получено, состояние записи меняется на Reachable.

5. *Probe* – состояние записи, при котором устройство отправляет сообщение Neighbor Solicitation через промежутки времени, определяемые таймером *RetransTimer* (по умолчанию 10 секунд). Если в течение трех последовательных передач сообщения Neighbor Solicitation получено сообщение Neighbor Advertisement, то запись переходит в состояние Reachable, в противном случае запись удаляется из NDP-таблицы.

Сообщения Neighbor Solicitation и Neighbor Advertisement используются не только для разрешения адресов, у них есть еще одно предназначение – *определение недоступности соседа (Neighbor Unreachability Detection, NUD)*. Функция NUD позволяет отслеживать состояние каналов связи между соседними узлами локальной сети. Операции функции NUD выполняются параллельно с отправкой пакетов соседним устройствам, и если между ними нет обмена данными, то сообщения Neighbor Solicitation и Neighbor Advertisement не отправляются.

6.11.1.1 Пример настройки разрешения адресов с помощью протокола NDP

Рассмотрим пример формирования NDP-таблицы (neighbor cache) на коммутаторе 3-го уровня DES-3810-28.

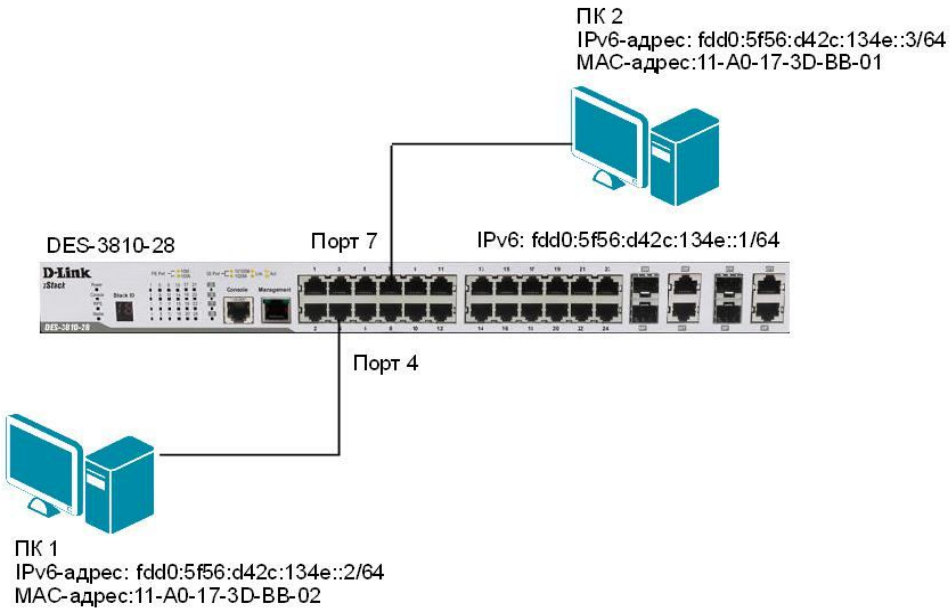


Рис. 6.37. Схема сети

Настройка коммутатора DES-3810-28

- Создать статическую запись для ПК 1 в NDP-таблице.

create ipv6 neighbor_cache ipif System fdd0:5f56:d42c:134e::2 11:A0:17:3D:BB:02

- Настроить время периодической отправки сообщений Neighbor Solicitation с интерфейса System для создания динамических записей в NDP-таблице.

config ipv6 nd ns ipif System retrans_time 400

- Посмотреть информацию о соседних устройствах, подключенных к интерфейсу System (NDP-таблицу).

show ipv6 neighbor_cache ipif System all

6.11.2 Определение дублирования адресов

При использовании механизма автоконфигурации IPv6-адреса необходимо определить, что адрес Link-Local, который сегментирован узлом, уже не используется другим узлом, т. е. проверить *дублирование адресов (Duplicate Address Detection, DAD)*. Для этого в сеть отправляется сообщение Neighbor Solicitation, и если в ответ на него получено сообщение Neighbor Advertisement, это означает, что данный адрес уже используется другим узлом. В этом случае процесс автоконфигурации завершается и требуется ручная настройка интерфейса.

6.11.3 Обнаружение маршрутизатора

Одной из важных функций протокола NDP является реализация процесса обнаружения узлами локальных маршрутизаторов – *Router Discovery*. При этом узлы локальной сети обнаруживают соседние маршрутизаторы (коммутаторы L3) и получают от них сетевые параметры, необходимые для автоконфигурации (рис. 6.38). Операция обнаружения узлами маршрутизаторов выполняется с помощью сообщений ICMPv6 Router Advertisement и Router Solicitation.

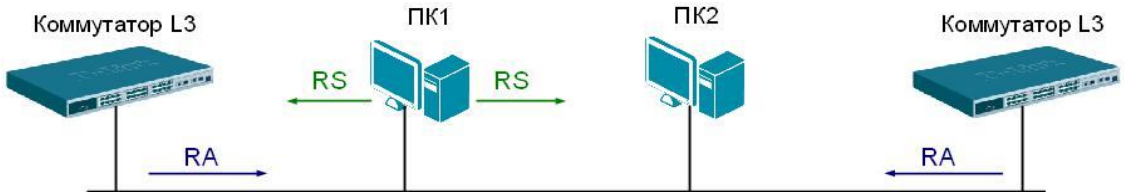


Рис. 6.38. Обнаружение локального коммутатора 3-го уровня

В процессе обнаружения маршрутизаторов (коммутаторов 3-го уровня) узлы выполняют следующие функции:

- *Рассылка объявлений.* Узлы прослушивают объявления Router Advertisement, передаваемые маршрутизаторами в локальной сети через определенные интервалы времени и обрабатывают их. Объявления содержат список префиксов, в том числе необходимых для автоконфигурации, а также могут включать информацию о шлюзе по умолчанию;
- *Генерация запросов.* При определенных условиях (например, узел загружается и ему требуются параметры для конфигурации интерфейса) узлы могут генерировать сообщения Router Solicitation. С помощью этого сообщения узел запрашивает любой локальный маршрутизатор о мгновенном предоставлении информации, т.е. отправке сообщения Router Advertisement;
- *Автоконфигурация.* Если в сети настроен механизм автоконфигурации Stateless autoconfiguration, то узел будет использовать информацию, полученную от локального маршрутизатора, чтобы автоматически сконфигурировать свой IPv6-адрес и другие сетевые параметры.

6.12 Понятие маршрутизации

Маршрутизация является одним из процессов, который выполняется на сетевом уровне модели OSI, и позволяет объединить IP-сегменты в единую сеть. Маршрутизация

выполняется *маршрутизаторами* или *коммутаторами 3-го уровня*, которые перенаправляют пакеты из одной IP-сети в другую, даже в том случае, если заранее неизвестно расположение получателя пакета.

Пакет, посылаемый из одной IP-сети в другую, достигает маршрутизирующего устройства, принимающего решение о его перенаправлении на основе IP-адреса назначения, который сравнивается с информацией, находящейся в *таблице маршрутизации*. Таблица маршрутизации хранится на маршрутизаторе и содержит записи, представляющие собой список наилучших маршрутов в соответствующие сети. В том случае, если к сети назначения имеется несколько путей, в таблицу маршрутизации будет помещен маршрут, у которого наилучшая метрика, определяемая на основании загрузки, полосы пропускания, задержки, стоимости или надежности канала связи.

Routing Table

IP Address/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
172.16.8.4/30	0.0.0.0	iptv-2	1	Local
172.16.8.8/30	172.16.8.6	iptv-2	2	RIP
172.16.8.12/30	0.0.0.0	stream	1	Local

Total Entries: 7

Рис. 6.39. Таблица маршрутизации

Существует четыре типа записей в таблице маршрутизации:

1. *Статический маршрут (Static Route)* – задается вручную системным администратором;
2. *Динамический маршрут (Dynamic Route)* – создается в процессе обмена маршрутизирующими устройствами маршрутной информацией;
3. *Маршрут по умолчанию (Default Route)* – задается вручную администратором в качестве пути, который используется в том случае, если другой маршрут к пункту назначения неизвестен;
4. *Локальный маршрут (Local Route)* – адрес непосредственно подключенной к интерфейсам маршрутизатора локальной сети. Задается в процессе конфигурирования устройства.

Каждая запись таблицы маршрутизации содержит следующую информацию:

- *адрес назначения (IP Address)* — адрес сети (в некоторых случаях узла) назначения;
- *маска сети (Netmask)* — маска, соответствующая адресу назначения (для сетей IPv4 маска /32 (255.255.255.255) позволяет указать единичный узел сети);
- *адрес шлюза (Gateway)* — сообщает маршрутизатору о том, что получатель пакета подключен непосредственно или доступен через другой маршрутизатор, который называется *следующим транзитным узлом (next hop)*;
- *интерфейс (Interface)* — идентификатор интерфейса, через который пакет покидает устройство;
- *метрика (Cost)* — числовой показатель, определяющий предпочтительность маршрута. Чем меньше значение метрики, тем более предпочтителен маршрут;
- *тип протокола (Protocol)* – информация о методе создания записи в таблице маршрутизации.

6.12.1 Процесс обработки пакета маршрутизирующим устройством

Предположим, что ПК 1, находящийся в сети 192.168.1.0/24 отправляет запрос серверу, расположенному в сети 172.11.10.0/16 (рис. 6.40).

Routing Table (SW1)			
IP Address	Gateway	Interface	Cost
192.168.1.0	0.0.0.0	Int1	1
192.168.2.0	0.0.0.0	Int2	1
172.11.10.0	192.168.2.2	Int2	2

Routing Table (SW2)			
IP Address	Gateway	Interface	Cost
172.11.10.0	0.0.0.0	Int3	1
192.168.2.0	0.0.0.0	Int2	1
192.168.1.0	192.168.2.1	Int2	2

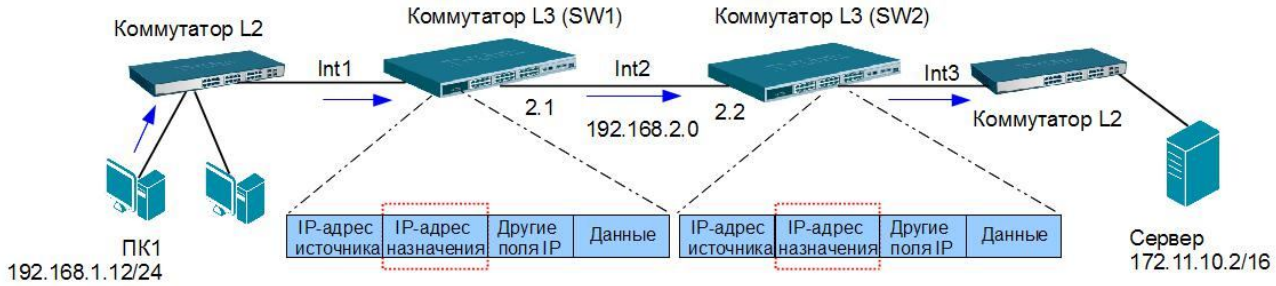


Рис. 6.40. Процесс обработки пакета маршрутизаторами

Коммутатор 3-го уровня SW1 получает кадр от ПК 1 на интерфейс Int1 и проверяет его целостность. Если кадр не поврежден, то коммутатор SW1 удаляет его заголовок и концевик, в противном случае кадр отбрасывается. Далее из заголовка полученного пакета маршрутизатор SW1 извлекает *IP-адрес назначения (Destination address)* и сравнивает его сетевую часть с записями в таблице маршрутизации. Если соответствие найдено, то данные передаются на нужный интерфейс маршрутизатора, в данном случае на интерфейс Int2 SW2. Если в таблице маршрутизации нет совпадений с сетевой частью IP-адреса и не определен шлюз по умолчанию, то пакет отбрасывается и отправителю передается ICMP-сообщение *Destination Unreachable (получатель недостижим)*. Затем интерфейс Int2 маршрутизатора SW1 формирует новый кадр, инкапсулируя в него пакет, и пересылает его следующему на пути маршрутизатору SW2, найденному в соответствии с таблицей маршрутизации.

Следует отметить, что когда пакет проходит маршрутизирующее устройство у него отбрасываются заголовок и концевик. Это связано с тем, что информация канального уровня служит для передачи данных между непосредственно подключенными устройствами, а информация сетевого уровня необходима для сквозной передачи данных через составную сеть.

После того, как маршрутизатор извлек из заголовка пакета IP-адрес назначения, он сравнивает сетевую часть IP-адреса с записями в таблице маршрутизации (рис.6.41) и принимает решение о дальнейшем пути пакета.

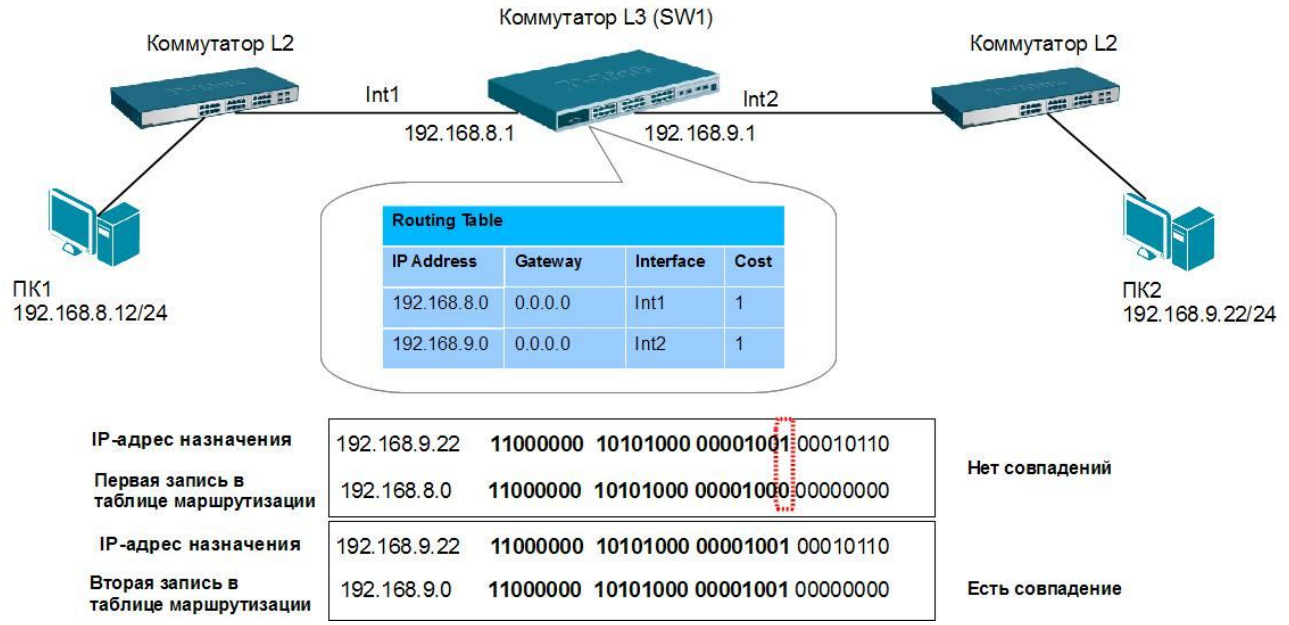


Рис. 6.41. Процесс поиска маршрута в таблице маршрутизации

6.12.2 Коммутация третьего уровня

Коммутаторы 3-го уровня имеют некоторые особенности, отличающие их от традиционных маршрутизаторов и коммутаторов 2-го уровня:

- одновременная поддержка функций маршрутизации и коммутации;
- обязательная поддержка механизма VLAN;
- реализация функций маршрутизации на аппаратном уровне с использованием ASIC.

Использование контроллеров ASIC является главной характеристикой, отличающей коммутаторы 3-го уровня от традиционных маршрутизаторов, так как при этом повышается производительность системы за счет выполнения операций аппаратно, благодаря чему не возникают накладные расходы, связанные с выборкой и интерпретацией хранимых команд. В связи с этим коммутаторы 3-го уровня маршрутизируют пакеты в среднем в 10-100 раз быстрее, чем традиционные маршрутизаторы.

6.12.3 Статическая и динамическая маршрутизация

Новые маршруты добавляются в таблицу маршрутизации *статически* или *динамически*. При статической маршрутизации записи о маршрутах к сети назначения добавляются вручную, в то время как при динамической маршрутизации записи заносятся и обновляются с помощью протоколов маршрутизации (routing protocols). У каждого из этих методов есть свои достоинства и недостатки.

Статическую маршрутизацию полезно использовать, когда к сети назначения имеется небольшое количество маршрутов и администратор вручную может задать наилучший маршрут, тем самым снижая нагрузку на процессор маршрутизатора. Также статическую маршрутизацию используют в том случае, если требуется скрыть маршрутную информацию, поскольку при динамической маршрутизации маршрутизирующие устройства обмениваются друг с другом обновлениями о маршрутах, которые могут быть перехвачены злоумышленниками. Если в силу каких-либо причин один из маршрутизаторов (коммутаторов L3), использующих статическую маршрутизацию, выходит из строя, он не сможет оповестить соседей о неисправности, и другие маршрутизаторы будут передавать пакеты по недоступному маршруту. В небольших сетях, где используется два или три

маршрутизатора, администратор может решить эту проблему достаточно быстро. В больших сетях предпочтительнее использовать динамическую маршрутизацию.

При динамической маршрутизации маршрутизирующие устройства сигнализируют соседям в случае обрыва соединения или обнаружения нового пути и автоматически обновляют сетевую топологию.

6.12.3.1 Пример настройки статической маршрутизации IPv4

Рассмотрим пример настройки статической IPv4-маршрутизации в сети, показанной на рис. 6.42. Сеть сегментирована с использованием VLAN. Требуется настроить маршрутизацию между VLAN v2 и v4 на коммутаторах 3-го уровня D-Link.

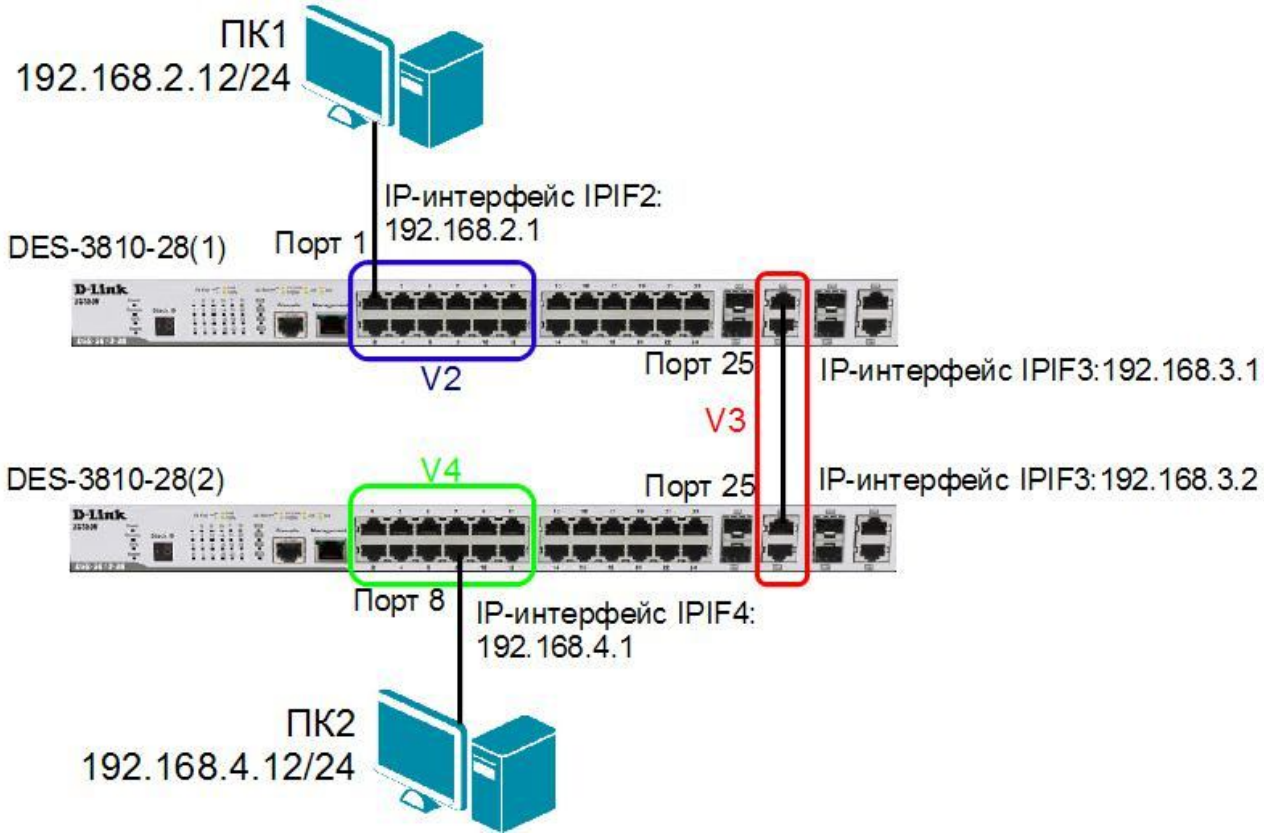


Рис. 6.42. Схема сети

Настройка коммутатора DES-3810-28 (1)

- Удалить порты коммутатора из VLAN по умолчанию для их использования в других VLAN.

config vlan default delete 1-24

- Создать VLAN v2 и v3 и добавить в соответствующие VLAN порты, которые необходимо настроить немаркированными.

create vlan v2 tag 2
config vlan v2 add untagged 1-12
create vlan v3 tag 3
config vlan v3 add untagged 25

- Создать IP-интерфейсы для VLAN v2 и v3 с именами IPIF2 и IPIF3 соответственно.

create ipif IPIF2 192.168.2.1/24 v2 state enable
create ipif IPIF3 192.168.3.1/24 v3 state enable

- Создать статический маршрут к сети 192.168.4.0/24.

create iproute 192.168.4.0/24 192.168.3.2

Настройка коммутатора DES-3810-28 (2)

- Удалить порты коммутатора из VLAN по умолчанию для их использования в других VLAN.

config vlan default delete 1-24

- Создать VLAN v4 и v3 и добавить в соответствующие VLAN порты, которые необходимо настроить немаркированными.

create vlan v4 tag 4
config vlan v4 add untagged 1-12
create vlan v3 tag 3
config vlan v3 add untagged 25

- Создать IP-интерфейсы для VLAN v4 и v3 с именами IPIF4 и IPIF3 соответственно.

create ipif IPIF4 192.168.4.1/24 v4 state enable
create ipif IPIF3 192.168.3.2/24 v3 state enable

- Создать статический маршрут к сети 192.168.2.0/24.

create iproute 192.168.2.0/24 192.168.3.1

6.12.3.2 Пример настройки статической маршрутизации IPv6

Рассмотрим пример настройки статической IPv6-маршрутизации между VLAN в пределах одного коммутатора D-Link (рис. 6.43).

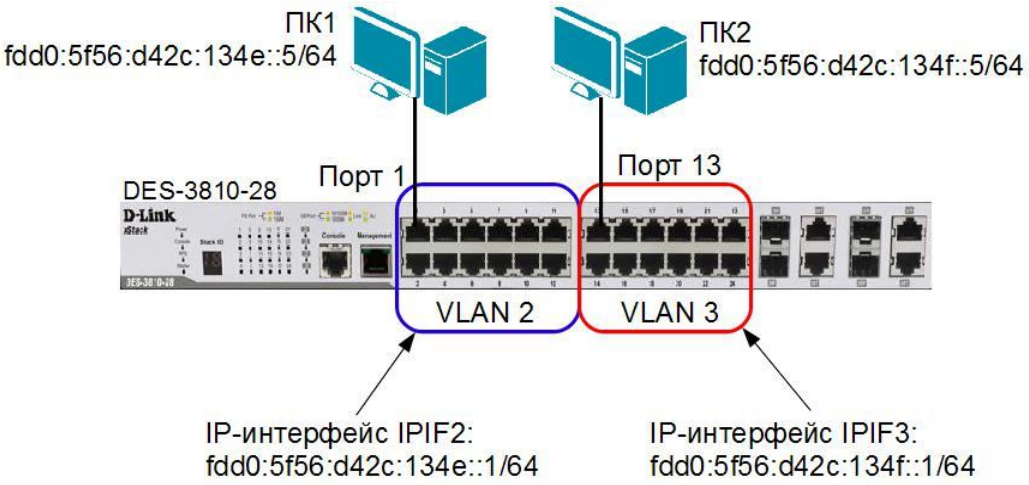


Рис. 6.43. Пример статической маршрутизации IPv6

Настройка коммутатора DES-3810-28

- Удалить порты коммутатора из VLAN по умолчанию для их использования в других VLAN.

config vlan default delete 1-24

- Создать VLAN v2 и v3 и добавить в соответствующие VLAN порты, которые необходимо настроить немаркированными.

create vlan vlan2 tag 2

config vlan vlan2 add untagged 1-12
create vlan vlan3 tag 3
config vlan vlan3 add untagged 13-24

- Создать IP-интерфейсы для VLAN v2 и v3 с именами IPIF2 и IPIF3 соответственно.

create ipif IPIF2 vlan2 state enable
create ipif IPIF3 vlan3 state enable

- Настроить IPv6-адрес для интерфейсов IPIF2 и IPIF3.

config ipif IPIF2 ipv6 ipv6address fdd0:5f56:d42c:134e::1/64
config ipif IPIF3 ipv6 ipv6address fdd0:5f56:d42c:134f::1/64

Внимание: при настройке маршрутизации в пределах одного коммутатора 3-го уровня D-Link, она начинает работать сразу после конфигурации IP-интерфейсов VLAN.

6.12.4 Протоколы динамической маршрутизации

Первоначально архитектура ядра сети интернет состояла из небольшого количества маршрутизаторов, которые хранили полную информацию о составной сети. По мере развития интернета стало увеличиваться количество сетей и соединяющих их маршрутизаторов. Это привело к тому, что количество маршрутной информации, обрабатываемой маршрутизаторами, стремительно увеличивалось. Таким образом, возникла необходимость перехода от централизованной к совершенно новой архитектуре, которая рассматривает глобальную сеть как набор независимых групп маршрутизаторов, так называемых автономных систем.

Автономная система (Autonomous system, AS) представляет собой группу маршрутизаторов (коммутаторов 3-го уровня) и IP-сетей, которые находятся под административным управлением. Например, сеть студенческого городка может быть автономной системой.

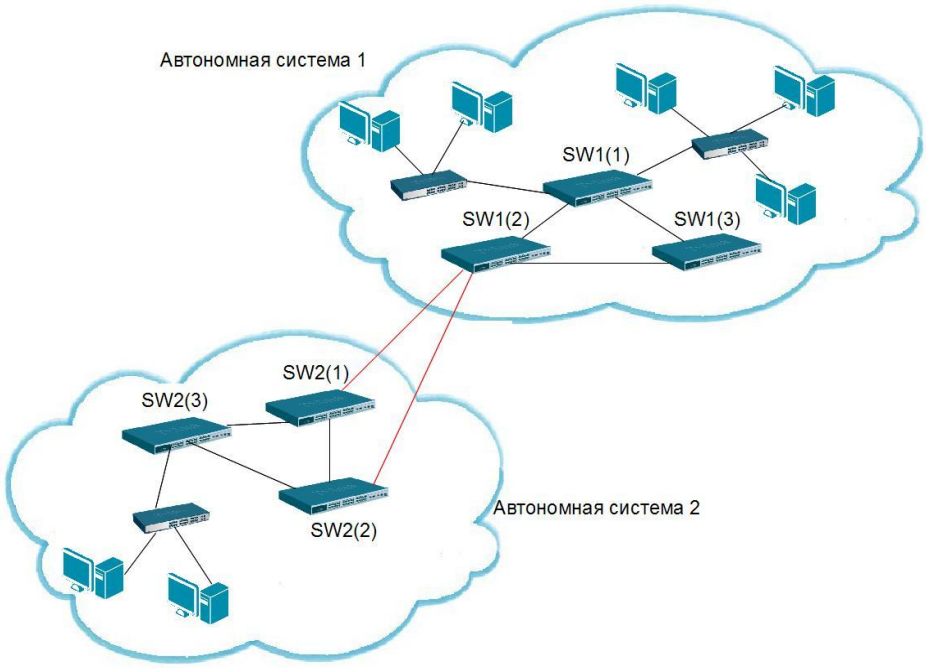


Рис. 6.44. Автономные системы

Для каждой автономной сиситемы выбираются протоколы динамической маршрутизации, которые позволяют обмениваться маршрутной информацией между

маршрутизирующими устройствами внутри автономной системы, так называемые *внутренние протоколы маршрутизации (Interior Routing Protocols)*. Существуют протоколы, используемые для обмена маршрутной информацией между автономными системами, которые называются *внешними протоколами маршрутизации (Exterior Routing Protocols)*.

Примеры внутренних протоколов маршрутизации:

- RIPv1 (Routing Information Protocol version 1) – описан в RFC 1058;
- RIPv2 (Routing Information Protocol version 2) – описан в RFC 2453;
- RIPng (Routing Information Protocol next generation) – описан в RFC 2080;
- OSPF (Open Shortest Path First) – описан в RFC 2328.

Самым известным протоколом внешней маршрутизации является BGP (Border Gateway Protocol), описанный в RFC 4271.

В основе каждого протокола маршрутизации лежит алгоритм, определяющий принцип работы и методы обработки маршрутной информации.

Алгоритм маршрутизации – это метод, который протокол маршрутизации использует для определения наилучшего маршрута к сети назначения и последующего включения его в таблицу маршрутизации. Алгоритмы маршрутизации для определения наилучшего маршрута используют различные *метрики*, отражающие число переходов, скорость прохождения пути, надежность пути, пропускную способность этого пути и т.д. Для того чтобы определить наилучший маршрут к сети назначения из всех имеющихся маршрутов, алгоритмы маршрутизации сравнивают их метрики. Чем меньше значение метрики, тем предпочтительнее выбранный маршрут. Наиболее часто в алгоритмах маршрутизации используются следующие метрики:

- *счетчик промежуточных узлов (Hop count)* или *число переходов* – количество маршрутизаторов (коммутаторов L3), через которые должен пройти пакет, прежде чем достигнет пункта назначения. При прохождении через маршрутизатор (коммутатор L3), значение счетчика узлов увеличивается на 1. Путь, для которого значение счетчика узлов равно 4, означает, что данные, отправленные по этому маршруту, пройдут через 4 маршрутизатора (коммутатора L3), прежде чем будут получены адресатом. Если существует несколько путей, маршрутизирующее устройство выбирает тот, для которого значение счетчика узлов наименьшее;
- *задержка передачи (Delay)* – время, требуемое на передачу пакета от отправителя к получателю;
- *надежность линии связи (Reliability)* – обычно обозначает относительное значение количества ошибок для каждого из каналов связи;
- *загруженность (Load)* – средняя загруженность канала связи;
- *пропускную способность (Bandwidth)* – пропускная способность канала связи;
- *стоимость (Cost)* – значение, вычисляемое обычно на основе пропускной способности, денежной стоимости или других единиц измерения, назначаемых администратором.

В зависимости от используемого алгоритма протоколы маршрутизации подразделяются на три класса:

- *дистанционно-векторные протоколы (Distance Vector Protocol)*;
- *протоколы с учетом состояния канала (Link State Protocol)*;
- *гибридные протоколы маршрутизации,* имеющие черты протоколов одного, так и другого класса.

6.13 Дистанционно-векторные протоколы маршрутизации

Дистанционно-векторные протоколы маршрутизации основаны на алгоритме Беллмана-Форда (Bellman-Ford) и используют его для поиска наилучшего маршрута к сети назначения. Каждый маршрут характеризуется двумя основными параметрами (рис. 6.45) – *расстоянием* (число переходов до сети назначения) и *вектором* (направление к сети назначения).

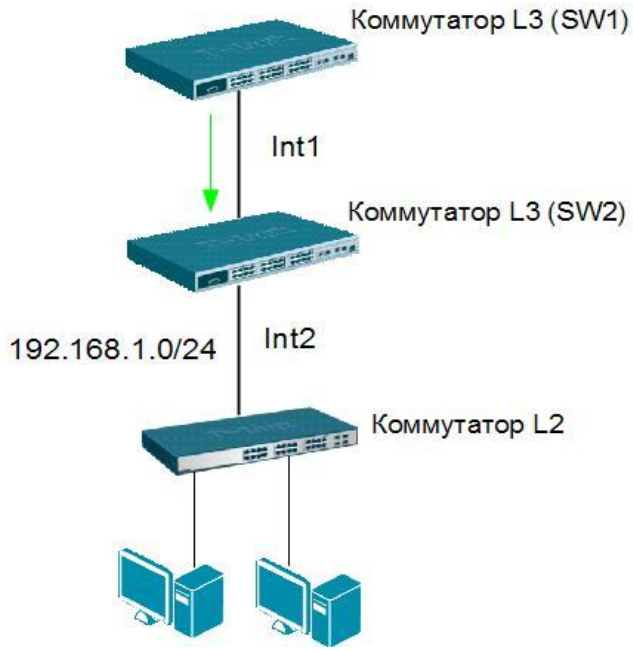


Рис. 6.45. Понятие вектора и расстояния

Для коммутатора SW1 сеть 192.168.1.0/24 доступна через один переход (расстояние), и пакет до этой сети должен быть передан на интерфейс Int1 коммутатора SW2 (направление).

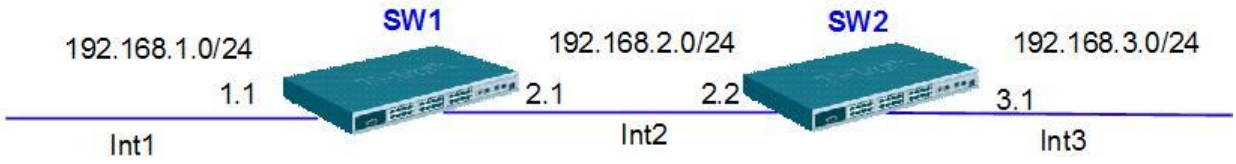
При использовании этого алгоритма маршрутизирующее устройство периодически (для протокола RIP каждые 30 секунд) пересылает всю или часть своей таблицы маршрутизации непосредственно подключенным маршрутизаторам. Получив таблицу маршрутизации от соседа, маршрутизирующее устройство обновляет свою таблицу маршрутизации, увеличивая метрику расстояния на 1. Далее эта таблица передается всем соседям и, таким образом, шаг за шагом информация о расстоянии распространяется по составной сети. Дистанционно-векторные протоколы рассылают периодические обновления, даже в том случае, если изменения в топологии сети не происходили.

6.13.1 Принцип работы дистанционно-векторного алгоритма маршрутизации

Работа маршрутизатора (коммутатора 3-го уровня), использующего дистанционно-векторный алгоритм маршрутизации, начинается с исследования своих соседей – непосредственно подключенных маршрутизирующих устройств. Как показано на рис. 6.46, первоначально (в момент времени t_0) в таблицах маршрутизации каждого маршрутизатора имеются записи о непосредственно подключенных к нему сетям с числом переходов Cost=1.

Далее маршрутизаторы направляют соседям широковещательные запросы с просьбой прислать свои таблицы маршрутизации. Соседние маршрутизаторы отвечают друг другу отправкой полных таблиц маршрутизации и сравнивают полученную маршрутную информацию со своей таблицей маршрутизации. В таблицу маршрутизации добавляются только маршруты к новым или уже известным сетям, но с лучшей метрикой. Обновляя свою таблицу (момент времени t_1), каждый маршрутизатор увеличивает значение метрики

маршрута на 1 для каждого добавляемого маршрута (в данном случае число переходов Cost=2). Это мера показывает, насколько далеко от маршрутизирующего устройства находится сеть назначения в данном направлении.



t_0

Routing Table (SW1)			
IP Address	Gateway	Interface	Cost
192.168.1.0	0.0.0.0	Int1	1
192.168.2.0	0.0.0.0	Int2	1

Routing Table (SW2)			
IP Address	Gateway	Interface	Cost
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	0.0.0.0	Int3	1

t_1

Routing Table (SW1)			
IP Address	Gateway	Interface	Cost
192.168.1.0	0.0.0.0	Int1	1
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	192.168.2.2	Int2	2

Routing Table (SW2)			
IP Address	Gateway	Interface	Cost
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	0.0.0.0	Int3	1
192.168.1.0	192.168.2.1	Int2	2

Рис. 6.46. Принцип работы дистанционно-векторного протокола

Чтобы лучше понять принцип работы дистанционно-векторного протокола маршрутизации, рассмотрим процесс обработки коммутатором 3-го уровня SW1 маршрута к сети 192.168.3.0/24, которая подключена к интерфейсу Int3 коммутатора 3-го уровня SW2.

Шаг 1. Сеть 192.168.3.0/24 напрямую подключена к коммутатору SW1, поэтому ее метрика равна 1.

Шаг 2. Коммутатор SW1, получив обновление от SW2, считает маршрут к сети 192.168.3.0/24 наилучшим, поскольку других маршрутов до этой сети у него нет.

Шаг 3. Коммутатор SW1 добавляет маршрут в свою таблицу маршрутизации и увеличивает значение метрики на 1 (Cost=2).

Шаг 4. Коммутатор SW1 для обнаруженного маршрута использует в качестве исходящего свой интерфейс Int2, поскольку обновление получено через него.

Шаг 5. Коммутатор SW1 для обнаруженного маршрута использует адрес 192.168.2.2 в качестве шлюза (следующего транзитного узла) для обнаружения маршрута, поскольку обновление получено от отправителя с этим IP-адресом.

6.13.2 Проблемы при функционировании дистанционно-векторного алгоритма маршрутизации

Алгоритм работы дистанционно-векторного протокола достаточно прост, но имеет свои недостатки – за счет медленной сходимости сети могут возникать *петли маршрутизации*. Под *сходимостью сети* подразумевают получение всеми маршрутизирующими устройствами информации о своей сети. Петли маршрутизации возникают тогда, когда два или более маршрутизатора (коммутатора 3-го уровня) пересылают пакеты по замкнутому пути, вследствие чего они никогда не доходят до нужного получателя. В сетях, где потоки данных значительны, петли маршрутизации могут приводить не только к потере пакетов, но и к неработоспособности всей сети. На рис. 6.47 показана схема, поясняющая эту проблему.

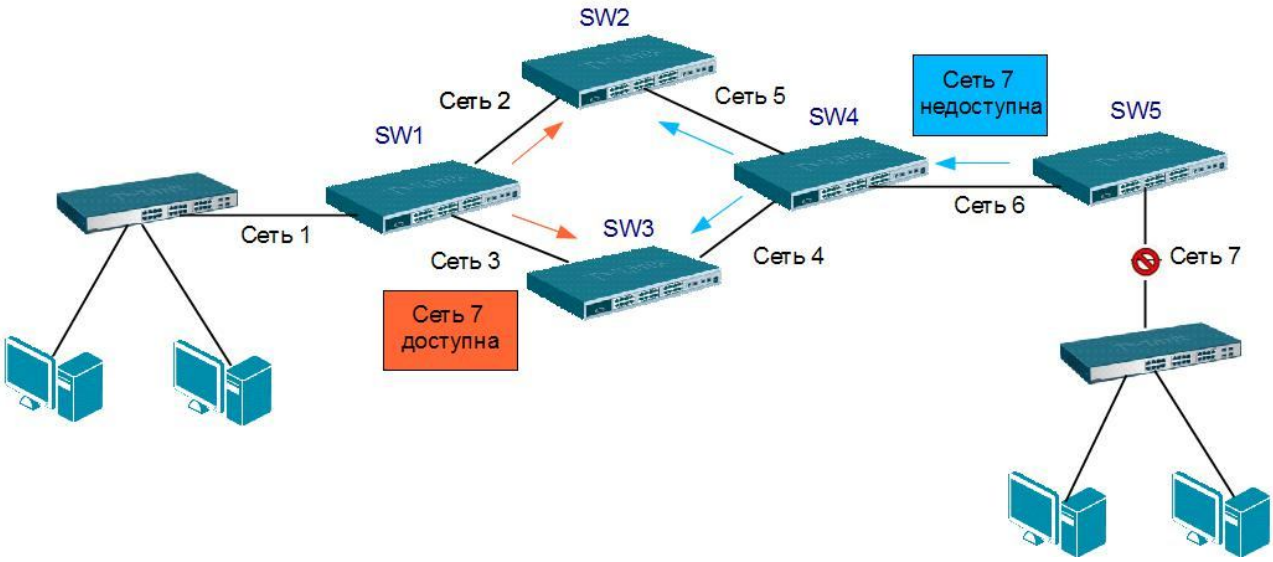


Рис. 6.47. Проблемы при работе дистанционно-векторного алгоритма маршрутизации

Предположим, что от коммутатора SW1 наилучший маршрут к сети 7 проходит через коммутатор SW2 (Cost=4). По какой-то причине в сети 7 произошел разрыв соединения. До этого события все коммутаторы 3-го уровня имели одинаковую информацию о топологии сети. После того, как коммутатор SW5 обнаружил, что сеть 7 больше не доступна, он отправляет коммутатору SW4 обновленную маршрутную информацию.

Коммутатор SW4 обновляет свою таблицу маршрутизации и перестает передавать данные в сеть 7, но коммутаторы SW2, SW3 и SW1 об этом пока не знают, так как еще не получили новую маршрутную информацию.

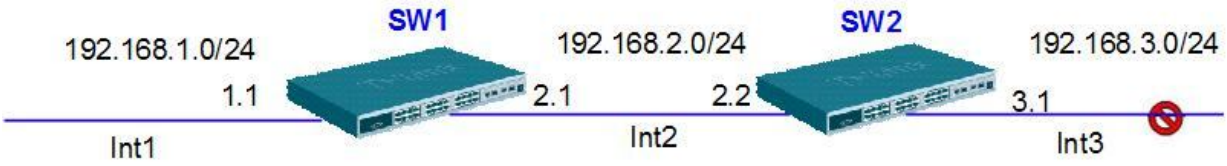
Во время очередного обновления, коммутатор SW1 отправляет свою таблицу маршрутизации SW2 и SW3, в которой маршрут к сети 7 лежит через коммутатор SW2 (Cost=4). Коммутатор SW3 обновляет свою таблицу маршрутизации, в которой маршрут к сети 7 будет лежать через коммутатор SW1 (Cost=5). При следующей рассылке маршрутной информации, коммутатор SW3 перешлет обновленную таблицу с неправильным маршрутом коммутатору SW4. Тот, посчитав, что появился альтернативный маршрут к сети 7, обновляет свою таблицу, в которой доступный маршрут будет проходить через коммутатор SW3 (Cost=6), и при очередной рассылке обновлений перешлет таблицу коммутаторам SW2 и SW5. Таким образом, теперь любой пакет, предназначенный сети 7 будет передаваться по кругу между коммутаторами SW1-SW2-SW4-SW3-SW1, т.е. появится петля маршрутизации. Поэтому основная задача любого протокола динамической маршрутизации заключается в том, чтобы как можно скорее исключить кольцевые маршруты из топологии.

Для решения этой проблемы используются следующие механизмы:

- ограничение максимального числа переходов;
- метод расщепления горизонта (Split Horizon);
- испорченный обратный маршрут (Poison reverse);
- установка таймеров удержания (Holddown timer);
- триггерные обновления (Triggered Update).

6.13.2.1 Ограничение максимального числа переходов

Для того чтобы сообщить о недоступности маршрута, дистанционно-векторные протоколы рассылают маршрутную информацию с максимальной метрикой, которая называется *метрикой бесконечности (infinity)*.



Routing Table (SW1)			
IP Address	Gateway	Interface	Cost
192.168.1.0	0.0.0.0	Int1	1
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	192.168.2.2	Int2	16

Routing Table (SW2)			
IP Address	Gateway	Interface	Cost
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	0.0.0.0	Int3	16
192.168.1.0	192.168.2.1	Int2	2

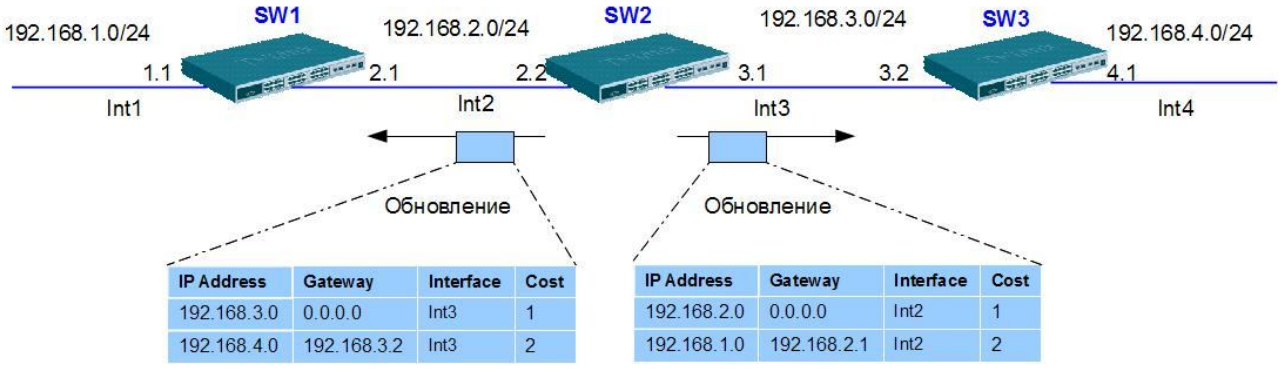
Рис. 6.48. Механизм ограничения максимального числа переходов

Внимание: в любом дистанционно-векторном протоколе определено максимальное значение метрики. Например, для протокола RIP максимальное значение метрики равно 15, если же значение метрики равно 16, то это означает, что сеть является недостижимой.

На рис. 6.48 показан следующий процесс: в сети 192.168.3.0/24, подключенной к коммутатору 3-го уровня SW2, произошел разрыв соединения. Коммутатор SW2 присваивает маршруту максимальное значение метрики (Cost=16) и во время очередного обновления отправляет таблицу коммутатору SW1, который получает обновленную информацию и хранит маршрут с недостижимой метрикой до тех пор, пока не истечет время соответствующего таймера, и маршрут не будет удален из таблицы маршрутизации.

6.13.2.2 Метод расщепления горизонта

Другим методом борьбы с петлями маршрутизации является метод *расщепления горизонта (Split Horizon)*, смысл которого заключается в том, что информация о маршруте никогда не передается тому маршрутизатору (коммутатору L3), от которого она была получена. Иными словами, когда маршрутизатор отправляет обновление в сеть, он опускает в нем любую информацию о маршрутах, полученных из этой сети.



IP Address	Gateway	Interface	Cost
192.168.3.0	0.0.0.0	Int3	1
192.168.4.0	192.168.3.2	Int3	2

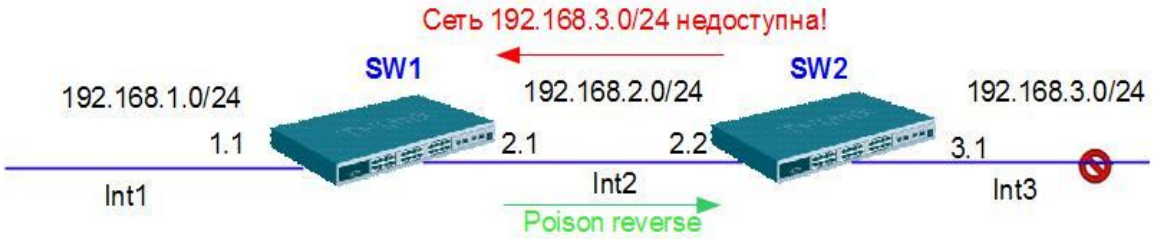
IP Address	Gateway	Interface	Cost
192.168.2.0	0.0.0.0	Int2	1
192.168.1.0	192.168.2.1	Int2	2

Рис. 6.49. Механизм расщепеления горизонта

Метод расщепления горизонта не всегда помогает решить проблему петель маршрутизации, особенно в случае, когда несколько маршрутизирующих устройств подключены не напрямую.

6.13.2.3 Метод испорченного обратного маршрута

Испорченный обратный маршрут (Posion reverse) является усовершенствованием метода расщепления горизонта. Если на маршрутизатор приходит информация о том, что маршрут до какой-то сети недоступен, то для этого маршрута, в отличие от метода расщепления горизонта, в обратном направлении пересылается маршрут с недостижимой метрикой, т. е. маршрут «портится».



Routing Table (SW1)

IP Address	Gateway	Interface	Cost
192.168.1.0	0.0.0.0	Int1	1
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	192.168.2.2	Int2	16

Routing Table (SW2)

IP Address	Gateway	Interface	Cost
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	0.0.0.0	Int3	16
192.168.1.0	192.168.2.1	Int2	2

Рис. 6.50. Метод испорченного обратного маршрута

6.13.2.4 Установка таймера удержания

Механизм установки таймера удержания (Holddown timer) позволяет маршрутизатору (коммутатору L3) запускать таймер при получении от соседа информации о недоступности сети и игнорировать в течение времени, установленного таймером, все полученные от соседних маршрутизаторов обновления о доступности маршрута с худшей метрикой.

Routing Table (SW1)

IP Address	Gateway	Interface	Cost
192.168.1.0	0.0.0.0	Int1	1
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	192.168.2.2	Int2	2
~~192.168.4.0~~	~~192.168.2.2~~	~~Int2~~	~~3~~

Routing Table (SW2)

IP Address	Gateway	Interface	Cost
192.168.2.0	0.0.0.0	Int2	1
192.168.3.0	0.0.0.0	Int3	1
192.168.1.0	192.168.2.1	Int2	2
~~192.168.4.0~~	~~192.168.3.2~~	~~Int3~~	~~2~~

Routing Table (SW3)

IP Address	Gateway	Interface	Cost
192.168.3.0	0.0.0.0	Int3	1
~~192.168.4.0~~	~~0.0.0.0~~	~~Int4~~	~~1~~
192.168.2.0	192.168.3.1	Int3	2
192.168.1.0	192.168.3.1	Int3	3

Рис. 6.51. Механизм установки таймера удержания

Рассмотрим принцип работы этого механизма. В сети 192.168.4.0/24 произошел обрыв соединения. Коммутатор 3-го уровня SW3 отправляет обновление коммутатору 3-го уровня SW2 о том, что эта сеть недоступна. SW2 получает от SW3 обновление, помечает маршрут как недоступный и запускает таймер удержания. Если в какой-то момент времени, до истечения периода времени, установленного таймером удержания, коммутатор SW2 получит от того же соседа обновление, которое сообщает, что ранее недоступная сеть теперь доступна, он помечает эту сеть как доступную и сбрасывает таймер удержания. Если до истечения таймера SW2 получит от SW1 обновление о доступности сети с лучшей метрикой по сравнению с той, которая записана в его таблице, он помечает маршрут как доступный и

сбрасывает таймер. Если SW2 получает обновление с худшей метрикой, то эта маршрутная информация игнорируется.

6.13.2.5 Триггерные обновления

Решение проблемы возникновения петель маршрутизации возможно при высокой скорости рассылки обновленной информации о разрыве соединения, что существенно снижает вероятность зацикливания пакетов. *Метод триггерных обновлений (triggered update)* позволяет маршрутизирующему устройству мгновенно, не дожидаясь очередного цикла обновления маршрутной информации, отправлять соседним маршрутизирующим устройствам информацию об аварийном маршруте. Этот прием может во многих случаях предотвратить передачу устаревших сведений об отказавшем маршруте, но перегружает сеть служебными сообщениями, поэтому триггерные объявления отправляются с некоторой задержкой. Триггерные обновления не позволяют избежать петель маршрутизации без использования дополнительных методов, например, метода испорченного обратного маршрута (Posion reverse).

6.14 Протокол RIP

Протокол RIP (Routing Information Protocol) является представителем класса внутренних протоколов стека TCP/IP. Этот протокол используется в небольших однородных сетях, т. е. имеющих одинаковые характеристики каналов связи, где самый длинный путь между любыми сетями составляет максимум 15 переходов.

Протокол RIP основан на дистанционно-векторном алгоритме маршрутизации и в качестве метрики при выборе маршрута использует *количество переходов (hops count)*. Он не учитывает ситуации, когда маршрут должен быть выбран на основе таких параметров, как загруженность канала, надежность или задержка передачи. Если маршрутизатор непосредственно подключен к сети, то расстояние до нее (количество переходов) равно 1. По умолчанию маршрутизаторы, использующие протокол RIP, отправляют на широковещательный адрес своим соседям обновления с маршрутной информацией каждые 30 секунд. При получении обновления от соседа, маршрутизатор заносит новые записи в таблицу маршрутизации и увеличивает значение метрики (число переходов) к соответствующей сети на 1.

Использование протокола RIP в ряде случаев может привести к петлям маршрутизации, поэтому для решения этой проблемы в нем предусмотрено использование механизмов расщепления горизонта, испорченного обратного маршрута и таймеров удержания.

В настоящее время существует три версии протокола:

- RIP версии 1 (RIPv1) используется для поддержки классовой адресации протокола IPv4;
- RIP версии 2 (RIPv2) используется для поддержки бесклассовой адресации IPv4;
- RIPng (next generation) используется для протокола IPv6.

6.14.1 Протокол RIPv1

Протокол RIPv1 позволяет использовать только *классовую (classfull)* маршрутизацию, поскольку не включает в маршрутные обновления информацию о маске подсети.

В RIPv1 определены два типа сообщений:

- *Request (запрос)* – сообщение, отправляемое маршрутизатору (коммутатору L3) с просьбой прислать часть или всю таблицу маршрутизации;
- *Response (ответ)* – сообщение, содержащее часть или всю таблицу маршрутизации.

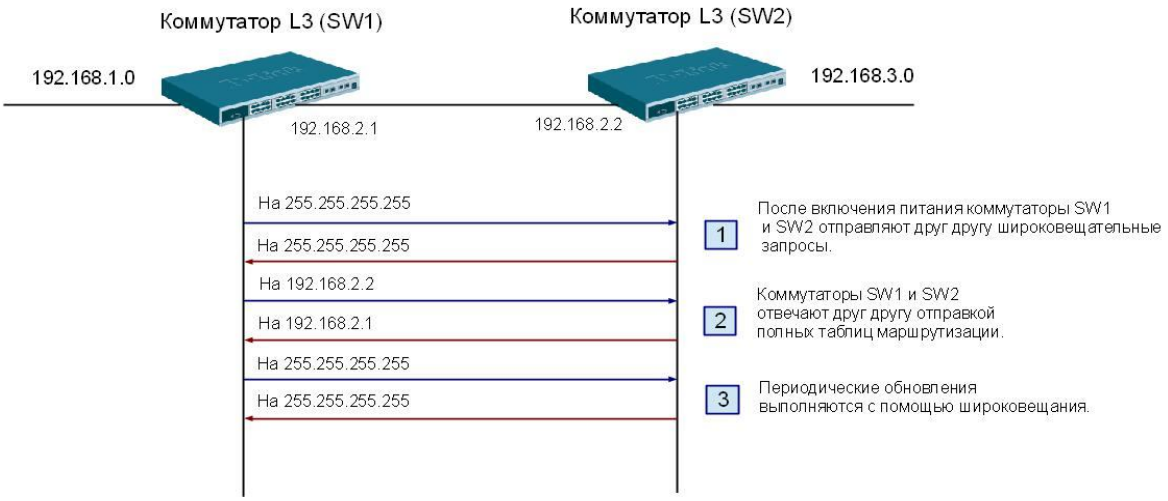


Рис. 6.52. Обмен RIP-сообщениями между коммутаторами 3-го уровня SW1 и SW2

При передаче сообщений используется протокол UDP (порт 520). Маршрутизатор (коммутатор 3-го уровня) отправляет RIP-запрос, если требуется маршрутная информация, например, при включении питания. После инициализации протокола RIP маршрутизатор обычно отправляет запросы в непосредственно подключенные к нему сети, чтобы запросить информацию у своих соседей.

При получении RIP-запроса маршрутизатор обрабатывает его и отправляет RIP-ответ, содержащий таблицу маршрутизации. При нормальной работе маршрутизаторы (коммутаторы L3) не рассылают RIP-запросы. Вместо этого они используют специальный таймер – *таймер обновлений (Update time)*, по истечении времени которого маршрутизирующие устройства широковещательно отправляют соседям RIP-ответ, т.е. обновление, содержащее таблицу маршрутизации. По умолчанию значение таймера обновления равно 30 секунд. Этот процесс гарантирует, что маршрутная информация будет рассылаться регулярно. Формат сообщения RIPv1 показан на рис. 6.53.

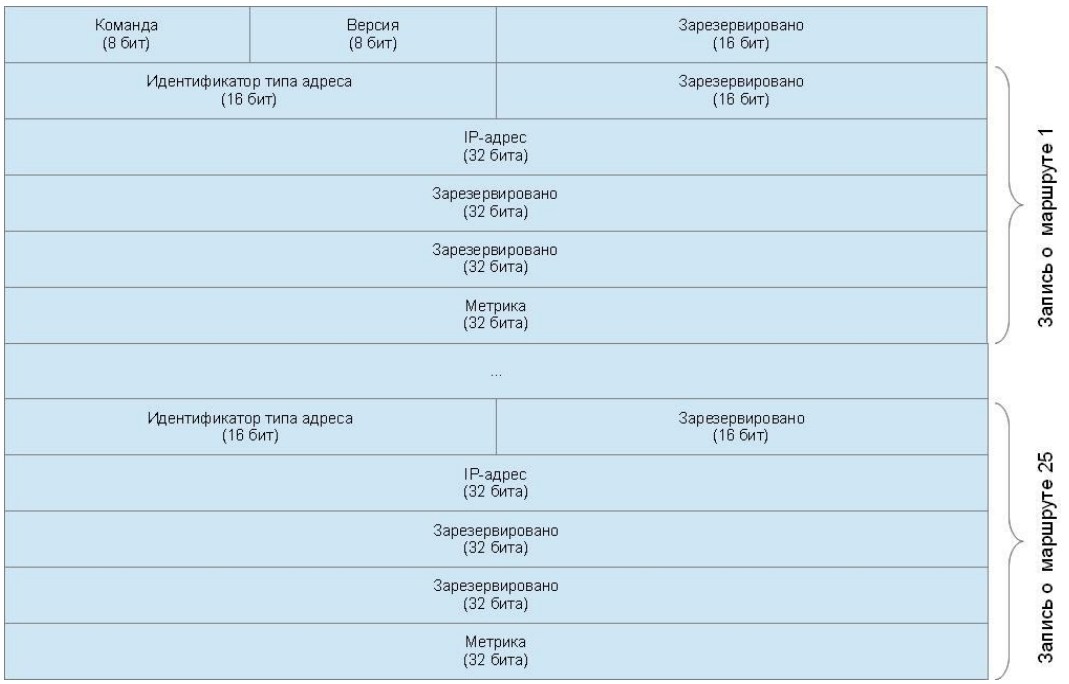


Рис. 6.53. Формат сообщения протокола RIPv1

Сообщение RIPv1 состоит из следующих полей:

- *Команда (Command):* значение равно 1 – запрос на получение частичной или полной таблицы маршрутизации; значение равно 2 – ответ, содержащий полную или частичную информацию из таблицы маршрутизации отправителя;
- *Версия (Version)* – равна 1 для RIPv1;

Далее идут записи о маршрутах, максимальное количество которых равно 25. Эти записи состоят из следующих полей:

- *Идентификатор типа адреса (Address Family Identifier)* – тип протокола, используемого в соответствующей сети. Для протокола IP значение равно 2;
- *IP-адрес (IP Address)* – IP-адрес сети назначения;
- *Метрика (Metric)* – расстояние до сети (число переходов), указанной в поле IP-адрес.

Для каждой записи в таблице маршрутизации существует *время старения (Timeout time)*, контролируемое таймером (по умолчанию 180 секунд). Таймер старения обнуляется каждый раз, когда маршрутизатор получает обновление с информацией о соответствующем маршруте. Если информация о каком-либо маршруте отсутствует в периодических обновлениях, то время, установленное таймером, истекает и маршрут помечается как недостижимый (значение метрики устанавливается равным 16).

Когда маршрут помечается как недостижимый, запускается таймер *«сборщик мусора» (Garbage-Collection time)*. Этот таймер отсчитывает время, по истечении которого недостижимый маршрут полностью удаляется из таблицы маршрутизации (по умолчанию 120 секунд). Значение таймеров по умолчанию, используемых протоколом RIP, приведено на рис. 6.54.

```
DES-3810-28:admin#show rip
Command: show rip

RIP Global State             : Enabled
Update Time                  : 30 seconds
Timeout Time                 : 180 seconds
Garbage Collection Time      : 120 seconds
```

Рис. 6.54. Значение таймеров, используемых протоколом RIP

Как говорилось ранее, протокол RIPv1 позволяет использовать только классовую маршрутизацию и не включает в маршрутные обновления информацию о маске подсети. Если к интерфейсу 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.

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

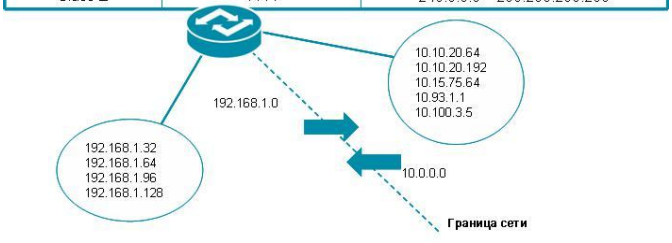


Рис. 6.55. Суммирование маршрутов на граничном маршрутизирующем устройстве

6.14.2 Протокол RIPv2

Протокол RIPv2 является расширением RIPv1 и определен в RFC 2453. Принцип работы RIPv2 в основном тот же, что и RIPv1. Главные отличия RIPv2 от RIPv1 состоят в следующем:

- *Поддержка бесклассовой адресации.* Протокол RIPv2 включает в маршрутные обновления информацию о маске подсети для каждого сетевого адреса, тем самым позволяя поддерживать маски переменной длины (VLSM) и бесклассовую маршрутизацию (CIDR).
- *Указание адреса следующего маршрутизатора.* Каждая запись включает IP-адрес маршрутизатора (коммутатора 3-го уровня), который может быть использован в качестве следующего транзитного маршрутизатора, предназначенного для передачи пакета в сеть назначения. Это позволяет повысить эффективность маршрутизации, избежав лишних пересылок.
- *Аутентификация.* Протокол RIPv2 предоставляет базовый механизм аутентификации, который позволяет маршрутизаторам (коммутаторам L3) идентифицировать другие маршрутизаторы, прежде чем принимать RIP-сообщения от них.
- *Метки маршрута (Route Tag)*. Каждая запись RIPv2 содержит поле *Route Tag (метка маршрута)*, в котором хранится дополнительная информация о маршруте. Это поле используется для идентификации автономной системы при работе с протоколами внешней маршрутизации.
- *Использование многоадресной рассылки.* Для уменьшения нагрузки на сеть, протокол RIPv2 позволяет рассылать маршрутные обновления не широковещательным методом, а на специальный групповой адрес 224.0.0.9.

Формат сообщения протокола RIPv2 в целом аналогичен формату RIPv1 (рис.6.56). Маршрутизаторы, использующие протокол RIPv1, могут принимать маршрутные обновления RIPv2.

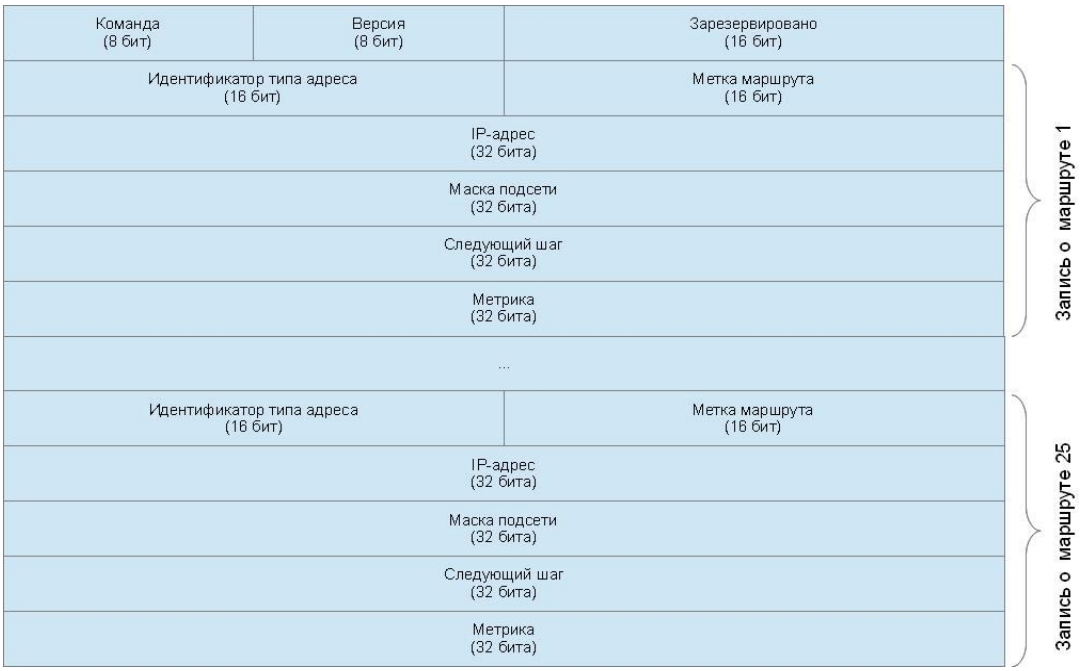


Рис. 6.56. Формат сообщения протокола RIPv2

В сообщение протокола RIPv2 добавлены следующие новые поля:

- *метка маршрута (Route tag)* – используется для идентификации автономной системы при работе с протоколами внешней маршрутизации;
- *маска подсети (Subnet Mask)*;

- *следующий шаг (Next Hop)* – IP-адрес следующего транзитного маршрутизатора (коммутатора 3-го уровня) на пути к сети назначения.

6.14.2.1 Пример настройки протокола RIPv2

Рассмотрим пример настройки протокола RIPv2 для сети, показанной на рис 6.57. Сеть построена на коммутаторах 3-го уровня DES-3810-28 и логически сегментирована с помощью VLAN, между которыми необходимо настроить маршрутизацию.

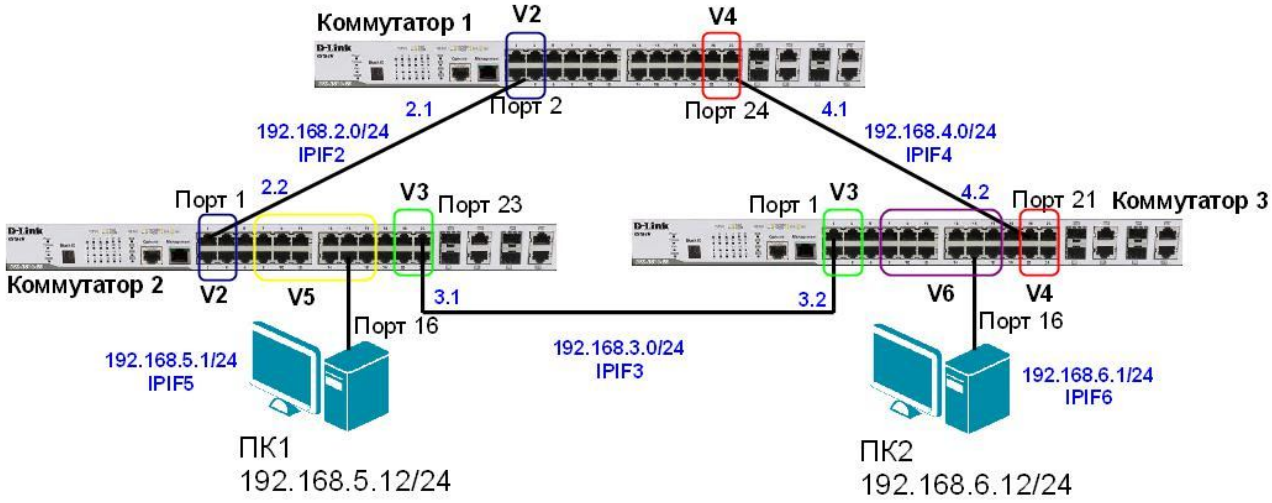


Рис. 6.57. Схема сети

Настройка коммутатора 1

- Создать VLAN.

config vlan default delete 1-24
create vlan v2 tag 2
config vlan v2 add tagged 1-4
create vlan v4 tag 4
config vlan v4 add tagged 21-24

- Создать IP-интерфейсы VLAN.

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

- Настроить протокол RIPv2 на всех интерфейсах коммутатора.

enable rip
config rip all tx_mode v2_only rx_mode v2_only state enable

Настройка коммутатора 2

- Создать VLAN.

config vlan default delete 1-24
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

- Создать IP-интерфейсы VLAN.

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

- Настроить протокол RIPv2 на всех интерфейсах коммутатора.

enable rip
config rip all tx_mode v2_only rx_mode v2_only state enable

Настройка коммутатора 3

- Создать VLAN.

config vlan default delete 1-24
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

- Создать IP-интерфейсы VLAN.

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

- Настроить протокол RIPv2 на всех интерфейсах коммутатора.

enable rip
config rip all tx_mode v2_only rx_mode v2_only state enable

Внимание: при настройке маршрутизации на коммутаторах 3-го уровня D-Link имена IP-интерфейсов одноименных VLAN разных коммутаторов, соединенных линией связи «точка-точка», должны быть одинаковыми.

6.14.3 Протокол RIPng

Протокол *RIP next generation (RIPng)* представляет собой новую версию протокола RIP, разработанную для поддержки IPv6, и определен в RFC 2080. Он поддерживает улучшения базовой версии протокола RIP, реализованные в RIPv2, и отличается тем, что использует адреса в формате IPv6. Основные особенности RIPng:

- *Поддержка бесклассовой адресации.* Вместо поля *Маска подсети* используется поле *Префикс*.
- *Указание адреса следующего маршрутизатора.* Из-за большого размера IPv6-адреса это поле является дополнительным; если требуется указать адрес транзитного маршрутизатора, он указывается в отдельной записи.
- *Аутентификация.* Протокол RIPng не имеет собственного механизма аутентификации и опирается на стандартную функцию IPSec, обязательно поддерживаемую в IPv6.
- *Использование многоадресной рассылки.* Сообщения отправляются на специальный групповой адрес – FF02::9.

Формат сообщения RIPng аналогичен формату RIPv1/v2 за исключением длины записей (рис. 6.58).

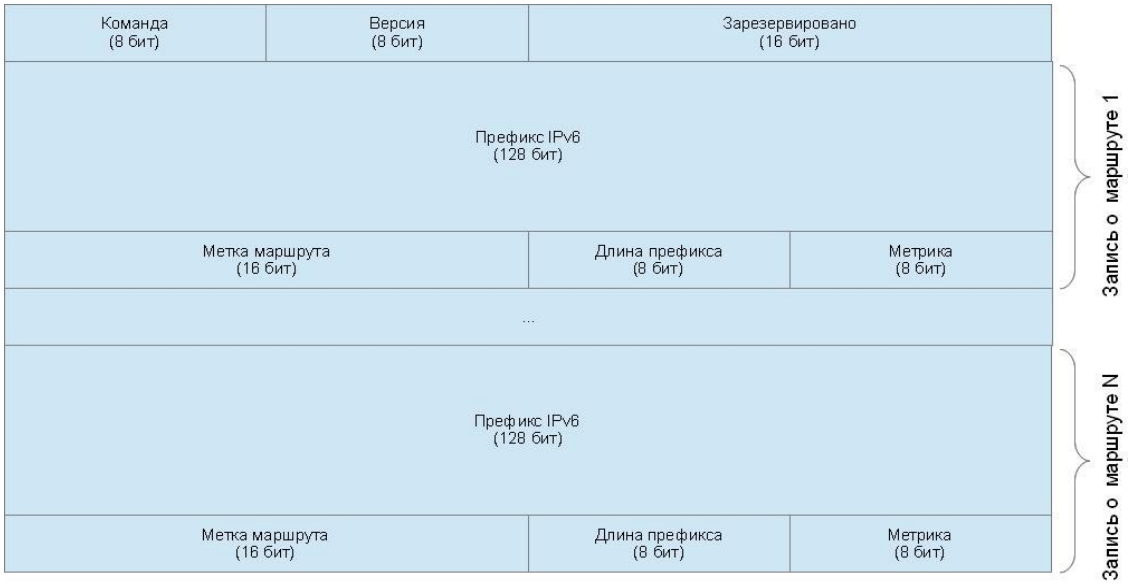


Рис. 6.58. Формат сообщения протокола RIPng

Максимальное количество записей сообщения RIPng не ограничено 25, как в RIPv1/v2, а ограничено только *MTU (Maximum Transmission Unit)* сети, через которую будет передаваться сообщение.

7. Качество обслуживания (QoS)

7.1 Модели QoS

Для поддержки передачи по одной сети трафика потоковых мультимедийных приложений (Voice over IP (VoIP), IPTV, видеоконференции, он-лайн игры и др.) и трафика данных с различными требованиями к пропускной способности, необходимы механизмы, обеспечивающие возможность дифференцирования и обработки различных типов сетевого трафика в зависимости от предъявляемых ими требований. Негарантированная доставка данных (*best effort service*), традиционно используемая в сетях, построенных на основе коммутаторов, не предполагала проведения какой-либо классификации трафика и не обеспечивала надежную доставку трафика приложений, гарантированную пропускную способность канала и определенный уровень потери пакетов. Для решения этой проблемы было введено такое понятие, как **качество обслуживания** (*Quality of Service, QoS*).

Функции качества обслуживания в современных сетях заключаются в обеспечении 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.

Можно выделить три модели реализации QoS в сети:

- **Негарантированная доставка данных (Best Effort Service)** – обеспечивает связь между узлами, но не гарантирует надежную доставку данных, время доставки, пропускную способность и определенный приоритет.
- **Интегрированные услуги (Integrated Services, IntServ)** – эта модель описана в RFC 1633 и предполагает предварительное резервирование сетевых ресурсов с целью обеспечения предсказуемого поведения сети для приложений, требующих для нормального функционирования гарантированной выделенной полосы пропускания на всем пути следования трафика. В качестве примера можно привести приложения IP-телефонии, которым для обеспечения приемлемого качества передачи голоса, требуется канал с минимальной пропускной способностью 64 Кбит/с (для кодека G.711).

  Модель IntServ использует сигнальный протокол RSVP (Resource Reservation Protocol, протокол резервирования ресурсов) для резервирования ресурсов для каждого потока данных, который должен поддерживаться каждым узлом на пути следования трафика. Эту модель также часто называют *жестким QoS* (*hard QoS*) в связи с предъявлением строгих требований к ресурсам сети.
- **Дифференцированное обслуживание (Differentiated Service, DiffServ)** – эта модель описана в RFC 2474, RFC 2475 и предполагает разделение трафика на классы на основе требований к качеству обслуживания. В архитектуре DiffServ каждый передаваемый пакет снабжается информацией, на основании которой принимается решение о его продвижении на каждом промежуточном узле сети, в соответствии с политикой обслуживания трафика данного класса (Per-Hop Behavior, PHB).

  Модель дифференцированного обслуживания занимает промежуточное положение между негарантированной доставкой данных и моделью IntServ и сама по себе не предполагает обеспечение гарантий предоставляемых услуг, поэтому дифференцированное обслуживание часто называют *мягким QoS* (*soft QoS*).

7.2 Приоритизация пакетов

Для обеспечения QoS на канальном уровне модели OSI коммутаторы поддерживают стандарт IEEE 802.1p. Стандарт IEEE 802.1p позволяет задать до 8 уровней приоритетов (от 0 до 7, где 7 – наивысший), определяющих способ обработки кадра, используя 3 бита поля приоритета тега IEEE 802.1Q.

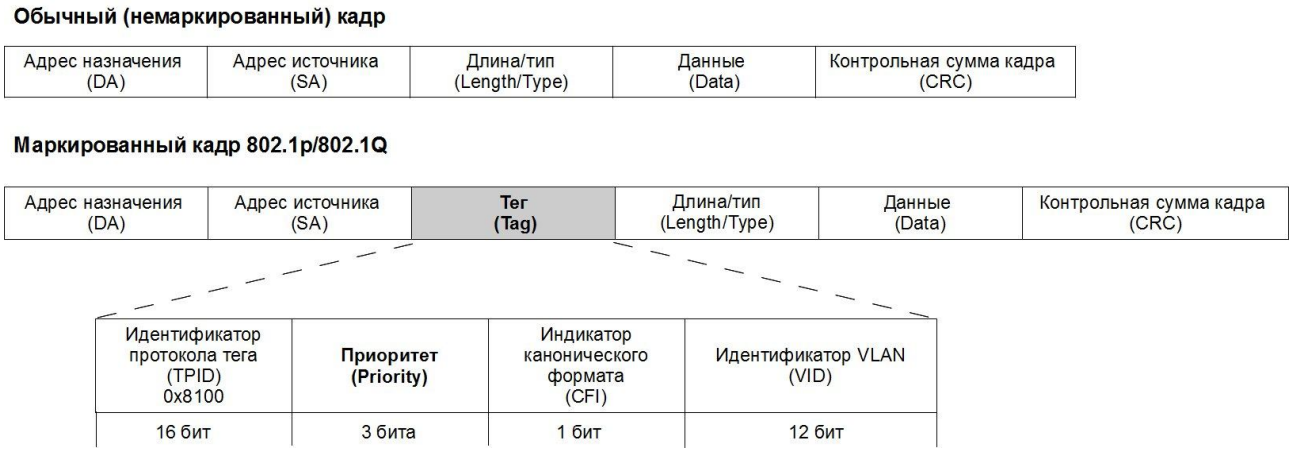


Рис. 7.1. Формат кадра 802.1Q с битами приоритета 802.1p

Для обеспечения QoS на сетевом уровне модели OSI в заголовке протокола IPv4 предусмотрено 8-битное поле ToS (Type of Service). Этот байт может быть заполнен либо значением приоритета IP Precedence, либо значением DSCP (Differentiated Services Code Point) в зависимости от решаемой задачи.

Поле IP Precedence имеет размерность 3 бита и может принимать значения от 0 до 7. Оно используется для указания относительного приоритета обработки пакета на сетевом уровне.

Поле DSCP было стандартизировано IETF с появлением модели DiffServ. Оно занимает 6 старших бит байта ToS и позволяют задать до 64 уровней приоритетов (от 0 до 63). По сути код DSCP является расширением 3-битового поля IP Precedence и обладает обратной совместимостью с IP-приоритетом.

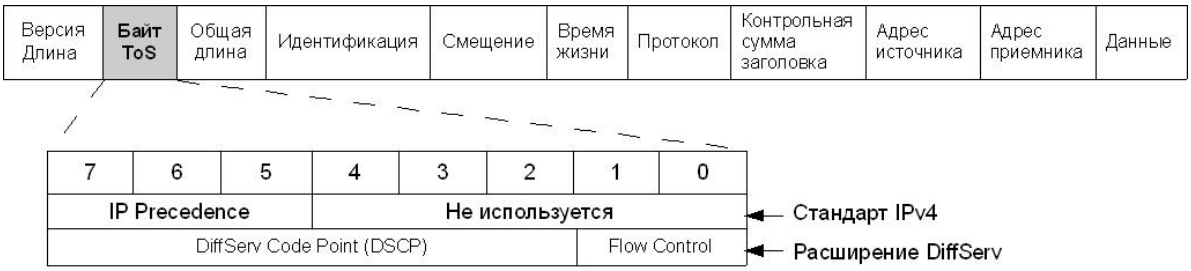


Рис. 7.2. Байт ToS заголовка IPv4

7.3 Классификация пакетов

Для обеспечения дифференцированного обслуживания трафика, коммутаторы поддерживают в зависимости от модели от 4 до 8 аппаратных очередей приоритетов на каждом из своих портов. Для обеспечения требуемой очередности передачи пакетов данных в коммутаторе необходимо настроить алгоритм обслуживания очередей и карту привязки приоритетов 802.1p, ToS, DSCP к очередям.

По умолчанию в коммутаторах D-Link используются следующие карты привязки пользовательских приоритетов 802.1p к аппаратным очередям:

- **4 очереди приоритетов**

Приоритет	*Номер очереди*
0	Q1
1	Q0
2	Q0
3	Q1
4	Q2
5	Q2
6	Q3
7	Q3

- **8 очередей приоритетов**

Приоритет	*Номер очереди*
0	Q2
1	Q0
2	Q1
3	Q3
4	Q4
5	Q5
6	Q6
7	Q6

Внимание: класс 7 в коммутаторах D-Link с поддержкой 8 очередей приоритетов зарезервирован для внутреннего использования и поэтому не настраивается.

В коммутаторах с поддержкой 4-х очередей приоритетов очереди нумеруются от 0 до 3, где очередь 3 обладает наивысшим приоритетом, очередь 0 – низшим. В коммутаторах с поддержкой 8-ми очередей приоритетов очереди нумеруются от 0 (низший приоритет) до 7 (наивысший приоритет).

Программное обеспечение коммутаторов позволяет настраивать карты привязки приоритетов 802.1p, ToS, DSCP к очередям в соответствии с требованиями пользователей.

Для того чтобы поместить пакет данных в одну из очередей приоритетов в соответствии с заданной политикой QoS, коммутатор анализирует содержимое одного или нескольких полей его заголовка – приоритет 802.1p, IP-приоритет или поле DSCP в байте ToS. Этот процесс называется *классификацией пакетов (packet classification)*.

Следует отметить, что при этом коммутатор не изменяет значения приоритетов внутри пакетов данных, а только определяет очередность и способ их обработки выходным портом, основываясь на реализованной в нем политике QoS.

В том случае, если на входной порт коммутатора поступает немаркированный кадр (заголовок кадра не содержит битов приоритета), то его классификация осуществляется на основе значения приоритета 802.1p по умолчанию, назначенного данному порту.

Также для классификации пакетов данных на основании различных параметров их заголовков, например MAC-адреса, IP-адреса, номера порта TCP/UDP, тега VLAN и т.д. могут использоваться списки управления доступом (Access Control List, ACL).

7.4 Маркировка пакетов

После процесса классификации коммутатор может осуществить *маркировку пакетов (packet marking)*. Маркировка пакетов определяет способ записи/перезаписи значений битов приоритета (DSCP, 802.1p или IP Precedence) входящих пакетов данных. Обычно процесс маркировки выполняется на граничных устройствах и позволяет последующим коммутаторам/маршрутизаторам использовать новое значение приоритета пакета для отнесения его к одному из поддерживаемых в сети классов обслуживания. Изменить

значения битов приоритета в заголовках входящих пакетов данных можно с помощью списков управления доступом.

Рис. 7.3. Маркировка пакетов

7.5 Управление перегрузками и механизмы обслуживания очередей

Наиболее часто перегрузка сети возникает в местах соединения коммутаторами сетей с разной полосой пропускания. В случае возникновения перегрузки сети пакеты начинают буферизироваться и распределяться по очередям. Порядок передачи через выходной интерфейс поставленных в очередь пакетов данных на основе их приоритетов определяется *механизмом обслуживания очередей* (*Queuing mechanism*), который позволяет управлять пропускной способностью сети при возникновении перегрузок.

Рис. 7.4. Возникновение перегрузки в сети

Механизм управления перегрузками (*Congestion management*) включает следующие механизмы обслуживания очередей:

- механизм FIFO (First-In, First-Out);
- очереди приоритетов (Priority Queuing);
- взвешенный алгоритм кругового обслуживания (Weighted Round Robin, WRR);
- настраиваемые очереди (Custom Queuing).

В коммутаторах D-Link для обслуживания очередей используются взвешенный алгоритм кругового обслуживания, очереди приоритетов и комбинации этих методов.

Механизм обслуживания очередей FIFO («первым пришел, первым ушел») передает пакеты, поставленные в очередь в том порядке, в котором они поступили в нее. Этот механизм не обеспечивает классификации пакетов и рассматривает их как принадлежащие одному классу.

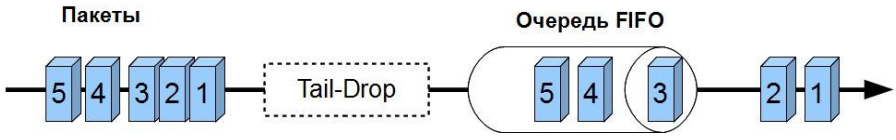


Рис. 7.5. Очередь FIFO

Очереди приоритетов со строгим режимом (*Strict Priority Queue*) предполагают передачу трафика строго в соответствии с приоритетом выходных очередей. В этом механизме предусмотрено наличие 4-х очередей – с высоким, средним, обычным и низким приоритетами обслуживания. Пакеты, находящиеся в очереди с высоким приоритетом, обрабатываются первыми. Пакеты из следующей по приоритету обслуживания очереди начнут передаваться только после того, как опустеет высокоприоритетная очередь. Например, пакеты из средней по приоритету очереди не будут передаваться до тех пор, пока не будут обслужены пакеты из высокоприоритетной очереди. Пакеты из очереди с нормальным приоритетом не начнут передаваться до тех пор, пока не опустеет очередь со средним приоритетом и т.д.

Следует отметить, что пакеты очереди с высоким приоритетом всегда получают предпочтение при обслуживании независимо от количества пакетов в других очередях и времени, прошедшего с момента передачи последнего пакета из очереди с низким приоритетом. В некоторых случаях это может привести к «зависанию» обслуживания низкоприоритетного трафика, т.е. пакеты из очередей с низким приоритетом будут долго не обрабатываться.

По умолчанию на коммутаторах D-Link настроены очереди приоритетов со строгим режимом.

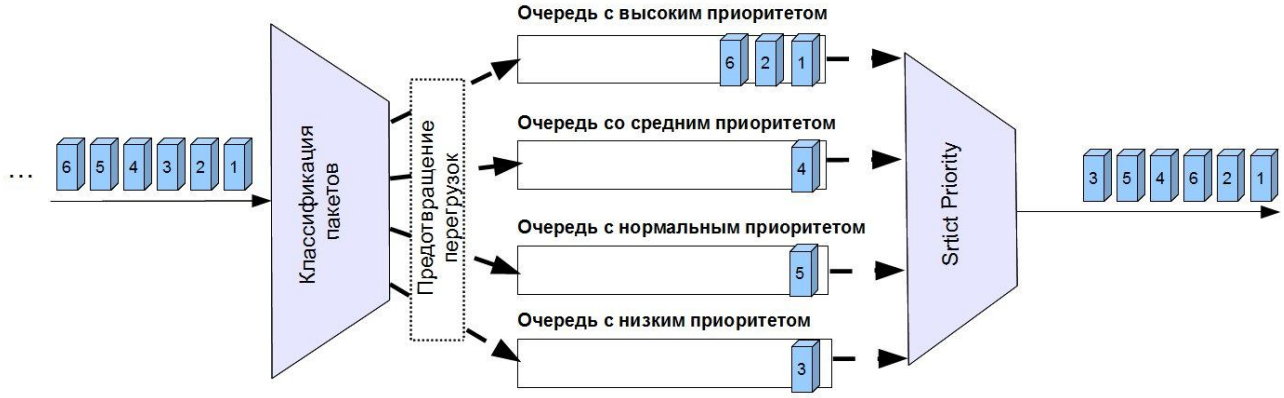


Рис. 7.6. Очереди приоритетов со строгим режимом

Еще одним механизмом обслуживания очередей является *взвешенный алгоритм кругового обслуживания* (Weighted Round Robin, WRR). Этот механизм исключает главный недостаток очередей приоритетов, обеспечивая обработку очередей в соответствии с назначенным им весом и предоставляя полосу пропускания для пакетов из низкоприоритетных очередей.

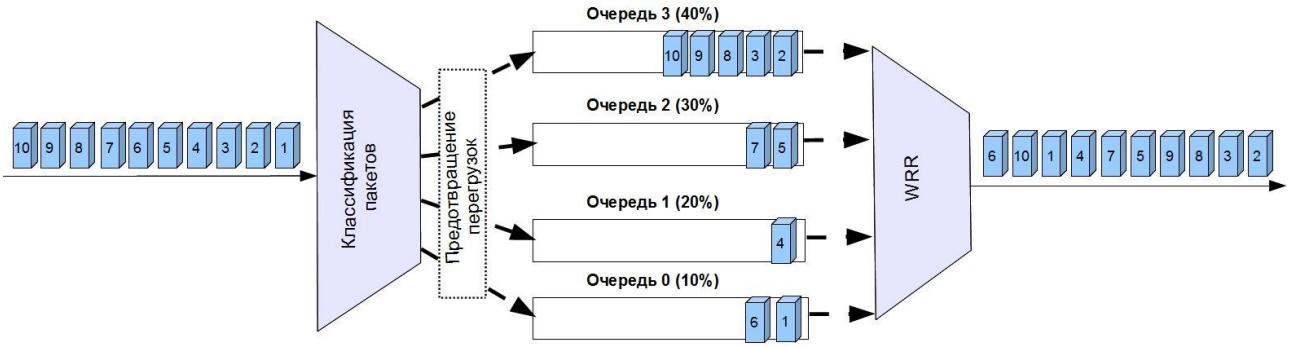


Рис. 7.7. Обслуживание очередей с использованием алгоритма WRR

Процесс обработки очередей осуществляется по круговому принципу, начиная с самой приоритетной очереди. Из каждой непустой очереди передается некоторый объем трафика, пропорциональный назначенному ей весу, после чего выполняется переход к следующей по убыванию приоритета очереди и т.д. по кругу.

7.6 Механизм предотвращения перегрузок

Механизм предотвращения перегрузок (*Congestion avoidance*) – это процесс выборочного отбрасывания пакетов с целью избежания перегрузок в сети в случае достижения выходными очередями своей максимальной длины (в пакетах).

Традиционной политикой обработки пакетов коммутаторами в случае переполнения всех выходных очередей является их отбрасывание, которое продолжается до тех пор, пока длина очередей не уменьшится за счет передачи находящихся в них пакетов. Такой алгоритм управления длиной выходных очередей получил название «*отбрасывание хвоста*» (*Tail-Drop*). Отбрасывание пакета будет служить сигналом о перегрузке сети источнику TCP-соединения, т.к. он не получит подтверждения о доставке пакета от приемника TCP-соединения. В этом случае он уменьшит скорость передачи путем уменьшения размера окна перегрузки до одного сегмента и перезапустит алгоритм *медленного старта* (*slow start*).

Поскольку коммутатор обрабатывает множество TCP-потоков в один момент времени, отбрасывание пакетов послужит сигналом о перегрузке тысячам источникам TCP-соединений, которые снизят скорость передачи. При этом почти все источники TCP-соединений будут использовать одинаковое время таймеров задержки перед началом увеличения скорости передачи. Значения этих таймеров достигнут своего лимита практически в одно и тоже время, что вызовет увеличение интенсивности трафика и переполнение очередей, которое приведет к отбрасыванию пакетов, и весь процесс повторится вновь.

Процесс, когда каждый источник TCP-соединения уменьшает и увеличивает скорость передачи одновременно с другими источниками TCP-соединений, получил название *эффекта глобальной синхронизации* (*global synchronization*). Эффект глобальной синхронизации приводит к неэффективному использованию полосы пропускания, а также к возрастанию задержки передачи пакетов.

Для решения проблемы поведения источников TCP-соединения в момент отбрасывания пакетов был разработан *алгоритм произвольного раннего обнаружения* (*Random Early Detection, RED*).

В отличие от алгоритма «отбрасывания хвоста», алгоритм RED отбрасывает поступающие пакеты вероятностно, на основе оценки среднего размера очередей. Средний размер очереди сравнивается с двумя пороговыми значениями – минимальным и максимальным. Если средний размер очереди превысит определенное минимальное пороговое значение, то пакеты начинают отбрасываться с некоторой вероятностью. Это позволяет избежать эффекта глобальной синхронизации, т.к. будут отбрасываться не все пакеты, а только пакеты произвольным образом выбранных потоков. Интенсивность отбрасывания пакетов возрастает прямо пропорционально возрастанию среднего размера очереди. Когда средний размер очереди превысит максимальное пороговое значение, алгоритм RED будет отбрасывать все пакеты, предназначенные для постановки в очередь.

В коммутаторах D-Link поддерживается *простой алгоритм произвольного раннего обнаружения* (*Simple Random Early Detection, SRED*), который является расширенной версией алгоритма RED, реализованной на основе ASIC, и выполняет вероятностное отбрасывание входящих «окрашенных» пакетов. «Окрашивание» пакетов позволяет реализовать разные политики обслуживания пакетов (различную вероятность отбрасывания) на основе их приоритетов. Так пакеты, «окрашенные» в зеленый цвет обладают наивысшим приоритетом. Пакеты «окрашенные» в желтый цвет – средним, в красный цвет – низшим приоритетом.

Алгоритм SRED позволяет задавать два пороговых значения размера для каждой очереди – минимальное и максимальное. Если длина очереди меньше минимального порогового значения, то пакеты будут помещаться в очередь. Если размер очереди будет находиться в интервале между минимальным и максимальным пороговыми значениями, т.е. будет наблюдаться умеренная перегрузка, то пакеты «окрашенные» в красные и желтые цвета будут отбрасываться с заданной вероятностью. Если длина очереди превысит максимальное пороговое значение, то пакеты любых цветов будут отбрасываться с заданной вероятностью. Т.е. алгоритм SRED обеспечивает возможность настройки более интенсивного отбрасывания пакетов низкоприоритетного трафика и менее интенсивного отбрасывания пакетов высокоприоритетного трафика.

В коммутаторах D-Link при настройке SRED существует возможность выбора из восьми значений скоростей (вероятностей) отбрасывания пакетов:

	Скорость отбрасывания
1	100%
2	6.25%
3	3.125%
4	1.5625%
5	0.78125%
6	0.390625%
7	0.1953125%
8	0.09765625%

7.7 Контроль полосы пропускания

Современные коммутаторы позволяют регулировать интенсивность трафика на своих портах с целью обеспечения функций качества обслуживания. Для этого они используют механизмы, называемые *Traffic Policing* (ограничение трафика) и *Traffic Shaping* (выравнивание трафика).

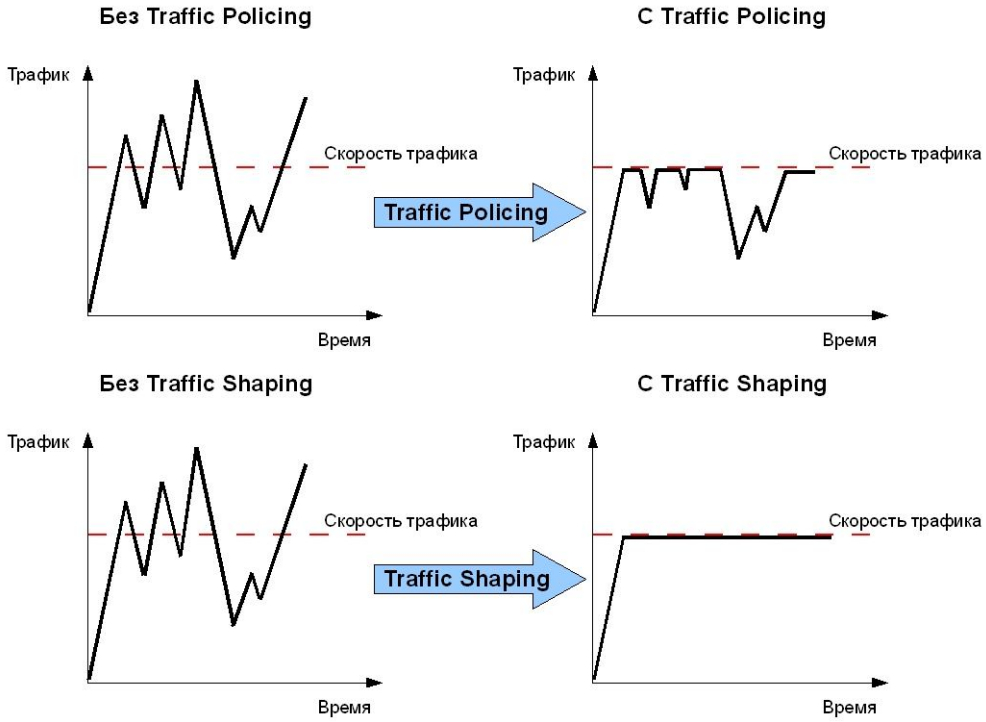


Рис. 7.8. Механизмы Traffic Policing и Traffic Shaping

Механизм Traffic Policing служит для ограничения скорости трафика, получаемого или отправляемого с интерфейса коммутатора. Когда эта функция активна, администратор может устанавливать различные пороговые значения скорости передачи на каждом из выходных портов коммутатора. Трафик, скорость которого меньше или равна пороговому значению, будет передаваться; трафик, скорость которого превышает пороговое значение, будет обрабатываться в соответствии с настроенной политикой, например, отбрасываться или маркироваться новым значением приоритета.

Основным средством, используемым для ограничения трафика, является хорошо известный алгоритм «*корзина маркеров*» (*token bucket*). Этот алгоритм предполагает наличие следующих параметров:

- **Согласованная скорость передачи (Committed Information Rate, CIR)** – средняя скорость передачи трафика через интерфейс коммутатора/маршрутизатора. Этот параметр также определяет скорость помещения маркеров в корзину.
- **Согласованный размер всплеска (Committed Burst Size, CBS)** – это объем трафика (в битах), на который может быть превышен размер корзины маркеров в отдельно взятый момент всплеска.
- **Расширенный размер всплеска (Extended Burst Size, EBS) –** это объем трафика (в битах), на который может быть превышен размер корзины маркеров в экстренном случае.

На рис. 7.9 показана схема реализации алгоритма «корзина маркеров» в рамках механизма Traffic Policing.

Размер стандартной корзины маркеров (максимальное число маркеров, которое она может вместить) равен согласованному размеру всплеска (CBS). Маркеры генерируются и помещаются в корзину с определенной скоростью (CIR). Если корзина полна, то поступающие избыточные маркеры отбрасываются. Для того чтобы передать пакет, из корзины вынимается число маркеров, равное размеру пакета в битах. Если маркеров в корзине достаточно, то пакет передается. Если размер пакета оказался больше, чем маркеров в корзине, то маркеры из корзины не извлекаются, а пакет рассматривается как «неудовлетворяющий» (non-conform) заданному профилю или избыточный. Для избыточных пакетов могут применяться различные способы обработки: они могут отбрасываться или перемаркировываться.

Рис. 7.9. Алгоритм «корзина маркеров» в рамках механизма Traffic Policing

Стандартная корзина маркеров не поддерживает экстренное увеличение размера всплеска, поэтому в такой реализации расширенный размер всплеска (EBS) равен согласованному размеру всплеска (CBS).

В корзине маркеров с возможностью экстренного увеличения размера всплеска расширенный размер всплеска (EBS) больше согласованного размера всплеска (CBS). Объем трафика (в битах), на который может быть превышен размер корзины, рассчитывается по формуле:

$$CBS = 1{,}5 \times CIR/8$$
$$EBS = 2 \times CBS$$

При такой реализации корзины маркеров, в случае нехватки маркеров, необходимых для передачи пакета, учитывается расширенный размер всплеска.

Механизм Traffic Shaping служит для сглаживания исходящего с интерфейсов коммутатора трафика. В отличие от механизма Traffic Policing, который в случае превышения скорости трафика заданного порогового значения может отбрасывать пакеты, механизм Traffic Shaping помещает избыточные пакеты в буфер.

В качестве средства выравнивания трафика, механизм Traffic Shaping также использует алгоритм «корзина маркеров». В соответствии с механизмом Traffic Shaping, из корзины вынимается число маркеров, равное размеру пакета в битах. Если в корзине имелось достаточное количество маркеров, то пакет передается. В противном случае пакет маркируется как неудовлетворяющий заданному профилю и ставится в очередь (буферизируется) для последующей передачи. Как только в корзине накопится количество маркеров, достаточное для передачи пакета, он будет передан.

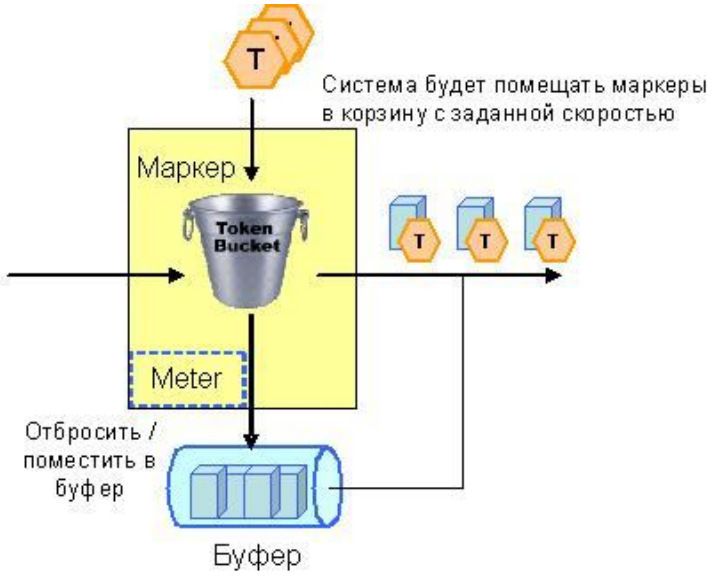


Рис. 7.10. Алгоритм «корзина маркеров» в рамках механизма Traffic Shaping

Следует отметить, что механизм Traffic Shaping вносит задержку в передачу трафика, что критично для приложений чувствительных к задержкам, таким как IP-телефония, потоковое видео и т.д. Однако этот механизм более дружествен к TCP-потокам, т.к. благодаря буферизации уменьшается количество отбрасываемых пакетов и число их повторных передач.

Для управления полосой пропускания входящего и исходящего трафика на портах Ethernet коммутаторы D-Link поддерживают функцию *Bandwidth control*, которая использует для ограничения скорости механизм Traffic Policing. Администратор может вручную устанавливать требуемую скорость соединения на порте в диапазоне от 64 Кбит/с до максимально поддерживаемой скорости интерфейса с шагом 64 Кбит/с.

В качестве примера приведем настройку ограничения скорости до 128 Кбит/с для трафика, передаваемого с интерфейса 5 коммутатора.

config bandwidth_control 5 tx_rate 128

Более гибким решением ограничения полосы пропускания является функция *per-flow Bandwidth control*, реализованная на старших моделях управляемых коммутаторов D-Link. Эта функция позволяет ограничивать полосу пропускания не всему трафику, получаемому или передаваемому с интерфейса коммутатора, а конкретным потокам данных, определенным администратором сети.

Функция per-flow Bandwidth control использует механизм списков управления доступом для просмотра определенного типа трафика и ограничения для него полосы пропускания. Весь этот процесс происходит на микросхемах портов ASIC. Таким образом, это не влияет на загрузку ЦПУ соответственно не снижает производительность коммутатора.

7.8 Пример настройки QoS

На рис. 7.11 приведена схема локальной сети, в которой пользователи 1 и 3 используют приложения IP-телефонии. Голосовому трафику пользователей 1 и 3 требуется обеспечить наивысшее качество обслуживания по сравнению с трафиком других приложений, выполняемых на компьютерах остальных пользователей сети.

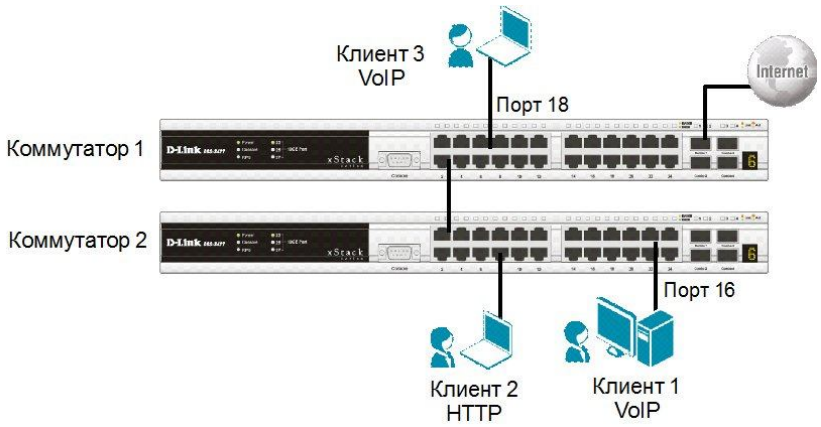


Рис. 7.11. Пример настройки QoS

Настройка коммутатора 1

• Для того чтобы внутри коммутатора могла обрабатываться информация о приоритетах 802.1p, состояние портов коммутатора, к которым подключены пользователи необходимо перевести из «немаркированные» в «маркированные».

config vlan default add tagged 1-6

• Изменить приоритет порта 18, к которому подключен пользователь 3, использующий приложения IP-телефонии с 0 (установлено по умолчанию) на 7. Пакеты с приоритетом 7 будут помещаться в очередь Q6, которая имеет наивысший приоритет обработки.

config 802.1p default_priority 18 7

Настройка коммутатора 2

• Изменить состояния портов с «немаркированные» на «маркированные»

config vlan default add tagged 1-6

• Изменить приоритет порта 16, к которому подключен пользователь 1, использующий приложения IP-телефонии с 0 (установлено по умолчанию) на 7. Пакеты с приоритетом 7 будут помещаться в очередь Q6, которая имеет наивысший приоритет обработки.

config 802.1p default_priority 16 7

Карта привязки приоритетов 802.1p к очередям и механизм обслуживания очередей не изменяются и используют параметры настроенные по умолчанию.

8. Функции обеспечения безопасности и ограничения доступа к сети

На сегодняшний день, для любого системного администратора одной из самых острых проблем остается обеспечение безопасности компьютерной сети. Казалось бы, такие задачи призваны решать межсетевые экраны, однако подчас первый удар принимают на себя именно коммутаторы. Хотя это и не основная их задача, тем не менее, на данный момент коммутаторы обладают широким функционалом для успешного решения подобного рода задач. Речь идет не только о защите сетей от атак извне, но и о всевозможных атаках внутри сети, таких как подмена DHCP-сервера, атаки типа DoS, ARP Spoofing, неавторизованный доступ и т.д. В некоторых случаях коммутаторы не способны полностью защитить сеть от подобного рода атак, но способны значительно ослабить угрозы их возникновения. Данная глава будет посвящена основным принципам обеспечения сетевой безопасности на базе оборудования D-Link.

D-Link предлагает комплексный подход к решению вопросов обеспечения безопасности *End-to-End Security* (E2ES), который включает в себя следующие решения:

- *Endpoint Security* (Защита конечного пользователя) –обеспечивает защиту внутренней сети от внутренних атак.
- *Gateway Security* (Защита средствами межсетевых экранов) – обеспечивает защиту внутренней сети от внешних атак.
- *Joint Security* (Объединенная безопасность) – связующее звено между Endpoint и Gateway Security, объединяющее использование межсетевых экранов и коммутаторов для защиты сети.

Решение Endpoint Security включает следующие функции, обеспечивающие аутентификацию и авторизацию пользователей, контроль над трафиком, узлами и их адресацией в сети.

- Функции аутентификации пользователей:
  - аутентификация IEEE 802.1X;
  - MAC-based Access Control (MAC);
  - WEB-based Access Control (WAC).
- Функции авторизации:
  - Guest VLAN.
- Функции контроля над трафиком:
  - Traffic Segmentation;
  - Access Control List (ACL).
- Функции контроля над подключением/адресацией узлов в сети:
  - Port Security;
  - IP-MAC- Port Binding (IMPB).
- Функции ослабления атак в сети:
  - Access Control List (ACL);
  - IP-MAC- Port Binding (IMPB);
  - Broadcast Storm Control;
  - ARP Spoofing Prevention;
  - LoopBack Detection (LBD).

Решение Joint Security включает в себя функции:

- Zone Defense;
- NAP.

Помимо основных функций безопасности, в коммутаторах D-Link реализованы дополнительные решения, позволяющие обнаруживать аномальные потоки кадров в сети

Ethernet и уменьшать загрузку ЦПУ в результате множественных широковещательных запросов, вызванных атаками типа ARP Flood:

- D-Link Safeguard Engine;
- Traffic Storm Control.

Прежде чем приступить к рассмотрению темы, уточним некоторые понятия.

Аутентификация – процедура проверки подлинности субъекта, на основе предоставленных им данных.

Авторизация - предоставление определенных прав лицу на выполнение некоторых действий.

Как правило, за аутентификацией следует авторизация.

8.1 Списки управления доступом (ACL)

Списки управления доступом (*Access Control List, ACL*) являются мощным средством фильтрации потоков данных без потери производительности, т.к. проверка содержимого пакетов данных выполняется на аппаратном уровне. Фильтруя потоки данных, администратор может ограничить типы приложений, разрешенных для использования в сети, контролировать доступ пользователей к сети и определять устройства, к которым они могут подключаться. Также ACL могут использоваться для определения политики QoS, путем классификации трафика и переопределения его приоритета.

ACL представляют собой последовательность условий проверки параметров пакетов данных. Когда сообщения поступают на входной порт, коммутатор проверяет параметры пакетов данных на совпадение с критериями фильтрации, определенными в ACL, и выполняет над пакетами данных одно из действий: Permit (Разрешить) или Deny (Запретить). Критерии фильтрации могут быть определены на основе следующей информации, содержащейся в пакете данных:

- порт коммутатора;
- MAC/ IP-адрес;
- тип Ethernet/ тип протокола;
- VLAN;
- 802.1p/ DSCP;
- порт TCP/ UDP (тип приложения);
- первые 80 байт пакета, включая поле данных.

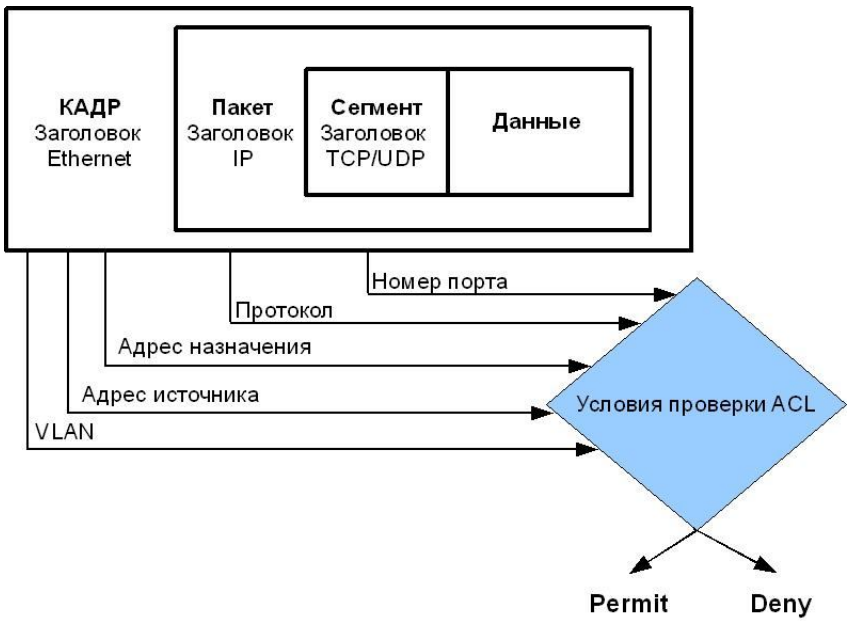


Рис. 8.1. Списки управления доступом (ACL)

Внимание: наборы критериев фильтрации ACL могут отличаться у разных моделей коммутаторов, поэтому прежде чем приступать к конфигурированию функции, необходимо ознакомиться с документацией на используемое устройство.

8.1.1 Профили доступа и правила ACL

Списки управления доступом состоят из *профилей доступа* (Access Profile) и *правил* (Rule). Профили доступа определяют типы критериев фильтрации, которые должны проверяться в пакете данных (MAC-адрес, IP-адрес, номер порта, VLAN и т.д.), а в правилах непосредственно указываются значения их параметров. Каждый профиль может состоять из множества правил.

Когда коммутатор получает кадр, он проверяет его поля на совпадение с типами критериев фильтрации и их параметрами, заданными в профилях и правилах. Последовательность, в которой коммутатор проверяет кадр на совпадение с параметрами фильтрации, определяется порядковым номером профиля (Profile ID) и порядковым номером правила (Rule ID). Профили доступа и правила внутри них работают последовательно, в порядке возрастания их номеров. Т.е. кадр проверяется на соответствие условиям фильтрации, начиная с первого профиля и первого правила в нем. Так кадр сначала будет проверяться на соответствие условиям, определенным в правиле 1 профиля 1. Если параметры кадра не подходят под условия проверки, то далее кадр будет проверяться на совпадение с условиями, определенными в правиле 2 профиля 1 и т.д. Если ни одно из правил текущего профиля не совпало с параметрами кадра, то коммутатор продолжит проверку на совпадение параметров кадра с условиями правила 1 следующего профиля. При первом совпадении параметров кадра с правилом, к пакету данных будет применено одно из действий, определенных в правиле: «Запретить», «Разрешить» или «Изменить содержимое поля пакета» (приоритет 802.1p/ DSCP). Дальше пакет данных проверяться не будет. Если ни одно из правил не подходит, применяется политика по умолчанию, разрешающая прохождение всего трафика.

Следует отметить, что коммутаторы имеют ограничения по количеству обрабатываемых профилей и правил. Информацию о максимальном количестве поддерживаемых профилей и правил можно найти в документации на используемое устройство.

Типы профилей доступа

В коммутаторах D-Link существует три типа профилей доступа: Ethernet, IP и Packet Content Filtering (фильтрация по содержимому пакета).

Профиль Ethernet (Ethernet Profile) позволят фильтровать кадры по следующим типам критериев:

- VLAN;
- MAC-адрес источника;
- MAC-адрес назначения;
- 802.1p;
- тип Ethernet.

Профиль IP (IP Profile) поддерживает следующие типы критериев фильтрации:

- VLAN;
- маска IP-источника;
- маска IP-назначения;
- DSCP;
- протокол (ICMP, IGMP, TCP, UDP);
- номер порта TCP/UDP.

Профиль фильтрации по содержимому пакета (Packet Content Filtering Profile) используется для идентификации пакетов, путем побайтного исследования их заголовков Ethernet.

Внимание: не все модели коммутаторов поддерживают Packet Content Filtering Profile. За информацией о поддержке функции необходимо обратиться к документации на используемый коммутатор.

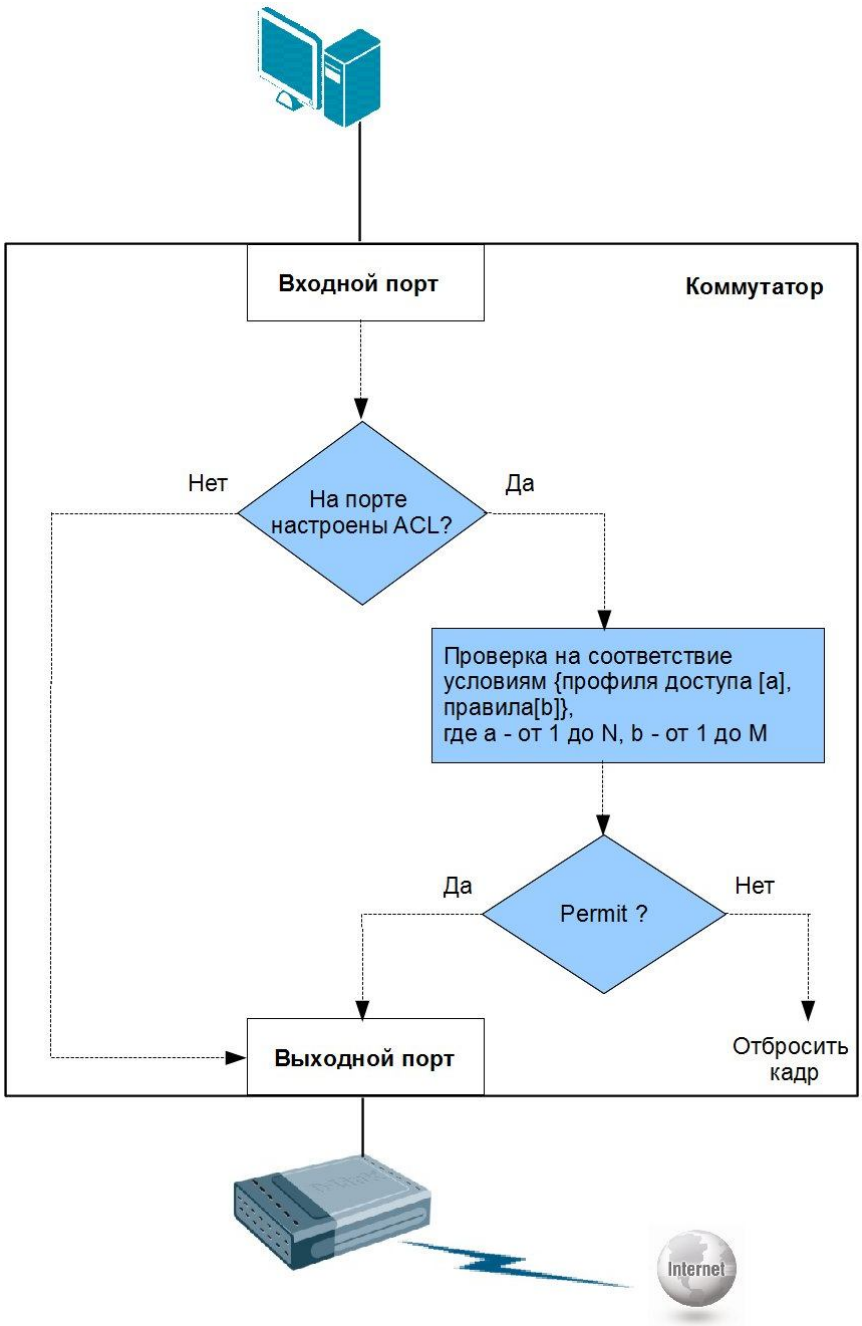


Рис. 8.2. Принцип работы ACL

Процесс создания профиля доступа

Процесс создание профиля доступа можно разделить на следующие основные шаги:

- Проанализируйте задачи фильтрации и определитесь с типом профиля доступа – Ethernet, IP или Packet Content Filtering.

- Определите стратегию фильтрации.
  Например:
  - отбрасывать пакеты данных некоторых узлов и принимать пакеты данных от всех остальных узлов – эта стратегия применима для сетевой среды с несколькими узлами/протоколами портов/подсетями, для которых необходимо выполнять фильтрацию;
  - принимать пакеты данных от некоторых узлов и отбрасывать пакеты данных всех остальных узлов – эта стратегия применима для сетевой среды с несколькими узлами/протоколами портов/подсетями, пакеты данных от которых разрешены в сети. Трафик остальных узлов будет отбрасываться.

  Основываясь на выбранной стратегии, определите, какая маска профиля доступа (Access Profile Mask) необходима, и создайте ее (команда *create access_profile*). Маска профиля доступа используется для указания, какие биты значений полей IP-адрес, MAC-адрес, порт TCP/UDP и т.д. должны проверяться в пакете данных, а какие игнорироваться.
- Добавьте правило профиля доступа (Access Profile Rule), связанное с этой маской (команда *config access_profile*).
- Правила профиля доступа проверяются в соответствии с номером access_id. Чем меньше номер, тем раньше проверяется правило. Если ни одно правило не сработало, пакет данных пропускается.
- В среде QoS, после того как срабатывает правило, перед отправкой пакета данных биты 802.1p/DSCP могут быть заменены на новые низко/высокоприоритетные значения.

Вычисление маски профиля доступа

Маска профиля доступа определяет, какие биты в значениях полей IP-адрес, MAC-адрес, порт TCP/UDP и т.д. приходящих на коммутатор кадров, должны проверяться, а какие игнорироваться. Биты маски имеют следующие значения:

«0» – означает игнорирование значения соответствующего бита поля пакета данных;

«1» – означает проверку значения соответствующего бита поля пакета данных.

Предположим, администратору сети необходимо запретить прохождение трафика от узла с MAC-адресом 01-00-00-00-AC-11. Маска профиля доступа для этого адреса будет равна FF-FF-FF-FF-FF-FF. Если необходимо запретить или разрешить прохождение через коммутатор трафика любого узла из подсетей 192.168.16.0/24 – 192.168.31.0/24, то маска профиля доступа будет вычисляться, как показано на рисунке ниже.

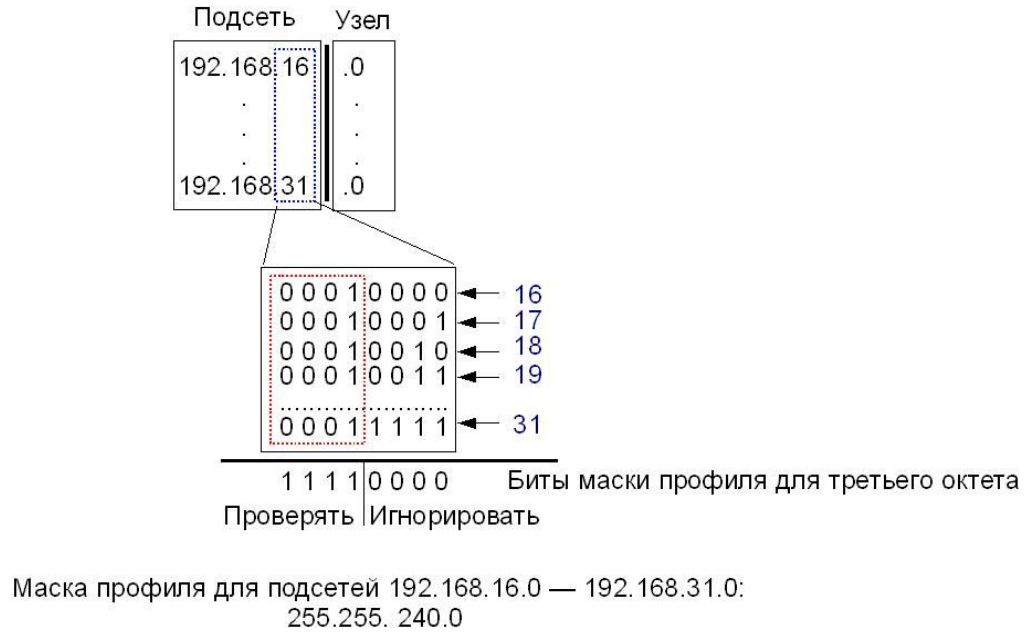


Рис. 8.3. Вычисление маски профиля

Первые два октета IP-адресов из проверяемого диапазона имеют одинаковое значение – «192.168». Они будут использоваться при проверке пакета, поэтому соответствующие биты маски содержат все 1. Последний октет IP-адреса, будет игнорироваться, т.к. нет заинтересованности в проверке индивидуальных адресов узлов подсетей. Поэтому последний октет маски профиля содержит все 0. В третьем октете значение маски будет равно 240 (11110000), т.к. оно охватывает все номера с 16 (00010000) до 31 (0001111), имеющие одинаковые значения (0001) первых четырех битов. Последние четыре бита третьего октета IP-адреса, маска профиля будет игнорировать, как малозначащие.

8.1.2 Примеры настройки ACL

Предположим, что администратору сети необходимо разрешить доступ в Интернет только некоторым пользователям, а остальным пользователям запретить. Пользователи идентифицируются по MAC-адресам их компьютеров.

В примере, показанном на рис. 8.4, пользователи ПК 1 и ПК 2 получат доступ в Интернет, т.к. их MAC-адреса указаны в разрешающем правиле 1. Как только пользователи других компьютеров попытаются выйти в Интернет, сработает правило 2, которое запрещает прохождение через коммутатор кадров с MAC-адресом назначения, равным MAC-адресу Интернет-шлюза.

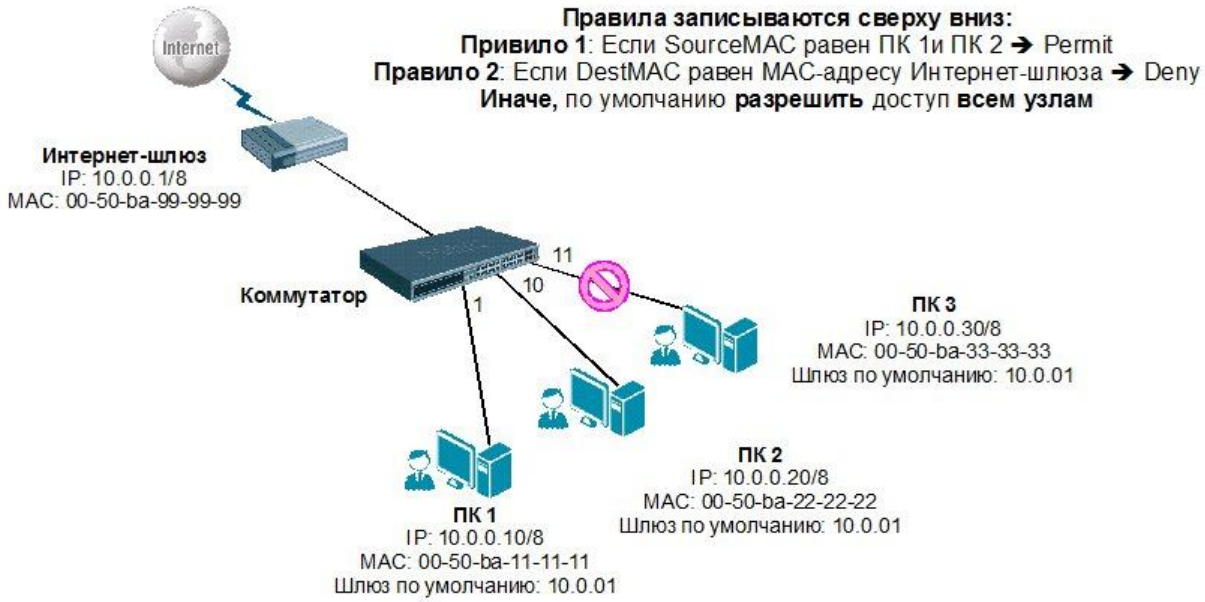


Рис. 8.4. Пример ACL для профиля Ethernet

Настройка коммутатора для профиля Ethernet

- Правило 1: если MAC-адрес источника SourceMAC равен MAC-адресам ПК 1 или ПК 2 – разрешить (Permit).

create access_profile ethernet source_mac FF-FF-FF-FF-FF-FF profile_id 1 profile_name Permit_Internet
config access_profile profile_id 1 add access_id 1 ethernet source_mac 00-50-ba-11-11-11 port 1 permit
config access_profile profile_id 1 add access_id 2 ethernet source_mac 00-50-ba-22-22-22 port 10 permit

- Правило 2: если MAC-адрес назначения DestMAC равен MAC-адресу Интернет-шлюза – запретить (Deny).

create access_profile ethernet destination_mac FF-FF-FF-FF-FF-FF profile_id 2 profile_name Deny_Internet
config access_profile profile_id 2 add access_id 1 ethernet destination_mac 00-50-ba-99-99-99 port 11 deny

- Иначе, по умолчанию разрешить доступ всем узлам.

В качестве второго примера приведем настройку ACL с профилем IP. Предположим, что администратору необходимо разрешить доступ в Интернет только пользователям с IP-адресами с 192.168.0.1/24 по 192.168.0.63/24. Остальным пользователем сети 192.168.0.0/24, с адресами не входящими в разрешенный диапазон, доступ в Интернет запрещен.

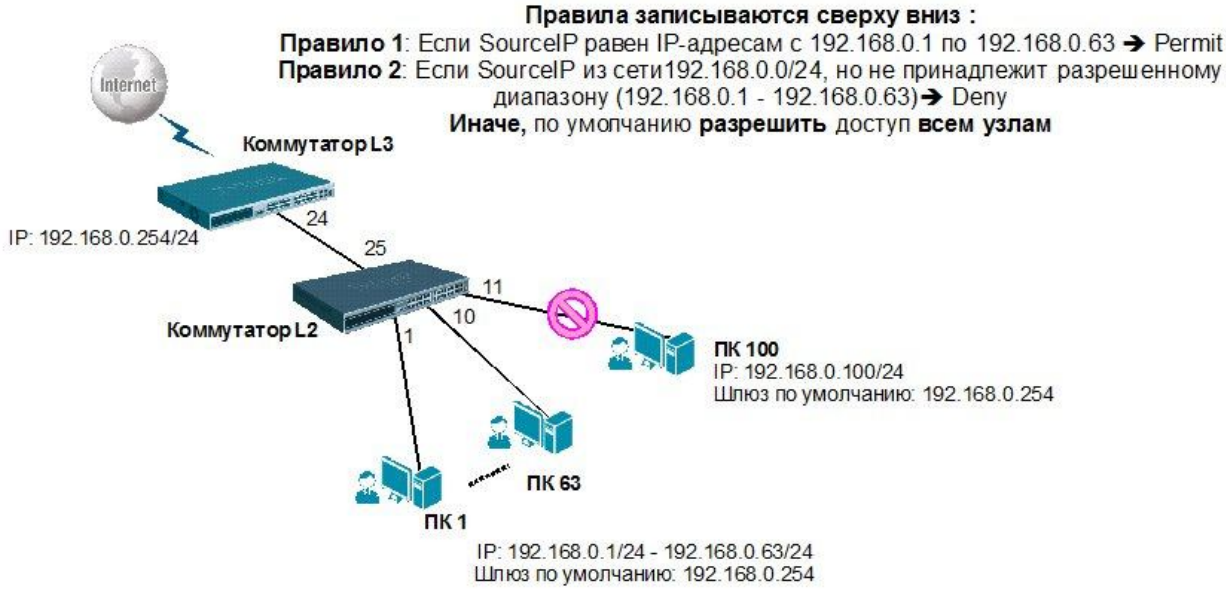


Рис. 8.5. Пример ACL для профиля IP

Настройка в коммутаторе L3 профиля IP

- Правило 1: если IP-адрес источника Source IP равен IP-адресам из диапазона с 192.168.0.1 по 192.168.0.63 – разрешить (Permit).

create access_profile ip source_ip_mask 255.255.255.192 profile_id 1
config access_profile profile_id 1 add access_id 1 ip source_ip 192.168.0.0 port 24 permit

- Правило 2: если IP-адрес источника Source IP принадлежит сети 192.168.0.0/24, но не входит в разрешенный диапазон адресов – запретить (Deny).

create access_profile ip source_ip_mask 255.255.255.0 profile_id 2
config access_profile profile_id 2 add access_id 1 ip source_ip 192.168.0.0 port 24 deny

- Иначе, по умолчанию разрешить доступ всем узлам.

8.2 Функции контроля над подключением узлов к портам коммутатора

В том случае, если какой-либо порт на коммутаторе активен, к нему может подключиться любой пользователь и получить несанкционированный доступ к сети. Этот пользователь может начать генерировать вредоносный трафик, который попадет в сеть и создаст проблемы внутри нее. Для защиты от подобных ситуаций, а также для контроля подключения узлов к портам, коммутаторы D-Link предоставляют функции безопасности, которые позволяют указывать MAC- и/или IP-адреса устройств, которым разрешено подключаться к данному порту, и блокировать доступ к сети узлам с неизвестными коммутатору адресами.

8.2.1 Функция Port Security

Функция **Port Security** позволяет настроить какой-либо порт коммутатора так, чтобы доступ к сети через него мог осуществляться только определенными устройствами. Устройства, которым разрешено подключаться к порту определяются по MAC-адресам. MAC-адреса могут быть изучены динамически или вручную настроены администратором сети. Помимо этого, функция Port Security позволяет ограничивать количество изучаемых портом MAC-адресов, тем самым, ограничивая количество подключаемых к нему узлов.

Внимание: для функции Port Security существуют ограничения по количеству MAC-адресов,

которые может обслуживать каждый порт. Эти ограничения различны для разных моделей коммутаторов. Для получения информации о максимальном количестве обслуживаемых портом MAC-адресов, необходимо обратиться к спецификации на используемое устройство.

Существует три режима работы функции Port Security:

- *Permanent* (Постоянный) – занесенные в таблицу коммутации MAC-адреса никогда не устаревают, даже если истекло время, установленное таймером FDB Aging Time или коммутатор был перезагружен.
- *Delete on Timeout* (Удалить при истечении времени) – занесенные в таблицу коммутации MAC-адреса устареют после истечения времени, установленного таймером FDB Aging Time и будут удалены.
  Если состояние канала связи на подключенном порте изменяется, MAC-адреса, изученные на нем, удаляются из таблицы коммутации, что аналогично выполнению действий при истечении времени, установленного таймером FDB Aging Time.
- *Delete on Reset* (Удалить при сбросе настроек) – занесенные в таблицу коммутации MAC-адреса будут удалены после перезагрузки коммутатора (этот режим используется по умолчанию).

При подключении неавторизованного пользователя к порту коммутатора, он будет заблокирован, а коммутатор отправит сообщение SNMP Trap или создаст запись в Log-файле, если администратор настроил выполнение этих действий. Порт коммутатора будет отбрасывать трафик, поступающий с неизвестного MAC-адреса.

Рис. 8.6. Функция Port Security

8.2.1.1 Пример настройки функции Port Security

В качестве примера рассмотрим ситуацию, показанную на рис. 8.6. На портах 1-3 управляемого коммутатора настроено ограничение по количеству подключаемых пользователей (к каждому порту может подключиться не более двух пользователей). MAC-адреса подключаемых пользователей изучаются портами 1-3 динамически.

Настройка коммутатора

config port_security ports 1-3 admin_state enabled max_learning_addr 2 lock_address_mode DeleteOnTimeout

В приведенном примере конфигурации используется режим Delete on Timeout. Это означает, что изученные на порте MAC-адреса будут удалены из таблицы коммутации по истечении времени, установленного таймером Aging Time, если по ним не было обращений (например, рабочая станция отключилась от сети). В этом случае к сети смогут подключиться новые пользователи, MAC-адреса, которых будут динамически изучены портом (рис. 8.7).

Рис. 8.7. Функция Port Security в режиме Delete on Timeout

При использовании режима работы Permanent, адреса изученные портом будут добавлены в статическую таблицу MAC-адресов, и храниться в ней даже после включения/выключения питания и перезагрузки коммутатора.

Функция Port Security оказывается весьма полезной при построении домовых сетей, сетей провайдеров Интернет и локальных сетей с повышенным требованием по безопасности, где требуется исключить доступ незарегистрированных рабочих станций к услугам сети.

Используя функцию Port Security можно полностью запретить динамическое изучение MAC-адресов указанными или всеми портами коммутатора. В этом случае доступ к сети получат только те пользователи, MAC-адреса которых указаны в статической таблице коммутации.

Настройка коммутатора

- Активизировать функцию Port Security на соответствующих портах и запретить изучение MAC-адресов (параметр *max_learning_addr* установить равным 0).

config port_security ports 1-24 admin_state enabled max_learning_addr 0

- Создать записи в статической таблице MAC-адресов (имя VLAN в примере “default”).

create fdb default 00-50-ba-00-00-01 port 2
create fdb default 00-50-ba-00-00-02 port 2
create fdb default 00-50-ba-00-00-03 port 2
create fdb default 00-50-ba-00-00-04 port 2
create fdb default 00-50-ba-00-00-05 port 8
..... (аналогично для всех требуемых портов)

8.2.2 Функция IP-MAC-Port Binding

Функция IP-MAC-Port Binding (IMPB), реализованная в коммутаторах D-Link, позволяет контролировать доступ компьютеров в сеть на основе их IP- и MAC-адресов, а также порта подключения. Администратор сети может создать записи («белый лист»), связывающие MAC- и IP-адреса компьютеров с портами подключения коммутатора. На основе этих записей, в случае совпадения всех составляющих, клиенты будут получать доступ к сети со своих компьютеров. В том случае, если при подключении клиента, связка MAC-IP-порт будет отличаться от параметров заранее сконфигурированной записи, то коммутатор заблокирует MAC-адрес соответствующего узла с занесением его в «черный лист».

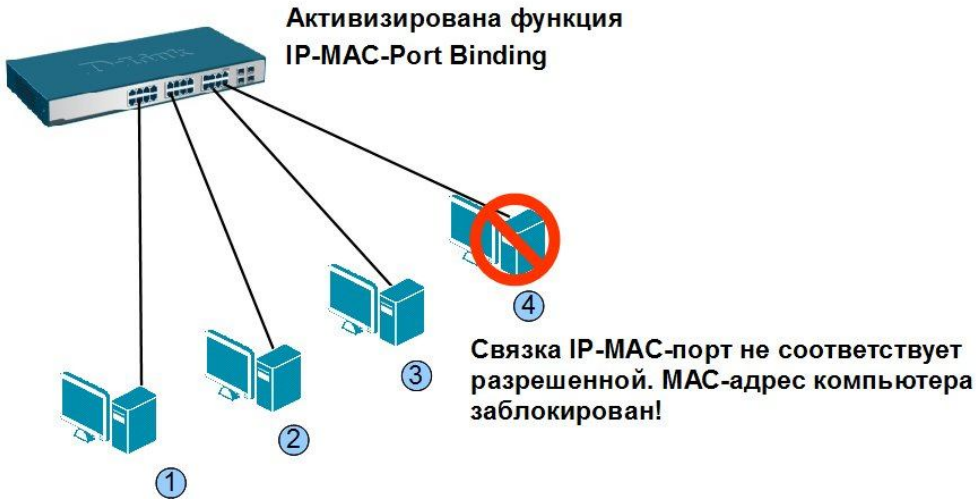


Рис. 8.8. Функция IP-MAC-Port Binding

Функция IP-MAC-Port Binding специально разработана для управления подключением узлов в сетях ETTH (Ethernet-To-The-Home) и офисных сетях. Помимо этого функция IMPB позволяет бороться с атаками типа ARP Spoofing, во время которых злонамеренные пользователи перехватывают трафик или прерывают соединение, манипулируя пакетами ARP.

Функция IP-MAC-Port Binding включает три режима работы: ARP mode (по умолчанию), ACL mode и DHCP Snooping mode.

ARP mode является режимом, используемым по умолчанию, при настройке функции IP-MAC-Port Binding на портах. При работе в режиме ARP коммутатор анализирует ARP-пакеты и сопоставляет параметры IP-MAC ARP-пакета с предустановленной администратором связкой IP-MAC. Если хотя бы 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).

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

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.

Внимание: режим DHCP Snooping отдельно от режимов ARP или ACL не используется.

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 ЦПУ 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.
-

8.2.2.1 Пример настройки функции IP-MAC-Port Binding

На рис. 8.9 показан пример работы функции IP-MAC-Port Binding в режиме ARP. Хакер инициировал атаку типа ARP Spoofing. Коммутатор обнаруживает, что на порт 10 приходят пакеты ARP, связка IP-MAC для которых отсутствует в «белом листе» IMPB, и блокирует MAC-адрес узла.

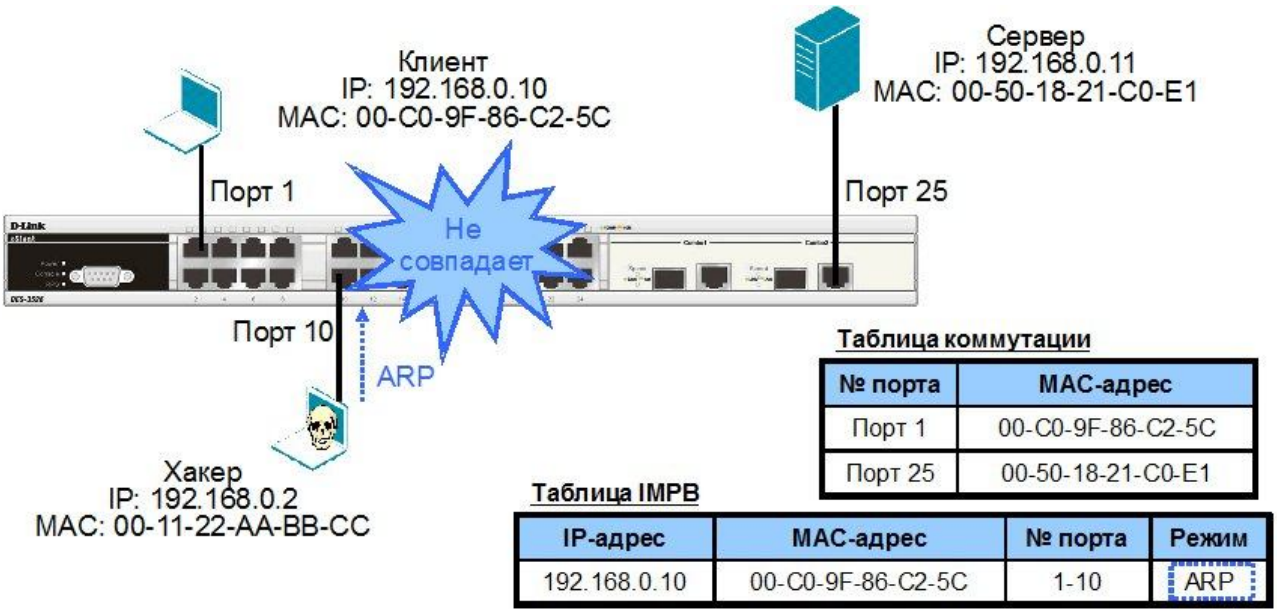


Рис. 8.9. Пример работы функции IP-MAC-Port Binding в режиме ARP

Настройка коммутатора

- Создать запись IP-MAC-Port Binding, связывающую IP-MAC-адрес узла с портами подключения, и указать режим работы функции.

create address_binding ip_mac ipaddress 192.168.0.10 mac_address 00-C0-9F-86-C2-5C ports 1-10 mode arp

- Активизировать функцию на требуемых портах и указать режим работы портов.

config address_binding ip_mac ports 1-10 state enable loose

На рис. 8.10 приведен пример работы функции IP-MAC-Port Binding в режиме DHCP Snooping. Коммутатор динамически создает запись IMPB после того, как клиент получит IP-адрес от DHCP-сервера.

Таблица записей DHCP Snooping

IP Address	MAC Address	Lease Time	Ports	Status
192.168.0.10	00-C0-9F-86-C2-5C	86390	1	Active
192.168.0.11	00-C0-9F-86-C2-5D	86395	10	Active

Таблица IP-MAC-Port Binding

IP Address	MAC Address	Ports	Mode
192.168.0.10	00-C0-9F-86-C2-5C	1	AUTO
192.168.0.11	00-C0-9F-86-C2-5D	10	AUTO

Рис. 8.10. Пример работы функции IP-MAC-Port Binding в режиме DHCP Snooping

Настройка коммутатора

- Активизировать функцию IP-MAC-Port Binding в режиме DHCP Snooping глобально на коммутаторе.

enable address_binding dhcp_snoop

- Указать максимальное количество создаваемых в процессе автоизучения записей IP-MAC на порт.

config address_binding dhcp_snoop max_entry ports 1-10 limit 10

- Активизировать функцию IP-MAC-Port Binding в режиме DHCP Snooping на соответствующих портах.

config address_binding ip_mac ports 1-10 state enable

8.3 Аутентификация пользователей 802.1X

Стандарт IEEE 802.1X (IEEE Std 802.1X-2010) описывает использование протокола EAP (Extensible Authentication Protocol) для поддержки аутентификации с помощью сервера аутентификации и определяет процесс инкапсуляции данных EAP, передаваемых между клиентами (запрашивающими устройствами) и серверами аутентификации. Стандарт IEEE 802.1X осуществляет контроль доступа и не позволяет неавторизованным устройствам подключаться к локальной сети через порты коммутатора.

Сервер аутентификации Remote Authentication in Dial-In User Service (RADIUS) проверяет права доступа каждого клиента, подключаемого к порту коммутатора, прежде чем разрешить доступ к любому из сервисов, предоставляемых коммутатором или локальной сетью.

До тех пор, пока клиент не будет аутентифицирован, через порт коммутатора, к которому он подключен, будет передаваться только трафик протокола Extensible Authentication Protocol over LAN (EAPOL). Обычный трафик начнет передаваться через порт коммутатора сразу после успешной аутентификации клиента.

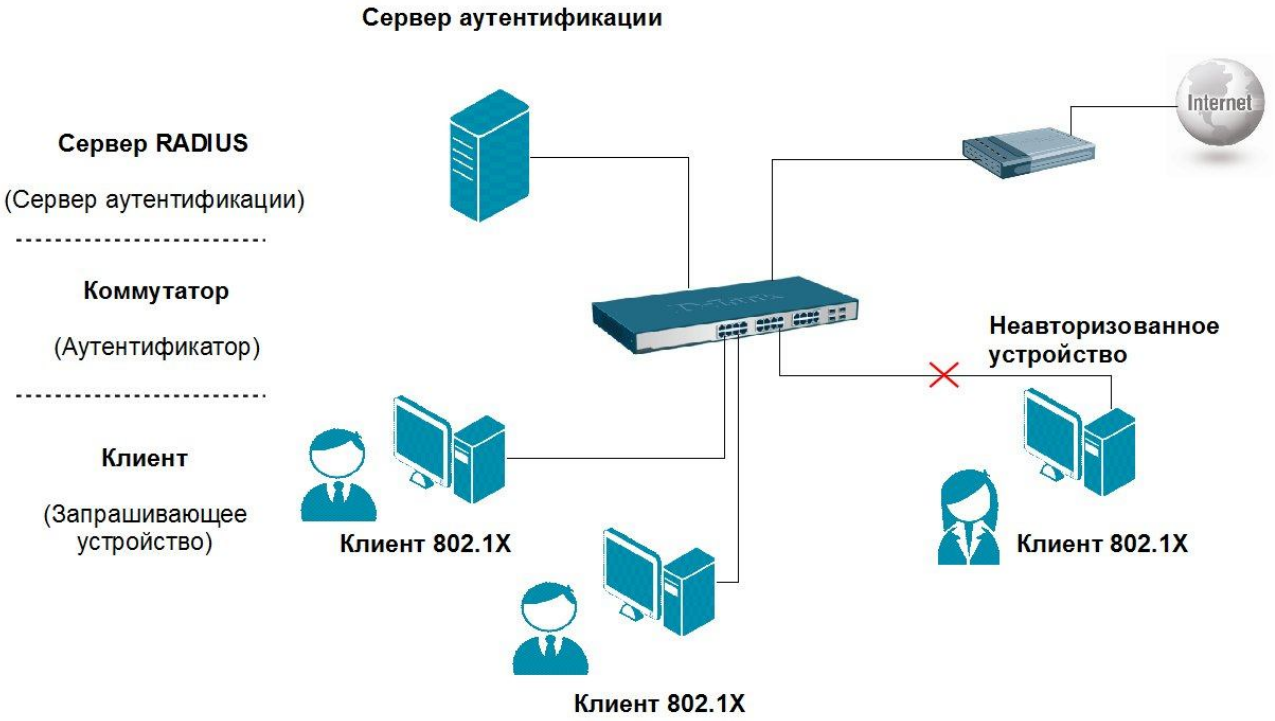


Рис. 8.11. Сеть с аутентификацией 802.1X

Внимание: протокол 802.1X не поддерживает работу на агрегированных каналах связи.

8.3.1 Роли устройств в стандарте 802.1X

В стандарте IEEE 802.1X определены следующие три роли, которые могут выполнять устройства:

- Клиент (Client/Supplicant);
- Аутентификатор (Authenticator);
- Сервер аутентификации (Authentication Server).

Клиент (Client/Supplicant) – это рабочая станция, которая запрашивает доступ к локальной сети и сервисам коммутатора и отвечает на запросы от коммутатора. На рабочей станции должно быть установлено клиентское ПО для 802.1X, например, то, которое встроено в ОС Microsoft Windows XP.

Рис. 8.12. Клиент 802.1X

Сервер аутентификации (Authentication Server) выполняет фактическую аутентификацию клиента. Он проверяет подлинность клиента и информирует коммутатор

предоставлять или нет клиенту доступ к локальной сети. RADIUS (Remote Authentication Dial-In User Service) работает в модели клиент/сервер, в которой информация об аутентификации передается между сервером RADIUS и клиентами RADIUS.

Рис. 8.13. Сервер аутентификации

Аутентификатор (Authenticator) управляет физическим доступом к сети, основываясь на статусе аутентификации клиента. Эту роль выполняет коммутатор. Он работает как посредник (Proxy) между клиентом и сервером аутентификации: получает запрос на проверку подлинности от клиента, проверяет данную информацию при помощи сервера аутентификации и пересылает ответ клиенту. Коммутатор поддерживает клиент RADIUS, который отвечает за инкапсуляцию и деинкапсуляцию кадров EAP, и взаимодействие с сервером аутентификации.

Рис. 8.14. Аутентификатор

Инициировать процесс аутентификации может или коммутатор или клиент.

Клиент инициирует аутентификацию, посылая кадр EAPOL-start, который вынуждает коммутатор отправить ему запрос на идентификацию. Когда клиент отправляет EAP-ответ со своей идентификацией, коммутатор начинает играть роль посредника, предающего кадры EAP между клиентом и сервером аутентификации до успешной или неуспешной аутентификации. Если аутентификация завершилась успешно, порт коммутатора становится авторизованным.

Схема обмена EAP-кадрами зависит от используемого метода аутентификации. На рис. 8.15 показана схема обмена, инициируемого клиентом, где сервером RADIUS используется метод аутентификации One-Time-Password (OTP).

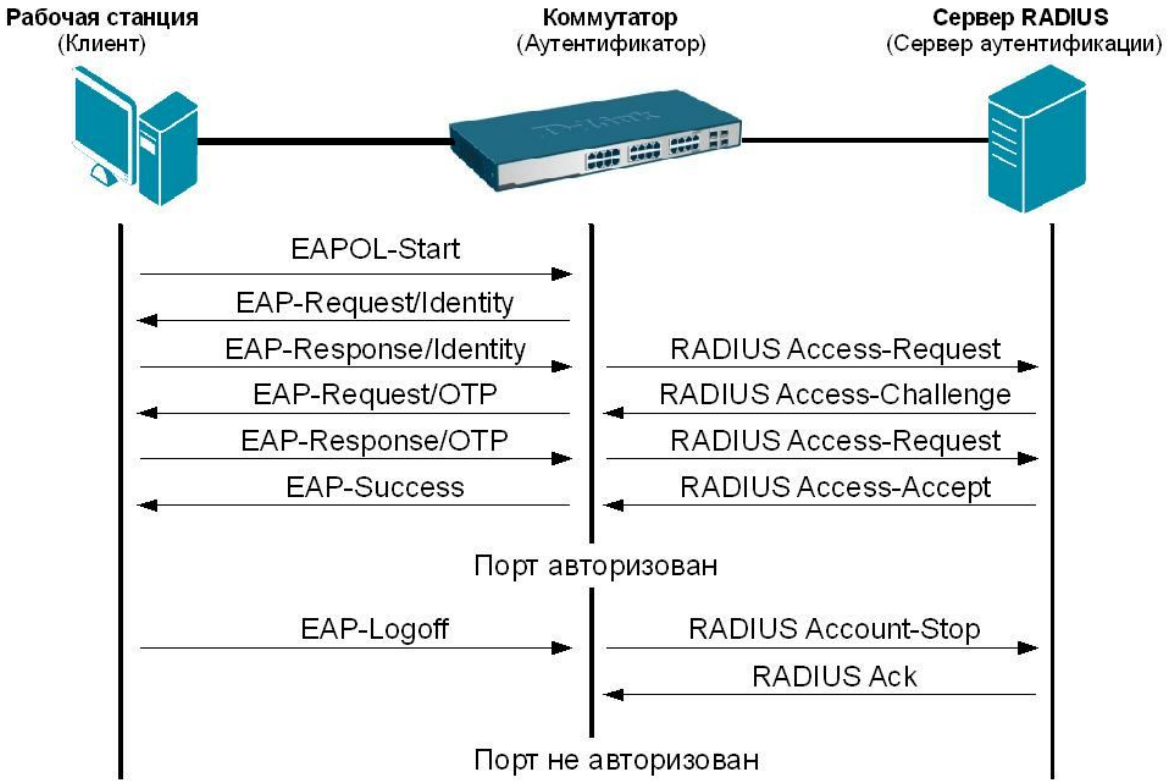


Рис. 8.15. Процесс аутентификации 802.1X

В коммутаторах D-Link поддерживаются две реализации аутентификации 802.1X:
- Port-Based 802.1X (802.1X на основе портов);
- MAC-Based 802.1X (802.1X на основе MAC-адресов).

8.3.2 Port-Based 802.1X

При аутентификации 802.1X на основе портов (Port-Based 802.1X), после того как порт был авторизован, любой пользователь, подключенный к нему, может получить доступ к сети.

Рассмотрим пример настройки функции Port-Based 802.1X для схемы, показанной на рис. 8.16.

Настройка коммутатора DES-3810-28

- Настроить проверку подлинности пользователей на сервере RADIUS.

config 802.1x auth_protocol radius_eap

- Настроить тип аутентификации 802.1X: port-based.

config 802.1x auth_mode port_based

- Настроить порты, к которым подключаются клиенты в качестве аутентификатора (на Uplink-портах к вышестоящим коммутаторам не следует настраивать режим «authenticator»).

config 802.1x capability ports 1-12 authenticator

- Активизировать функцию 802.1X.

enable 802.1x

- Настроить параметры сервера RADIUS.

config radius add 1 192.168.0.10 key 123456 default

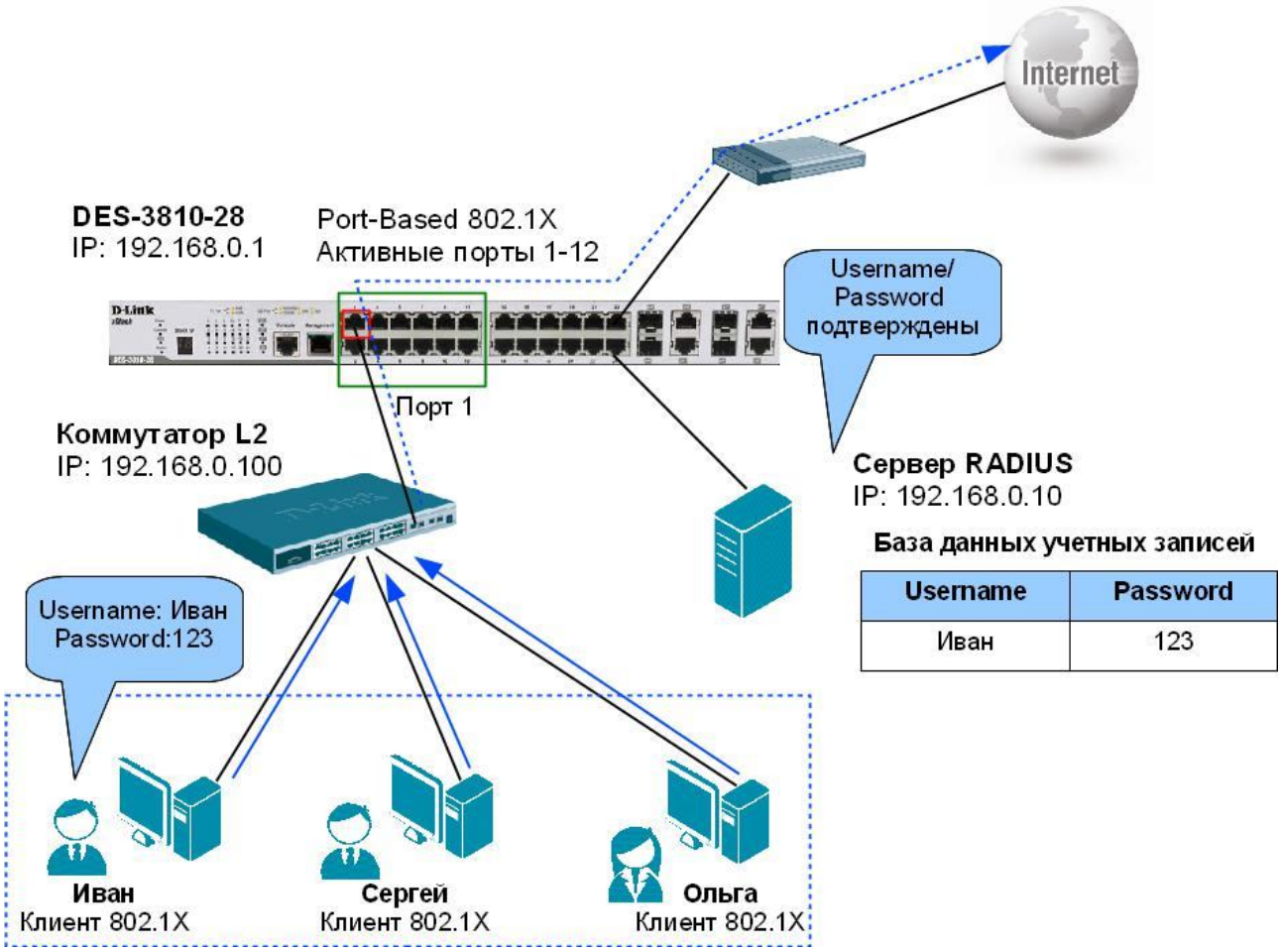


Рис. 8.16. Аутентификация 802.1X на основе портов

8.3.3 MAC-Based 802.1X

В отличие от аутентификации 802.1X на основе портов, где один порт, авторизированный клиентом, остается открытым для всех клиентов, аутентификация 802.1X на основе MAC-адресов (MAC-Based 802.1X) – это аутентификация множества клиентов на одном физическом порте коммутатора. При аутентификации 802.1X на основе MAC-адресов проверяются не только имя пользователя/пароль, подключенных к порту коммутатора клиентов, но и их количество. Количество подключаемых клиентов ограничено максимальным количеством MAC-адресов, которое может изучить каждый порт коммутатора. Для функции MAC-Based 802.1X количество изучаемых MAC-адресов указывается в спецификации на устройство. Сервер аутентификации проверяет имя пользователя/пароль, и если информация достоверна, аутентификатор (коммутатор) открывает логическое соединение на основе MAC-адреса клиента. При этом если достигнут предел, изученных портом коммутатора MAC-адресов, новый клиент будет заблокирован. Рассмотрим пример настройки функции MAC-Based 802.1X для схемы показанной на рис. 8.17.

Настройка коммутатора DES-3810-28

- Настроить проверку подлинности пользователей на сервере RADIUS.

config 802.1x auth_protocol radius_eap

- Настроить тип аутентификации 802.1X: MAC-based.

config 802.1x auth_mode mac_based

- Настроить порты, к которым подключаются клиенты в качестве аутентификатора.

config 802.1x capability ports 1-12 authenticator

- Активизировать функцию 802.1X.

enable 802.1x

- Настроить параметры сервера RADIUS.

config radius add 1 192.168.0.10 key 123456 default

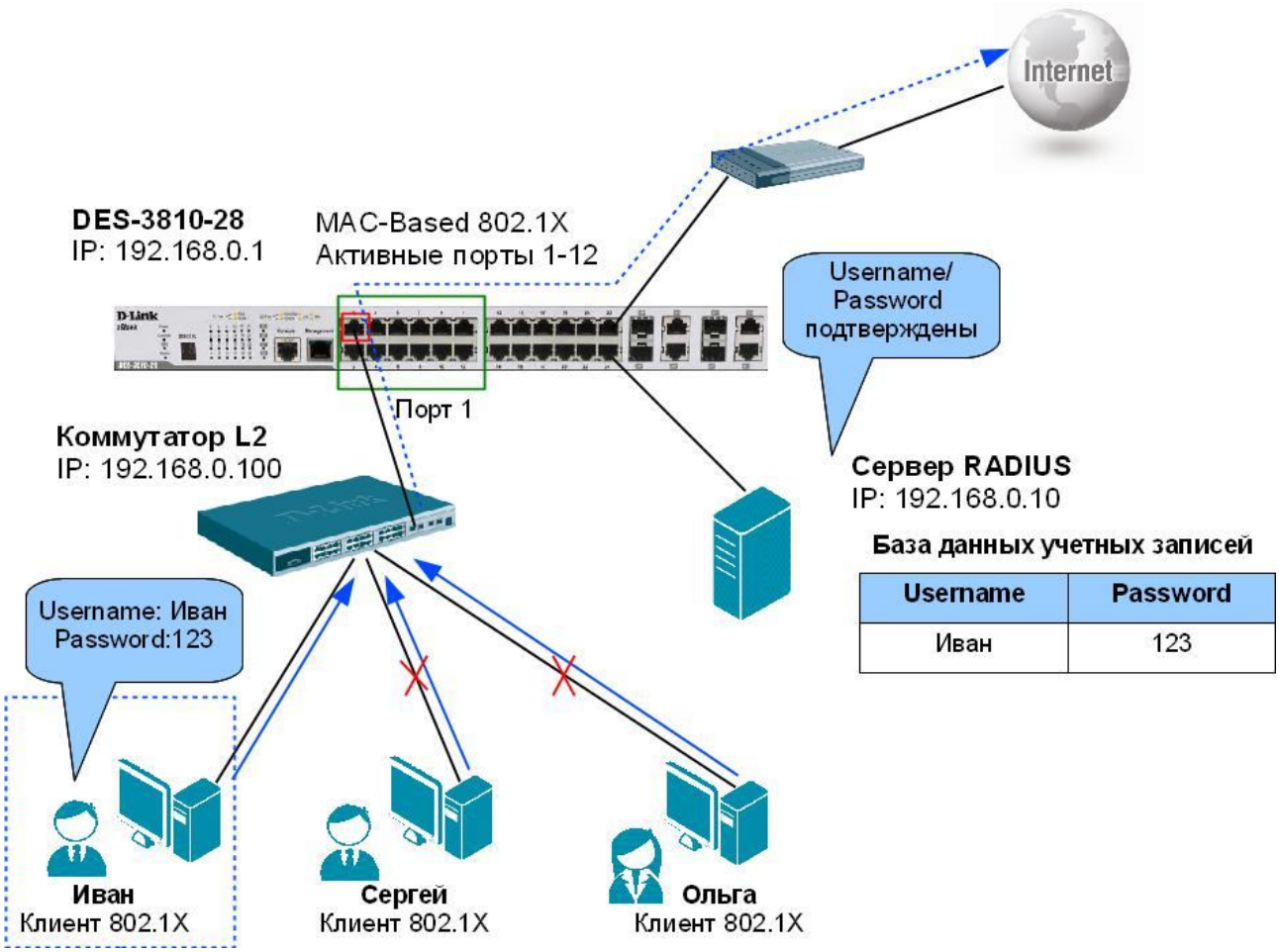


Рис. 8.17. Аутентификация 802.1X на основе MAC-адресов

Следует отметить, что коммутатор может выполнять роль сервера аутентификации. В этом случае база данных учетных записей пользователей будет храниться локально на самом коммутаторе. На рис. 8.18 показана локальная аутентификация 802.1X на основе MAC-адресов.

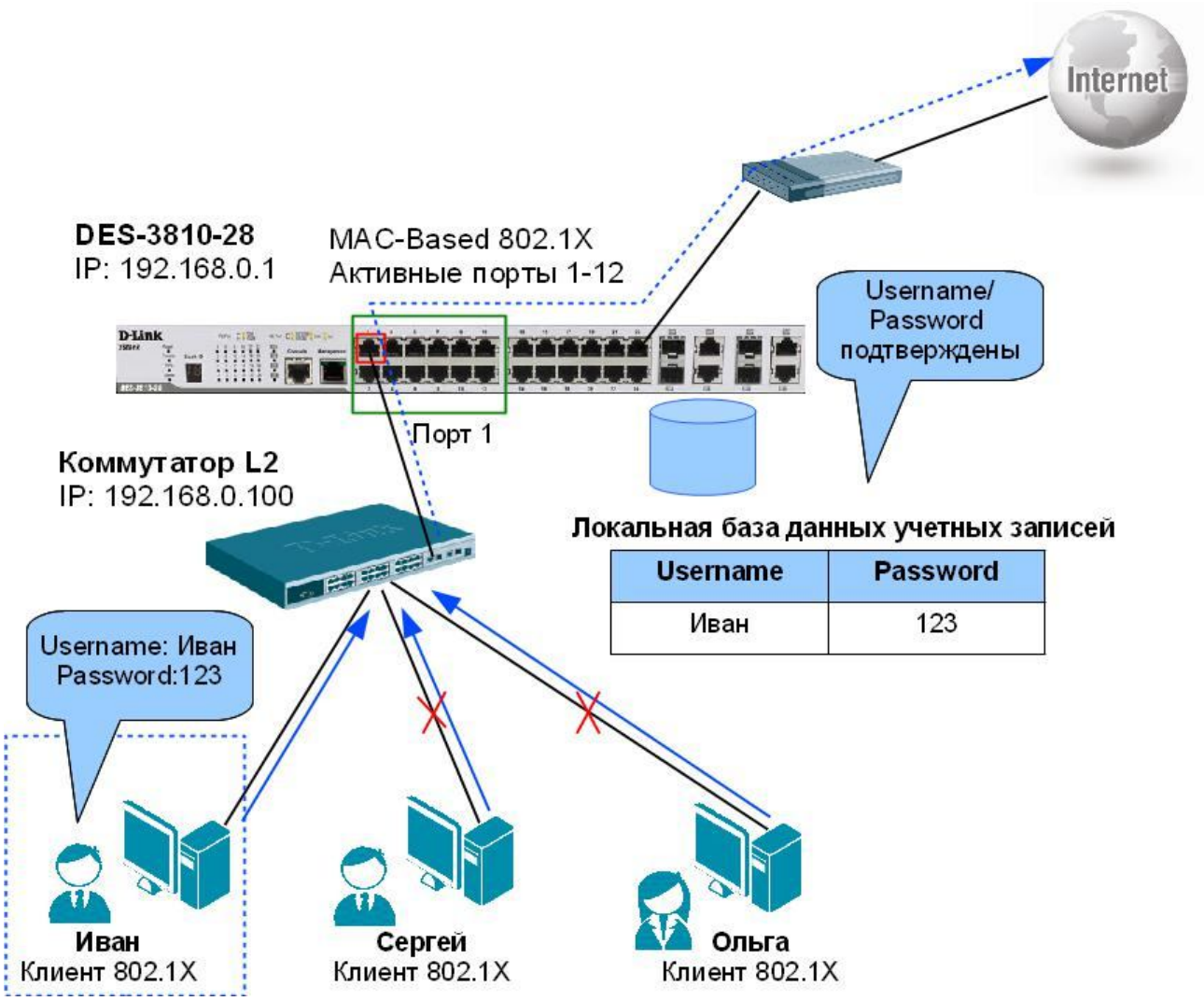


Рис. 8.18. Аутентификация 802.1X на основе MAC-адресов с использованием локальной базы данных учетных записей пользователей

8.3.4 Состояние портов коммутатора

Состояние порта коммутатора определяется тем, получил или не получил клиент право доступа к сети. Первоначально порт находится в *неавторизованном* состоянии. В этом состоянии он запрещает прохождение всего входящего и исходящего трафика за исключением пакетов EAPOL. Когда клиент аутентифицирован, порт переходит в *авторизованное* состояние, позволяя передачу через него любого трафика.

Возможны следующие варианты, когда клиент или коммутатор не поддерживают 802.1X.

Если клиент, который не поддерживает 802.1X, подключается к неавторизованному порту 802.1X, коммутатор посылает клиенту запрос на авторизацию. Поскольку в этом случае, клиент не ответит на запрос, порт останется в неавторизованном состоянии, и клиент не получит доступ к сети.

Когда клиент с поддержкой 802.1X подключается к порту, на котором не поддерживается протокол 802.1X, он начинает процесс аутентификации, посылая кадр EAPOL-start. Не получив ответа, клиент посылает запрос определенное количество раз. Если после этого ответ не получен, клиент, считая, что порт находится в авторизованном состоянии, начинает передавать данные.

В случае, когда и клиент, и коммутатор поддерживают 802.1X, при успешной аутентификации клиента, порт переходит в авторизованное состояние и начинает передавать

все кадры клиента. Если в процессе аутентификации возникли ошибки, порт остается в неавторизованном состоянии, но аутентификация может быть восстановлена. Если сервер аутентификации не может быть достигнут, коммутатор может повторно передать запрос. Если от сервера не получен ответ после определенного количества попыток, клиенту будет отказано в доступе к сети из-за ошибок аутентификации. Чтобы вероятность такой ситуации была минимальной, на коммутаторе можно настроить параметры нескольких серверов RADIUS.

Когда клиент завершает сеанс работы, он посылает сообщение EAPOL-logoff, переводящее порт коммутатора в неавторизованное состояние.

Если состояние канала связи порта переходит из активного (up) в неактивное (down) или получен кадр EAPOL-logoff, порт возвращается в неавторизованное состояние.

8.4 802.1X Guest VLAN

Функция 802.1X Guest VLAN используется для создания гостевой VLAN с ограниченными правами для пользователей не прошедших аутентификацию. Когда клиент подключается к порту коммутатора с активизированной аутентификацией 802.1X и функцией Guest VLAN, происходит процесс аутентификации (локально или удаленно с использованием сервера RADIUS). В случае успешной аутентификации клиент будет помещен в VLAN назначения (Target VLAN) в соответствии с предустановленным на сервере RADIUS параметром VLAN. Если этот параметр не определен, то клиент будет возвращен в первоначальную VLAN (в соответствии с настройками порта подключения).

В том случае, если клиент не прошел аутентификацию, он помещается в Guest VLAN с ограниченными правами и доступом.

Более наглядно данный процесс приведен на блок-схеме (рис. 8.19).

Члены Guest VLAN могут взаимодействовать друг с другом в пределах этой VLAN, даже если они не прошли аутентификацию 802.1X. После успешного прохождения аутентификации, член Guest VLAN может быть перемещен в VLAN назначения (Target VLAN) в соответствии с атрибутом VLAN, указанном на сервере RADIUS.

Внимание: функция Guest VLAN поддерживается только для аутентификации 802.1X на базе портов.

Следует отметить, что, используя функцию 802.1X Guest VLAN, клиентам можно предоставлять ряд ограниченных сервисов *до прохождения* процесса аутентификации 802.1X. Например, клиент может загрузить с сервера и установить необходимое программное обеспечение 802.1X.

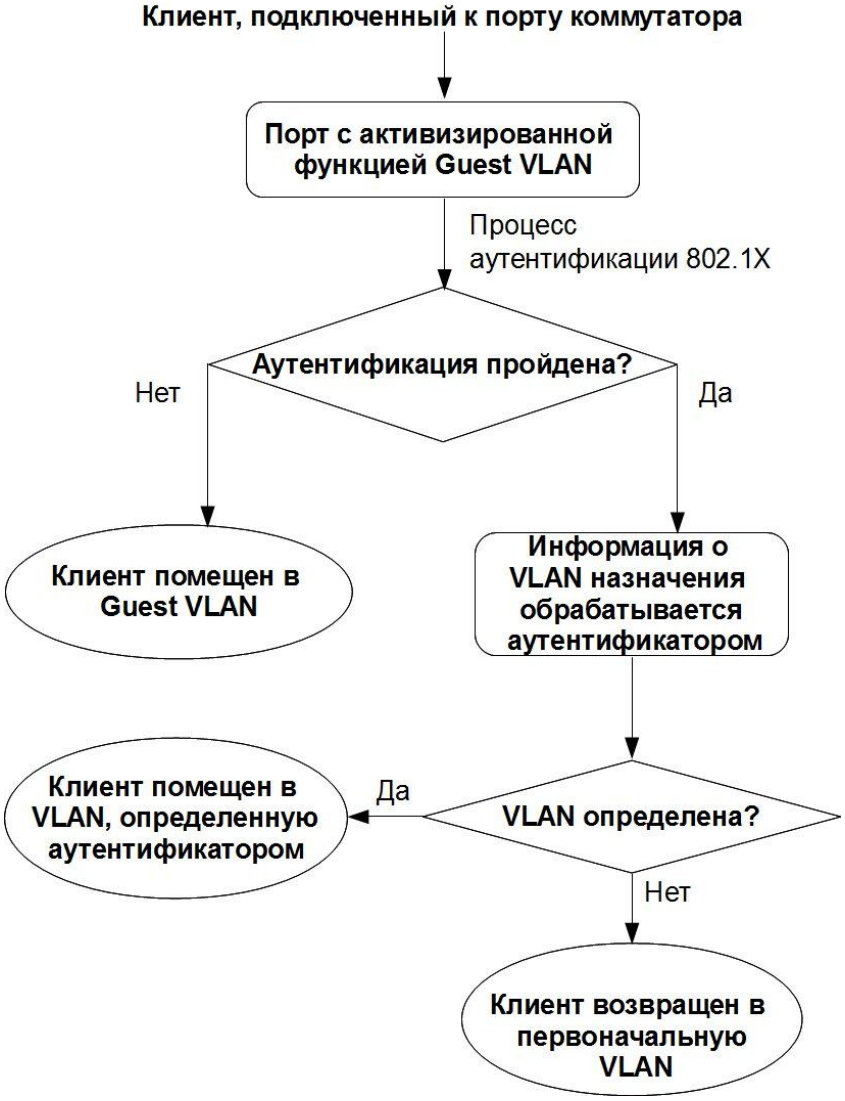


Рис. 8.19. Процесс аутентификации с использованием Guest VLAN

Рассмотрим пример, показанный на рис. 8.20. До аутентификации клиент 1 находится в Guest VLAN и имеет доступ к рабочим станциям, расположенным в ней, и общедоступному Web/FTP-серверу. После успешной аутентификации клиента 1, порт коммутатора, к которому он подключен, будет добавлен в VLAN 10, и клиент 1 сможет получить доступ к конфиденциальной информации, хранящейся на FTP-сервере VLAN 10.

Если клиент не прошел аутентификацию 802.1X, он останется в Guest VLAN с ограниченными правами.

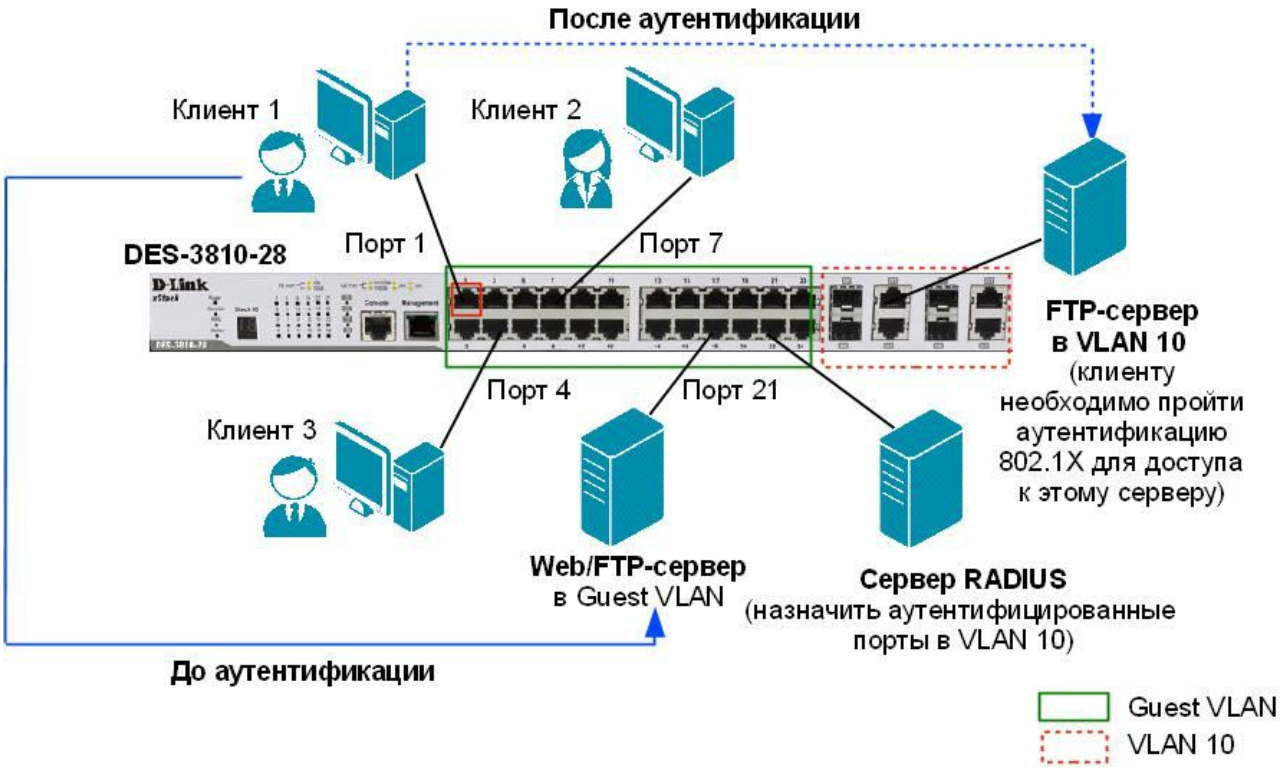


Рис. 8.20. Ресурсы доступные клиенту до и после аутентификации 802.1X при использовании Guest VLAN

8.4.1 Пример настройки 802.1X Guest VLAN

В качестве примера использования и настройки функции 802.1X Guest VLAN, рассмотрим схему сети компании (рис. 8.21), в которой неаутентифицированным пользователям, находящимся в VLAN 10, разрешен доступ в Интернет. После успешной аутентификации пользователей, порты к которым они подключены, будут добавлены в VLAN 20.

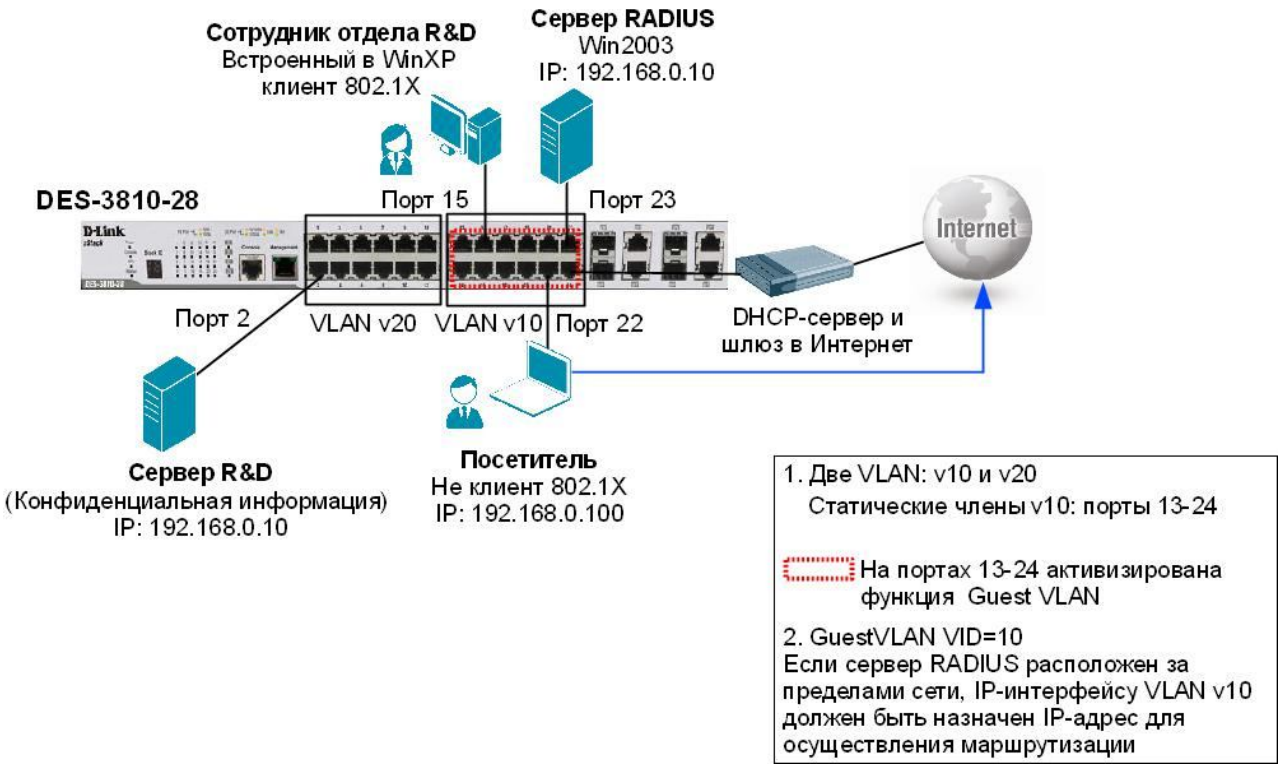


Рис. 8.21. Пример использования 802.1X Guest VLAN

Настройка коммутатора DES-3810-28

- Создать на коммутаторе VLAN v10 и v20.

config vlan default delete 1-24
create vlan v10 tag 10
config vlan v10 add untagged 13-24
create vlan v20 tag 20
config vlan v20 add untagged 1-12
config ipif System ipaddress 192.168.0.1/24 vlan v10

- Активизировать функции 802.1X и Guest VLAN.

enable 802.1x
create 802.1x guest_vlan v10
config 802.1x guest_vlan ports 13-24 state enable

- Настроить коммутатор в качестве аутентификатора и задать параметры сервера RADIUS.

config 802.1x capability ports 13-24 authenticator
config radius add 1 192.168.0.10 key 123456 default

Настройка параметров на сервере RADIUS включает установку следующих пользовательских атрибутов:

Tunnel-Medium-Type (65) = 802
Tunnel-Pvt-Group-ID (81) = 20 ←VID
Tunnel-Type (64) = VLAN

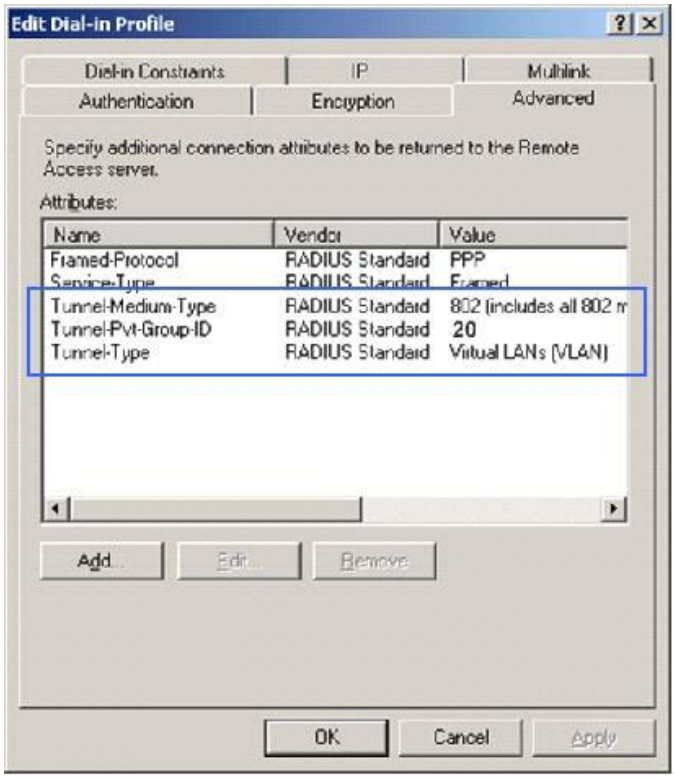


Рис. 8.22. Пользовательские атрибуты на сервере RADIUS

Проверить конфигурацию коммутатора можно с помощью следующих команд:

```
DGS-3810-28#show 802.1x auth_configuration
Command: show 802.1x auth_configuration

802.1X                    : Enabled
Authentication Mode       : Port_based
Authentication Protocol   : RADIUS_EAP

Port number     : 1
Capability      : None
AdminCrlDir     : Both
OpenCrlDir      : Both
Port Control    : Auto
QuietPeriod     : 60 sec
TxPeriod        : 30 sec
Supp Timeout    : 30 sec
Server Timeout  : 30 sec
MaxReq          : 2 times
ReAuthPeriod    : 3600 sec
ReAuthenticate  : Disabled

DGS-3810-28#show 802.1x guest_vlan
Command: show 802.1x guest_vlan

Guest VLAN Setting
------------------------------
Guest VLAN : v10
```

```
Enable Guest VLAN Ports : 13-24

DGS-3810-28#show radius
Command: show radius

Idx IP Address    Auth-Port  Acct-Port  Status  Key
--- ------------  ---------- ---------  ------- --------
 1  192.168.0.10  1812       1813       Active  123456

Total Entries: 1
```

Пока клиент, подключенный к порту 22, не прошел аутентификацию, текущая конфигурация VLAN и состояние аутентификации 802.1X на коммутаторе будут следующими:

```
DGS-3810-28#show vlan
VID   : 1   VLAN Name : default
VLAN Type  : Static   Advertisement : Enabled
Member Ports          : 25-27
Static Ports          : 25-27
Current Tagged Ports  :
Current Untagged Ports: 25-27
Static Tagged Ports   :
Static Untagged Ports : 25-27
Forbidden Ports       :

VID   : 10  VLAN Name : v10
VLAN Type  : Static   Advertisement : Disabled
Member Ports          : 13-24
Static Ports          : 13-24
Current Tagged Ports  :
Current Untagged Ports: 13-24
Static Tagged Ports   :
Static Untagged Ports : 13-24
Forbidden Ports :

VID   : 20  VLAN Name : v20
VLAN Type  : Static   Advertisement : Disabled
Member Ports          : 1-12
Static Ports          : 1-12
Current Tagged Ports  :
Current Untagged Ports: 1-12
Static Tagged Ports   :
Static Untagged Ports : 1-12
Forbidden Ports       :

Total Entries : 3

DGS-3810-28#show 802.1x auth_state
Command: show 802.1x auth_state

Port Auth PAE State    Backend State Port State
---- ---------------   ------------- --------------
1    ForceAuth         Success       Authorized
2    ForceAuth         Success       Authorized
3    ForceAuth         Success       Authorized
4    ForceAuth         Success       Authorized
```

```
5     ForceAuth         Success        Authorized
6     ForceAuth         Success        Authorized
7     ForceAuth         Success        Authorized
8     ForceAuth         Success        Authorized
9     ForceAuth         Success        Authorized
10    ForceAuth         Success        Authorized
11    ForceAuth         Success        Authorized
12    ForceAuth         Success        Authorized
13    Disconnected      Idle           Unauthorized
14    Disconnected      Idle           Unauthorized

……
```

22 Connecting Idle Unauthorized

После аутентификации клиента текущие настройки VLAN и состояние аутентификации 802.1X изменятся следующим образом:

```
DGS-3810-28#show vlan
VID   : 1   VLAN Name : default
VLAN Type  : Static   Advertisement : Enabled
Member Ports          : 25-27
Static Ports          : 25-27
Current Tagged Ports  :
Current Untagged Ports: 25-27
Static Tagged Ports   :
Static Untagged Ports : 25-27
Forbidden Ports :

VID   : 10  VLAN Name : v10
VLAN Type  : Static   Advertisement : Disabled
Member Ports          : 13-21,23-24
Static Ports          : 13-21,23-24
Current Tagged Ports  :
Current Untagged Ports: 13-21,23-24
Static Tagged Ports   :
Static Untagged Ports : 13-21,23-24
Forbidden Ports :

VID   : 20  VLAN Name : v20
VLAN Type  : Static   Advertisement : Disabled
Member Ports          : 1-12,22
Static Ports          : 1-12,22
Current Tagged Ports  :
Current Untagged Ports: 1-12,22
Static Tagged Ports   :
Static Untagged Ports : 1-12,22
Forbidden Ports :

Total Entries : 3

DGS-3810-28#show 802.1x auth_state
Command: show 802.1x auth_state

Port Auth PAE State    Backend State Port State
---- --------------    ------------- --------------
```

```
1     ForceAuth          Success        Authorized
2     ForceAuth          Success        Authorized
3     ForceAuth          Success        Authorized
4     ForceAuth          Success        Authorized
5     ForceAuth          Success        Authorized
6     ForceAuth          Success        Authorized
7     ForceAuth          Success        Authorized
8     ForceAuth          Success        Authorized
9     ForceAuth          Success        Authorized
10    ForceAuth          Success        Authorized
11    ForceAuth          Success        Authorized
12    ForceAuth          Success        Authorized
13    Disconnected       Idle          Unauthorized
14    Disconnected       Idle          Unauthorized

.............

22    Authenticated      Idle           Authorized
```

8.5 Функции защиты ЦПУ коммутатора

При возникновении в сети многоадресных или широковещательных штормов, вызванных неправильной настройкой оборудования или сетевыми атаками, может возникнуть проблема, связанная с перегрузкой ЦПУ коммутатора и его недоступностью для выполнения важных сетевых задач. В коммутаторах D-Link реализованы функции **Safeguard Engine** и **CPU Interface Filtering**, обеспечивающие защиту ЦПУ от обработки нежелательных пакетов и повышающих общую отказоустойчивость и доступность сети.

8.5.1 Функция Safeguard Engine

Функция **Safeguard Engine** специально разработана для обеспечения доступности коммутатора в ситуациях, когда в результате наводнения сети вредоносным трафиком, его ЦПУ испытывает сильную загрузку. В результате этого ЦПУ коммутатора не может надлежащим образом обрабатывать пакеты протоколов STP/RSTP/MSTP, IGMP, предоставлять административный доступ через Web-интерфейс, CLI, SNMP и выполнять другие задачи, требующие обработки на ЦПУ. Функция Safeguard Engine позволяет идентифицировать и приоритизировать направляемый для обработки на ЦПУ трафик (например, ARP-широковещание, пакеты с неизвестным IP-адресом назначения и т.д.) с целью отбрасывания нежелательных пакетов для сохранения функциональности коммутатора.

Когда коммутатор с включенной функцией Safeguard Engine получает большое количество пакетов, предназначенных для обработки на ЦПУ и превышающее установленное верхнее пороговое значение *Rising Threshold*, он переходит в режим высокой загрузки (**Exhausted mode**). Находясь в этом режиме, коммутатор может выполнять одно из следующих действий для уменьшения загрузки ЦПУ:

- прекращение получения всех ARP-пакетов и широковещательных IP-пакетов (при работе функции в строгом режиме (*strict mode*));
- ограничение полосы пропускания для получаемых ARP-пакетов и широковещательных IP-пакетов, путем ее динамического изменения (при работе функции в нестрогом режиме (*fuzzy mode*)).

При нормализации работы сети и снижении количества нежелательных пакетов до установленного нижнего порогового значения *Falling Threshold*, коммутатор выйдет из режима высокой загрузки, и механизм Safeguard Engine перестанет функционировать.

Следует отметить, что при переключении коммутатора в режим Exhausted могут возникать следующие побочные эффекты:

- При работе функции Safeguard Engine в строгом режиме, будет невозможно осуществлять административный доступ к коммутатору уровня 2, так как этот режим предусматривает отбрасывание всех ARP-запросов, поступающих на интерфейс ЦПУ. Для решения этой проблемы в статической ARP-таблице управляющей рабочей станции можно создать запись, связывающую MAC-адрес коммутатора с IP-адресом его интерфейса управления. В этом случае рабочей станции не потребуется отправлять ARP-запрос коммутатору.
- При работе функции Safeguard Engine в строгом режиме на коммутаторе уровня 3, помимо невозможности административного доступа, также может быть нарушена маршрутизация между подключенными к нему подсетями, т.к. будут отбрасываться ARP-запросы, поступающие не только на интерфейс ЦПУ, но и на IP-интерфейсы коммутатора.
- Преимуществом нестрогого режима работы функции Safeguard Engine является то, что в нем не просто отбрасываются все ARP-пакеты или широковещательные IP-пакеты, а динамически изменяется полоса пропускания для них. Таким образом, даже при серьезной вирусной эпидемии, коммутатор уровня 2/3 будет доступен по управлению, а коммутатор уровня 3, в том числе, сможет обеспечивать маршрутизацию между подсетями.

8.5.1.1 Пример настройки функции Safeguard Engine

В качестве примера использования функции Safeguard Engine рассмотрим ситуацию, когда одна из рабочих станций, подключенных к коммутатору, постоянно рассылает ARP-пакеты с очень высокой скоростью. Загрузка ЦПУ коммутатора при этом меняется от нормальной до 90%. При устранении причины, вызвавшей лавинную генерацию ARP-пакетов на рабочей станции, загрузка ЦПУ снизится до нормы.

Для защиты ЦПУ от подобных ситуаций и снижения его загрузки, на коммутаторе можно настроить функцию Safeguard Engine.

Настройка коммутатора

- Активизируйте функцию Safeguard Engine.

config safeguard_engine state enable

- Задайте нижнее и верхнее пороговые значения (указываются значения в процентах от загрузки ЦПУ), при которых будет происходить переключение между нормальным режимом работы и режимом Exhausted. Укажите режим работы функции.

config safeguard_engine utilization rising 40 falling 25 mode strict

8.5.2 Функция CPU Interface Filtering

Стандартные списки управления доступом выполняют фильтрацию трафика входящего/исходящего через порты на аппаратном уровне и не могут фильтровать потоки данных, предназначенные для обработки на ЦПУ, например, запросы ICMP, отправляемые на IP-адрес управления коммутатором. В случае возникновения большого количества таких пакетов, производительность коммутатора может сильно снизиться из-за высокой загрузки ЦПУ.

Функция **CPU Interface Filtering,** поддерживаемая на старших моделях коммутаторов D-Link, является еще одним решением, позволяющим ограничивать пакеты, поступающие для обработки на ЦПУ, путем фильтрации нежелательного трафика на программном уровне. По своей сути функция CPU Interface Filtering представляет собой списки управления доступом к интерфейсу ЦПУ и обладает аналогичными стандартным ACL принципами работы и конфигурации.

Несмотря на то, что функция CPU Interface Filtering позволяет контролировать и фильтровать нежелательный трафик, для своей работы она использует центральный процессор. В случае сильной атаки, центральный процессор будет использовать все свои ресурсы для фильтрации вредоносного трафика, что приведет к снижению производительность коммутатора.

Поэтому для уменьшения влияния сетевых атак на ЦПУ коммутатора рекомендуется настраивать обе функции – Safeguard Engine и CPU Interface Filtering.

8.5.2.1 Пример настройки функции CPU Interface Filtering

В качестве примера рассмотрим задачу, в которой необходимо настроить коммутатор таким образом, чтобы пакеты ICMP, передаваемые компьютером ПК 2, не отправлялись на обработку на ЦПУ, но при этом ПК 2 мог передавать данные другим устройствам, например ПК 1.

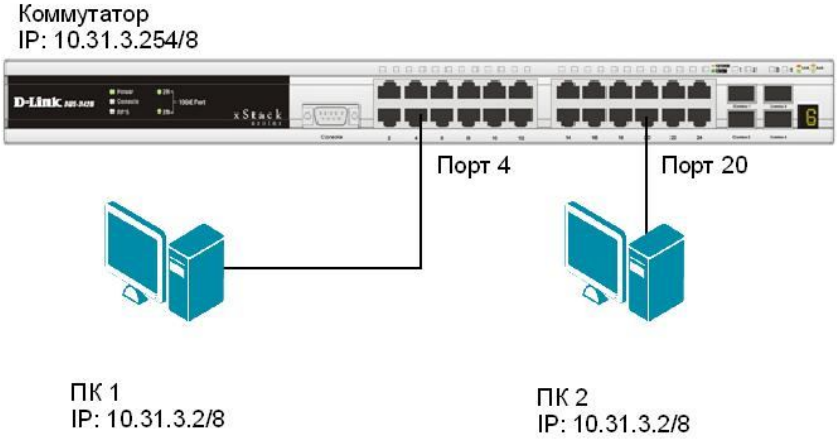


Рис. 8.23. Схема сети

Настройка коммутатора

- Активизируйте функцию CPU Interface Filtering глобально на коммутаторе.

enable cpu_interface_filtering

- Создайте профиль доступа для интерфейса ЦПУ.

create cpu access_profile ip source_ip_mask 255.255.255.128 icmp profile_id 1

- Создайте правило для профиля доступа.

config cpu access_profile profile_id 1 add access_id 1 ip source_ip 10.31.3.2 icmp deny

9. Многоадресная рассылка

В современных IP-сетях существует три способа отправки пакетов от источника к приемнику:

- одноадресная передача (*Unicast*);
- широковещательная передача (*Broadcast*);
- многоадресная рассылка *(Multicast*).

При *одноадресной передаче* поток данных передается от узла-отправителя на индивидуальный IP-адрес конкретного узла-получателя. *Широковещательная передача* предусматривает доставку потока данных от узла-отправителя – множеству узлов-получателей, подключенных к сети, используя широковещательный IP-адрес.

Многоадресная рассылка обеспечивает доставку потока данных группе узлов на IP-адрес *группы многоадресной рассылки*. У этой группы нет физических или географических ограничений: узлы могут находиться в любой точке мира. Узлы, которые заинтересованы в получении данных для определенной группы, должны присоединиться к этой группе (подписаться на рассылку) при помощи протокола IGMP (Internet Group Management Protocol, межсетевой протокол управления группами). После этого пакеты многоадресной рассылки IP, содержащие в поле назначения заголовка групповой адрес, будут поступать на этот узел и обрабатываться.

Многоадресная рассылка имеет ряд преимуществ при работе таких приложений как видеоконференции, корпоративная связь, дистанционное обучение, видео и аудио-трансляции и т.д., т.к. позволяет значительно повысить эффективность использования полосы пропускания и распределения информации среди больших групп получателей. Во-первых, отправитель может один раз передать единственную копию пакета данных всем членам группы, а не рассылать множество его копий. Во-вторых, благодаря передаче только одной копии пакета снижается перегрузка канала связи.

Одним из недостатков многоадресной рассылки является то, что она использует в качестве протокола транспортного уровня протокол UDP, который не гарантирует успешную доставку пакетов, в отличие от протокола TCP.

9.1 Адресация многоадресной IP-рассылки

Источник многоадресного трафика направляет пакеты многоадресной рассылки не на индивидуальные IP-адреса каждого из узлов-получателей, а на групповой IP-адрес. Групповые адреса определяют произвольную группу IP-узлов, присоединившихся к этой группе и желающих получать адресованный ей трафик.

Агентство IANA (Internet Assigned Numbers Authority, Агентство по выделению имен и уникальных параметров протоколов Интернет), которое управляет назначением групповых адресов, выделило для многоадресной рассылки адреса IPv4 класса D в диапазоне от 224.0.0.0 до 239.255.255.255. Адреса, назначенные IANA, приведены в таблице ниже. Более подробную информацию о зарегистрированных адресах можно получить на Web-сайте: http://www.iana.org/assignments/multicast-addresses/multicast-addresses.xhtml#multicast-addresses-12

Таблица 10 **Назначенные IANA диапазоны адресов многоадресной рассылки**

Диапазон	**Описание**
224.0.0.0 - 224.0.0.255	Блок управления локальной сети (Local Network Control Block). Адреса этого диапазона зарезервированы для использования сетевыми протоколами в сегментах локальных сетей.
224.0.1.0 - 224.0.1.255	Межсетевой блок управления (Internetwork Control Block). Адреса из этого диапазона используются для трафика управления протоколов, который может быть передан через Интернет.
224.0.2.0 - 224.0.255.255	Блок AD-HOC I (AD-HOC Block I). Используется для приложений, которые не попадают в блок управления локальной сетью и межсетевой блок управления.

Продолжение таблицы 10

224.1.0.0 - 224.1.255.255	Зарезервировано
224.2.0.0 - 224.2.255.255	Блок SDP/SAP (SDP/SAP Block). Этот диапазон адресов используется для приложений, которые получают адреса через протокол SAP для использования через приложения подобные SDR.
224.3.0.0 - 224.4.255.255	Блок AD-HOC II (AD-HOC Block II). Используется для приложений, которые не попадают в блок управления локальной сетью и межсетевой блок управления.
224.5.0.0 - 224.255.255.255	Зарезервировано
225.0.0.0 - 231.255.255.255	Зарезервировано
232.0.0.0 - 232.255.255.255	Блок специфичной для источника многоадресной рассылки (Source-Specific Multicast Block). Этот диапазон адресов зарезервирован для протокола SSM, который представляет собой расширение протокола PIM.
233.0.0.0 - 233.251.255.255	Блок GLOP (GLOP Block). Этот диапазон адресов зарезервирован для использования в качестве адресов, статически определяемых организациями с зарезервированным номером автономной системы.
233.252.0.0 - 233.255.255.255	Блок AD-HOC III (AD-HOC Block III). Этот диапазон известен как расширенный GLOP (EGLOP, Extended GLOP).
234.0.0.0 - 238.255.255.255	Зарезервировано
239.0.0.0 - 239.255.255.255	Блок административно ограниченных адресов (Administratively Scoped Block). Эти адреса могут локально использоваться внутри домена.

Использование групповых IP-адресов из блока с административным ограничением наиболее удобно при организации многоадресной рассылки в локальной сети предприятия или организации. В соответствии с RFC 2365 «Administratively Scoped IP Multicast» подсеть 239.192.0.0/14 выделена для частного использования и определена как локальная область организации IPv4.

Формат IP-адреса класса D показан на рис. 9.1. Первые 4 бита адреса всегда равны 1110, остальные 28 бит используются для идентификации конкретной группы получателей многоадресного трафика.

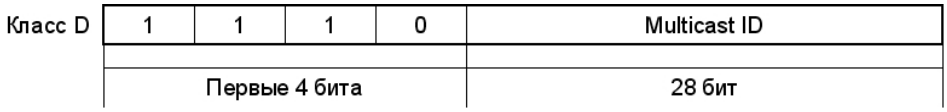


Рис. 9.1. Формат IP-адреса класса D

9.2 MAC-адреса групповой рассылки

Как правило, рабочие станции локальной сети получают и обрабатывают кадры только в случае совпадения MAC-адреса назначения кадра с их собственным MAC-адресом или если MAC-адрес – широковещательный. При использовании многоадресной рассылки необходимо, чтобы несколько узлов могли получать поток данных с общим MAC-адресом. Одним из способов, позволяющим достичь этого является преобразование группового IP-адреса в MAC-адрес.

В спецификации IEEE 802.3 определена возможность указания типа MAC-адреса назначения: индивидуальный или групповой (широковещательный или многоадресный). Для этого используется первый бит поля адреса назначения (Destination Address) кадра Ethernet. Если значение бита равно 1, это указывает на то, что кадр предназначен для группы или для всех узлов сети (широковещательный адрес имеет вид 0xFF-FF-FF-FF-FF-FF).

MAC-адрес групповой рассылки начинается с префикса, состоящего из 24 бит – 0x01-00-5E. Следующий 25-й бит (или бит высокого порядка) приравнивается к 0. Последние 23 бита MAC-адреса формируются из 23 младших бит группового IP-адреса. Это проиллюстрировано на рис. 9.2.

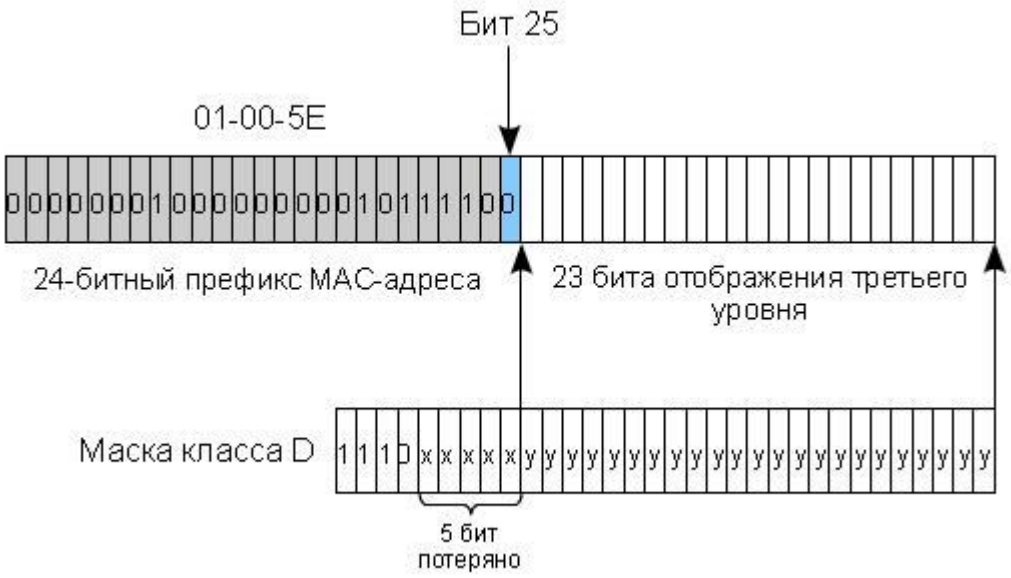


Рис. 9.2. Преобразование группового IP-адреса в адрес MAC-адрес групповой рассылки

Поскольку при преобразовании теряются 5 битов 1-го октета IP-адреса, получившийся адрес не является уникальным. Каждому MAC-адресу соответствует 32 IP-адреса групповой рассылки. Это необходимо учитывать при назначении IP-адресов многоадресной рассылки.

В протоколе IPv6 при использовании многоадресной передачи данных также необходимо, чтобы несколько узлов могли получать поток данных с общим MAC-адресом. MAC-адрес групповой передачи протокола IPv6 начинается с префикса, состоящего из 16 бит – 0x33-33. Следующие 32 бита формируются из последних 32 бит идентификатора многоадресной группы (Group ID). Например:

- FF02::2 =33-33-00-00-00-02;
- FF02::1:ff5c:b300= 33-33-ff-5c-b3-00.

9.3 Подписка и обслуживание групп

Сам по себе многоадресный трафик не знает ничего о том, где находятся его адресаты. Как и для любого приложения для этого нужны протоколы.

Протокол IGMP используется для динамической регистрации отдельных узлов в многоадресной группе локальной сети. Узлы сети определяют принадлежность к группе, посылая IGMP-сообщения на свой локальный многоадресный маршрутизатор. По протоколу IGMP маршрутизаторы (коммутаторы L3) получают IGMP-сообщения и периодически посылают запросы, чтобы определить, какие группы активны или неактивны в данной сети.

В общем случае протокол IGMP определяет следующие типы сообщений:

- запрос о принадлежности к группе (Membership Query);
- ответ о принадлежности к группе (Membership Report);
- сообщение о выходе из группы (Leave Group Message).

В настоящее время существуют три версии протокола IGMP:

- IGMP версии 1 (IGMP v1, описан в RFC 1112);
- IGMP версии 2 (IGMP v2, описан в RFC 2236);
- IGMP версии 3 (IGMP v3, описан в RFC 3376).

Протокол IGMP используется только в сетях с адресацией IPv4, так как в сетях с адресацией IPv6 групповая передача пакетов реализована по-другому.

9.4 Управление многоадресной рассылкой на 2-м уровне модели OSI (IGMP Snooping)

Когда коммутатор 2-го уровня получает многоадресный трафик он начинает передавать кадры через все порты, т.к. не находит записи о MAC-адресе в своей таблице коммутации. Это противоречит основному назначению коммутатора, которое заключается в ограничении трафика и передаче его только тем портам, к которым подключены получатели.

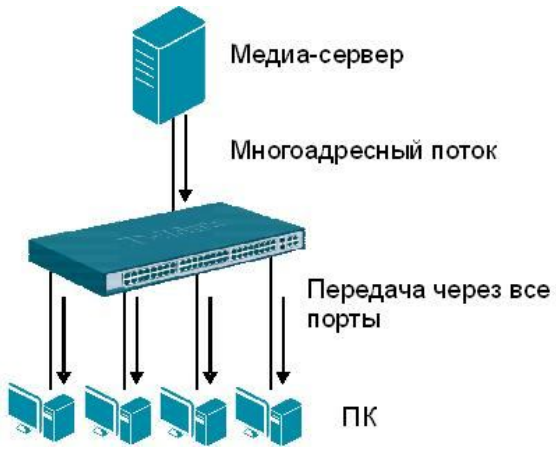


Рис. 9.3. Передача многоадресного трафика без поддержки управления им на коммутаторе

Управление многоадресной рассылкой на коммутаторе 2-го уровня может быть выполнено двумя способами.

Первый способ заключается в создании статических таблиц коммутации для портов, к которым не подключены подписчики многоадресных групп. Это позволяет ограничить многоадресный трафик и передавать его только через те порты, к которым подключены узлы-подписчики. Однако этот способ не позволяет динамически отслеживать добавление или исключение членов из многоадресной группы.

Вторым способом, позволяющим решить проблему лавинной передачи (flooding) многоадресных кадров и динамически отслеживать состояние подписки узлов, является функция **IGMP Snooping (**IGMP-прослушивание).

IGMP Snooping – это функция второго уровня модели OSI, которая позволяет коммутаторам изучать членов многоадресных групп, подключенных к его портам, прослушивая IGMP-сообщения (запросы и ответы), передаваемые между узлами-подписчиками и маршрутизаторами (коммутаторами уровня 3) сети.

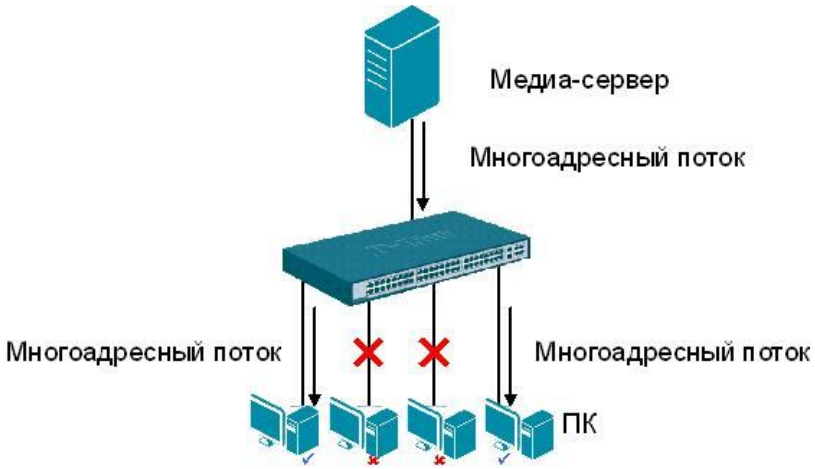


Рис. 9.4. Передача многоадресного трафика с поддержкой IGMP Snooping

Когда узел, подключенный к коммутатору, хочет вступить в многоадресную группу или отвечает на IGMP-запрос, полученный от маршрутизатора (коммутатора уровня 3) многоадресной рассылки, он отправляет IGMP-ответ, в котором указан адрес многоадресной группы. Коммутатор просматривает информацию в IGMP-ответе и создает в своей ассоциативной таблице коммутации IGMP Snooping запись для этой группы (если она не существует). Эта запись связывает порт, к которому подключен узел-подписчик, порт, к которому подключен маршрутизатор (коммутатор уровня 3) многоадресной рассылки, и MAC-адрес многоадресной группы.

Если коммутатор получает IGMP-ответ для этой же группы от другого узла данной VLAN, то он добавляет номер порта в уже существующую запись ассоциативной таблицы коммутации IGMP Snooping.

Формируя таблицу коммутации многоадресной рассылки, коммутатор осуществляет передачу многоадресного трафика только тем узлам, которые в нем заинтересованы.

Рассмотрим пример работы функции IGMP Snooping для сети, показанной на рис. 9.5.

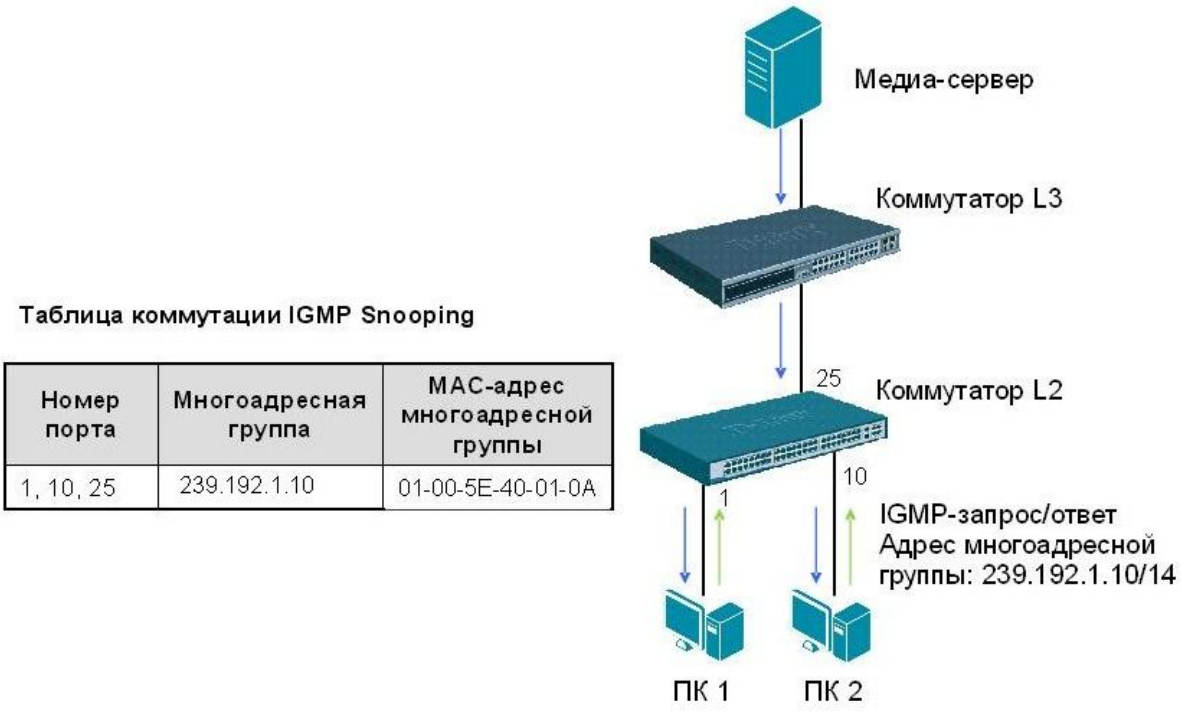


Рис. 9.5. Процесс создания таблицы коммутации IGMP Snooping

Коммутатор L3 отправляет IGMP-запрос о принадлежности к группе коммутатору L2, который рассылает его через все порты, за исключением порта-получателя. ПК 1 хочет вступить в многоадресную группу 239.192.1.10 и отправляет IGMP-ответ на адрес группы, указывая в качестве многоадресного MAC-адреса назначения 0x01-00-5E-40-01-0A. Процессор коммутатора L2 анализирует IGMP-ответ и создает в ассоциативной таблице коммутации IGMP Snooping (в первоначальный момент времени она пуста) запись для MAC-адреса 0x01-00-5E-40-01-0A, эквивалентного групповому адресу 239.192.1.10. Также в эту запись заносится информация о портах, к которым подключены ПК 1 и коммутатор L3.

ПК 2 хочет вступить в многоадресную группу 239.192.1.10 и отправляет IGMP-ответ на адрес группы, не дожидаясь получения очередного IGMP-запроса. Коммутатор L2 анализирует IGMP-ответ и добавляет порт 10, к которому подключен ПК 2, в уже существующую запись для MAC-адреса 0x01-00-5E-40-01-0A.

В результате порты 1, 10 и 25 ассоциированы с многоадресным MAC-адресом 0x01-00-5E-40-01-0A.

Когда коммутатор получает IGMP-сообщение о выходе узла из группы, он удаляет номер порта, к которому подключен этот узел, из соответствующей записи таблицы коммутации IGMP Snooping.

Функция IGMP Snooping сильно загружает центральный процессор и может снизить производительность коммутатора. Поэтому в коммутаторах обычно используются специализированные микросхемы ASIC, которые проверяют IGMP-сообщения на аппаратном уровне.

9.4.1 Пример настройки IGMP Snooping

Рассмотрим пример настройки функции IGMP Snooping на коммутаторах D-Link. На рис. 9.6 показана схема сети, в которой реализован сервис многоадресной рассылки. Клиенты, среди которых имеются подписчики многоадресной рассылки, подключены к коммутаторам второго уровня.

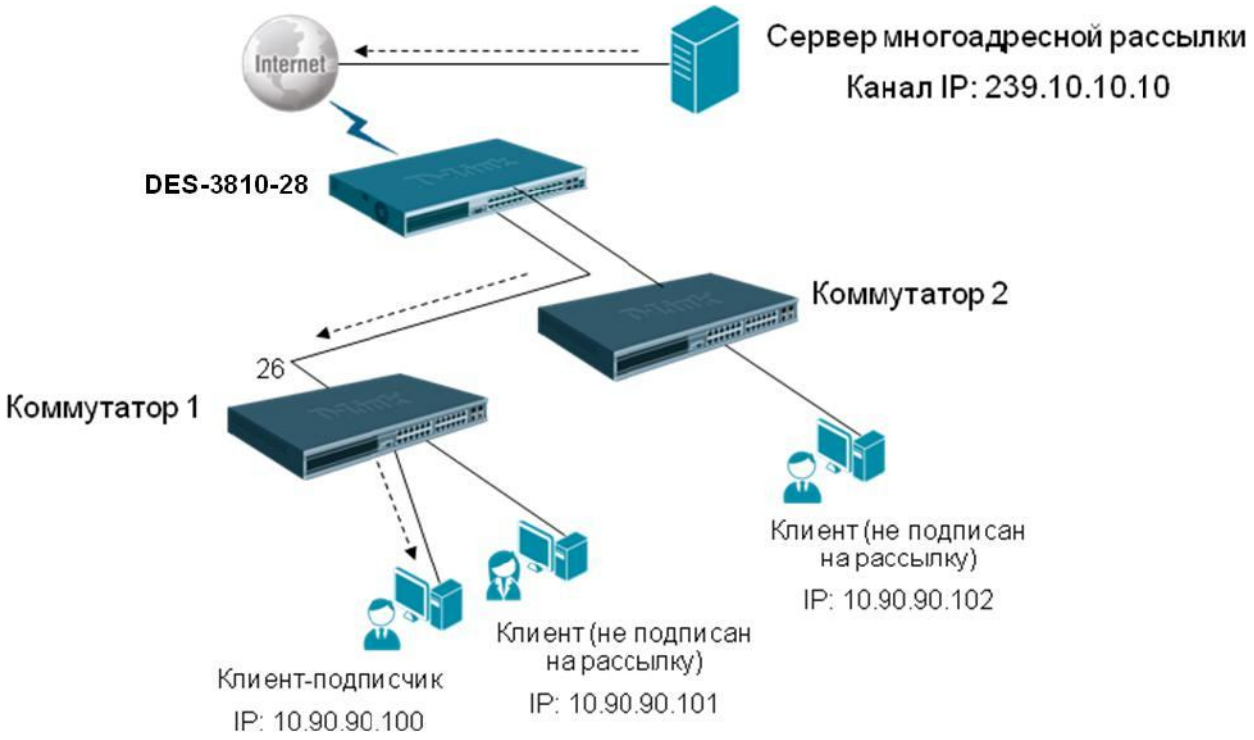


Рис. 9.6. Схема сети

Настройка коммутатора 1

- Активизировать функцию IGMP Snooping глобально на коммутаторе.

enable igmp_snooping

- Активизировать функцию IGMP Snooping в указанной VLAN (в данном примере VLAN по умолчанию).

config igmp_snooping vlan default state enable

- Включить фильтрацию многоадресного трафика, чтобы избежать его передачи узлам, не являющимся подписчиками многоадресной рассылки.

config multicast vlan_filtering_mode vlan default filter_unregistered_groups

9.5 Функция IGMP Snooping Fast Leave

Функция IGMP Snooping Fast Leave, активизированная на коммутаторе, позволяет мгновенно исключить порт из таблицы коммутации IGMP Snooping при получении им сообщения о выходе из группы. Это позволяет прекратить передачу по сети ненужных потоков данных и более эффективно использовать полосу пропускания. Функция IGMP Snooping Fast Leave полезна в приложениях IPTV, т.к. благодаря ней можно уменьшить время отклика, когда пользователи переключаются между телевизионными каналами.

Следует отметить, что порт будет удален из таблицы коммутации IGMP Snooping только в том случае, если к нему больше не подключено ни одного узла-подписчика.

9.5.1 Пример настройки IGMP Snooping Fast Leave

Рассмотрим пример настройки функции IGMP Snooping Fast Leave. На рис. 9.7 приведена схема сети, в которой на коммутаторе 2, активизирована функция IGMP Snooping Fast Leave. Все порты коммутатора 2 находятся в VLAN по умолчанию (Default VLAN). К одному из портов коммутатора подключен узел-подписчик многоадресной рассылки.

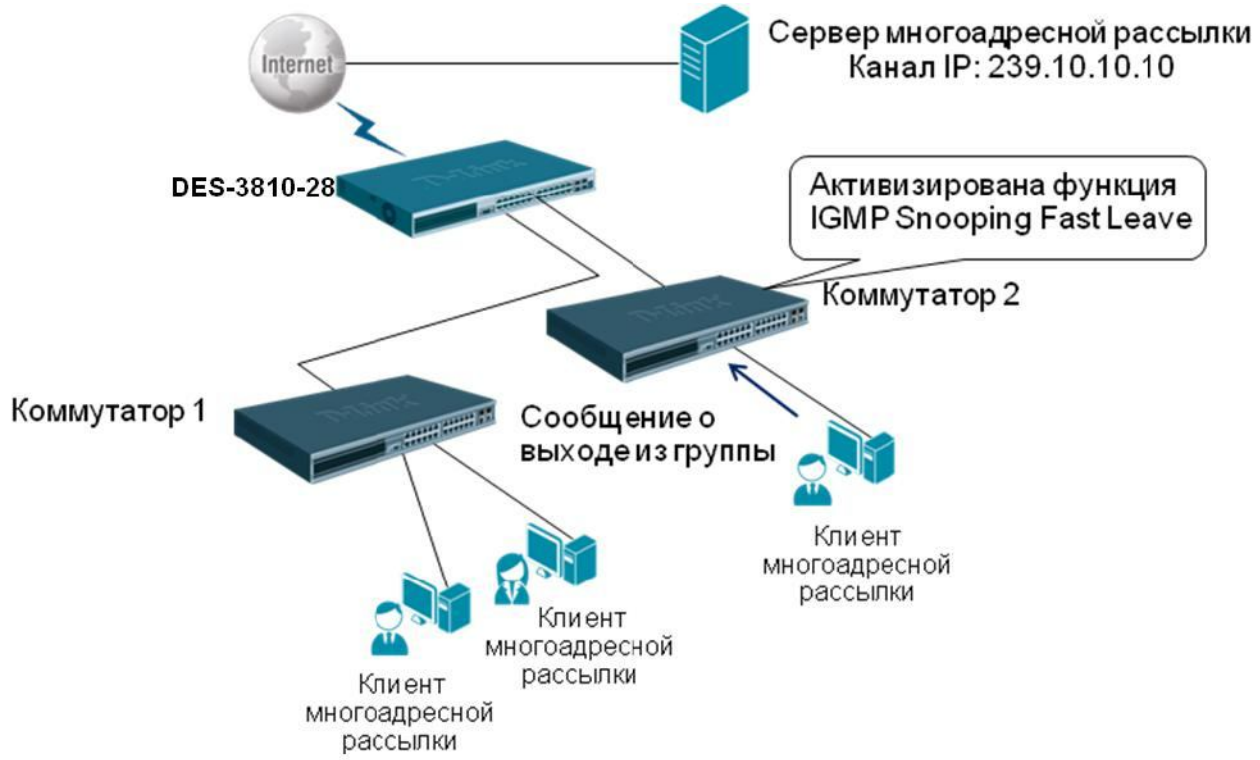


Рис. 9.7. Схема сети

Настройка коммутатора 2

- Активизировать функцию IGMP Snooping глобально на коммутаторе и в указанной VLAN (в данном примере VLAN по умолчанию). Включить фильтрацию многоадресного трафика.

enable igmp_snooping
config igmp_snooping vlan default state enable
config multicast vlan_filtering_mode vlan default filter_unregistered_groups

- Активизировать функцию IGMP Snooping Fast Leave в указанной VLAN.

config igmp_snooping vlan default fast_leave enable

10. Функции управления коммутаторами

10.1 Управление множеством коммутаторов

Независимое управление множеством коммутаторов требует выделения каждому устройству отдельного IP-адреса, что ведет к неэкономному использованию адресного пространства и необходимости запоминания администратором сети IP-адреса каждого коммутатора. D-Link предлагает два подхода к управлению множеством коммутаторов:

- физическое стекирование коммутаторов;
- виртуальное стекирование коммутаторов.

Оба эти подхода предполагают объединение коммутаторов в физическую или логическую группу, которая будет управляться через единый IP-адрес.

10.1.1 Объединение коммутаторов в физический стек

При физическом стекировании коммутаторы представляют собой одно логическое устройство, что обеспечивает удобство управления и мониторинга их параметров. Для управления коммутаторами можно использовать интерфейс командной строки (CLI), Web-интерфейс, Telnet, протокол SNMP, и только одному коммутатору (мастеру-коммутатору) потребуется присвоение управляющего IP-адреса.

Передача данных между коммутаторами стека ведется в полнодуплексном режиме. Коммутаторы могут быть объединены в стек либо кольцевой топологии, либо линейной топологии. Одним из преимуществ стека кольцевой топологии над стеком линейной топологии является поддержка технологии определения оптимального пути передачи пакетов. Эта технология позволяет достичь полного использования полосы пропускания и повысить отказоустойчивость стека.

Внимание: технология определения оптимального пути используется для передачи только одноадресных пакетов.

В примере, приведенном на рис. 10.1, показано, что данные от коммутатора 2 передаются не по кругу (через коммутаторы 3, 4, 5 и т.д.), а непосредственно в направлении коммутатора 9 (через коммутаторы 1,12,11,10). При этом следует отметить, что весь трафик в стеке передается одновременно, и локальный трафик не оказывает влияния на трафик, циркулирующий внутри стека (Рис. 10.2).

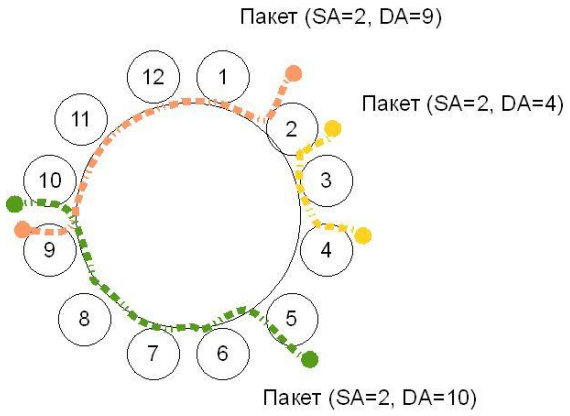


Рис. 10.1. Пример выбора оптимального пути передачи пакета в стеке типа «кольцо»

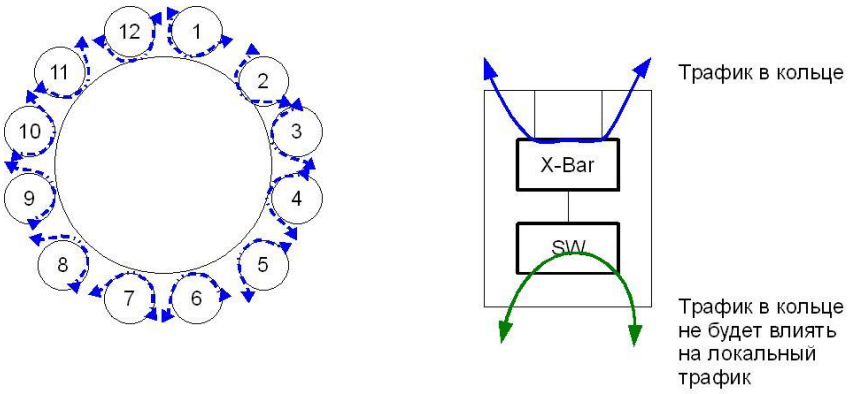


Рис. 10.2. Потоки трафика в стеке

В стеке линейной топологии данные передаются только в одном направлении, и выход из строя какого-либо коммутатора стека повлияет на его работу.

В стекируемых коммутаторах D-Link для повышения отказоустойчивости и производительности стека, реализованы следующие механизмы:

- Механизм *Resilient Master Technology (RMT)* обеспечивает непрерывную работу стека при выходе какого-либо устройства из строя, замене, добавлении и удалении коммутаторов, а также позволяет автоматически назначать нового мастера-коммутатора, в случае неработоспособности текущего, и автоматически восстанавливать работу стека.
- Механизм *Cross Device Trunking (CDT)* позволяет объединять несколько физических портов разных коммутаторов стека в один агрегированный канал с повышенной полосой пропускания. При этом такая логическая магистраль будет продолжать функционирование, даже если какой-либо порт или коммутатор выйдут из строя.
- Технология *SmartRoute* позволяет копировать таблицы коммутации 3-го уровня, хранимые на мастере-коммутаторе, на все другие устройства стека (в том случае, если стек построен на коммутаторах L3). Благодаря этому, каждый коммутатор стека может маршрутизировать трафик локально, не пересылая его на мастер-коммутатор, что уменьшает потребление полосы пропускания между коммутаторами и повышает отказоустойчивость стека.

Роли коммутаторов стека

Каждому коммутатору стека присваивается определенная роль. Эти роли могут быть вручную настроены администратором сети на каждом коммутаторе или определены стеком автоматически. Существуют 3 роли, которые могут быть назначены коммутаторам стека.

Основной мастер (*Primary Master*) – основной мастер-коммутатор является ведущим устройством стека и единой точкой управления. Он следит за нормальной работой стека, топологией, назначает идентификаторы устройствам стека (Box ID), синхронизирует конфигурации и передает команды другим коммутаторам. Роль основного мастера может быть присвоена коммутатору вручную, путем назначения наивысшего приоритета администратором сети или определена автоматически в процессе выборов.

Резервный мастер (*Backup Master*) – резервный мастер дублирует основной мастер-коммутатор и в случае его выхода из строя, берет на себя функции основного мастера. Резервный мастер-коммутатор следит за состоянием соседних коммутаторов стека, основного мастера-коммутатора и выполняет его команды. Роль резервного мастера может быть назначена коммутатору вручную, путем присвоения ему второго по значению наивысшего приоритета до физического объединения устройств в стек или автоматически во время выборов.

Ведомый (*Slave*) – ведомыми являются все остальные коммутаторы стека. Ведомые коммутаторы выполняют операции, требуемые основным мастером, следят за состоянием соседних коммутаторов стека и топологией, следуют командам резервного мастера, когда он становится основным. Ведомые коммутаторы принимают участие в процессе выбора нового резервного мастера, в случае если:

- резервный мастер стал основным мастером;
- резервный мастер вышел из строя или удален из стека;
- оба, и основной, и резервный мастер вышли из строя или удалены из стека.

Выборы основного мастера-коммутатора стека

После того как коммутаторы объединены в стек, каждое устройство начнет собирать информацию (такую как приоритет, MAC-адрес) о соседних коммутаторах и сохранять ее во временной базе данных топологии стека (self temp stacking topology database). Далее коммутаторы приступят к выбору основного мастера-коммутатора стека. Основной мастер выбирается путем сравнения приоритетов (по умолчанию приоритет 32) и MAC-адресов коммутаторов стека.

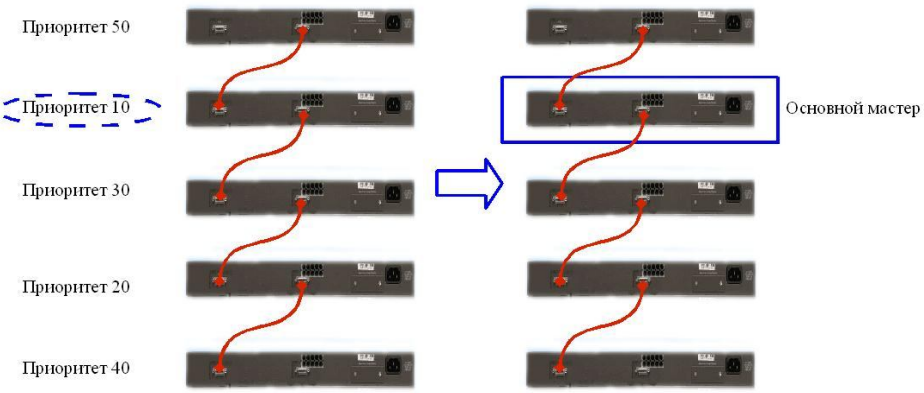


Рис. 10.3. Процесс выбора основного мастера на основании приоритетов

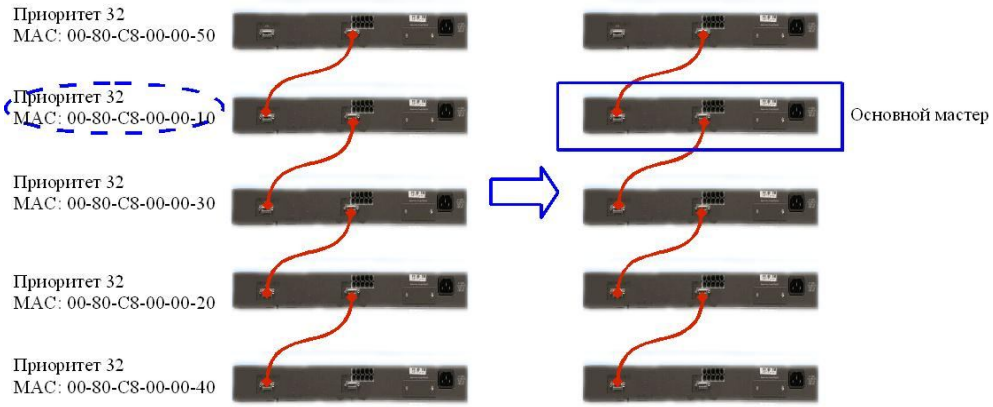


Рис. 10.4. Процесс выбора основного мастера на основании MAC-адресов при равном значении приоритетов

Основным мастером становится коммутатор с наименьшим значением приоритета. Если приоритеты коммутаторов равны, то будет выбран коммутатор с наименьшим значением MAC-адреса.

После выбора основного мастера, начинается процесс выбора резервного мастера из оставшихся устройств стека по аналогичному сценарию.

Как только выбраны основной и резервный мастер, всем коммутаторам стека будут присвоены порядковые номера Box ID. Если в настройках коммутатора параметр Box ID установлен в Auto, основной мастер назначит каждому устройству стека порядковый номер в

соответствии с правилами автоматического назначения номеров (основному мастеру в автоматическом режиме присваивается номер 1). Эта информация о топологии будет разослана всем устройствам стека.

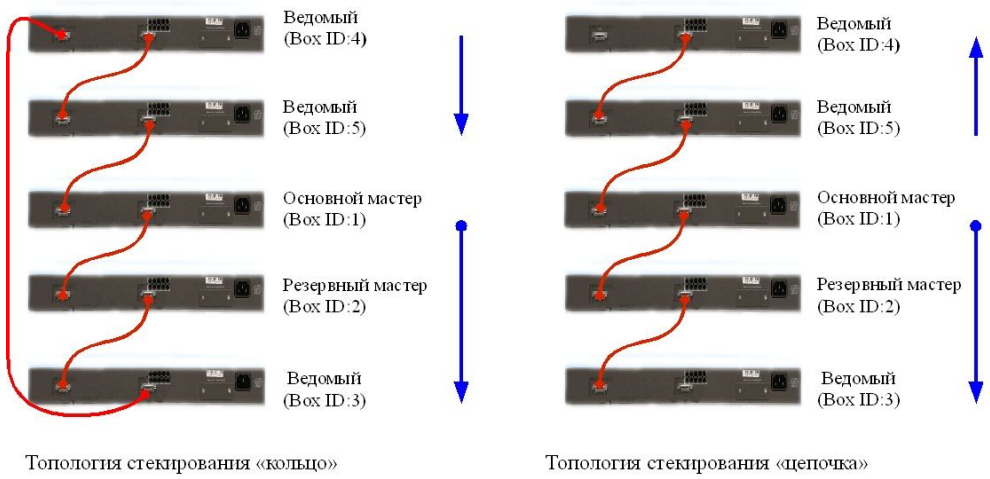


Рис. 10.5. Назначение порядковых номеров в автоматическом режиме

Когда значение параметра Box ID установлено в статический режим (STATIC), присвоение порядковых номеров коммутаторам будет осуществляться вручную администратором сети. При этом настройки любого коммутатора с новым, отличным от предыдущего, значением Box ID, будут возвращены к заводским настройкам по умолчанию. Если в процессе изучения топологии стека возник конфликт идентификационных номеров коммутаторов, устройства с одинаковыми Box ID переводятся в автономный режим работы. Все порты для стекирования отключаются, и система выдает сообщение о конфликте Box ID.

Следует отметить, что основной мастер-коммутатор и резервный мастер-коммутатор не имеют фиксированных номеров, т.е. основному мастеру может быть присвоен Box ID отличный от 1.

При работе в смешанном режиме, когда на коммутаторах используется автоматическое и статическое назначение номеров, действует следующее правило.

1) Сначала основной мастер собирает информацию о Box ID всех коммутаторов стека.
2) Выполняется проверка на наличие конфликтов Box ID.
3) Если конфликт не обнаружен, то основной мастер сначала присвоит идентификационные номера устройствам, работающим в статическом режиме, а затем устройствам, работающим в автоматическим режиме.

Изменение топологии стека

В случае добавления или удаления коммутатора(ов) из стека или изменении топологии, протокол стекирования, реализованный в устройствах, быстро обнаружит изменения и синхронизирует информацию о новой топологии стека.

Когда новый коммутатор добавляется в работающий стек, ему присваивается роль ведомого или резервного мастера, в зависимости от значений приоритета и MAC-адреса.

Если в стек добавляется только одно устройство, то процесс выбора основного мастера не запускается.

При удалении одного или нескольких коммутаторов из работающего стека, оставшиеся коммутаторы удаляют информацию о выбывших устройствах из своих баз данных топологии стека.

При удалении резервного мастера запускается процесс выборов нового резервного мастера. Этот процесс также запускается при удалении или выходе из строя основного мастера, т.к. резервный мастер становится в этом случае основным. При этом для

минимизации нарушения работы сети, новому основному мастеру присваивается тот же IP-адрес, который был у предыдущего (Box ID не меняется).

При удалении обоих мастеров (основного и резервного) мгновенно инициируется процесс выбора нового основного и резервного мастеров из оставшихся коммутаторов стека.

Когда происходит изменение топологии стека – с линейной на кольцевую или наоборот, состояние устройств не изменяется.

Пример настройки стекирования

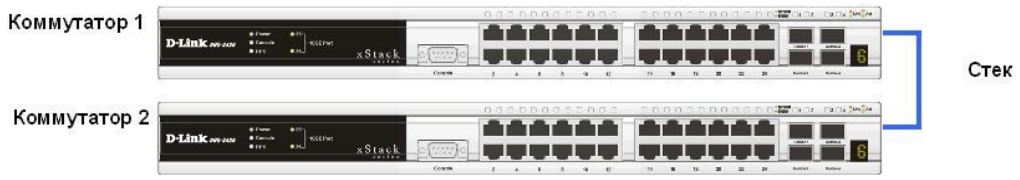


Рис. 10.6. Коммутаторы, объединенные в стек

В качестве примера приведем настройку двух коммутаторов, объединенных в физический стек.

Настройка коммутатора 1
config stacking mode enable
config box_priority current_box_id 1 priority 1

Настройка коммутатора 2
config stacking mode enable
config box_priority current_box_id 1 new_box_id 2
config box_priority current_box_id 2 priority 2

10.1.2 Виртуальный стек. Технология Single IP Management (SIM)

Технология **Single IP Management (SIM)** является простым и удобным способом сетевого управления. Она разработана для управления группой коммутаторов, называемых *SIM-группой*, как единым устройством. При этом для управления SIM-группой требуется только один IP-адрес, который назначается выделенному коммутатору группы (Commander switch).

Технология SIM позволяет:

- Устранить ограничения на модели коммутаторов, объединяемых в стек.
- Уменьшить количество управляющих IP-адресов в сети.
- Устранить необходимость использования специализированных модулей и кабелей, предназначенных для стекирования; преодолеть ограничения, связанные с длиной кабелей в стеке.

В отличие от стеков, построенных с использованием традиционных методов стекирования, виртуальный стек на основе технологии SIM не ограничивается 6-ю или 12-ю коммутаторами. В SIM-группу может входить до 32-х коммутаторов любых моделей, поддерживающих функции Single IP Management. Это означает, что виртуальный стек может включать коммутаторы разного типа, от недорогих коммутаторов 2-го уровня до высокопроизводительных коммутаторов на основе шасси (для ядра сети).

Объединение коммутаторов в SIM-группу не требует использования специальных соединительных кабелей. Трафик, передаваемый между устройствами стека, проходит через интерфейсы Fast Ethernet, Gigabit Ethernet или 10 Gigabit Ethernet по обычным медным или оптическим кабелям. Отказ от использования специализированных стекирующих кабелей позволяет преодолеть ограничения, связанные с их длиной. Устройства SIM-группы могут

быть подключены друг к другу через промежуточные устройства, не поддерживающие технологию SIM. Объединение коммутаторов в SIM-группу не влияет на их нормальное функционирование.

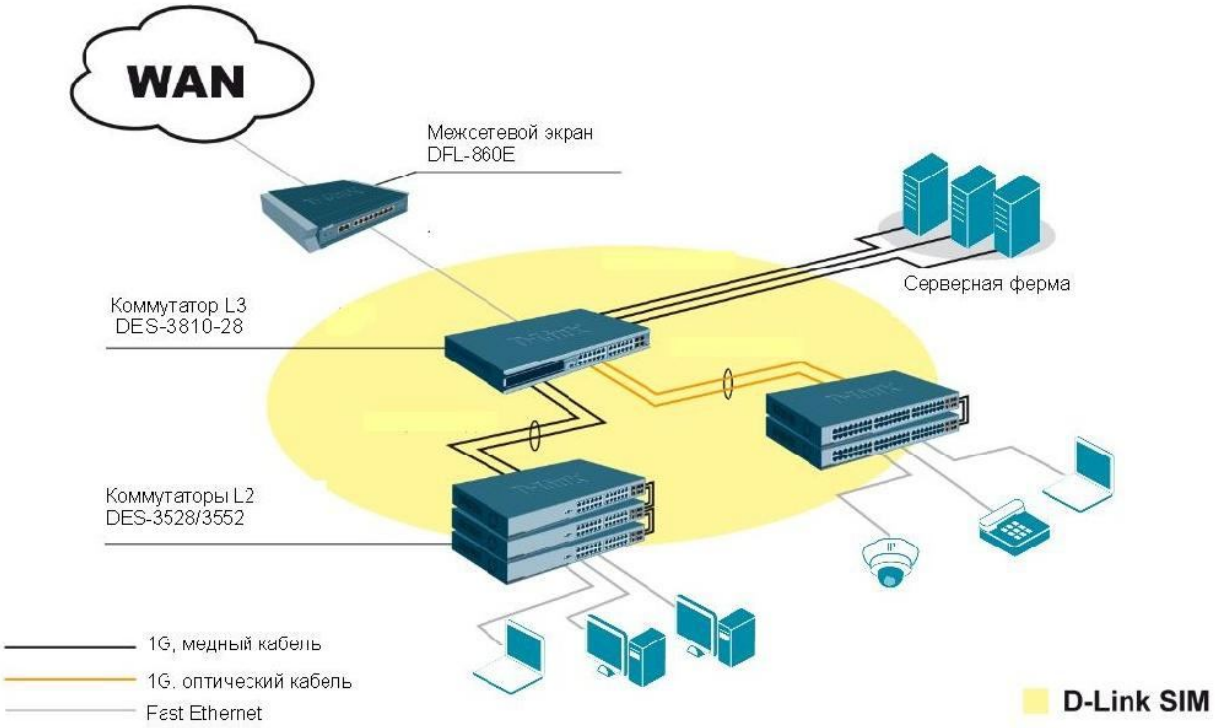


Рис. 10.7. Технология SIM

Внимание: SIM является дополнительной функцией коммутаторов и может быть активизирована или отключена через Web-интерфейс или CLI. По умолчанию эта функция на коммутаторе отключена.

Технология SIM предусматривает три роли, которые могут быть назначены коммутаторам группы:

1) **Commander Switch (CS)** – это коммутатор, который вручную настраивается администратором сети как управляющее устройство SIM-группы. В SIM-группе может быть только один Commander Switch. CS обладает следующими характеристиками:
   - ему присвоен IP-адрес;
   - он не является Commander Switch или Member Switch другой SIM-группы;
   - он подключен к коммутаторам Member Switch через управляющую VLAN (VLAN, к которой привязан управляющий интерфейс коммутатора System. По умолчанию управляющей VLAN является default VLAN).
2) **Member Switch (MS)** – коммутатор, который вступил в SIM-группу и доступен через Commander Switch. Member Switch обладает следующими характеристиками:
   - он не является Commander Switch или Member Switch другой SIM-группы;
   - он подключен к Commander Switch через управляющую VLAN.
3) **Candidate Switch (CaS)** – это коммутатор, который готов вступить в SIM-группу (стать Member Switch) используя либо автоматический метод, либо ручную настройку. Коммутатор, настроенный как Candidate Switch не является членом SIM-группы. Он обладает следующими характеристиками:
   - он не является Commander Switch или Member Switch другой SIM-группы;

- он подключен к Commander Switch через управляющую VLAN.

Все коммутаторы, объединенные в одну SIM-группу должны принадлежать одной IP-подсети (широковещательному домену). В пределах одной подсети может быть создано несколько SIM-групп, но при этом каждый коммутатор должен принадлежать только одной SIM-группе. Каждая группа может содержать до 32 коммутаторов (от 0 до 31), включая Commander Switch (его номер 0). Если в сети настроено несколько VLAN, SIM-группа будет использовать только управляющую VLAN любого коммутатора.

По умолчанию, после активизации функции SIM всем коммутаторам присваивается роль «Candidate Switch». Это означает, что они смогут стать членами SIM-группы (Member Switch), как только получат запрос от Commander Switch.

После того, как одному из коммутаторов группы была присвоена роль «Commander Switch», он начинает формировать SIM-группу, добавляя в нее новых членов. Для этого Commander Switch просматривает список кандидатов (Candidate Switch) и отправляет им периодические запросы. Кандидат отправляет Commander Switch ответ, содержащий информацию о нем, что позволяет ему стать членом SIM-группы. Если коммутатор-кандидат имел ранее сконфигурированный пароль, он не сможет стать членом группы до тех пор, пока не будут введены его аутентификационные данные.

Можно добавлять членов группы, используя Web-интерфейс. Функционал SIM встроен в Web-интерфейс управления коммутаторов и не требует установки дополнительного ПО.

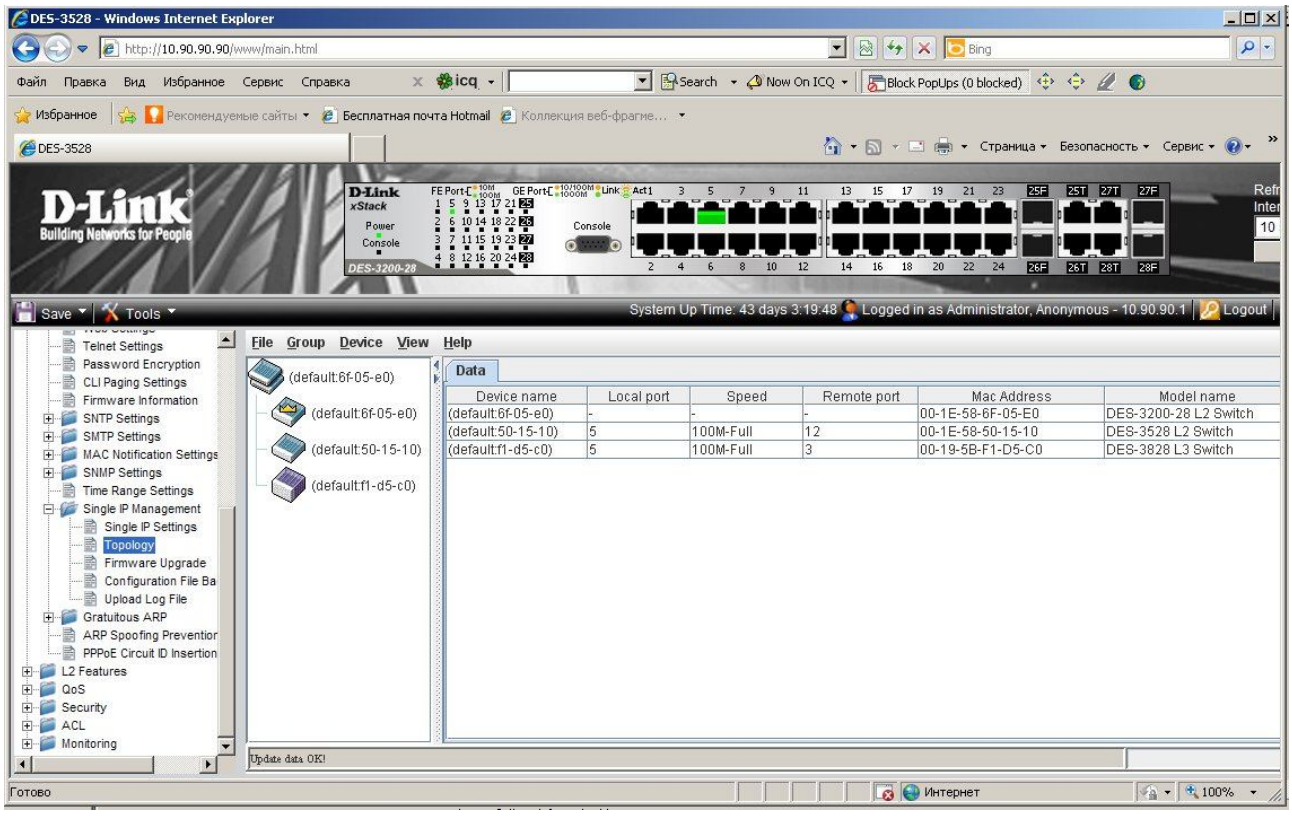


Рис. 10.8. Функция SIM в Web-интерфейсе

После настройки Commander Switch в папке Single IP Management Web-интерфейса станет доступна опция Topology. Эта опция позволяет настраивать коммутаторы и управлять ими внутри SIM-группы. При выборе пункта View окна Topology появится топологическая карта, показывающая, как подключены устройства внутри SIM-группы. Топологическая карта автоматически обновляется каждые 20 секунд.

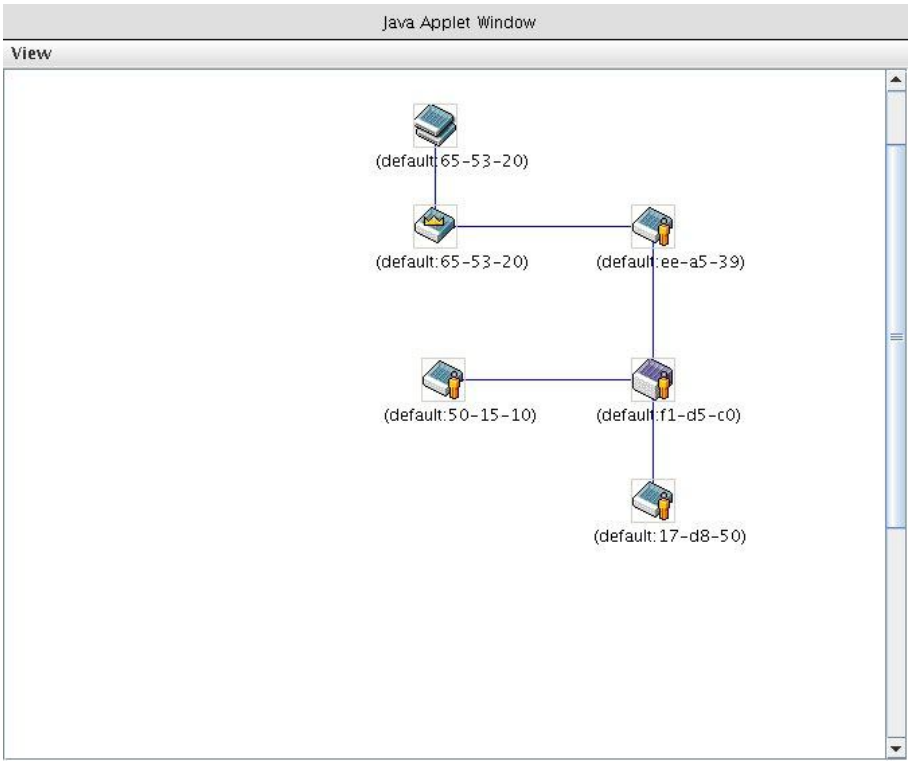


Рис. 10.9. Топологическая карта

Используя топологическую карту, администратор может получать детальную информацию о группе, просматривать краткую информацию о каждом коммутаторе SIM-группы, настраивать его, добавлять и удалять устройства из SIM-группы.

В топологической карте используются следующие иконки для обозначения Commander Switch, Member Switch и Candidate Switch.

Рис. 10.10. Иконки, используемые для представления устройств в топологической карте

10.2 Протокол SNMP

Протокол SNMP (Simple Network Management Protocol) является протоколом 7 уровня модели OSI, который специально разработан для управления и мониторинга сетевых устройств. Протокол SNMP входит в стек протоколов TCP/IP и позволяет администраторам сетей получать информацию о состоянии устройств сети, обнаруживать и исправлять неисправности и планировать развитие сети.

В настоящее время существует три версии протокола SNMP: SNMPv1 (RFC 1157), SNMPv2c (RFC 1901-1908) и SNMP v3 (RFC 3411-3418). Эти версии отличаются предоставляемым уровнем безопасности при обмене данными между менеджером и агентом SNMP. Коммутаторы D-Link поддерживают все три версии протокола.

10.2.1 Компоненты SNMP

Сеть, управляемая по протоколу SNMP, основывается на архитектуре «клиент/сервер» и состоит из трех основных компонентов: менеджера SNMP, агента SNMP, базы управляющей информации.

Менеджер SNMP (SNMP Manager) – это программное обеспечение, установленное на рабочей станции управления, наблюдающее за сетевыми устройствами и управляющее ими.

Агент SNMP (SNMP Agent) – это программный модуль для управления сетью, который находится на управляемом сетевом устройстве (маршрутизаторе, коммутаторе, точке доступа, Интернет-шлюзе, принтере и т.д.). Агент обслуживает базу управляющей информации и отвечает на запросы менеджера SNMP.

База управляющей информации (Management Information Base, MIB) – это совокупность иерархически организованной информации, доступ к которой осуществляется посредством протокола управления сетью.

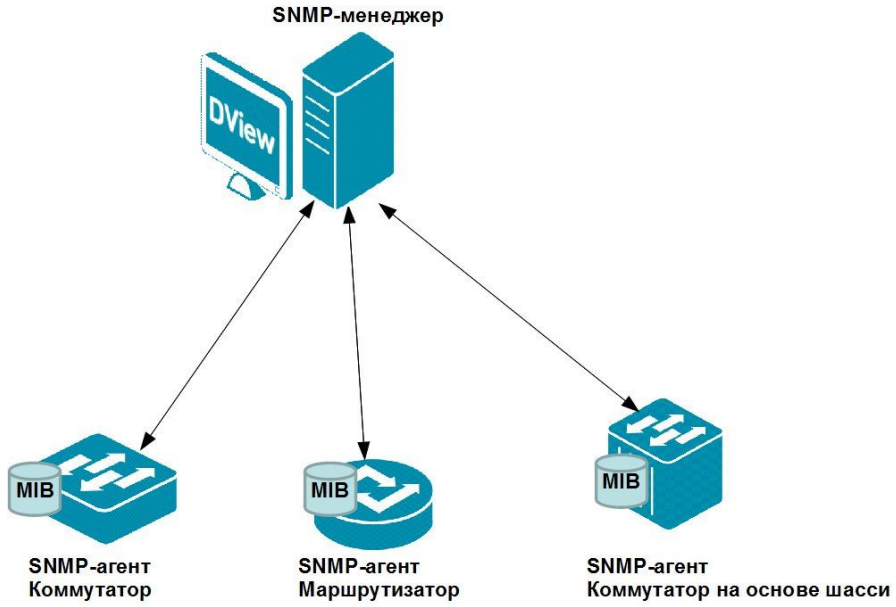


Рис. 10.11. Компоненты SNMP

Менеджер взаимодействует с агентами при помощи протокола SNMP с целью обмена управляющей информацией. В основном, это взаимодействие реализуется в виде периодического опроса менеджером множества агентов, которые предоставляют доступ к информации.

10.2.2 База управляющей информации SNMP

Базы управляющей информации описывают структуру управляющей информации устройств и состоят из управляемых объектов (переменных). *Управляемый объект* (или MIB-объект) – это одна из нескольких характеристик управляемого сетевого устройства (например, имя системы, время, прошедшее с ее перезапуска, количество интерфейсов устройства, IP-адрес и т.д.).

Обращение к управляемым объектам MIB происходит посредством идентификаторов объекта (Object IDentifier, OID). Каждый управляемый объект имеет уникальный идентификатор в пространстве имен OID и контролируется агентством IANA. Пространство имен OID можно представить в виде иерархической структуры с корнем без названия, идентификаторы верхних уровней которой отданы организациям, контролирующим стандартизацию, а идентификаторы нижних уровней определяются самими этими организациями. Каждая ветвь дерева OID нумеруется целыми числами слева направо, начиная с единицы. Идентификатор представляет собой последовательность целых десятичных цифр, разделенных точкой, записанных слева направо и включает полный путь от корня до управляемого объекта. Числам могут быть поставлены в соответствие текстовые строки для удобства восприятия. В целом, структура имени похожа на систему доменных имен Интернет (Domain Name System, DNS).

Каждая MIB (в настоящее время основными стандартами на базы управляющей информации для протокола SNMP являются MIB-I и MIB-II) определяет набор переменных, т. е. определенную ветку дерева OID, описывающую управляющую информацию в определенной области. Например, ветка 1.3.6.1.2.1.1 (символьное эквивалентное имя: iso.org.dod.internet.mgmt.mib-2.system) описывает общую информацию о системе.

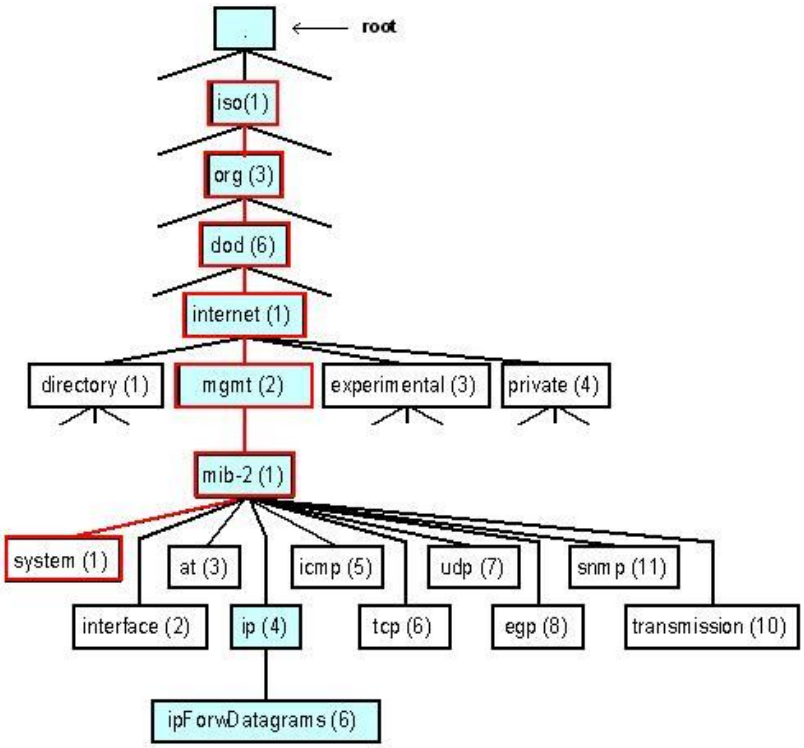


Рис. 10.12. Пространство имен OID

Производители сетевого оборудования определяют частные ветви пространства имен OID (группа объектов private (4)), куда помещают управляемые объекты для своей продукции. Так D-Link в качестве вершины иерархии для своей продукции использует OID равный 1.3.6.1.4.1.171.

Помимо непосредственного описания данных, необходимо вести операции над ними. Первоначальная спецификация MIB-I определяла только операции чтения значений переменных. Спецификация MIB-II дополнительно определяет операции изменения или установки значений управляемых объектов.

10.2.3 Типы сообщений протокола SNMP

Протокол SNMP является простым протоколом типа «запрос – ответ», т.е. на каждый запрос менеджера, агент должен дать ответ. В протоколе определено несколько типов сообщений, которыми обмениваются менеджер и агент:

- **Get-Request** – запрос значений одного или нескольких объектов;
- **Get-Next-Request** – запрос значения следующего объекта, в соответствии с алфавитным порядком идентификаторов OID;
- **Set-Request** – запрос на изменение значения одного или нескольких объектов;
- **Get(Set)-Reply** – получение ответа от агента на сообщение Get-Request, Get-Next-Request или Set-Request.

Сообщение **Trap** (ловушка) используется агентом SNMP для асинхронного сообщения менеджеру SNMP о событии, происходящем на управляемом сетевом устройстве. События могут быть серьезные, например, перезагрузка устройства или менее серьезные, например, изменение состояния порта.

Версия SNMP v.2 добавляет к этому набору команду GetBulk, которая позволяет менеджеру получить несколько переменных за один запрос.

При передаче управляющих сообщений, в качестве протокола транспортного уровня используется протокол UDP.

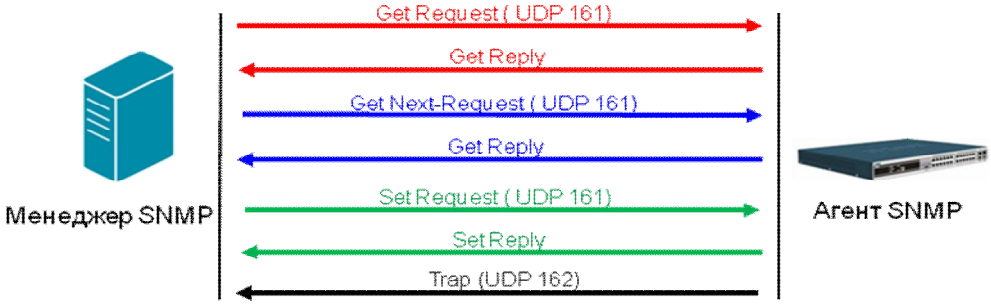


Рис. 10.13. Типы сообщения протокола SNMP

10.2.4 Безопасность SNMP

В протоколе SNMP v.1 и v.2c предусмотрена аутентификация пользователей, которая выполняется с помощью строки сообщества (Community string). Строки Community string функционирует подобно паролю, который разрешает удаленному менеджеру SNMP доступ к агенту. Менеджер и агент SNMP должны использовать одинаковые строки Community string, т.к. все пакеты от менеджера SNMP не прошедшего аутентификацию будут отбрасываться. В коммутаторах с поддержкой SNMP v.1 и v.2c используются следующие Community string по умолчанию:

- **public** – позволить авторизованной рабочей станции читать (право «read only») MIB-объекты;
- **private** – позволить авторизованной рабочей станции читать и изменять (право «read/write») MIB-объекты.

Внимание: Community string передаются по сети в открытом виде.

Протокол SNMP v.3 использует более сложный процесс аутентификации, который разделен на две части. Первая часть – обработка списка и атрибутов пользователей, которым

позволено функционировать в качестве менеджера SNMP. Вторая часть описывает действия, которые каждый пользователь из списка может выполнять в качестве менеджера SNMP.

На коммутаторе SNMP можно создавать группы со списками пользователей (менеджеров SNMP) и настраивать для них общий набор привилегий. Помимо этого для каждой группы может быть установлена версия используемого ей протокола SNMP. Таким образом, можно создать группу менеджеров SNMP, которым позволено просматривать информацию с правом «read only» или получать ловушки (trap), используя SNMP v.1, в то время как другой группе можно настроить наивысший уровень привилегий с правами «read/write» и возможность использования протокола SNMP v.3.

При использовании протокола SNMP v.3, отдельным пользователям или группам менеджеров SNMP может быть разрешено или запрещено выполнять определенные функции SNMP-управления. Помимо этого в SNMP v.3 доступен дополнительный уровень безопасности, при котором SNMP-сообщения могут шифроваться при передаче по сети.

10.2.5 Пример настройки протокола SNMP

На рис. 10.14 показана схема сети, в которой управление коммутатором может выполняться через управляющую консоль SNMP по протоколу SNMP v.2. В случае обнаружения SNMP-агентом коммутатора каких-либо неполадок, он будет отправлять сообщения Trap менеджеру SNMP. В целях повышения безопасности строки Community string по умолчанию удалены.

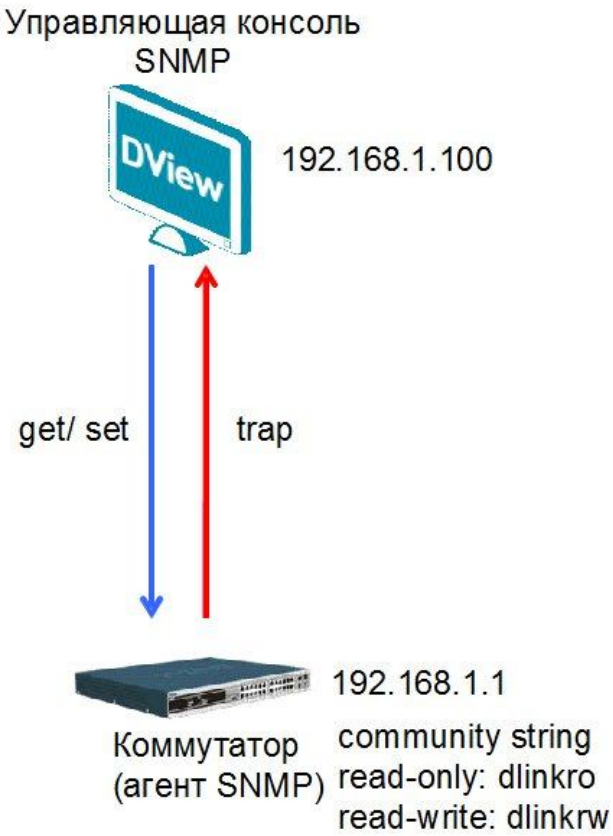


Рис. 10.14. Схема сети

Настройка коммутатора

• Активизировать функцию SNMP глобально на коммутаторе.

enable snmp

• Удалить строки Community string по умолчанию и создать новые строки Community string.

delete snmp community public
delete snmp community private
create snmp community dlinkro view CommunityView read_only
create snmp community dlinkrw view CommunityView read_write

• Задать параметры получателя сообщений Trap от агента и активизировать функцию отправки сообщений Trap.

create snmp host 192.168.1.100 v2c dlinkrw
enable snmp traps
enable snmp authenticate_traps

10.3 RMON (Remote Monitoring)

Спецификация *RMON* MIB (Remote MONitoring, удаленный мониторинг) была разработана сообществом IETF для поддержки мониторинга и анализа протоколов в локальных сетях. Первая версия RMON v.1 (RFC 2819) основывается на мониторинге информации сетей Ethernet и Token Ring. Ее расширением является RMON v.2 (RFC 2021), которая добавила к уже имеющимся средствам мониторинга, поддержку мониторинга на сетевом уровне и уровне приложений модели OSI.

Реализация RMON основывается на модели клиент/сервер. На устройствах мониторинга, называемых в терминологии RMON «зондами» (probe), установлено специальное программное обеспечение – агент RMON, которое собирает информацию и анализирует пакеты. Зонды действуют как серверы, а приложения сетевого управления, установленные на станциях управления сетью, исполняют роль клиентов. Агенты RMON могут размещаться как на автономных устройствах, так и встраиваться в коммутаторы, маршрутизаторы и другие сетевые устройства. Станция управления сетью и распределенные зонды RMON взаимодействуют по сети по протоколу SNMP.

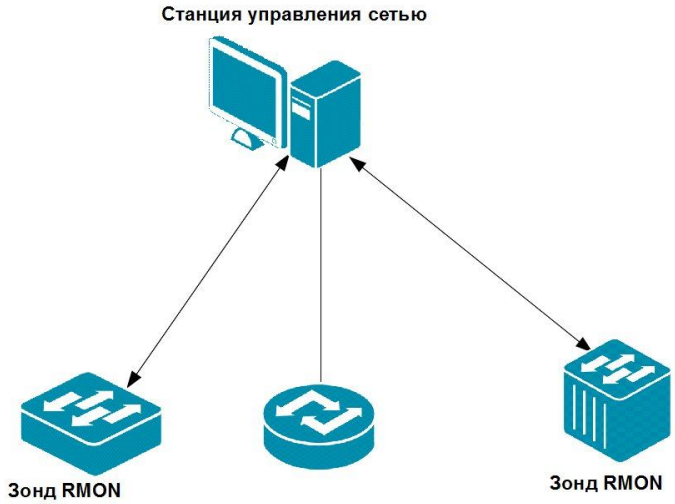


Рис. 10.15. Консоль и зонд RMON

Несмотря на то, что RMON является расширением протокола SNMP, он отличается от него тем, что зонды RMON могут самостоятельно выполнять сбор и обработку данных. Это позволяет сократить трафик SNMP в сети и нагрузку на станцию управления, причем информация будет передаваться на станцию, только когда это необходимо. Расположенные в различных частях сети приложения RMON могут одновременно взаимодействовать и получать информацию от одного и того же зонда.

RMON поставляет информацию в группы RMON MIB, каждая из которых поддерживает определенный набор данных, удовлетворяющих общим требованиям

мониторинга сети. RMON v.1 содержит десять групп RMON MIB, а RMON v.2 добавляет к ним еще девять групп RMON MIB. Вместе RMON v.1 и RMON v.2 позволяют собирать статистику о трафике на всех уровнях модели OSI. Из-за того, что при выполнении обработки данных на ресурсы устройств мониторинга ложится большая нагрузка, то производители реализуют на оборудовании ограниченный набор групп RMON MIB. Обычно агент RMON поддерживает только группы statistics, history, alarm и event.

Таблица 11 **Группы мониторинга RMON v.1**

№	Группа RMON	Функция
1	Statistics	Содержит статистические данные, измеренные датчиком на каждом интерфейсе устройства, для которого проводится мониторинг.
2	History	Периодическая запись статистических выборок из сети и их хранение для дальнейшего использования.
3	Alarm	Периодическое извлечение статистических выборок из переменных в датчике и их сравнение с заранее выбранными пороговыми значениями. Если наблюдаемые значения выходят за границы пороговых, генерируется событие.
4	Host	Содержит статистические данные, связанные с каждым узлом, обнаруженным в сети.
5	Hosts top N	Содержит отсортированные данные по указанному числу узлов в порядке убывания их статистики.
6	Matrix	Содержит статистику по диалогам между парами узлов, в том числе о величине трафика и количестве ошибок в обоих направлениях.
7	Filter	Содержит условия фильтрации пакетов.
8	Capture	Содержит пакеты, захваченные интерфейсом в соответствии с условиями фильтрации.
9	Event	Протоколирование событий и определение действий при их наступлении.
10	Token Ring	Расширенная статистика для сетей Token Ring.

Таблица 12 **Группы мониторинга RMON v.2**

№	Группа RMON	Функция
1	Protocol Directory	Список протоколов, для которых зонд может осуществлять мониторинг пакетов.
2	Protocol Distribution	Статистика трафика для каждого протокола с информацией о распределении и тенденциях в использовании протоколов.
3	Address Map	Соответствия между адресами сетевого уровня и MAC-адресами, обнаруженные зондом на интерфейсе.
4	Network Layer Host	Статистика трафика передаваемого от и к каждому обнаруженному зондом узлу.
5	Network Layer Matrix	Содержит статистику по диалогам между парами узлов на сетевом уровне.
6	Application Layer Host	Статистика трафика передаваемого от и к каждому обнаруженному зондом узлу по протоколам.
7	Application Layer Matrix	Статистика трафика по диалогам между парами узлов на сетевом уровне по протоколам.
8	User History	Периодические выборки для определенных пользователем переменных.
9	Probe Configuration	Удаленная конфигурация параметров зонда.

Для настройки RMON на коммутаторах D-Link второго уровня требуется активизировать эту функцию глобально на коммутаторе с помощью команды *enable rmon*.

При настройке функции на коммутаторе третьего уровня, необходимо дополнительно активизировать протокол SNMP и определить Community string.

10.4 Функция Port Mirroring

Функция *Port Mirroring* (Зеркалирование портов) позволяет отображать (копировать) кадры, принимаемые и отправляемые портом-источником (Source port) на целевой порт (Target port) коммутатора, к которому подключено устройство мониторинга с целью анализа проходящих через интересующий порт пакетов. Эта функция полезна администраторам для мониторинга и поиска неисправностей в сети.

Следует отметить, что целевой порт и порт-источник должны принадлежать одной VLAN и иметь одинаковую скорость работы. В том случае, если скорость порта-источника будет выше скорости целевого порта, то коммутатор снизит скорость порта-источника до скорости работы целевого порта.

На рис. 10.16 показан пример, в котором трафик, передаваемый и получаемый портами 2 и 4, будет зеркалироваться на порт 1, к которому подключено устройство мониторинга.

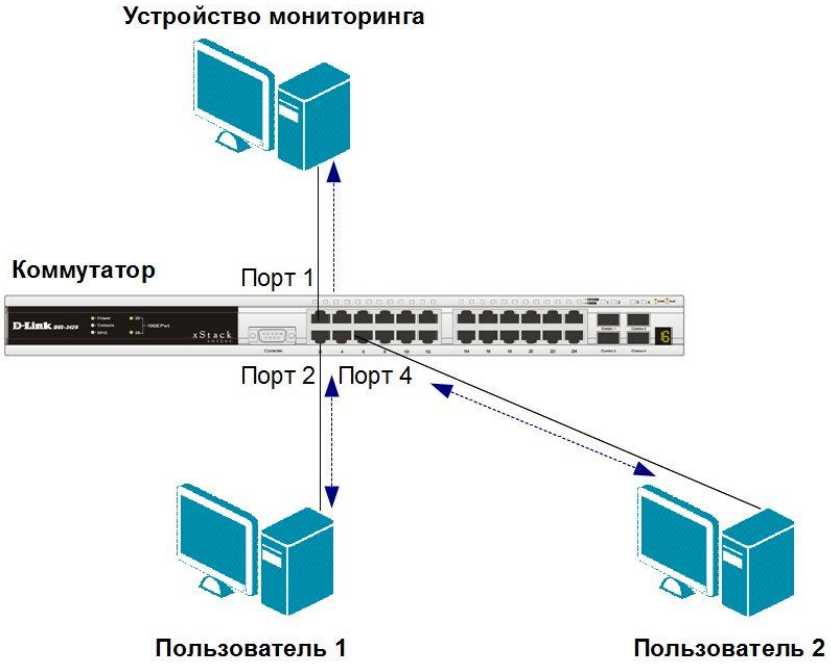


Рис. 10.16. Функция Port Mirroring

Настройка коммутатора
config mirror port 1 add source ports 2, 4 both
enable mirror

11. Обзор коммутаторов D-Link

Исходя из решаемой задачи и также учитывая размер сети, объем трафика и требуемый функционал, можно подобрать требуемые коммутаторы D-Link. Производимые D-Link устройства можно классифицировать по принадлежности к трем уровням иерархической модели сети. Это помогает пользователям определить, какое оборудование оптимально использовать для решения поставленной задачи в конкретной сети.

11.1 Неуправляемые коммутаторы

Неуправляемые коммутаторы (*Unmanaged Switches*) D-Link являются идеальным решением для развертывания сетей небольших рабочих групп или домашних сетей (SOHO, Small-Office-Home-Office). Также их можно использовать на уровне доступа сетей малых предприятий. Эти коммутаторы просты в установке и поддерживают, в зависимости от модели, такие функции как Green Ethernet, диагностика кабеля, управление потоком (IEEE 802.3x), автоматическое определение полярности кабелей (MDI/MDIX), возможность передачи Jumbo-фреймов и приоритизацию трафика.

Неуправляемые коммутаторы не поддерживают функции управления и обновления программного обеспечения.

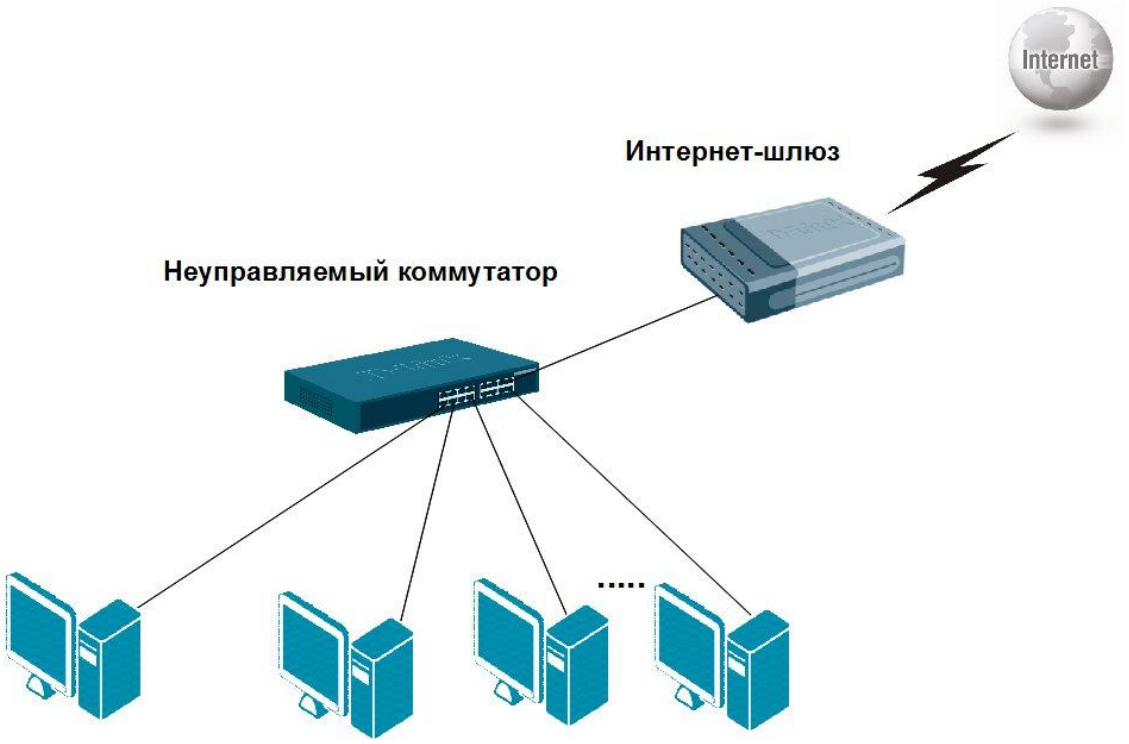


Рис. 11.1. Неуправляемые коммутаторы D-Link в сети небольшой рабочей группы

Неуправляемые коммутаторы D-Link представлены сериями DES-10xx, DES-10xxA, DES-10xxD/RU, DGS-10xxD/RU, DGS-10xxD и DGS-10xxA. Серия DES-10xx включает в себя модели неуправляемых коммутаторов Fast Ethernet с различным количеством портов 10/100 Мбит/с (от 5 до 48) в настольном и стоечном исполнении. Модели DES-1026G и DES-1050G этой серии также оснащены двумя портами Gigabit Ethernet.

Серия DES-10xxA состоит из экономичных неуправляемых коммутаторов с различным количеством портов 10/100 Мбит/с (от 5 до 24).

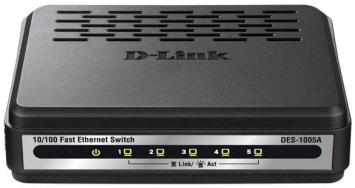


Рис. 11.2. Коммутатор DES-1005A

Серии коммутаторов улучшенного дизайна DES-10xxD/RU и DGS-10xxD/RU включают коммутаторы с 5 и 8 портами Fast и Gigabit Ethernet соответственно.

Серия DGS-10xxA включает в себя модели неуправляемых коммутаторов Gigabit Ethernet с различным количеством портов 10/100/1000 Мбит/с (от 5 до 24). Коммутаторы DGS-1016D и DGS-1024D с 16 и 24 портами 10/100/1000 Мбит/с поддерживают стандарт IEEE 802.1p и четыре аппаратных очереди приоритетов на каждом физическом порте.

Коммутаторы серий DGS-10xxD/RU, DGS-10xxA и DGS-10xxD поддерживают технологию Green Ethernet. Эта энергосберегающая технология позволяет сократить расходы на электроэнергию, при этом, не оказывая влияния на производительность и функциональность устройств. Технология Green Ethernet может регулировать потребление электроэнергии, основываясь на определении состояния канала связи и длины кабеля. Когда коммутатор с поддержкой этой технологии определяет, что питание подключенного к нему компьютера отключено, то переводит соответствующий порт в режим сохранения энергии (power standby mode). Также коммутатор может регулировать энергопотребление путем анализа длины кабеля Ethernet. Т.к. в большинстве случаев для подключения пользователей домашних/офисных сетей используются кабели длиной менее 20 м, энергопотребление может быть снижено.

Благодаря уменьшению энергопотребления (до 80%), выделяется меньше тепла, что увеличивает срок эксплуатации устройства и снижает эксплуатационные расходы.

Также данная технология подразумевает использование материалов, не наносящих вред окружающей среде.

Помимо технологии Green Ethernet, в коммутаторах серий DGS-10xxD/RU и DGS-10xxD реализована поддержка функции диагностики кабеля (Cable Diagnostic). Эта функция позволяет пользователям определять состояние кабеля по индикаторам, расположенным на передней панели коммутатора. С помощью нее можно определить следующие повреждения кабеля:

- разомкнутая цепь (Open Circuit) – оборвана жила кабеля Ethernet или кабель не подключен;
- короткое замыкание (Short Circuit) – короткое замыкание пары кабеля (два проводника касаются друг друга);
- неправильная терминация кабеля (Improper Termination) – сопротивление между кабелем и его разъемом не совпадает или сопротивление больше чем 100 Ом.

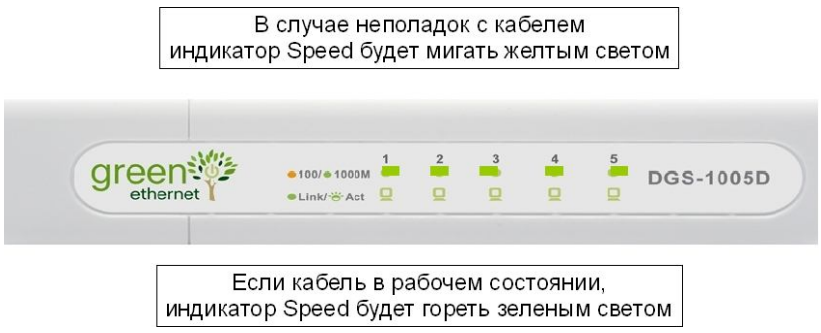


Рис. 11.3. Функция диагностики кабеля

Функция диагностики кабеля сканирует все порты Ethernet и определяет состояние каждого подключенного кабеля. Во время этого процесса индикатор каждого порта последовательно мигает зеленым светом. Первоначальное сканирование порта требует около 10 секунд. Если обнаруживается повреждение кабеля, индикатор соответствующего порта будет мигать желтым светом около 5 секунд. Далее коммутатор автоматически перезагрузится и продолжит работу в обычном режиме. Этот процесс займет около 2-х секунд.

Внимание: для неуправляемых коммутаторов повреждение кабеля не определяется в режиме обычной работы. Диагностика кабеля осуществляется только во время загрузки или цикла выключения/включения коммутатора.

11.2 Коммутаторы серии Smart

Традиционно существуют только 2 категории коммутаторов: неуправляемые и управляемые. Однако D-Link предлагает еще одну, промежуточную категорию – **настраиваемые коммутаторы** (*Smart Switches*). Эти коммутаторы предназначены для использования на уровне доступа сетей малых и средних предприятий (*Small-to-Medium Business*, *SMB*).

Настраиваемые коммутаторы D-Link представлены сериями: Easy Smart, Smart и SmartPro.

Серии Easy Smart включает следующие модели коммутаторов: DES-1100-16, DES-1100-24 и DES-1100-26.

DES-1100-16	16 портов 10/100Base-TX
DES-1100-24	24 порта 10/100Base-TX
DES-1100-26	24 порта 10/100Base-TX, 2 комбо-порта 1000Base-T/SFP

Коммутаторы DES-1100-16 и DES-1100-24 помещены в металлический корпус компактного размера (11”), коммутатор DES-1100-26 имеет корпус стандартного размера для установки в 19” телекоммуникационную стойку. Коммутаторы предоставляют пользователям возможность настраивать такие функции как 802.1Q VLAN, приоритизацию 802.1p, IGMP Snooping с помощью интуитивно понятных средств управления, например Web-интерфейса или утилиты SmartConsole. По сравнению с коммутаторами серии Smart, устройства Easy Smart обладают ограниченным функционалом.

Третье поколение коммутаторов серии Smart представлено продуктовыми линейками DES-1210-xx и DGS-1210-xx. Серия DES-1210-xx включает модели DES-1210-08P, DES-1210-10, DES-1210-28, DES-1210-28P и DES-1210-52.

DES-1210-08P	24 портов PoE 10/100Base-TX
DES-1210-10	8 порта 10/100Base-TX, 2 комбо-порта 1000Base-T/SFP
DES-1210-28	24 порта 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-1210-28P	24 порта PoE 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-1210-52	48 портов 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP

Коммутатор DES-1210-08P помещен в металлический компактный корпус размером 7,5”, корпуса остальных устройств данной серии имеют стандартный размер для установки в стойку (19”). В коммутаторах моделей DES-1210-08P и DES-1210-28P реализована поддержка технологии Power over Ethernet (PoE) (стандарты IEEE 802.3af и IEEE 802.3at

(PoE+)), которая позволяет передавать питание по неиспользуемым парам кабеля Ethernet одновременно с передачей данных. Коммутаторами DES-1210-28P поддерживается функция Smart Fan, благодаря которой вентилятор может автоматически изменять скорость работы при определенной температуре, обеспечивая непрерывную, надежную и экологичную работу устройства. Функционал коммутаторов серии DES-1210-xx включает поддержку функции агрегирования каналов, SafeGuard Engine, IGMP Snooping, протоколов STP и RSTP, 802.1Q и 802.1p. Для разертывания системы видеонаблюдения и выделения приложений VoIP в отдельную подсеть коммутаторами DES-1210-xx поддерживаются функции Auto Surveillance VLAN (ASV) и Auto Voice VLAN. Для повышения безопасности сети администраторами сети могут использоваться списки контроля доступа (ACL). Функции управления коммутаторов DES-1210-xx включают Web-интерфейс, утилиту SmartConsole, упрощенный интерфейс командной строки и протокол SNMP.

Рис. 11.4. Коммутатор DES-1210-28

Серия DGS-1210-xx включает четыре модели: DGS-1210-10P, DGS-1210-20, DGS-1210-28, DGS-1210-28P и DGS-1210-52.

DGS-1210-10P	8 порта 10/100/1000Base-T, 2 комбо-порта 1000Base-T/SFP
DGS-1210-20	16 портов 10/100/1000Base-T, 4 порта SFP
DGS-1210-28	24 порта 10/100/1000Base-T, 4 порта SFP
DGS-1210-28P	24 порта PoE 10/100/1000Base-T, 4 порта SFP
DGS-1210-52	48 портов 10/100/1000Base-T, 4 порта SFP

Модели DGS-1210-10P и DGS-1210-28P поддерживают технологию Power over Ethernet (PoE). Все устройства за исключением DGS-1210-10P помещены в металлический корпус размером для установки в стойку (19”). Размер корпуса DGS-1210-10P составляет 13”. Коммутаторами DGS-1210-28P и DGS-1210-52 поддерживается функция Smart Fan.

Коммутаторы серии Smart поддерживают функции обеспечения отказоустойчивости (STP, RSTP, 802.3ad), безопасности (ACL, 802.1x, Port Security, SafeGuard Engine, Smart IMPB), сегментации сети (802.1Q, Traffic Segmentation, Auto Surveillance VLAN (ASV), Auto Voice VLAN), качества обслуживания (802.1p, Bandwidth control), мониторинга трафика (Port Morroring) и диагностики кабеля. Управление коммутаторами может осуществляться через Web-интерфейс, утилиту SmartConsole, упрощенный интерфейс командной строки и протокол SNMP. Помимо этого, коммутаторы серии DGS-1210-xx поддерживают технологию Green Ethernet и Jumbo-фреймы.

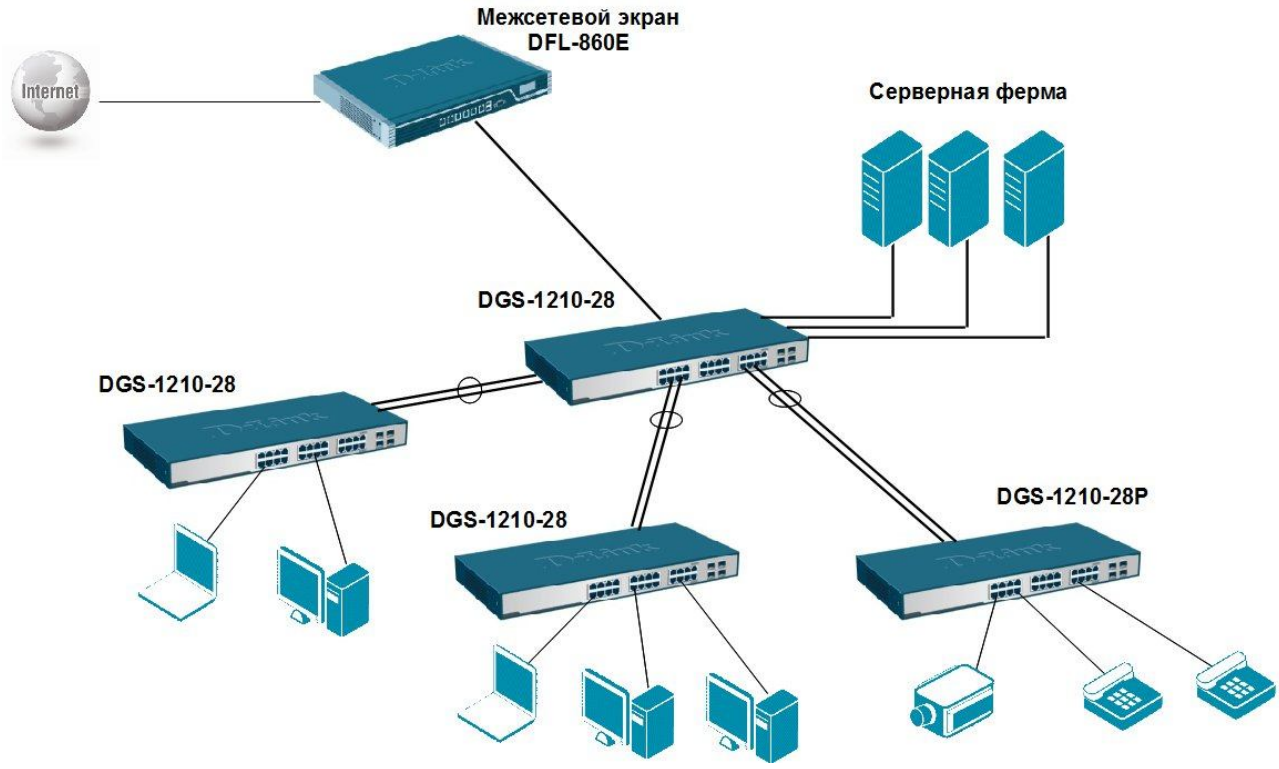


Рис. 11.5. Пример использования коммутаторов серии Smart в сетях SMB

Серия коммутаторов SmartPro может использоваться на уровне распределения небольших сетей малых и средних предприятий, т.к. в отличие от коммутаторов серий Easy Smart и Smart поддерживает статическую маршрутизацию. В дополнение к функционалу аналогичному серии Smart, коммутаторы серии SmartPro поддерживают виртуальное стекирование по технологии SIM.

Серия SmartPro представлена следующими моделями коммутаторов: DGS-1500-20, DGS-1500-28, DGS-1500-28P и DGS-1500-52.

DGS-1500-20	16 портов 10/100/1000Base-T, 4 порта SFP
DGS-1500-28	24 порта 10/100/1000Base-T, 4 порта SFP
DGS-1500-28P	24 порта PoE 10/100/1000Base-T, 4 порта SFP
DGS-1500-52	48 портов 10/100/1000Base-T, 4 порта SFP

Коммутаторами DGS-1500-28P и DGS-1500-52 поддерживается функция Smart Fan.

Рис. 11.6. Коммутаторы серии DGS-1500-xx

11.3 Управляемые коммутаторы

Управляемые коммутаторы (*Managed Switches*) по сравнению с неуправляемыми коммутаторами и коммутаторами серии Smart, являются сложными устройствами, поддерживающими расширенный набор функций 2 и 3 уровня модели OSI. Такие устройства предоставляют большой выбор интерфейсов, обладают высокоскоростной внутренней магистралью, возможностью установки дополнительных модулей и физического стекирования. Управление коммутаторами может осуществляться посредством Web-интерфейса, командной строки (CLI), протокола SNMP, Telnet и т.д.

Серия бюджетных управляемых коммутаторов Fast Ethernet 2 уровня DES-1228/ME, DES-3028/52 может использоваться на уровне доступа сетей малых, средних и крупных предприятий, а также в сетях провайдеров для предоставления услуг Triple Play. Коммутаторы поддерживают базовый и расширенный функционал 2 уровня, обеспечивающий возможность управления доступом пользователей, контроля полосы пропускания, сегментации сети, управления широковещательными пакетами, многоадресной рассылкой. Коммутаторы DES-1228/ME также поддерживают функцию диагностики кабеля.

DES-1228/ME	24 порта 10/100Base-TX, 2 порта SFP, 2 комбо-порта 1000Base-T/SFP
DES-3028	24 порта 10/100Base-TX, 2 порта 10/100/1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-3028P	24 порта PoE 10/100Base-TX, 2 порта 10/100/1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-3052	48 портов 10/100Base-TX, 2 порта 10/100/1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-3052P	48 портов PoE 10/100Base-TX, 2 порта 10/100/1000Base-T, 2 комбо-порта 1000Base-T/SFP

Коммутаторы Fast Ethernet 2 уровня серии DES-3200-xx предназначены для использования на уровне доступа сетей провайдеров, предоставляющих услуги по подключению к сети Интернет посредством технологий Metro Ethernet (Ethernet-To-The-Home, ETTH и Fiber-To-The-Home, FTTH) и реализующих сервисы Triple Play.

Рис. 11.7. Коммутаторы серии DES-3200-xx

DES-3200-10	8 портов 10/100Base-TX, 2 комбо-порта 1000Base-T/SFP
DES-3200-18	16 портов 10/100Base-TX, 2 комбо-порта 1000Base-T/SFP
DES-3200-26	24 порта 10/100Base-TX, 2 комбо-порта 1000Base-T/SFP
DES-3200-28	24 порта 10/100Base-TX, 2 комбо-порта 1000Base-T/SFP, 2 порта SFP
DES-3200-28F	24 порта SFP, 4 комбо-порта 1000Base-T/SFP

Серия коммутаторов Fast Ethernet 2 уровня DES-3528/3552 предназначена для использования на уровне доступа сетей крупных предприятий и Metro Ethernet с сервисами Triple Play. Коммутаторы поддерживают физическое стекирование по Ethernet, статическую

маршрутизацию, функции управления многоадресной рассылкой, расширенные функции безопасности и виртуальных локальных сетей VLAN. Устройства легко интегрируются с коммутаторами L3 уровня ядра для формирования многоуровневой сетевой структуры с высокоскоростной магистралью и централизованными серверами.

DES-3528	24 порта 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-3528DC	24 порта 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP (питание 48В DC)
DES-3528P	24 порта PoE 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP
DES-3552	48 портов 10/100Base-TX, 2 порта 1000Base-T, 2 комбо-порта 1000Base-T/SFP

Коммутаторы Fast Ethernet 3 уровня серии DES-3810-xx являются новым поколением мультисервисных коммутаторов D-Link, предназначенным для использования на уровне доступа сетей крупных предприятий и Metro Ethernet, в которых реализованы сервисы Triple Play и VPN. В настоящее время серия представлена моделью DES-3810-28. Одной из особенностей коммутаторов серии DES-3810-xx является то, что в них встроены два разных образа программного обеспечения: Standard Image (SI) и Enhanced Image (EI). В стандартной прошивке реализованы такие функции как качество обслуживания (QoS), включая механизм Traffic Shaping, Q-in-Q VLAN, маршрутизация пакетов IPv4, многоадресная рассылка, Ethernet OAM и множество функций безопасности. Расширенная прошивка включает поддержку маршрутизации IPv6, протоколов BGP и MPLS. Также коммутаторы серии DES-3810-xx поддерживают функцию Switch Resource Management (SRM), позволяющую администратору оптимизировать ресурсы коммутатора при его использовании в различных сетевых средах.

Рис. 11.8. Коммутатор DES-3810-28

DES-3810-28	24 порта 10/100Base-TX, 4 комбо-порта 1000Base-T/SFP

Коммутаторы Gigabit Ethernet 2 уровня серии DGS-3120-xx могут использоваться как на уровне доступа, так и на уровне субагрегации сетей SOHO/SMB и Metro Ethernet. В коммутаторах реализован базовый и расширенный функционал 2 уровня; поддерживается физическое стекирование через порты 10GE, подключение резервных источников питания, функция диагностики кабеля. Благодаря высокой плотности портов SFP коммутаторы DGS-3120-24SC и DGS-3120-24SC-DC обеспечивают возможность гибкого подключения по оптике к магистрали сети и серверам в сетях провайдеров услуг. Коммутаторы DGS-3120-24PC и DGS-3120-48PC поддерживают спецификацию передачи питания по Ethernet (стандарты IEEE 802.3af и IEEE 802.3at (PoE+)).

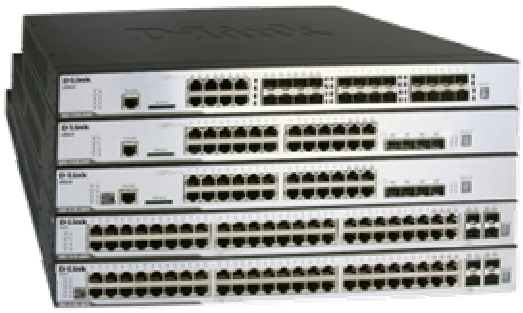


Рис.11.9. Коммутаторы серии DGS-3120-xx

DGS-3120-24TC	20 портов 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP
DGS-3120-48TC	44 порта 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP
DGS-3120-24PC	20 портов PoE 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP
DGS-3120-48PC	44 порта PoE 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP
DGS-3120-24SC	8 комбо-портов 1000Base-T/ SFP, 16 портов SFP
DGS-3120-24SC-DC	8 комбо-портов 1000Base-T/ SFP, 16 портов SFP (питание 48В DC)

Малопортовые коммутаторы Gigabit Ethernet 2 уровня серии DGS-3200-xx обладают полным набором функций, позволяющих обеспечить безопасность, контроль доступа, отказоустойчивость и управляемость сети, и предназначены для использования на уровне доступа сетей средних и крупных предприятий. Коммутаторы DGS-3200-10 и DGS-3200-16 помещены в компактный корпус размером 11”. В коммутаторе DGS-3200-10 применяется безвентиляторная технология, благодаря чему он бесшумно работает. В коммутаторе DGS-3200-16 используется технология автоматической вентиляции.

Рис. 11.10. Коммутаторы серии DGS-3200-xx

DGS-3200-10	8 портов 10/100/1000Base-T, 2 комбо-порта 1000Base-T/ SFP
DGS-3200-16	14 портов 10/100/1000Base-T, 2 комбо-порта 1000Base-T/ SFP
DGS-3200-24	20 портов 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP

Коммутаторы Gigabit Ethernet 2 уровня серии DGS-3420-xx предназначены для использования на уровне распределения сетей крупных предприятий и Metro Ethernet.

DGS-3420-26SC	20 портов SFP, 4 комбо-порта 1000Base-T/ SFP, 2 слота 10GE SFP+
DGS-3420-28SC	20 портов SFP, 4 комбо-порта 1000Base-T/ SFP, 4 слота 10GE SFP+
DGS-3420-28TC	20 портов 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP, 4 порта 10GE SFP+
DGS-3420-28PC	20 портов 10/100/1000Base-T PoE, 4 комбо-порта 1000Base-T(PoE)/ SFP, 4 порта 10GE SFP+
DGS-3420-52T	48 портов 10/100/1000Base-T, 4 порта 10GE SFP+
DGS-3420-52P	48 портов 10/100/1000Base-T PoE, 4 порта 10GE SFP+

Коммутаторы обеспечивают высокую плотность портов для подключения рабочих мест, оснащены слотами SFP и SFP+ для гибкого подключения по оптике на скоростях Gigabit и 10 Gigabit Ethernet соответственно, обладают высокопроизводительной внутренней магистралью, поддерживают возможность физического стекирования (до 12 устройств в стеке) через порты 10GE и подключение резервных источников питания.

Среди функциональных возможностей можно выделить поддержку статической маршрутизации IPv4/IPv6 и протокола RIP, расширенные функции безопасности, качества обслуживания, виртуальных локальных сетей и управления. Также в коммутаторах реализована поддержка функции ERPS (Ethernet Ring Protection Switching), которая обеспечивает защиту от колец Ethernet в коммутируемой сети.

Коммутаторы Gigabit Ethernet 2 уровня серии DGS-3700-xx спроектированы с учетом требований операторов телекоммуникационных услуг при построении сетей Metro Ethernet. Коммутаторы поддерживают широкий набор функций безопасности, качества обслуживания, управления многоадресными пакетами и обеспечивают расширенную поддержку VLAN. Коммутаторы обладают модульной архитектурой ядра, поддерживают возможность выбора источника питания (постоянного или переменного тока) и расширенный диапазон рабочих температур.

DGS-3700-12	8 портов 10/100/1000Base-T, 4 комбо-порта 1000Base-T/SFP
DGS-3700-12G	8 портов SFP, 4 комбо-порта 1000Base-T/SFP

Семейство маршрутизирующих управляемых коммутаторов Gigabit Ethernet 3 уровня с поддержкой портов 10GE DGS-36xx обладает высокой производительностью и предназначено для использования на уровнях распределения и ядра крупных корпоративных сетей, сетей предприятий малого и среднего бизнеса (SMB) и городских сетей Metro Ethernet. Благодаря расширенной поддержке функций многоадресной передачи данных, среди которых IGMP v.3, PIM SM и PIM DM, коммутаторы позволяют значительно повысить эффективность предоставляемых операторами связи таких услуг, как видео по требованию (VoD), IP-телевидение (IPTV) и телевидение высокой четкости (HDTV). Коммутаторы поддерживают протоколы маршрутизации BGP, OSPF, RIP v.1/2, возможность создания статических и плавающих статических маршрутов IP v4/v6.

DGS-3612	8 портов 10/100/1000Base-T, 4 комбо-порта SFP /1000Base-T
DGS-3612G	8 портов SFP, 4 комбо-порта SFP /1000Base-T
DGS-3627	20 портов 10/100/1000Base-T, 4 комбо-порта 1000Base-T/SFP, 3 слота расширения
DGS-3627G	20 портов SFP, 4 комбо-порта SFP /1000Base-T, 3 слота расширения
DGS-3650	44 порта 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP, 2 слота расширения

Высокопроизводительные коммутаторы Gigabit Ethernet 3 уровня с поддержкой портов 10 GE серии DGS-3610-xx обладают расширенным функционалом, включая поддержку BGP, и могут применяться на магистрали сетей Metro Ethernet и крупных предприятий.

Рис. 11.11. Коммутатор DGS-3610-26

DGS-3610-26	12 портов 10/100/1000Base-T, 12 комбо-портов 1000Base-T/ SFP, 2 слота расширения
DGS-3610-26G	12 портов SFP, 12 комбо-портов 1000Base-T/ SFP, 2 слота расширения

Семейство высокопроизводительных коммутаторов Gigabit Ethernet 3 уровня с поддержкой портов 10GE серии DGS-3620-xx предназначено для использования на уровне ядра сетей крупных предприятий и Metro Ethernet. Коммутаторы поддерживают физическое стекирование через порты 10GE SFP+ и позволяют объединить в стек до 12 устройств. Коммутаторы поддерживают расширенные функции 2 уровня, включая ERPS, Q-in-Q VLAN, 802.3ah Ethernet Link OAM, Optical Transceiver Digital Diagnostic Monitoring (DDM). Функции 3 уровня включают поддержку протоколов маршрутизации OSPF v.2/3, RIP v.1/2/ng для IP v4/v6, протоколов многоадресной рассылки IGMP v.1/2/3, PIM SM и PIM DM. Расширенные функции безопасности обеспечивают возможность контроля доступа к ресурсам сети и предотвращения распространенных атак типа ARP Spoofing. Расширенный набор средств и функций управления позволяет администратору сети производить гибкую настройку сети и следить за ее состоянием.

Рис. 11.12. Коммутатор DGS-3620-28TC

DGS-3620-28SC	20 портов SFP, 4 комбо-порта 1000Base-T/ SFP, 4 порта 10GE SFP+
DGS-3620-28TC	20 портов 10/100/1000Base-T, 4 комбо-порта 1000Base-T/ SFP, 4 порта 10GE SFP+
DGS-3620-28PC	20 портов 10/100/1000Base-T PoE, 4 комбо-порта 1000Base-T(PoE)/ SFP, 4 порта 10GE SFP+
DGS-3620-52T	48 портов 10/100/1000Base-T, 4 порта 10GE SFP+
DGS-3620-52P	48 портов 10/100/1000Base-T PoE, 4 порта 10GE SFP+

Модульные коммутаторы 3 уровня серии DGS-66xx представляют собой высокопроизводительные устройства с высокой плотностью портов, предназначенные для использования на уровне субядра сетей крупных предприятий, сетей небольших операторов связи, а также для организации широкополосного доступа в Интернет в крупных торговых комплексах и бизнес-центрах. В настоящее время серия представлена 4-слотовым шасси. Широкий выбор модулей позволяет обеспечить гибкость при подключении пользователей. В максимальной конфигурации шасси поддерживает до 144 гигабитных портов или до 24 портов 10GE. Шасси поддерживает расширенный набор функций 2 уровня. Функции 3 уровня включают поддержку маршрутизации OSPF v.2/3, RIP v.1/2/ng для IP v4/v6, BGP. Расширенные функции управления, мониторинга и сбора статистики, включая sFlow, LLDP, IPv6 Neighbor Discover (ND), DHCP relay option 82/60/61 предоставляют администратору сети

возможность следить за состоянием сети и анализировать причины возникновения в ней ошибок и узких мест.

Коммутаторы 3 уровня на основе шасси серии DES-72xx являются высокопроизводительными устройствами с высокой плотностью портов, предназначенными для уровня ядра сетей крупных предприятий и Metro Ethernet. Устанавливая в шасси модули расширения, пользователи могут получить до 384 гигабитных портов, до 32 портов 10GE, до 192 портов SFP или их комбинаций. Коммутаторы поддерживают богатый набор функций 2 и 3 уровня, включая поддержку протоколов BGP, MPLS (Multi-protocol Label Switching), функции IPFIX, позволяющей получать статистику о сетевом трафике.

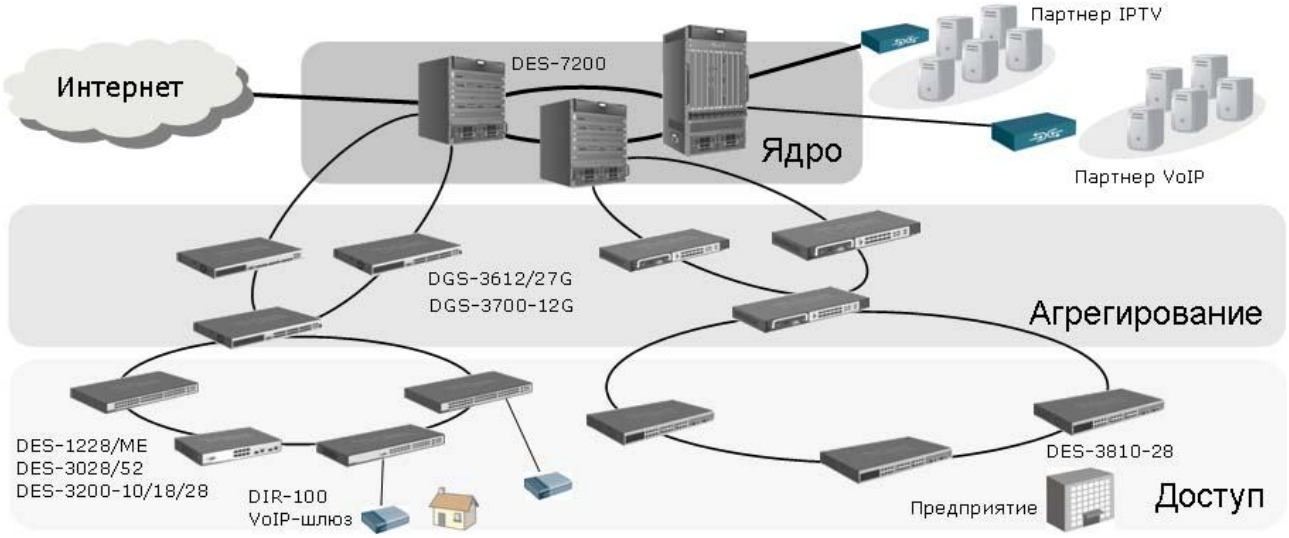


Рис. 11.13. Типовая схема применения коммутаторов D-Link

ГЛОССАРИЙ

A

AAA (англ. Authentication, Authorization, Accounting**)**. Функция, которая представляет собой комплексную структуру организации доступа пользователя в сеть. Она включает следующие базовые процессы:

- **Аутентификация (Authentication)**. Процедура проверки подлинности субъекта, на основе предоставленных им данных.
- **Авторизация (Authorization)**. Предоставление определенных прав лицу на выполнение некоторых действий.
- **Учет (Accounting)**. Слежение за использованием пользователем сетевых ресурсов.

Access layer. Уровень доступа. Уровень доступа является нижним уровнем иерархической модели сети и управляет доступом пользователей и рабочих групп к ресурсам объединенной сети. Основной задачей уровня доступа является создание точек входа/выхода пользователей в сеть.

ACL (англ. Access Control List**)**. Списки управления доступом. Списки управления доступом являются средством фильтрации потоков данных на аппаратном уровне. Используя ACL можно ограничить типы приложений, разрешенных для использования в сети, контролировать доступ пользователей к сети и определять устройства, к которым они могут подключаться. Также ACL могут использоваться для определения политики QoS, путем классификации трафика и переопределения его приоритета.

Agent. Агент. В модели клиент-сервер – часть системы, выполняющая подготовку информации и обмен ею между клиентской и серверной частью. Применительно к SNMP, термин агент означает программный модуль для управления сетью, который находится на управляемом сетевом устройстве (маршрутизаторе, коммутаторе, точке доступа, Интернет-шлюзе, принтере и т.д.). Агент обслуживает базу управляющей информации и отвечает на запросы менеджера SNMP.

Auto-negotiation. Автосогласование. Функция, обеспечивающая механизм автоматической настройки портов мультискоростных устройств. Устройства, поддерживающие функцию автосогласования, могут определять режимы работы партнеров по соединению, оповещать их о своих режимах работы и выбирать наилучший режим для совместного функционирования.

ARP (англ. Address Resolution Protocol**)**. Протокол разрешения адресов. Протокол, используемый для динамического преобразования IP-адресов в физические (аппаратные) MAC-адреса устройств локальной сети TCP/IP. В общем случае ARP требует передачи широковещательного сообщения всем узлам, на которое отвечает узел с соответствующим запросу IP-адресом.

ASIC (англ. Application Specific Integrated Circuit**)**. Специализированная для решения конкретной задачи интегральная схема (ИС). Современные контроллеры ACIS часто содержат на одном кристалле 32-битные процессоры, блоки памяти, включая ROM, RAM, EEPROM, Flash, и встроенное программное обеспечение. Такие ASIC получили название System-on-a-Chip (SoC).

B

Backbone. Магистраль, часть сети, по которой передается основной трафик, и которая является чаще всего источником и приемником трафика других сетей.

Backplane. Объединительная плата. Физическое соединение между интерфейсным процессором или платой, шинами данных и шинами распределения питания системного блока устройства.

Bandwidth. Полоса пропускания, доступная или занимаемая для передачи потока данных, измеряется в Кбит/с, Мбит/с, Гбит/с.

BGP (англ. Border Gateway Protocol). Протокол пограничных шлюзов. Обеспечивает основную динамическую маршрутизацию в сети Интернет. Регламентируется RFC 4271 и другими.

BOOTP (англ. Bootstrap Protocol**).** Протокол загрузки. Сетевой протокол, используемый для удаленной загрузки бездисковых рабочих станций. Позволяет им автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Регламентируется RFC 951 и другими.

BPDU (англ. Bridge Protocol Data Unit). Блоки данных протокола моста. Служебные кадры протокола связующего дерева (Spanning Tree Protocol), которые посылаются через заданные интервалы времени для обмена информацией между мостами.

Bridge. Мост. Устройство, соединяющее две физических сети и передающее кадры из одной сети в другую. Мосты работают на канальном уровне модели OSI.

Broadcast. Широковещание. Система доставки пакетов, при которой копия каждого пакета передается всем узлам, подключенным к сети.

Broadcast storm. Широковещательный шторм. Множество одновременных широковещательных рассылок в сети, которые, как правило, поглощают доступную полосу пропускания сети и могут вызвать отказ сети.

Bus topology. Шинная топология. Топология сети, при которой в качестве среды передачи используется единый кабель (он может состоять из последовательно соединенных отрезков), к которому подключаются все сетевые устройства.

C

CBS (англ. Committed Burst Size). Согласованный размер всплеска. В алгоритме «корзина маркеров» – объем трафика, на который может быть превышен размер корзины маркеров в отдельно взятый момент всплеска. Также см. CIR и EBS.

CDT (англ. Cross Device Trunking). Функция объединения нескольких физических портов разных коммутаторов физического стека в один агрегированный канал с повышенной полосой пропускания. См. также Link Aggregation.

Channel. Канал. Путь передачи [электрических] сигналов между двумя или несколькими точками. Используются также термины: link, line, circuit и facility.

Chassis. Шасси. Специальная конструкция для установки модулей и других компонент, образующих вместе единое устройство. Шасси обеспечивает питание и соединяющую модули магистраль.

CIOQ (англ. Combined Input and Output Queued). Тип буферизации в коммутаторах с комбинированными входными и выходными очередями. Буферы памяти подключаются как к входным, так и выходным портам

CIR (англ. Committed Information Rate). Согласованная скорость передачи. В алгоритме «корзина маркеров» – средняя скорость передачи трафика через интерфейс коммутатора/маршрутизатора. Также см. CBS и EBS.

CLI (англ. Command Line Interface**)**. Интерфейс командной строки. Позволяет пользователю взаимодействовать с операционной системой настраиваемого устройства путем ввода команд и параметров.

Client. Клиент. Узел или программное обеспечение (внешнее устройство), которое запрашивает у сервера некоторые сервисы.

Collision. Коллизия. Возникает в сети Ethernet, когда два узла одновременно ведут передачу. Передаваемые ими по физическому носителю кадры сталкиваются и разрушаются.

Collision domain. Домен коллизий. Часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части сети эта коллизия возникла.

Console port. Консольный порт. Порт на коммутаторе, к которому подключается терминальное или модемное соединение. Он преобразует параллельное представление данных в последовательное, которое используется при передаче данных. Этот порт используется для выделенного локального управления через консоль.

Core layer. Уровень ядра. Уровень ядра находится на самом верху иерархической модели сети и отвечает за надежную и быструю передачу больших объемов данных. Трафик, передаваемый через ядро, является общим для большинства пользователей. Сами пользовательские данные обрабатываются на уровне распределения, который, при необходимости, пересылает запросы к ядру.

CoS (англ. Class of Service). Класс обслуживания. Способ классификации и приоритизации пакетов на основе типа приложения или других методов классификации (802.1p, ToS, DiffServ) для обеспечения качества обслуживания в сети.

Cut-through. Коммутация без буферизации. Способ коммутации, при котором коммутатор копирует в буфер только MAC-адрес приемника (первые 6 байт после префикса) и сразу начинает передавать кадр, не дожидаясь его полного приема. Коммутация без буферизации уменьшает задержку, но проверку на ошибки не выполняет.

CVLAN (англ. Customer VLAN ID). В Q-in-Q – идентификатор VLAN, используемый в сетях пользователей. См. также SP-VLAN.

D

D-View. Программное обеспечение SNMP компании D-Link, используемое для управления и мониторинга сетевого оборудования.

Desktop switch. Настольный коммутатор. Настольные коммутаторы предназначены для размещения на столах. Иногда они могут оснащаться, входящими в комплект поставки, скобами для крепления на стену. Обычно такие коммутаторы обладают корпусом обтекаемой формы с относительно небольшим количеством фиксированных портов, внешним или внутренним блоком питания и небольшими ножками (обычно резиновыми) для обеспечения вентиляции нижней поверхности устройства.

DHCP (англ. Dynamic Host Configuration Protocol). Протокол динамической конфигурации узла. Сетевой протокол, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Является расширением протокола BOOTP. Регламентируется RFC 2131 и другими.

Diffserv (англ. Differentiated Services). Простой метод классификации, управления и предоставления качества обслуживания в современных IP-сетях. Использует для своей работы поле DSCP. Регламентируется RFC 2475, 3260.

Distribution layer. Уровень распределения/агрегации. Средний уровень иерархической модели сети, который иногда называют уровнем рабочих групп, является связующим звеном между уровнями доступа и ядра.

DNS (англ. Domain Name System). Система доменных имен. Компьютерная распределенная система для получения информации о доменах. Чаще всего используется для получения IP-адреса по имени хоста (компьютера или устройства), получения информации о маршрутизации почты, обслуживающих узлах для протоколов в домене.

DoS (англ. Denial-of-service). Атака типа «отказ в обслуживании».

Double VLAN. См. Q-in-Q.

DSCP (англ. Differentiated Services Code Point). Поле в заголовке IP-пакета, используемое для классификации (приоритизации) передаваемой информации. Регламентируется RFC 2774 и другими.

E

E2ES (англ. End-to-End Security). Дословно «Безопасность от края до края». Концепция комплексной защиты сети предприятия.

EBS (англ. Extended Burst Size). Расширенный размер всплеска. В алгоритме «корзина маркеров» – объем трафика, на который может быть превышен размер корзины маркеров в экстренном случае. Также см. CBS и CIR.

EAP (англ. Extensible Authentication Protocol**)**. Расширяемый протокол аутентификации. Протокол, поддерживающий множество механизмов аутентификации.

ECTP (англ. Ethernet Configuration Testing Protocol). Служебный протокол, используемый для работы функции LoopBack detection.

Enterprise. Крупные предприятия. Название сегмента рынка электроники. Обычно характеризует устройства, предназначенные для использования в сетях крупных предприятий с численностью сотрудников более 1000 человек.

Ethernet. Стандарт организации локальных сетей (ЛВС), описанный в спецификациях IEEE и других организаций. IEEE 802.3. Ethernet использует полосу 10 Мбит/с и метод доступа к среде CSMA/CD. Наиболее популярной реализацией Ethernet является 10Base-T. Развитием технологии Ethernet является Fast Ethernet (100 Мбит/с), Gigabit Ethernet (1 Гбит/с), 10 Gigabit Ethernet (10 Гбит/с).

ETTH (англ. Ethernet to the Home). Ethernet до дома (квартиры). Цель решения ETTH заключается в передаче данных, речи и видео по простой и недорогой сети Ethernet.

F

FDB (англ. Forwarding DataBase). Таблицы коммутации. Таблица коммутации создается коммутатором в процессе работы и содержит данные о соответствии MAC-адреса узла порту коммутатора.

FIFO (англ. First Input First Output). Тип очереди «первым пришел, первым ушел».

FTTH (англ. Fiber to the Home). Оптический кабель до дома (квартиры). Цель решения FTTH заключается в передаче данных, речи и видео по простой и недорогой сети, чаще всего Ethernet. Уникальным преимуществом данного решения является то, что использование Ethernet с оптическим волокном в качестве среды передачи данных позволяет обеспечить доступ к сети непосредственно из помещений клиентов услуг на высоких скоростях.

Filtering. Фильтрация. Процесс проверки пакетов данных в сети и определения адресатов для принятия решения о дальнейшей пересылке (данная локальная сеть, удаленная локальная сеть) или отбрасывании пакета. Фильтрация пакетов выполняется мостами, коммутаторами и маршрутизаторами.

Flooding. Лавинная передача. Способ передачи трафика, используемый в коммутаторах и мостах, при котором полученный интерфейсом трафик пересылается всем другим интерфейсам этого устройства.

Flow control. Управление потоком. Методы, используемые для контроля над передачей данных между двумя точками сети и позволяющие избегать потери данных в результате переполнения приемных буферов.

Forwarding. Продвижение. Процесс продвижения пакета к месту его назначения посредством сетевого устройства.

Fragment-free. Коммутация с исключением фрагментов. Этот метод коммутации является компромиссным решением между методами store-and-forward и cut-through switching. Коммутатор принимает в буфер первые 64 байта кадра, что позволяет ему отфильтровывать коллизионные кадры перед их передачей.

Frame. Кадр. Единица информации на канальном уровне сетевой модели. В ЛВС кадр представляет собой единицу данных подуровня MAC, содержащую управляющие данные и пакет сетевого уровня. Иногда для обозначения кадров используется термин пакет, но

термины кадр или фрейм никогда не используются для обозначения пакетов сетевого уровня. Кадр обычно содержит ограничители, управляющие поля, адреса, контрольную сумму и собственно информацию.

FTP (англ. File Transfer Protocol). Протокол передачи файлов. Протокол FTP относится к протоколам прикладного уровня стека TCP/IP и предназначен для передачи файлов в компьютерных сетях. FTP позволяет подключаться к серверам FTP, просматривать содержимое каталогов и загружать файлы с сервера или на сервер.

Full duplex. Дуплексная передача. Одновременная передача данных между станцией-отправителем и станцией- получателем.

G

GBIC (англ. Gigabit Interface Converter). Спецификация SFF-8053 комитета SFF на компактные сменные интерфейсные модули, описывающая конвертеры гигабитного интерфейса.

GVRP (англ. GARP VLAN Registration Protocol). В стандарте IEEE 802.1Q протокол GVRP определяет способ, посредством которого коммутаторы обмениваются информацией о сети VLAN, чтобы автоматически зарегистрировать членов VLAN на портах во всей сети. Позволяет динамически создавать и удалять VLAN на магистральных портах коммутаторов, автоматически регистрировать и исключать атрибуты VLAN.

GUI (англ. Graphical User Interface). Графический интерфейс пользователя. Метод взаимодействия между пользователем и компьютером, при котором пользователь может вызывать различные функции, указывая на графические элементы (кнопки) вместо ввода команд с клавиатуры.

H

Half duplex. Полудуплексная передача. Способность канала в каждый момент времени только передавать или принимать информацию. Прием и передача, таким образом, должны выполняться поочередно.

HDMI (англ. High-Definition Multimedia Interface). Цифровой интерфейс, использующийся в некоторых коммутаторах D-Link для физического стекирования.

HOL (англ. Head-Of-Line blocking). Блокировка первым в очереди. Блокировка возникает в том случае, когда коммутатор пытается одновременно передать пакеты из нескольких входных очередей на один выходной порт. При этом пакеты, находящиеся в начале этих очередей блокируют все остальные пакеты, находящиеся за ними.

I

IANA (англ. Internet Assigned Numbers Authority). Агентство по выделению имен и уникальных параметров протоколов Интернет.

ICMP (англ. Internet Control Message Protocol). Межсетевой протокол управляющих сообщений. Сетевой протокол, входящий в стек протоколов TCP/IP. В основном ICMP

используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или узел, или маршрутизатор не отвечают. Также на ICMP возлагаются некоторые сервисные функции. Регламентируется RFC 792 и другими.

IEEE (англ. Institute of Electrical and Electronic Engineers). Институт инженеров по электротехнике и радиоэлектронике. Профессиональная организация, основанная в 1963 году для координации разработки компьютерных и коммуникационных стандартов. Институт подготовил группу стандартов 802 для локальных сетей. Членами IEEE являются ANSI и ISO.

IGMP (англ. Internet Group Management Protocol). Межсетевой протокол управления группами. Протокол IGMP используется для динамической регистрации отдельных узлов в многоадресной группе локальной сети. Узлы сети определяют принадлежность к группе, посылая IGMP-сообщения на свой локальный многоадресный маршрутизатор. Регламентируется RFC 1112, 2236, 3376 и другими.

IMPB (англ. IP-MAC-Port Binding). Функция коммутаторов D-Link, позволяющая контролировать доступ компьютеров в сеть на основе их IP- и MAC-адресов, а также порта подключения.

IntServ (англ. Integrated Services). Интегрированные услуги. Модель приоритизации, предполагающая предварительное резервирование сетевых ресурсов с целью обеспечения предсказуемого поведения сети для приложений, требующих гарантированной выделенной полосы пропускания на всем пути следования трафика. Регламентируется RFC 1633 и другими.

IP (англ. Internet Protocol). IP-протокол. Часть стека протоколов TCP/IP. Описывает программную маршрутизацию пакетов и адресацию устройств. Стандарт используется для передачи через сеть базовых блоков данных и дейтаграмм IP. Обеспечивает передачу пакетов без организации соединений и гарантии доставки. Регламентируется RFC 791 и другими.

IP address. IP-адрес. Адрес для протокола IP – 32 битовое (4 байта) значение, определенное в STD 5 (RFC 791) и используемое для представления точек подключения в сети TCP/IP. IP-адрес состоит из номера сети (network portion) и номера хоста (host portion) – такое разделение позволяет сделать маршрутизацию более эффективной. Обычно для записи IP-адресов используют десятичную нотацию с разделением точками. Новая версия протокола IPv6 использует 128-разрядные адреса, позволяющие решить проблему нехватки адресного пространства.

ISO (англ. International Organization for Standardization). Международная организация по стандартизации.

ISO/OSI (англ. Open Systems Interconnection Reference Model). Эталонная модель взаимодействия открытых систем (OSI), разработанная организацией ISO.

ISP (англ. Internet Service Provider). Поставщик услуг Интернет.

L

LACP (англ. Link Aggregation Control Protocol). Протокол управления агрегированным каналом, регламентируемый в стандарте IEEE 802.3ad. См. также Link Aggregation.

LBD (англ. LoopBack Detection). Функция коммутаторов D-Link, блокирующая коммутационные петли на пользовательских портах.

L2 switch. Коммутатор 2-го уровня. Анализирует входящие кадры, принимает решение об их дальнейшей передаче и передает их получателю на основе MAC-адресов канального уровня модели OSI.

L3 switch. Коммутатор 3-го уровня. Выполняет L2 коммутацию в пределах рабочей группы (точно так же, как коммутатор 2-го уровня) и маршрутизацию между различными подсетями или виртуальными локальными сетями.

LAN (англ. Local Area Network). Локальная сеть. Высокоскоростная компьютерная сеть, покрывающая относительно небольшую площадь. Локальные сети объединяют рабочие станции, периферийные устройства, терминалы и другие устройства, находящиеся в одном здании или на другой небольшой территории.

Latency. Задержка. Временная задержка между моментом, когда устройство получило пакет, и моментом, когда пакет был отправлен на порт назначения.

Link Aggregation. Агрегирование каналов связи. Объединение нескольких физических портов в одну логическую магистраль на канальном уровне модели OSI с целью образования высокоскоростного канала передачи данных и повышения отказоустойчивости.

Load Balancing. Балансировка нагрузки. Распределение процесса выполнения заданий между несколькими устройствами сети с целью оптимизации использования ресурсов и сокращения времени вычисления.

M

MAC address. MAC-адрес. Стандартный адрес канального уровня, который требуется задавать для каждого порта или устройства, подключенного к локальной сети. Другие устройства используют эти адреса для обнаружения специальных сетевых портов, а также для создания и обновления таблиц маршрутизации и структур данных. Длина MAC-адреса составляет 6 байтов, а их содержимое регламентируется IEEE. MAC-адреса также называют аппаратными или физическими адресами.

MAC (англ. MAC-based Access Control). Функция коммутаторов D-Link, позволяющая проводить аутентификацию пользователей через протокол IEEE 802.1X, используя в качестве источника аутентификации MAC-адрес сетевой платы пользователя.

Managed switch. Управляемый коммутатор. Управляемые коммутаторы являются сложными устройствами, позволяющими выполнять расширенный набор функций 2 и 3 уровня модели OSI. Управление коммутаторами может осуществляться посредством Web-интерфейса, командной строки (CLI), протокола SNMP, Telnet и т.д.

MIB (англ. Management Information Base). База управляющей информации. Совокупность иерархически организованной информации, доступ к которой осуществляется посредством

протокола управления сетью SNMP. База управляющей информации состоит из управляемых объектов (MIB-объектов), значения которых могут быть изменены или извлечены с помощью команд SNMP и сетевой системы управления (например, D-Link D-View) с GUI-интерфейсом.

MDI (англ. Medium Dependent Interface). Ethernet-порт абонентского устройства, например, сетевой карты ПК.

MDIX (англ. Medium Dependent Interface with Crossover). Ethernet-интерфейс с перекрёстным подключением цепей приема и передачи. Используется в Ethernet-коммутаторах.

MSTP (англ. Multiple Spanning Tree Protocol). Является расширением протокола RSTP и позволяет настраивать отдельное связующее дерево для любой VLAN или группы VLAN, создавая множество маршрутов передачи трафика, и позволяя осуществлять балансировку нагрузки. Первоначально протокол MSTP был определен в стандарте IEEE 802.1s, но позднее был добавлен в стандарт IEEE 802.1Q-2003. Протокол MSTP обратно совместим с протоколами STP и RSTP.

MTU (англ. Maximum Transmission Unit). Модуль передачи максимального размера. Максимальный размер (в байтах) пакета данных, который можно передать через данный интерфейс.

Multicast. Многоадресная рассылка. Доставка потока данных группе узлов на IP-адрес группы многоадресной рассылки.

Multicast address. Групповой адрес. Общий адрес, который относится к некоторой группе нескольких сетевых устройств.

Multicast group. Группа многоадресной рассылки. Динамически определенная группа IP-узлов, идентифицируемая одним групповым IP-адресом.

Multicast router. Многоадресный маршрутизатор. Маршрутизатор, используемый для получения IGMP-ответов и периодической отправки IGMP-запросов о принадлежности узлов к многоадресной группе, чтобы определить, какие группы многоадресной рассылки активны или неактивны в данной сети.

N

NAP (англ. Network Access Protection). Защита доступа к сети. Технология компании Microsoft для управления доступом клиентских компьютеров к сетевым ресурсам на основе удостоверения компьютера и соответствия корпоративным политикам.

Network Address. Сетевой адрес. Адрес сетевого уровня, который относится к логическому, а не к физическому сетевому устройству. Он также называется протокольным адресом (protocol address).

Node. Узел. Точка присоединения к сети, устройство, подключенное к сети.

Non-blocking switch fabric. «Неблокирующая» коммутирующая матрица. Матрица, у которой производительность и QoS не зависят от типа трафика, коммутируемого через матрицу и производительность равна сумме скоростей всех портов.

NNI (англ. Network-to-Network Interface). Интерфейс «сеть-сеть». В Q-in-Q – роль порта, который подключается к внутренней сети провайдера или другим граничным коммутаторам.

NVRAM (англ. NonVolatile RAM). Энергонезависимое ОЗУ. Оперативное запоминающее устройство, содержимое которого сохраняется при отключении питания.

O

OID (англ. Object Identifier). В протоколе SNMP – идентификатор объекта в базе MIB.

OSI. См. ISO/OSI.

OSPF (англ. Open Shortest Path First). Протокол динамической маршрутизации для IP-сетей. Регламентируется RFC 2328, 5340 и другими.

P

Packet. Пакет. Группа битов, включающая данные и служебные поля, представленные в соответствующих форматах, и передаваемая целиком. Структура пакета зависит от протокола. В общем случае пакет включает 3 основных элемента: управляющую информацию (адрес получателя и отправителя, длина пакета и т.п.), передаваемые данные, биты контроля и исправления ошибок.

PCF (англ. Packet Content Filtering, также ACL PCF). Фильтрация по содержимому пакета. Тип ACL, побайтно обрабатывающий заголовок кадра. Тип заголовка (Ethernet, IP, или любой другой) при этом не имеет значения, обрабатываются все его поля одновременно.

PDU (англ. Protocol Data Unit). Модуль данных протокола. Термин OSI для пакетов данных.

Ping (англ. Packet INternet Groper). Проверка доступности адресата. Эхо-сообщение протокола ICMP и ответ на него. Инструмент, используемый для проверки доступности адресата в IP-сетях.

PoE (англ. Power over Ethernet). Технология передачи питания по кабелю «витая пара» в сетях Ethernet. Регламентируется стандартом IEEE 802.3af.

PoE Plus (англ. Power over Ethernet Plus, также PoE+). Технология передачи питания по кабелю «витая пара» в сетях Ethernet. Является расширением технологии PoE и обеспечивает подачу большей мощности. Регламентируется стандартом IEEE 802.3at.

Port density. Плотность портов. Количество портов на шасси.

Port Security. Безопасность портов. Функция, применяемая в коммутаторах для обеспечения контроля над подключением узлов к их портам.

PPPoE (англ. PPP over Ethernet). Реализация протокола PPP для сетей Ethernet. Регламентируется RFC 2516 и другими.

Proxy ARP (англ. Proxy Address Resolution Protocol)**.** Агент протокола разрешения адресов. Вариант протокола ARP, в котором промежуточное устройство (например, маршрутизатор) посылает ответ ARP от имени конечного узла запрашивающему устройству.

PVID (англ. Port VLAN ID). Идентификатор порта VLAN.

Q

QoS (англ. Quality of Service). Качество обслуживания. Показатель эффективности системы передачи данных, который отражает качество передачи.

Q-in-Q (или QinQ)**.** Расширение стандарта IEEE 802.1Q. Позволяет добавлять в маркированные кадры Ethernet второй тег IEEE 802.1Q. Регламентируется стандартом IEEE 802.1ad.

R

RADIUS (англ. Remote Authentication Dial-In User Service). Служба аутентификации удаленных пользователей. Протокол для реализации аутентификации, авторизации и сбора сведений об использованных ресурсах, разработанный для передачи сведений между центральной платформой и оборудованием. Регламентируется RFC 2865 и другими.

Rack mounted switch. Коммутаторы, монтируемые в телекоммуникационную стойку. Коммутаторы в стоечном исполнении высотой 1U обладают корпусом для монтажа в 19” стойку, встроенным блоком питания и фиксированным количеством портов.

RED (англ. Random Early Detection). В приоритизации – алгоритм произвольного раннего обнаружения, позволяющий избегать перегрузок в сети.

Redundancy. Избыточность. Дублирование устройств, сервисов и соединений. В случае неисправности позволяет избыточным устройствам, службам и соединениям выполнять функции исправных.

Redundant system. Избыточная система. Компьютер, маршрутизатор, коммутатор или другая система, которая содержит два или более экземпляра наиболее важных подсистем, таких как дисководы, центральные процессоры или источники питания.

Reliability. Надежность. В общем случае свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, хранения и транспортирования.

RIP (англ. Routing Information Protocol). Протокол динамической маршрутизации для IP-сетей. Регламентируется RFC 1058, 2453 и другими.

RIPng. Протокол RIP для протокола IPv6. Регламентируется RFC 2080.

RJ-45 (RJ45). Унифицированный разъем, используемый в телекоммуникациях, имеет 8 контактов и защелку.

RMON (англ. Remote MONitoring). Удаленный мониторинг. Спецификация RMON MIB, разработанная сообществом IETF для поддержки мониторинга и анализа протоколов в

локальных сетях. Первая версия RMON v.1 основывается на мониторинге информации сетей Ethernet и Token Ring. Ее расширением является RMON v.2, которая добавила к уже имеющимся средствам мониторинга, поддержку мониторинга на сетевом уровне и уровне приложений модели OSI. Регламентируется RFC 2819, 2819 и другими.

RMT (англ. Resilient Master Technology). Технология, обеспечивающая непрерывную работу физического стека при выходе какого-либо устройства из строя, замене, добавлении и удалении коммутаторов.

Router. Маршрутизатор. Устройство сетевого уровня, отвечающее за принятие решений о выборе одного из нескольких путей передачи сетевого трафика. Маршрутизаторы отправляют пакеты из одной сети в другую на основе информации сетевого уровня.

Routing. Маршрутизация. Процесс выбора оптимального пути для передачи сообщения.

RSTP (англ. Rapid Spanning Tree Protocol). Протокол RSTP является развитием протокола STP. Первоначально был определен в стандарте IEEE 802.1w-2001, сейчас определен в стандарте IEEE 802.1D-2004.

S

Segment. Сегмент. 1. Секция сети, ограниченная мостами, маршрутизаторами или коммутаторами. 2. В LAN с шинной топологией – непрерывная электрическая цепь, часто соединенная с другими сегментами при помощи повторителей. 3. Термин, используемый в спецификации TCP для описания одиночного модуля транспортного уровня.

SIM (англ. Single IP Management). Технология виртуального стекирования, применяемая в управляемых коммутаторах D-Link.

SFP (англ. Small Form Factor Pluggable). Промышленный стандарт модульных компактных приемопередатчиков (трансиверов), используемых для передачи данных.

Smart switch. Настраиваемый коммутатор. Настраиваемые коммутаторы позволяют настраивать определенные параметры сети, используя Web-интерфейс или компактный интерфейс командной строки (Compact Command Line Interface, CLI), доступный через Telnet.

SMB (англ. Small-to-Medium Business). Малые и средние предприятия. Название сегмента рынка электроники. Характеризизует устройства, предназначенные для использования в сетях малых и средних предприятий с численностью сотрудников от 100 до 999 человек.

SNMP (англ. Simple Network Management Protocol). Простой протокол управления сетью. Протокол 7 уровня модели OSI, который специально разработан для управления и мониторинга сетевых устройств. Протокол SNMP входит в стек протоколов TCP/IP и позволяет получать информацию о состоянии устройств сети, обнаруживать и исправлять неисправности и планировать развитие сети. Регламентируется RFC 1157, 1901-1908, 3411-3418 и другими.

SOHO (англ. Small Office, Home Office). Малый/домашний офис. Название сегмента рынка электроники. Как правило, характеризует устройства, предназначенные для домашнего

использования или использования в небольших офисах, и не рассчитанные на производственные нагрузки.

SP-VLAN (англ. Service Provider VLAN ID). В Q-in-Q – идентификатор VLAN, используемый в сети ISP. См. также CVLAN.

SRED (англ. Simple Random Early Detection). В приоритизации – алгоритм произвольного раннего обнаружения, позволяющий избегать перегрузок в сети. Является расширением алгоритма RED.

SSH (англ. Secure Shell). Безопасная оболочка. Сетевой протокол сеансового уровня, позволяющий производить удаленное управление операционной системой и туннелирование TCP-соединений. Регламентируется RFC 4253 и другими.

SSL (англ. Secure Sockets Layer). Уровень защищенных сокетов. Криптографический протокол, обеспечивающий безопасную передачу данных по сети Интернет. Регламентируется RFC 2246, 4346 и другими.

SST (англ. Single Spanning Tree Bridge). Мост, поддерживающий только единственное связующее дерево. Это единственное связующее дерево может поддерживать протоколы STP или RSTP.

STA (англ. Spanning Tree Algorithm). Алгоритм построения связующего дерева. Алгоритм, используемый протоколом связующего дерева для построения связующего дерева.

Stack. Стек. Группа сетевых устройств, которые объединены в одно логическое устройство с целью увеличения количества портов, удобства управления и мониторинга.

STP (англ. Spanning Tree Protocol). Протокол связующего дерева. Стандарт IEEE 802.1D-1998, использующий алгоритм связующего дерева и позволяющий самообучающемуся мосту динамически обрабатывать коммутационные петли в сетевой топологии путем создания связующего дерева. Мосты обнаруживают петли путем обмена сообщениями BPDU с другими мостами и ликвидируют петли посредством блокирования выбранных мостовых интерфейсов.

Store-and-forward. Коммутация с промежуточным хранением. Методика коммутации пакетов, согласно которой кадры полностью обрабатываются перед их отправкой через соответствующий порт. Обработка включает расчет CRC и проверку адреса приемника. Кроме того, кадры необходимо временно хранить до тех пор, пока не станут доступными сетевые ресурсы (например, свободный канал) для передачи сообщения. Эта технология противоположна коммутации без буферизации (cut-through).

Switch. Коммутатор. Сетевое устройство, которое фильтрует, пересылает и направляет кадры в зависимости от их адреса приемника. Коммутатор работает на канальном уровне модели OSI.

Switch capacity. Производительность коммутирующей матрицы. Производительность определяется как общая полоса пропускания (bandwidth), обеспечивающая коммутацию без отбрасывания пакетов трафика любого типа (одноадресного, многоадресного, широковещательного).

Switch fabric. Коммутирующая матрица. Коммутирующая матрица представляет собой чипсет, соединяющий множество входов с множеством выходов на основе фундаментальных технологий и принципов коммутации.

T

Tag. Тег. Идентификационная информация, в том числе и номер.

TCP (англ. Transmission Control Protocol). Протокол управления передачей. Ориентированный на соединение протокол транспортного уровня, обеспечивающий надежную дуплексную передачу данных. TCP входит в набор протоколов TCP/IP. Регламентируется RFC 675, 793, 2581 и другими.

Telnet. Стандартный протокол виртуального терминала из набора протоколов TCP/IP. Протокол Telnet используется для удаленного терминального соединения, что дает возможность пользователям подключаться к удаленным системам и использовать их ресурсы, как если бы они работали через обычный терминал. Регламентируется RFC 15, 854 и другими.

TFTP (англ. Trivial File Transfer Protocol). Простейший протокол передачи файлов. Упрощенная версия протокола FTP, который позволяет компьютерам обмениваться файлами по сети. Регламентируется RFC 1350 и другими.

Throughput. Пропускная способность. Объем информации, поступающей и, возможно, проходящей через определенный участок сети в определенный момент времени.

ToS (англ. Type of Service). Тип сервиса. Поле в заголовке протокола IP, используемое для обеспечения QoS.

TPID (англ. Tag Protocol Identifier). Идентификатор протокола тегирования в кадрах протоколов IEEE 802.1Q и IEEE 802.1ad.

Traffic Policing. Ограничение трафика. Механизм Traffic Policing служит для ограничения входящего и исходящего трафика в соответствии с установленными пороговыми значениями скорости. Допускается всплеск трафика. См. также Traffic Shaping.

Traffic Segmentation. Сегментация трафика. Функция, используемая в коммутаторах для разграничения доменов на уровне 2.

Traffic Shaping. Выравнивание трафика. Механизм Traffic Shaping служит для выравнивания исходящего трафика с целью предотвращения перегрузки канала и удовлетворения требования постащика услуг. См. также Traffic Policing.

Trap. Ловушка. Тревожное сообщение (alarm message), которое устройство, находящееся под мониторингом, посылает управляющей станции при возникновении тревожных условий. Условия тревоги могут включать ошибки устройств, сетевые ошибки, изменения состояний и переход заданных пороговых значений.

Trunk. Магистраль. Физическое и логическое соединение между двумя коммутаторами, по которому передается сетевой трафик.

U

UDP (англ. User Datagram Protocol). Протокол дейтаграмм пользователя. Протокол транспортного уровня, не требующий подтверждения соединения. Входит в набор протоколов TCP/IP. UDP обеспечивает обмен дейтаграммами без подтверждения и гарантий доставки.

UNI (англ. User-to-Network Interface). В Q-in-Q – роль порта, через который будет осуществляться взаимодействие граничного коммутатора провайдера с клиентскими сетями.

Unmanaged switch. Неуправляемый коммутатор. Неуправляемые коммутаторы не поддерживают функции настройки и управления, имеют уже предустановленную функциональность. Данные коммутаторы применяются там, где характеристики необходимые в сети стандартные, и не требуют дополнительных настроек.

V

VID (VLAN ID)**.** Идентификатор VLAN.

VoIP (англ. Voice over IP). IP-телефония. Система связи, обеспечивающая передачу речевого сигнала по любым IP-сетям.

VLAN (англ. Virtual LAN). Виртуальная локальная сеть. Группа устройств, принадлежащих одной или нескольким локальным сетям и сконфигурированных таким образом (при помощи программного обеспечения), что обмен данными между ними происходит так, как будто они подключены к одному кабелю, хотя на самом деле находятся в разных сегментах локальной сети. VLAN основаны на логическом соединении.

VT100. Тип терминала, который использует символы ASCII. Терминалы VT100 представляют информацию в текстовом виде.

X

XFP (англ. 10 Gigabit Small Form Factor Pluggable**)**. Протоколо-независимый оптический трансивер горячей замены, обычно работающий на длинах волны 850 нм, 1310 нм или 1550 нм на скорости 10 Гбит/с в стандартах SONET/SDH, Fibre Channel, Gigabit Ethernet, 10 Gigabit Ethernet, включая каналы WDM.

W

WAC (англ. Web-based Access Control). Функция комматуторов D-Link, используемая для аутентификации пользователей при их попытке подключиться к сети Интернет через коммутатор. Процесс аутентификации использует протокол HTTP. Коммутатор может выступать в качестве сервера аутентификации и выполнять аутентификацию на основе локальной базы данных, или быть клиентом RADIUS и использовать для аутентификации протокол IEEE 802.1X.

WDM (англ. Wavelength Division Multiplexing). Спектральное уплотнение каналов. Технология, позволяющая одновременно передавать несколько информационных каналов по одному оптическому волокну на разных несущих частотах.

Список литературы

1. Бараш Л. Коммутаторы в локальных сетях (параметры для сравнения), архитектура... // КО. — 1997. — № 12.
2. History of LAN Switching. http://www.myipaddressinfo.com
3. http://en.wikipedia.org – Википедия.
4. Руководство по технологиям объединенных сетей, 3-е издание.: Пер. с англ.– М.:Издательский дом «Вильямс», 2002.
5. Upgrading and Repairing Networks, Third Edition. Scott Mueller– Que, 2002.
6. Руководства пользователя коммутаторов D-Link. ftp://ftp.dlink.ru/
7. Н. Жилкина. Сменные интерфейсы.// «Журнала сетевых решений/ LAN». —2004 —№05
8. Network Processors. Panoc C. Lekkas. – The Mc-Graw-Hill Companies, 2003.
9. Evolution: 20 years of switching fabric. Ori Aruj, Dune Networks. http://www.commsdesign.com
10. On-chip Global Interconnects for Networking ASICs. http://www.lsi.com
11. Shared-Memory Fabrics Meet 10-Gbit Backplane Demands. Andreas D. Bovopoulos and Micha Zeiger, TeraChip, Inc. http://www.commsdesign.com
12. Matching Output Queueing with a Combined Input Output Queued Switch. Shang-Tse Chuang, Ashish Goel, Nick McKeown, Balaji Prabhakar. http://www-rcf.usc.edu
13. An improved algorithm for CIOQ switches. Yossi Azar, Yossi Richter. http://portal.acm.org
14. http://www.sciencedirect.com – сайт научной базы данных «SciVerse ScienceDirect»
15. Компьютерные сети. Принципы, технологии, протоколы. В.Г. Олифер, Н.А. Олифер. – СПб:Питер, 2000.
16. http://www.ieee.org – сайт института инженеров по электротехнике и электронике (IEEE, Institute of Electrical and Electronics Engineers)
17. Учебные материалы компании D-Link. ftp://ftp.dlink.ru/pub/Trainings/
18. http://www.olifer.co.uk/ - Телекоммуникационные технологии. Сайт Натальи и Виктора Олифер
19. Качество обслуживания в сетях IP.: Вегешна Шринивас. Пер. с англ.– М.:Издательский дом «Вильямс», 2003.
20. Internet RFC/STD/FYI/BCP Archives. http://www.faqs.org/rfcs
21. An Analysis of the Variation in the RED Algorithm. Andrew Haugh. http://www.bcri.ucc.ie/FILES/PUBS/BCRI_69.pdf