# Учебное пособие:

# Коммутаторы локальных сетей D-Link


**Первое издание**

Москва, 2004

**Оглавление**

# 1. ВВЕДЕНИЕ

Типичная сеть состоит из узлов (компьютеров), соединенных средой передачи данных (кабельной или беспроводной) и специализированным сетевым оборудованием, таким как маршрутизаторы, концентраторы или коммутаторы. Все эти компоненты сети, работая вместе, позволяют пользователям пересылать данные с одного компьютера на другой, возможно даже в другую часть света.

Коммутаторы – фундаментальная часть большинства современных сетей. Используя *микросегментацию*, они дают возможность одновременно посылать по сети информацию множеству пользователей. Микросегментация позволяет создать частные или выделенные сегменты – по одной рабочей станции на сегмент (к порту коммутатора подключается не сегмент, а только рабочая станция). Каждая рабочая станция, при этом, получает доступ сразу ко всей полосе пропускания, и ей не приходится конкурировать с другими станциями. Если оборудование работает в дуплексном режиме, то исключаются коллизии.

Существует множество различных типов коммутаторов и сетей. Коммутаторы, которые обеспечивают выделенное соединение для каждого узла внутренней сети компании, называются *коммутаторами локальных сетей (LAN Switches)*. Им и посвящен данный курс лекций.

## 1.1 Преимущества использования коммутаторов LAN в сетях

В большинстве первых локальных сетей использовались концентраторы для организации соединения между рабочими станциями сети. По мере роста сети, появлялись следующие проблемы:

- Маштабируемость сети (Scalability) – в сети, построенной на концентраторах, ограниченная совместно используемая полоса пропускания сильно затрудняет рост сети без потери производительности, а современные приложения требуют большую полосу пропускания, чем раньше.
- Задержка (Latency) – количество времени, которое требуется пакету, чтобы достичь пункта назначения. Т.к. каждый узел в сети, построенной на концентраторах должен ждать появления возможности передачи данных во избежании коллизий, то задержка может значительно увеличиться при наращивании количества узлов в сети. Сбой в сети (Network failure) – в обычной сети, одно устройство, подключенное к концентратору, может вызвать проблемы у остальных устройств, подключенных к нему из-за несоответствия скоростей работы (100 Мбит/с сетевой адаптер и 10 Мбит/с концентратор) или большого числа широковещательных сообщений (broadcast). Коммутаторы могут быть сконфигурированы для ограничения количества широковещательных пакетов.
- Коллизии (Collisions) – в полудуплексном Ethernet используется метод Carrier Sense Multiple Access /Collision Detection (CSMA/CD) для доступа к разделяемой среде передачи данных. При этом способе доступа, узел не сможет отправить свой пакет до тех пор, пока не убедиться, что среда передачи свободна. Если два узла обнаружили, что среда передачи свободна и начали передачу в одно и тоже время, возникает коллизия и пакет теряется. Часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части сети эта коллизия возникла, называется *доменом коллизий (collision domain)*. Сеть Ethernet, построенная на концентраторах, всегда образует один домен коллизий.

Простая замена концентраторов на коммутаторы позволяет значительно повысить эффективность локальных сетей, при этом не требуется замена кабельной проводки или сетевых адаптеров. Коммутаторы делят сеть на отдельные логические сегменты, создавая при этом отдельные небольшие по размеру домены коллизий на каждом порту.

Разделение большой сети на несколько автономных сегментов при помощи коммутаторов имеет несколько преимуществ. Поскольку перенаправлению подвергается только часть трафика, коммутаторы уменьшают трафик, принимаемый устройствами во всех сегментах сети. Коммутаторы увеличивают фактический размер сети, позволяя подключать к ней удаленные станции, которые иначе подключить нельзя. Это достигается возможностью работы коммутатора в режиме полного дуплекса, благодаря которому нет необходимости определять коллизию в сети.

Еще одно существенное преимущество коммутаторов над концентраторами следующее. Все узлы, подключенные к концентратору, делят между собой всю полосу пропускания. Коммутаторы предоставляют каждому узлу (если он подключен непосредственно к порту коммутатора) отдельную полосу пропускания, чем уменьшают вероятность коллизий в сетевых сегментах.

Например, если к 10 Мбит/с концентратору подключено 10 устройств, то каждый узел получит пропускную способность равную менее 1 Мбит/с (10/N Мбит/с, где N-количество рабочих станций), даже если не все устройства будут передавать данные. Если вместо концентратора поставить коммутатор, то каждый узел сможет функционировать на скорости 10 Мбит/с.

До появления коммутаторов, сети Ethernet были *полудуплексными*, т.е. только одно устройство могло передавать данные в любой момент времени в одном домене коллизий. Коммутация позволила сети Ethernet работать в полнодуплексном режиме.

*Полнодуплексный режим* – это дополнительная возможность одновременной двухсторонней передачи по линии связи "точка – точка" на MAC - подуровне. Функционально дуплексная передача намного проще полудуплексной, т.к. она не вызывает в среде передачи коллизий, не требует составления расписания повторных передач и добавления битов расширения в конец коротких кадров. В результате не только увеличивается время, доступное для передачи данных, но и *удваивается* полезная полоса пропускания канала, поскольку каждый канал обеспечивает полноскоростную одновременную двустороннюю передачу.

Технология коммутации представляет новый шаг в развитии локальных сетей. В данный момент коммутаторы являются идеальным решением для увеличения пропускной способности локальной сети.

## 2. Технологии коммутации

### 2.1.1 **Коммутация 2-го уровня**

Коммутаторы работают на канальном уровне модели OSI. Они анализируют входящие кадры, принимают решение об их дальнейшей передаче на основе MAC - адресов, и передают кадры пунктам назначения. Основное преимущество коммутаторов – 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, поскольку пакет данных не претерпевает изменений. Передача кадра в коммутаторе может осуществляться специализированным контроллером, называемым Application-Specific Integrated Circuits (ASIC). Эта технология, разработанная для коммутаторов, позволяет поддерживать гигабитные скорости с небольшой задержкой.

Существуют 2 основные причины использования коммутаторов 2-го уровня – сегментация сети и объединение рабочих групп. Высокая производительность коммутаторов позволяет разработчикам сетей значительно уменьшить количество узлов в физическом сегменте. Деление крупной сети на логические сегменты повышает производительность сети (за счет уменьшения объема передаваемых данных в отдельных сегментах), а также гибкость построения сети, увеличивая степень защиты данных, и облегчает управление сетью.

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

Таким образом, очевидно, что для повышения производительности сети необходима функциональность 3-го уровня OSI модели.

### 2.1.2 Коммутация 3-го уровня

Коммутация 3-го уровня – это аппаратная маршрутизация, где передача пакетов обрабатывается контроллерами ASIC. В отличие от коммутаторов 2-го уровня, коммутаторы 3-го уровня принимают решения на основе информации сетевого уровня, а не на основе MAC - адресов. Основная цель коммутации 3-го уровня – получить скорость коммутации 2-го уровня и масштабируемость маршрутизации. Обработку пакетов коммутатор 3-го уровня выполняет таким же образом, как и маршрутизатор:

- на основе информации 3-го уровня (сетевых адресов) определяет путь к месту назначения пакета
- проверяет целостность заголовка 3-го уровня, вычисляя контрольную сумму
- проверяет время жизни пакета
- обрабатывает и отвечает на любую дополнительную информацию
- обновляет статистику в Информационной базе управления (Management Information Base -MIB)
- обеспечивает управление безопасностью (если необходимо)
- обеспечивает необходимое качество сервиса (QoS) для мультимедийных приложений чувствительных к задержкам передачи

Основное отличие между маршрутизаторами и коммутаторами 3-го уровня заключается в том, что в основе коммутации 3-го уровня лежит аппаратная реализация. В маршрутизаторах общего назначения коммутация пакетов обычно выполняется программным образом. Т.к. коммутаторы 3-го уровня обычно быстрее и дешевле маршрутизаторов, то их использование в локальных сетях очень привлекательно.

В качестве примеров коммутаторов 3-го уровня можно привести D-Link DES-3326S и DES-3326SR, DES-3350SR, DES-6300, DES-6500.

### 2.1.3 Коммутация 4-го уровня

Коммутация 4-го уровня основывается на аппаратной маршрутизации сетевого уровня, которая отвечает за управляющую информацию 4-го уровня. Информация в заголовках пакета обычно включает адресацию сетевого уровня, тип протокола 3-го уровня, время жизни (TTL) и контрольную сумму. В пакете также содержится информация о протоколах верхних уровней, такая как тип протокола и номер порта.

Простое определение коммутации 4-го уровня – это возможность принимать решение о передаче пакета, основываясь не только на MAC или IP адресах, но и на параметрах 4-го уровня, таких как номер порта TCP/UDP.

Маршрутизаторы умеют управлять трафиком, основываясь на информации транспортного уровня. Одним из методов является создание расширенных списков доступа (extended access lists).

Когда коммутаторы выполняют функции 4-го уровня, они читают поля TCP и UDP внутри заголовка и определяют, какой тип информации передается в этом пакете. Администратор сети может запрограммировать коммутатор обрабатывать трафик в соответствии с приоритетом приложений. Эта функция позволяет определить качество сервиса для конечных пользователей. Когда задано качество сервиса, коммутация 4-го уровня будет выделять, например, трафику видеоконференции, большую полосу пропускания по сравнению, например, с почтовым сообщением или пакетом FTP.

Коммутация 4-го уровня необходима, если выбранная политика предполагает разделение управления трафиком по приложениям или требуется учет количества трафика,

вырабатываемого каждым приложением. Однако следует заметить, что коммутаторам, выполняющим коммутацию 4-го уровня, требуется возможность определять и хранить большое число таблиц коммутации, особенно если коммутатор используется внутри ядра корпоративной сети.

## 2.2 Технологическая реализация коммутаторов

Коммутаторы ЛВС отличаются большим разнообразием возможностей и, следовательно, цен. Одной из причин столь больших различий является то, что они предназначены для решения различных классов задач. Коммутаторы высокого класса должны обеспечивать высокую производительность и плотность портов, а также поддерживать широкий спектр функций управления. Такие устройства зачастую кроме традиционной коммутации на MAC-уровне выполняют функции маршрутизации. Простые и дешевые коммутаторы имеют обычно небольшое число портов и не способны поддерживать функции управления.

Одним из основных различий является используемая в коммутаторе архитектура. Поскольку большинство современных коммутаторов работают на основе патентованных контроллеров ASIC, устройство этих микросхем и их интеграция с остальными модулями коммутатора (включая буферы ввода-вывода) играет важнейшую роль. Коммутаторы, реализующие также функции сетевого уровня (маршрутизацию), оснащены, как правило, RISC-процессорами для выполнения ресурсоемких программ маршрутизации.

Контроллеры ASIC для коммутаторов ЛВС делятся на 2 класса - большие ASIC, способные обслуживать множество коммутируемых портов (один контроллер на устройство) и небольшие контроллеры ASIC, обслуживающие несколько портов и объединяемые в матрицы коммутации. Вопросы масштабирования и стратегия разработчиков коммутаторов в области организации магистралей и/или рабочих групп определяет выбор ASIC и, следовательно, - скорость продвижения коммутаторов на рынок.

Существует 3 варианта архитектуры коммутаторов:

- На основе коммутационной матрицы (cross-bar);
- С разделяемой многовходовой памятью (shared memory);
- На основе общей высокоскоростной шины.

Часто эти три способа взаимодействия комбинируются в одном коммутаторе.

### Коммутаторы на основе коммутационной матрицы

*Коммутационная матрица (cross-bar)* - основной и самый быстрый способ взаимодействия процессоров портов, именно он был реализован в первом промышленном коммутаторе локальных сетей. Однако, реализация матрицы возможна только для определенного числа портов, причем сложность схемы возрастает пропорционально квадрату количества портов коммутатора.

На рисунке показана блок-схема коммутатора с архитектурой, используемой для поочередного соединения пар портов. В любой момент такой коммутатор может обеспечить организацию только одного соединения (пара портов). При невысоком уровне трафика не требуется хранение данных в памяти перед отправкой в порт назначения. Однако, коммутаторы cross-bar требуют буферизации на входе от каждого порта, поскольку в случае использования единственного возможного соединения коммутатор блокируется (рисунок 1). Несмотря на малую стоимость и высокую скорость продвижения на рынок, коммутаторы класса cross-bar слишком примитивны для эффективной трансляции между низкоскоростными интерфейсами Ethernet или Token Ring и высокоскоростными портами ATM и FDDI.

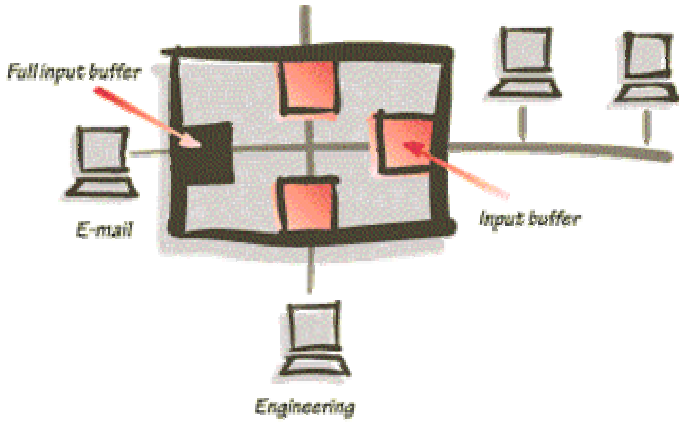



Рисунок 1. Блокировка коммутатора на основе коммутационной матрицы

**Коммутаторы с разделяемой памятью**

Коммутаторы с разделяемой памятью имеют общий входной буфер для всех портов. Буферизация данных перед их рассылкой приводит к возникновению задержки. Однако, коммутаторы с разделяемой памятью, как показано на рисунке 2 не требуют организации специальной внутренней магистрали для передачи данных между портами, что обеспечивает им более низкую цену по сравнению с коммутаторами на базе высокоскоростной внутренней шины.

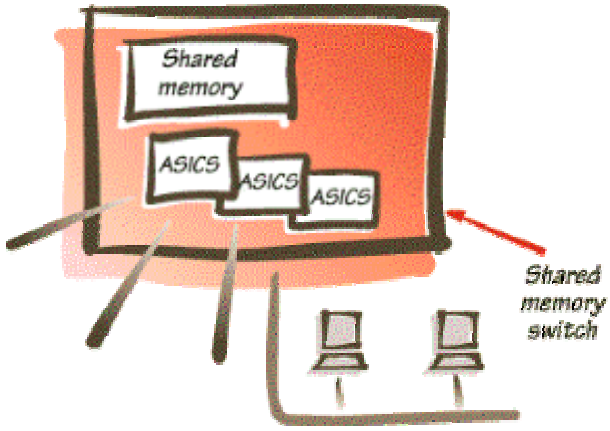



Рисунок 2. Архитектура коммутатора с разделяемой памятью

**Коммутаторы с общей шиной**

Коммутаторы с общей шиной (backplane) используют для связи процессоров портов высокоскоростную шину, используемую в режиме разделения времени. На рисунке 3 показана блок-схема коммутатора с высокоскоростной шиной, связывающей контроллеры ASIC. После того, как данные преобразуются в приемлемый для передачи по шине формат, они помещаются на шину и далее передаются в порт назначения.

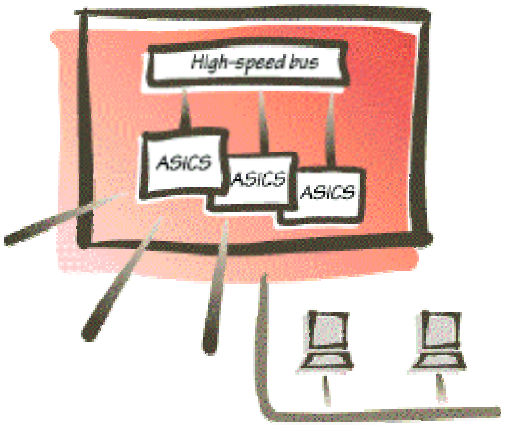



Рисунок 3 . Коммутатор с высокоскоростной шиной

Для того, чтобы шина не была узким местом коммутатора, ее производительность должна быть, по крайней мере, в N/2 раз выше скорости поступления данных во входные блоки процессоров портов. Кроме этого, кадр должен передаваться по шине небольшими частями, по несколько байт, чтобы передача кадров между несколькими портами происходила в псевдопараллельном режиме, не внося задержек в передачу кадра в целом. Размер такой ячейки данных определяется производителем коммутатора. Поскольку шина может обеспечивать одновременную передачу потока данных от всех портов, такие коммутаторы часто называют "неблокируемыми" (non-blocking), т.е. они не создают пробок на пути передачи данных.

### 2.2.1 Методы коммутации

Коммутаторы можно классифицировать по методам передачи кадров.

При *коммутации с промежуточным хранением (store-and-forward)* – коммутатор копирует весь кадр в буфер и только затем его передает. Перед отправкой фрейма читаются его адрес назначения и адрес источника, если надо, к ним применяется соответствующий фильтр и только после этого кадр передается на выходной порт. Естественно, что этот способ передачи связан с задержками, при этом, чем больше кадр, тем больше времени требуется на его прием. Во время приема кадра происходит его проверка на наличие ошибок.

*Коммутация «на лету» (cut-through)* – коммутатор локальной сети копирует во внутренние буферы только адрес приемника (первые 6 байт после префикса) и сразу начинает передавать кадр, не дожидаясь его полного приема. Это режим уменьшает задержку, но проверка на ошибки в нем не выполняется. Существует две формы коммутации «на лету»:

*Коммутация с быстрой передачей (fast-forward switching)* – эта форма коммутации предлагает низкую задержку за счет того, что кадр начинает передаваться немедленно, как только будет прочитан адрес назначения. Передаваемый кадр может содержать ошибки. В этом случае сетевой адаптер, которому предназначен этот кадр, отбросит его, что вызовет необходимость повторной передачи этого кадра. Другая форма коммутации уменьшает количество пакетов передаваемых с ошибками.

*Коммутация со свободными фрагментами (fragment-free switching)*– фильтрует коллизионные кадры, перед их передачей. В правильно работающей сети, коллизия может произойти во время передачи первых 64 байт. Поэтому, все кадры, с длиной больше 64 байт считаются правильными. Этот метод коммутации ждет, пока полученный кадр не будет проверен на предмет коллизии, и только после этого, начнет его передачу.

### 2.2.2 Конструктивное исполнение коммутаторов

В конструктивном отношении коммутаторы делятся на следующие типы:

- автономные коммутаторы с фиксированным количеством портов;
- модульные коммутаторы на основе шасси;
- коммутаторы с фиксированным количеством портов, собираемые в стек.

Первый тип коммутаторов обычно предназначен для организации небольших рабочих групп.

Модульные коммутаторы на основе шасси чаще всего предназначены для применения на магистрали сети. Поэтому они выполняются на основе какой-либо комбинированной схемы, в которой взаимодействие модулей организуется по быстродействующей шине или же на основе быстрой разделяемой памяти большого объема. Модули такого коммутатора выполняются на основе технологии «hot swap», то есть допускают замену на ходу, без выключения коммутатора, так как центральное коммуникационное устройство сети не должно иметь перерывов в работе. Шасси обычно снабжается резервными источниками питания и резервными вентиляторами в тех же целях. Примерами модульных коммутаторов D-Link могут служить DES-1200M, , DES-6000, DES-6300, DES-6500, DES-7000.

Рисунок 4. DES-7000

С технической точки зрения определенный интерес представляют стековые коммутаторы. Эти устройства представляют собой коммутаторы, которые могут работать автономно, так как выполнены в отдельном корпусе, но имеют специальные интерфейсы, которые позволяют их объединять в общую систему, работающую как единый виртуальный коммутатор. Говорится, что в этом случае отдельные коммутаторы образуют стек. Компания D-Link производит стековые коммутаторы 2-го и 3-го уровня, примерами которых могут служить DES-3226S, DES-3326S, DGS-3312SR, DES-3624 и др.

Существуют 2 подхода для объединения коммутаторов в стек.

*Вариант 1:* Стек типа «кольцо»

В качестве примера рассмотрим коммутаторы DES-3326S. Один специальный интерфейс для стекирования подключается к вышележащему коммутатору, а второй - к нижележащему, при этом самый нижний и самый верхний коммутатор в стеке также объединяются. Коммутаторы объединяются высокоскоростной шиной с производительностью 1 Гбит/с. Структура стека на коммутаторах D-Link DES-3326S, соединяемых по скоростным специальным портам, показана на рисунке 6.

Рисунок 5. Стек на коммутаторах DES-3326S

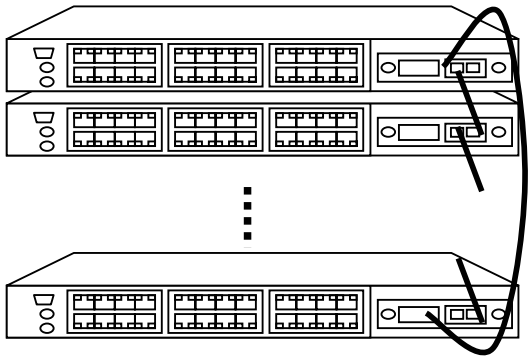


Рисунок 6. Стек коммутаторов, объединяемых по высокоскоростным каналам стандартным способом (в кольцо)

*Вариант 2*: Соединение «точка-точка» - стек типа «звезда»
Получить такой стек, можно используя коммутаторы D-Link DGS-3212SR и/или DGS-3312SR в качестве агрегирующего устройства. Они позволяют объединить в стек до 12 устройств DES-3226S по топологии «звезда» получив до 288 портов 10/100 Мбит/с Fast Ethernet и 12 портов Gigabit Ethernet, и управлять ими как единым сетевым узлом. В результате получается более производительное решение, по сравнению с предыдущим вариантом, поскольку каждое соединение точка-точка является полнодуплексным 2Гбит/с соединением.

Рисунок 7. Стек типа «звезда» на коммутатора DGS-3212SR (в центре) и DES-3226S

### 2.2.3 Понятие неуправляемых, управляемых и настраиваемых коммутаторов

Коммутаторы можно классифицировать по управлению.

*Управляемые коммутаторы* – поддерживают широкий набор функций управления и настройки, включающие Web-интерфейс управления, интерфейс командной строки (CLI), Telnet, SNMP, TFTP и др. В качестве примера можно привести коммутаторы D-Link DES-3226, DES-3226S, DES-3250TG, DES-6300, DGS-3212SR, DGS-3224SR и др.

*Неуправляемые коммутаторы* – функции управления и настройки не поддерживают. Примером могут служить коммутаторы D-Link серии DxS-10xx.

*Настраиваемые коммутаторы* занимают промежуточную позицию между ними. Эти коммутаторы позволяют выполнять настройку определенных параметров, но не поддерживают удаленное управление по SNMP и Telnet. Примером таких коммутаторов являются DES-1218R/26R/26G.

Большинство современных управляемых коммутаторов обеспечивают возможность конфигурирования на основе Web, что позволяет использовать в качестве станции управления любой компьютер, оснащенный Web-браузером, независимо от операционной системы.

Также стоит отметить возможность обновления программного обеспечения коммутатора (за исключением неуправляемых). Это обеспечивает более долгий срок службы устройств, т.к. позволяет добавлять новые функции либо устранять имеющиеся ошибки по мере выхода новых версий ПО, что существенно облегчает и удешевляет использование устройств, т.к. как правило, новые версии поставщики распространяют 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 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 - «на лету» или с промежуточным хранением;
- размер буфера (буферов) кадров;
- производительность внутренней шины;
- производительность процессора или процессоров;
- размер внутренней адресной таблицы.

#### Скорость фильтрации и скорость продвижения

Скорость фильтрации и продвижения кадров - это две основные характеристики производительности коммутатора. Эти характеристики являются интегральными показателями, они 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;
- просмотр адресной таблицы с целью нахождения порта для адреса назначения кадра;
- уничтожение кадра, если его порт назначения и порт источника принадлежат одному логическому сегменту.

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 адресной таблицы с целью нахождения порта для адреса назначения кадра;
- передача кадра в сеть через найденный по адресной таблице 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* 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* 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, - просмотра адресной таблицы, принятия решения о продвижении и получения доступа к среде выходного порта.

Величина вносимой коммутатором задержки зависит от режима его работы. Если коммутация осуществляется «на лету», 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).

Коммутатор - это многопортовое устройство, поэтому для него принято все приведенные выше характеристики (кроме задержки передачи кадра) давать в двух вариантах. Первый вариант - суммарная производительность коммутатора при одновременной передаче трафика по всем его портам, второй вариант - производительность, приведенная в расчете на один порт. Обычно производители коммутаторов указывают общую максимальную пропускную способность устройства.

**Размер адресной таблицы**

Максимальная емкость адресной таблицы определяет предельное количество MAC-адресов, с которыми может одновременно оперировать коммутатор. В таблице коммутации для каждого порта хранятся только те наборы адресов, с которыми он работал в последнее время.

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 4К до 8К. Коммутаторы крупных рабочих групп поддерживают таблицу MAC адресов емкостью от 8К до 16К, а коммутаторы магистралей сетей – как правило, от 16К до 32 К адресов.

Недостаточная емкость адресной таблицы может служить причиной замедления работы коммутатора и засорения сети избыточным трафиком. Если адресная таблица коммутации полностью заполнена, а порт встречает новый адрес источника в поступившем пакете,

коммутатор должен вытеснить из таблицы какой-либо старый адрес и поместить на его место новый. Эта операция сама по себе отнимет часть времени, но главные потери производительности будут наблюдаться при поступлении кадра с адресом назначения, который пришлось удалить из адресной таблицы. Так как адрес назначения кадра неизвестен, то коммутатор должен передать этот кадр на все остальные порты. Эта операция будет создавать лишнюю работу для многих процессоров портов, кроме того, копии этого кадра будут попадать и на те сегменты сети, где они совсем не обязательны.

**Объем буфера кадров**

Внутренняя буферная память коммутатора нужна для временного хранения кадров данных в тех случаях, когда их невозможно немедленно передать на выходной порт. Буфер предназначен для сглаживания кратковременных пульсаций трафика. Ведь даже если трафик хорошо сбалансирован и производительность процессоров портов, а также других обрабатывающих элементов коммутатора достаточна для передачи средних значений графика, это не гарантирует, что их производительности хватит при пиковых значениях нагрузок. Например, трафик может в течение нескольких десятков миллисекунд поступать одновременно на все входы коммутатора, не давая ему возможности передавать принимаемые кадры на выходные порты.

При кратковременном многократном превышении среднего значения интенсивности трафика (а для локальных сетей часто встречаются значения коэффициента пульсации трафика в диапазоне 50-100) возможны потери кадров. Одним из методов борьбы с этим служит буфер большого объема. Чем больше объем этой памяти, тем менее вероятны потери кадров при перегрузках, хотя при несбалансированности средних значений трафика буфер все равно рано или поздно переполнится. Другой метод – управление потоком (*Flow control*).

Обычно коммутаторы, предназначенные для работы в ответственных частях сети, имеют буферную память в несколько десятков или сотен килобайт на порт. Хорошо, когда эту буферную память можно перераспределять между несколькими портами, так как одновременные перегрузки по нескольким портам маловероятны. Дополнительным средством защиты может служить общий для всех портов буфер в модуле управления коммутатором. Такой буфер обычно имеет объем в несколько мегабайт.

# 3. Продукты компании D-Link

## 3.1 Трехуровневая иерархическая модель сети

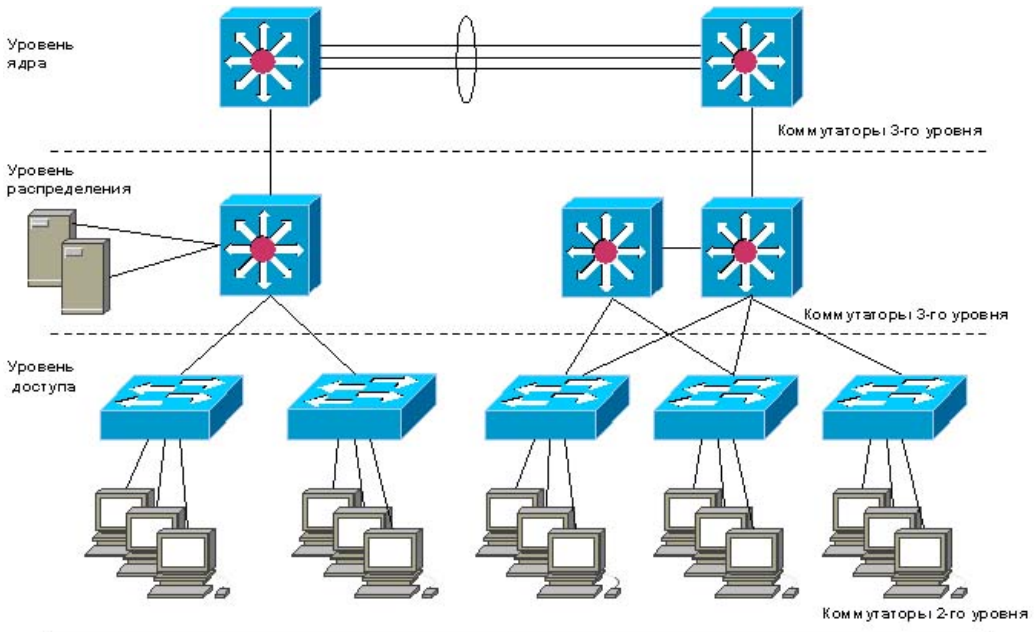



Рисунок 8. Трехуровневая модель сети.

Иерархическая модель определяет подход к проектированию сетей и включает в себя три логических уровня (рисунок 8):

- уровень доступа;
- уровень распределения;
- уровень ядра.

Для каждого уровня определены свои функции. Три уровня не обязательно предполагают наличия 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 OSI модели, 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**

*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**.

*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 адресов
- переход от одной технологии к другой (например, от 100Base-TX к 1000Base-T)
- объединение полос пропускания низкоскоростных каналов доступа в высокоскоростные магистральные каналы.

**Уровень доступа**

*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:

- продолжение (начиная с уровня распределения) управления доступом и политиками сети
- создание отдельных доменов коллизий (сегментация)
- подключение рабочих групп к уровню распределения
- уровень доступа использует технологию коммутируемых локальных сетей.

## 3.2 Продукты D-Link

### Коммутаторы уровня доступа

Уровень доступа является ближайшим к пользователю уровнем и предоставляет ему доступ к ресурсам сети. Размещенные на этом уровне коммутаторы должны поддерживать подключение отдельных компьютеров к объединенной сети.

Коммутаторы уровня доступа D-Link представлены следующими моделями:

**DES-1010G/1026G** – неуправляемые коммутаторы, которые обеспечивают каналы связи скоростью 10/100Мбит/с, имеют 2 порта 1000Мбит/с и возможность подключения до 26 пользователей для сетей малых и средних офисов.

**DGS-1005D/08D/16TL/24TL** – неуправляемые коммутаторы, которые обеспечивают гигабитные каналы связи для высокоскоростного подключения серверов и рабочих станций.

**DES-12xxR и DGS-12xxT** – настраиваемые коммутаторы, которые обеспечивают коммутируемые каналы 10/100 Мбит/с и 10/100/1000Мбит/с и поддерживающие до 24 пользователей и 2 порта Gigabit Ethernet для серверов.

**DES-3226/3226S/DHS-3226** – управляемые коммутаторы, предоставляющие при объединении в стек возможность подключения до 192 пользователей с помощью 10/100 Мбит/с каналов связи и 8 серверов через порты Gigabit Ethernet.

**DGS-3212SR/3312SR**- управляемые гигабитные коммутаторы, предоставляющие сетевому администратору гибкость в использовании их либо в качестве агрегирующего устройства стека (Master Switch), либо автономных модульных устройств, поддерживающих до 12 медных и оптических портов Gigabit Ethernet.

**Коммутаторы уровня распределения**

Коммутаторы уровня распределения служат местом концентрации для нескольких коммутаторов уровня доступа и должны справляться с большими объемами передаваемых данных.

Такие возможности имеют следующие коммутаторы D-Link:

**DES-3226S/3326S, DES-3250TG, DES-3350SR, DGS-3324SR**– многофункциональные, управляемые коммутаторы, которые поддерживают от 24 до 48 портов 10/100Мбит/с и от 2 до 4 портов 10/100/1000Мбит/с.

**DGS-3212SR/3312SR**- многофункциональные управляемые гигабитные коммутаторы, которые можно использовать в качестве агрегирующего устройства стека (Master Switch) и объединить с их помощью в стек до 12 коммутаторов DES-3226S, получив до 288 портов 10/100BASE-TX и до 12 портов Gigabit Ethernet.

**DES-6000/6300** – коммутаторы этой серии поддерживают до 128 портов 10/100 Мбит/с, до 96 оптических портов 100Base-FX, до 16 портов Gigabit Ethernet. Коммутаторы этой серии являются эффективным решением для уровня распределения. Они поддерживают большое количество интерфейсов для разных сред передачи и разных скоростей, имеют возможности резервирования и обладают функциональностью, необходимой этому уровню (фильтрация, маршрутизация, управление доступом).

**Коммутаторы уровня ядра**

Уровень ядра имеет высокую производительность. К коммутаторам это уровня можно отнести следующие модели:

**DES-6000/6300** – модульные высокопроизводительные коммутаторы, предназначенные для работы в сетях операторов связи.

**DES-7000** –Ethernet/EoVDSL коммутатор предназначенный для сетей крупных операторов связи, предоставляющий высокопроизводительную коммутацию и высокий уровень доступности.

# 4. Настройка коммутатора

## 4.1 Подключение к коммутатору

Перед тем, как начать настройку коммутатора, необходимо установить физическое соединение между коммутатором 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, устройству необходимо назначить IP-адрес.

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

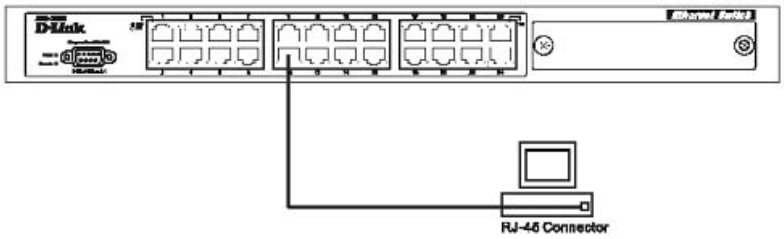



Рисунок 9. Подключение компьютера к коммутатору

Для подключения к другому коммутатору так же можно использовать любой четырехпарный кабель UTP категории 5, 5e, при условии, что порты коммутатора поддерживают автоматическое определение полярности. В противном случае надо использовать кроссовый кабель.

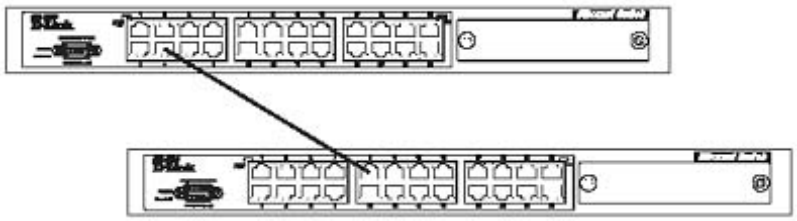


Рисунок 10. Подключение коммутатора к обычному (не -Uplink) порту коммутатора с помощью прямого или кроссового кабеля.

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 описание сигналов индикаторов необходимо обратиться к руководству пользователя коммутатора конкретной модели).

### 4.1.1 Подключение к локальной консоли коммутатора

Управляемые коммутаторы D-Link имеют консольный порт, который с помощью кабеля стандарта RS-232, входящему в комплект поставки, 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). Она может использоваться для установки и управления коммутатором, даже если нет подключения к сети.

После подключения к консольному порту необходимо следует запустить эмулятор терминала (например, программу HyperTerminal в Windows). В программе следует установить следующие параметры:

| | |
|---|---|
| Baud rate: | 9,600 |
| Data width: | 8 bits |
| Parity: | none |
| Stop bits: | 1 |
| Flow Control: | none |

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

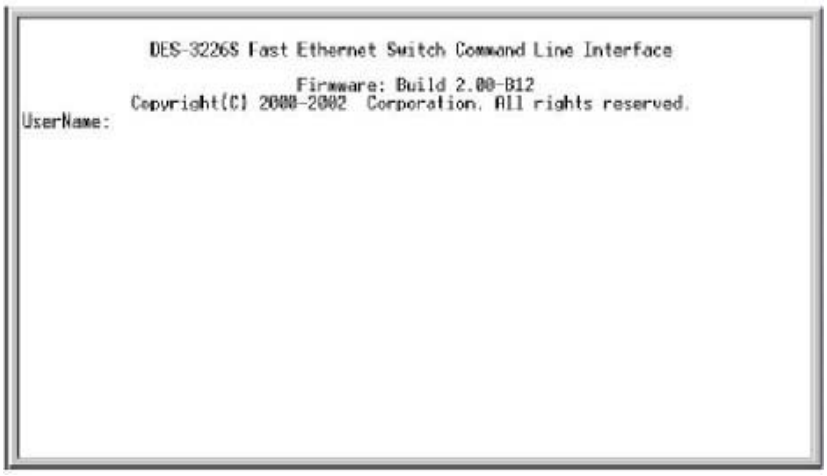



Рисунок 11. Первоначальное окно консоли.

Более старые модели коммутаторов, например, DES-3226, DHS-3226 имеют систему меню. Поэтому при подключении к коммутатору по консоли, окно будет таким, как приведено на рисунке ниже.

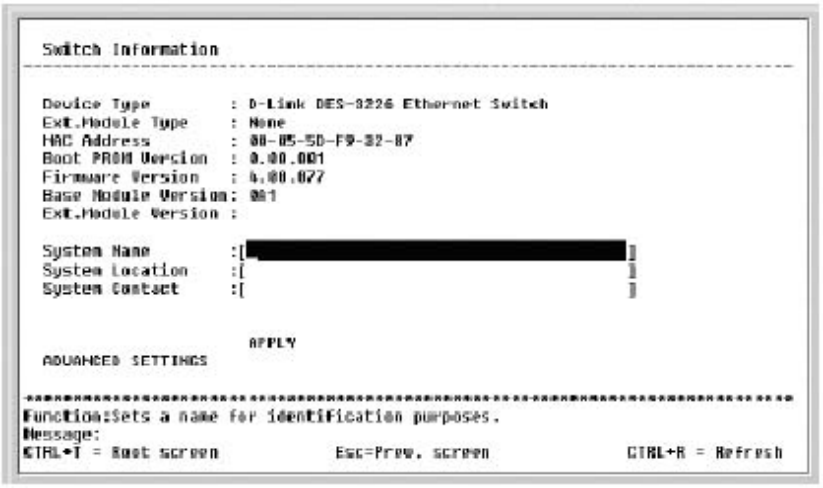



Рисунок 12. Система меню коммутатора.

Настройка коммутатора с помощью системы меню рассматриваться не будет, т.к. все современные модели коммутаторов поддерживают настройку с помощью интерфейса командной строки.

Если оно не появилось, нажмите Ctrl+r , чтобы обновить окно.

Коммутатор предложит ввести пароль. Первоначально не существует имени пользователя и пароля, поэтому нажмите дважды клавишу Enter, после чего в командной строке появится следующее приглашение, например DES-3226S#. Теперь можно вводить команды.


Рисунок 13. Строка приглашения для ввода команд CLI коммутатора.

### 4.2 Начальная конфигурация коммутатора

### 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 для того, чтобы вывести на экран список всех команд данного уровня.


Рисунок 14. 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 config, введите в командной строке:

DES-3226S**#config** + 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.


Рисунок 15. Результат вызова помощи о возможных параметрах команды config

**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 двух уровней привилегий: *Admin* или *User*. Учетная запись *Admin* имеет наивысший уровень привилегий.

Создать учетную запись пользователя можно с помощью следующих команд CLI:

DES-3226S# **create account admin/user** <*username*>

(знак «/» означает ввод или одного параметра, или другого)

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 от 0 до 15 символов.

После успешного создания учетной записи на экране появится слово *Success*.

Ниже приведен пример создания учетной записи с уровнем привилегий «Admin» и Username «dlink»:

DES-3226S#create account admin dlink

Command: create account admin dlink

Enter a case-sensitive new password:****

Enter the new password again for confirmation:****

Success.

DES-3226S#

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

DES-3226S# **config account** <*username*>

Ниже приведен пример создания нового пароля для учетной записи dlink:

DES-3226S#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:

DES-3226S# **show account**.

Ниже приведен пример выполнения этой команды.

DES-3226S#show account

Command: show account

Current Accounts:

Username Access Level

--- ---

dlink Admin

Удалить учетную запись можно, выполнив команду **delete account** <*username*>. На рисунке приведен пример удаления учетной записи dlink.

DES-3226S#delete account dlink

Command: delete account dlink

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:

DES-3226S# **config ipif System dhcp,**

DES-3226S# **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 .

Пример использования команды:

DES-3226S#config ipif System ipaddress 10.48.74.122/8

Command: config ipif System ipaddress 10.48.74.122/8

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, что автоопределение будет действовать некорректно и потребуется ручная установка скорости и режима.

Для установки параметров портов на коммутаторе D-Link можно воспользоваться командой **config ports**.

Ниже приведен пример установки скорости равной 10Мбит/с, дуплексного режима работы, обучения и состояния для портов коммутатора с 1 по 3.

```
DES-3226S#config ports 1-3 speed 10_full learning enable state enable
Command: config ports 1-3 speed 10_full learning enable state enable

Success
```

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

```
DES-3226S#show ports
Command show ports:

Port  Port       Settings                Connection              Address
      State Speed/Duplex/FlowCtrl  Speed/Duplex/FlowCtrl Learning
------ -------- ---------------------  ---------------------- --------
1      Enabled   Auto/Enabled            Link Down             Enabled
2      Enabled   Auto/Enabled            Link Down             Enabled
3      Enabled   Auto/Enabled            Link Down             Enabled
4      Enabled   Auto/Enabled            Link Down             Enabled
5      Enabled   Auto/Enabled            Link Down             Enabled
……..
10   Enabled   Auto/Enabled          100M/Full/802.3x      Enabled
```

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

```
DES-3226S#save
Command: save

Saving all settings to NV-RAM... 100%
done.
DES-3226S#
```

**Шаг 5**. Перезагрузка коммутатора с помощью команды reboot.

```
DES-3226S#reboot
Command: reboot
Are you sure 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**.

DES-3226S#reset config

Command: reset config

Success

**Шаг 6**. Просмотр конфигурации коммутатора.

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

DES-3226S#show switch

Command: show switch

Device Type : DES-3226S Fast-Ethernet Switch
Module Type : DES-332GS 1-port GBIC Gigabit Ethernet and 1 Stacking Port
Unit ID : 1
MAC Address : DA-10-21-00-00-01
IP Address : 10.41.44.22 (Manual)
VLAN Name : default
Subnet Mask : 255.0.0.0
Default Gateway : 0.0.0.0
Boot PROM Version : Build 0.00.001
Firmware Version : Build 4.00-B30
Hardware Version : 1B1
Device S|N :
System Name : DES-3226S_#3
System Location : 7th_flr_east_cabinet
System Contact : Julius_Erving_212-555-6666
Spanning Tree : Disabled
GVRP : Disabled
IGMP Snooping : Disabled
TELNET : Enabled (TCP 23)
WEB : Enabled (TCP 80)
RMON : Enabled
Asymmetric VLAN : Enabled

### 4.3 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 и обеспечивает графическое представление коммутатора в режиме реального времени и 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.

Для того, чтобы подключиться к HTTP серверу на коммутаторе необходимо выполнить следующие шаги, используя интерфейс командной строки:

1. Назначить коммутатору IP-адрес из диапазона адресов Вашей сети, используя команду:
   DES-3226S# **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

2. Проверить правильность настройки IP-адреса коммутатора с помощью команды:
   DES-3226S# **show ipif**

3. На рабочей станции запустить Web-браузер, в командной строке которого ввести IP- адрес коммутатора.

## 5. Дополнительные функции 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, вполне логично включить в него дополнительные функции, полезные при построении современных, расширяемых, надежных и гибких сетей. Большинство современных коммутаторов, независимо от производителя, поддерживают несколько дополнительных возможностей, отвечающих общепринятым стандартам. Среди них самые распространенные и наиболее используемые сегодня это:

1. Технологии Виртуальных Сетей – VLAN
2. Поддержка протокола Spanning Tree IEEE 802.1d и Rapid Spanning Tree IEEE 802.1w
3. Объединение каналов Ethernet
4. Поддержка SNMP – управления
5. Обеспечение функции Port Security, или привязка MAC-адреса к определенному порту
6. Поддержка 802.1x
7. Протоколы группового вещания
8. Управление потоком и др.

В настоящее время одной из самых важных характеристик любой компьютерной сети помимо производительности является надежность каналов связи — в связи с развитием электронного бизнеса и повышения роли компьютерной связи при ведении практически уже любого рода коммерческой деятельности, даже в небольшой сети может стать причиной колоссальных убытков компании. Поэтому следует обратить особое внимание на те функции сетевого оборудования, которые позволяют обеспечивать отказоустойчивость сети, ее надежность и защищенность от несанкционированного доступа.

## 6. Виртуальные локальные сети VLAN

Всем коммутируемым сетям присуще одно ограничение. Поскольку коммутатор не имеет дел с протоколами сетевого уровня, он не может знать, куда направлять их широковещательные пакеты. Хотя трафик с конкретными адресами (соединения "точка-точка") изолирован парой портов, широковещательные пакеты передаются во всю сеть (на каждый порт). *Широковещательные пакеты* – это пакеты, 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 широковещательные пакеты могут возникать из-за некорректно работающего сетевого адаптера. Широковещательные пакеты могут привести к насыщению 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 ЛВС (Virtual LAN, VLAN)**.

**Виртуальной 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 пакетов и вызываемых ими следствий, которые могут развиться в широковещательные штормы и существенно снизить производительность сети.

Итак, 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 усилить безопасность сети, определив с помощью фильтров, настроенных на коммутаторе или маршрутизаторе, политику взаимодействия пользователей из разных виртуальных сетей.

### 6.1 Типы VLAN

В коммутаторах могут использоваться три типа VLAN:

1. VLAN на базе портов
2. VLAN на базе MAC-адресов.
3. VLAN на основе меток в дополнительном поле кадра – стандарт IEEE 802.1q

### 6.2 VLAN на базе портов.

При использовании VLAN на базе портов, 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.

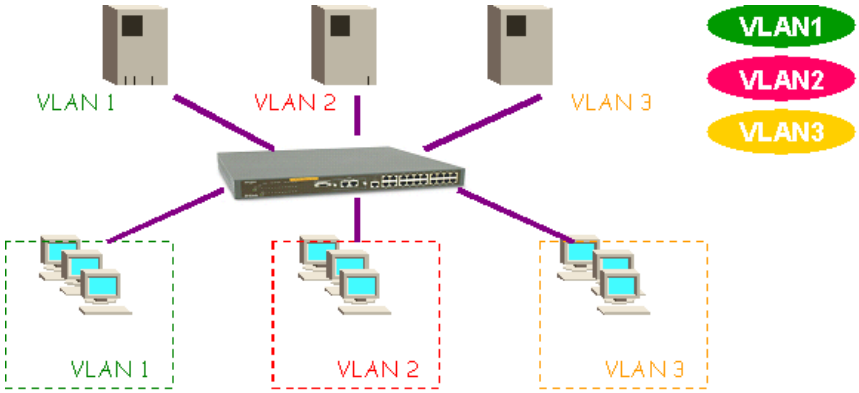



Рисунок 16. VLAN на базе портов.

Основные характеристики:

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

каждому порту, находящемуся в одной VLAN, присвоить один и тот же 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 в новой VLAN. Таким образом, VLAN обеспечивают гибкость при перемещениях, изменениях и наращивании сети.

4. **Каждый порт может входить только в один VLAN.** Поэтому для объединения виртуальных подсетей – как внутри одного коммутатора, так и между двумя коммутаторами, нужно использовать сетевой уровень. Один из портов каждого VLAN подключается к интерфейсу маршрутизатора, который создает таблицу маршрутизации для пересылки пакетов из одной подсети в другую (IP адреса подсетей должны быть разными).

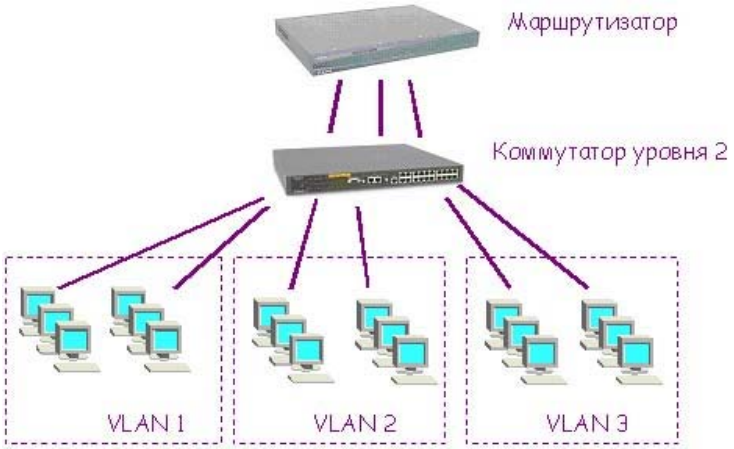



Рисунок 17. Объединение VLAN на 3-м уровне.

Недостатком такого решения является то, что один порт каждого VLAN'а необходимо подключать к маршрутизатору, при этом порты и кабели используются очень расточительно, плюс затраты на маршрутизатор. Решить данную проблему можно двумя способами: во-первых, использовать коммутаторы, которые на основе фирменного решения позволяют включать порт в несколько VLAN. Второе решение заключается в использовании коммутаторов 3-го уровня.

### 6.3 VLAN на базе MAC-адресов.

Следующий способ, который используется для образования виртуальных сетей, основан на группировке MAC-адресов. При существовании в сети большого количества узлов этот способ требует выполнения большого количества ручных операций от администратора. Однако он оказывается более гибким при построении виртуальных сетей на основе нескольких коммутаторов, чем способ группировки портов. Группирование MAC-адресов в сеть на каждом коммутаторе избавляет от необходимости их связи несколькими портами, однако, требует выполнения большого количества ручных операций по маркировке MAC-адресов на каждом коммутаторе сети.

Широковещательные домены на базе MAC-адресов, позволяют физически перемещать станцию (подключать к любому порту коммутатора), позволяя оставаться ей в одном и том же широковещательном домене без каких-либо изменений в настройках конфигурации

Настройка виртуальной сети на основе MAC-адресов может отнять много времени - представьте себе, что вам потребуется связать с VLAN адреса 1000 устройств. Кроме того, MAC-адреса "наглухо зашиты" в оборудование, и может потребоваться много времени на выяснение адресов устройств в большой, территориально распределенной сети.

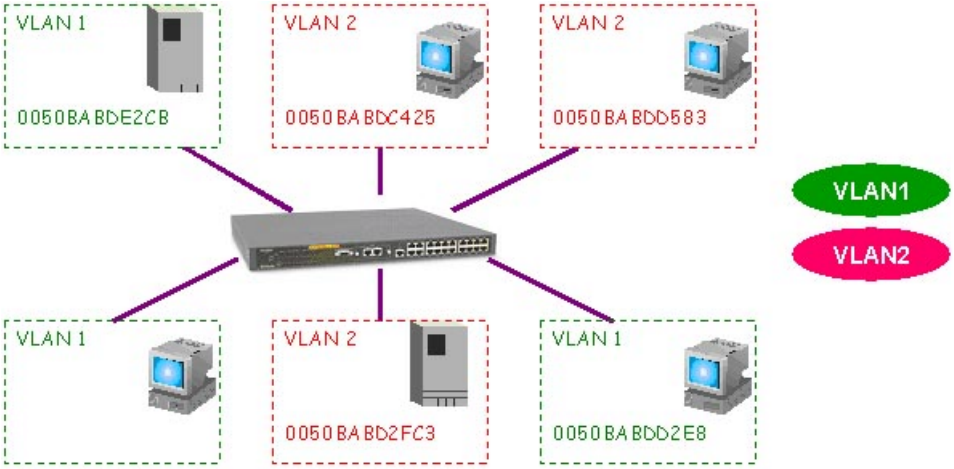



Рисунок 18. VLAN на базе MAC-адресов.

### 6.4 VLAN на базе меток – стандарт 802.1q.

Описанные два подхода основаны только на добавлении дополнительной информации к адресным таблицам моста и не используют возможности встраивания информации о принадлежности кадра к виртуальной сети в передаваемый кадр. Метод организации VLAN на основе меток – тэгов, использует дополнительные поля кадра для хранения информации о принадлежности кадра при его перемещениях между коммутаторами сети.

Стандарт IEEE 802.1q определяет изменения в структуре кадра Ethernet, позволяющие передавать информацию о VLAN по сети.

К кадру Ethernet добавлены четыре байта. Первые 2 байта с фиксированным значение 0x8100 определяют, что кадр содержит тег протокола 802.1q/802.1p. Остальные 2 байта содержат следующую информацию:

- 3 бита приоритета передачи кодируют до восьми уровней приоритета (от 0 до 7, где 7-наивысший приоритет), которые используются в стандарте 802.1p;
- 1 бит Canonical Format Indicator (CFI), который зарезервирован для обозначения кадров сетей других типов (Token Ring, FDDI), передаваемых по магистрали Ethernet;
- 12-ти битный идентификатор VLAN, определяющий, какой VLAN принадлежит трафик.

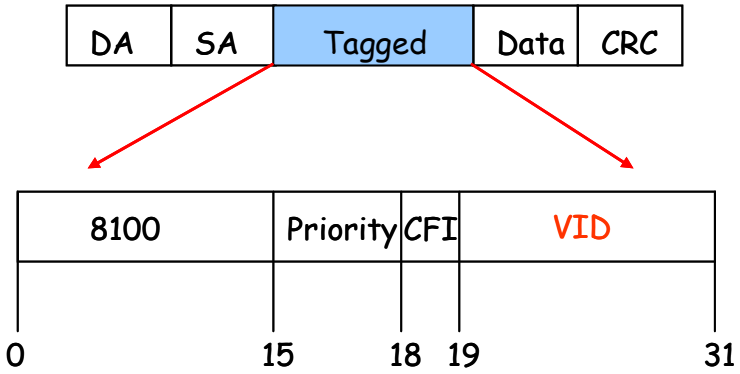



Рисунок 19. Маркированный кадр Ethernet.

С точки зрения удобства и гибкости настроек, VLAN на основе меток является лучшим решением. 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 802.1q. Способность добавления меток позволяет VLAN распространяться через множество 802.1q-совместимых коммутаторов по одному физическому соединению.

2. Позволяет активизировать алгоритм покрывающего 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, т.о., автоматически предотвращается возникновение петель в сети.
3. Способность VLAN 802.1q добавлять и извлекать метки из заголовков пакетов позволяет VLAN работать с коммутаторами и сетевыми адаптерами серверов и рабочих станций, которые не распознают метки.
4. Устройства разных производителей, поддерживающие стандарт могут работать вместе, т.е. не зависимо от какого-либо фирменного решения.
5. Не нужно применять маршрутизаторы, чтобы связать подсети на сетевом уровне, достаточно включить нужные порты в несколько VLAN для возможности обмена трафиком. Например, для обеспечения доступа к серверу из различных VLAN, нужно включить порт коммутатора, к которому подключен сервер во все подсети . Единственное ограничение – сетевой адаптер сервера должен поддерживать стандарт IEEE 802.1q.

В силу указанных свойств, VLAN на базе тэгов используются на практике гораздо чаще остальных типов, поэтому остановимся подробно на принципах работы такой схемы и вариантов, которые можно с ее помощью организовать.

Существуют два основных понятия для понимания IEEE 802.1q VLAN:

1. VLAN-идентификатор порта - Port VLAN ID (PVID)
2. Номер VLAN ID (VID)

**PVID** определят, в какую VLAN коммутатор направит **немаркированный** пакет с подключенного к порту сегмента, когда пакет нужно **передать** на другой порт. С другой стороны, пользователь может определить порт, как входящий в несколько VLAN, позволяя сегменту, подключенному к данному порту **принимать маркированные** пакеты от нескольких VLAN в сети. В этом случае, для дальнейшей обработки пакета используется поле **VID** в кадре Ethernet, определяющее, в какую VLAN будет отправлен этот пакет. Таким образом, эти два параметра контролируют способность порта принимать и передавать VLAN-трафик и различия между ними обеспечивают сегментацию сети с одновременным сохранением возможности получать доступ к общим сетевым ресурсам из различных VLAN..

Для примера рассмотрим ситуацию (рисунок 19): Порт 1 входит в VLAN 1 и имеет PVID=1. Если пакет нужно передать на другой порт, например Порт 3 (найденный обычным способом в таблице коммутатора), то коммутатор, прежде чем передать пакет смотрит, входит ли Порт 3 в VLAN 1, и может ли соответственно получать пакеты, предназначенные для этого VLAN. Если Порт 3 не является членом VLAN 1, то пакет отбрасывается коммутатором и соответственно не будет передан получателю. Если Порт 3 входит в VLAN 1, то пакет будет передан. Таким образом, Порт 1 может передавать и принимать пакеты для VLAN 1, т.к. его PVID=1. Порт 3, у которого PVID может быть другим, может принимать пакеты из VLAN 1, т.к. входит в этот VLAN, но он не может передавать пакеты в VLAN 1, пока его PVID не будет установлен в 1.

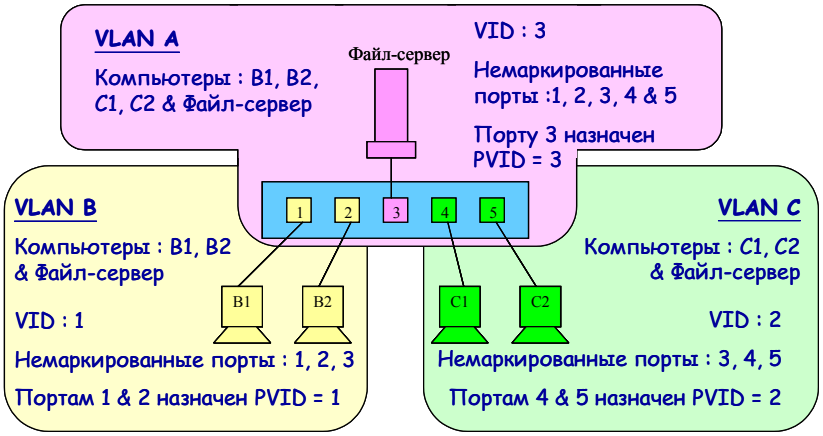



Рисунок 19. Пример.

VLAN'ы могут работать между несколькими коммутаторами в вашей сети. Следует учитывать два момента: во-первых, поддерживает ли коммутатор стандарт IEEE 802.1q и должны ли быть VLAN-пакеты маркированы – *Tagged* или немаркированы – *Untagged*.
Вот определения некоторых терминов, необходимых для понимания работы VLAN в сети:

- **Tagging (Маркировка пакета) –** процесс добавления информации о принадлежности к 802.1q VLAN в заголовок кадра. Порты, на которых включена маркировка пакетов, могут добавлять в заголовки всех передаваемых пакетов номер VID, информацию о приоритете и пр. Если пакет приходит на порт уже маркированным, то данный пакет не изменяется и таким образом при пересылке сохраняется вся информация о VLAN. Маркировка пакетов в основном применяется для пересылки пакетов между устройствами, поддерживающими стандарт 802.1q VLAN.
-

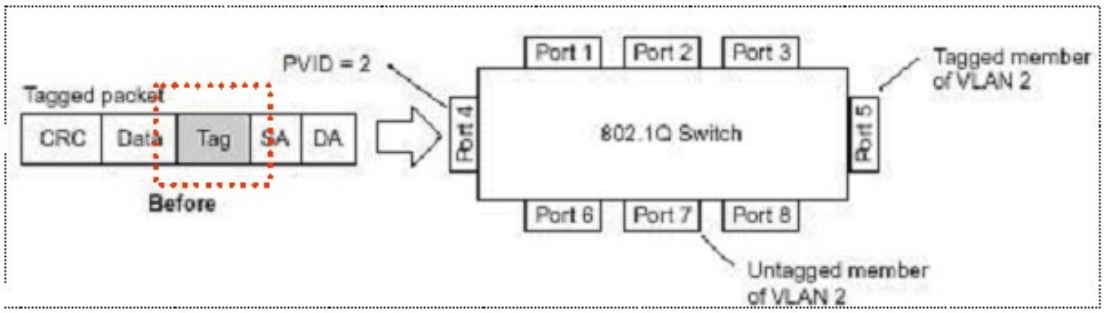



Рисунок 20. Маркированный пакет.

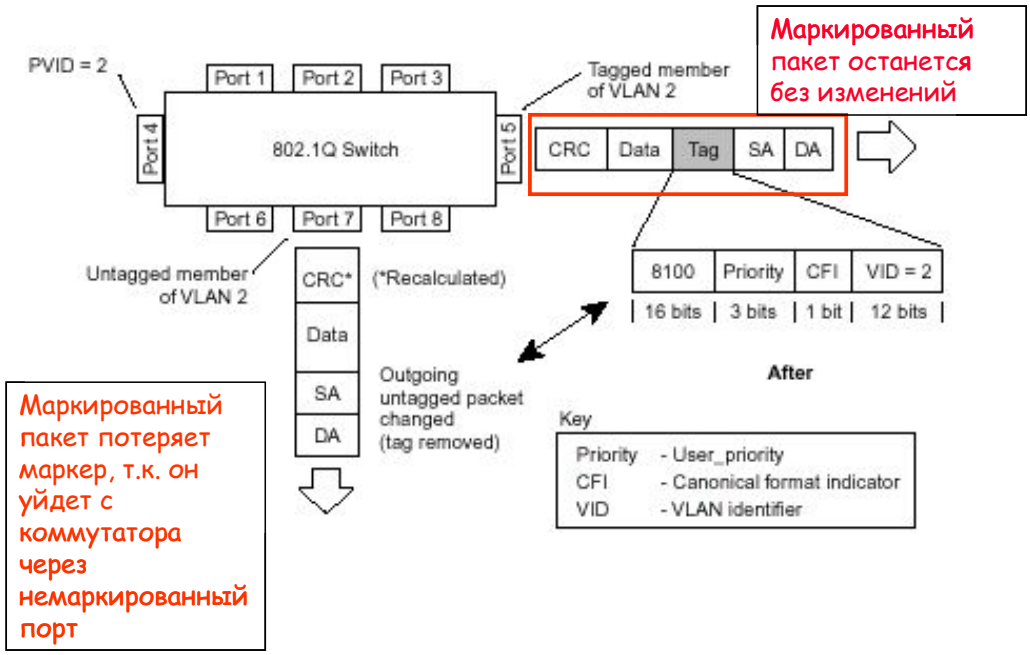



Рисунок 21. Маркированный пакет, выходящий через маркированный и немаркированный порт.

- **Untagging** –Процесс извлечения информации 802.1q VLAN из заголовка пакета. Порты, на которых включена данная функция, извлекают все информацию, касающуюся VLAN из заголовков, как входящих, так и исходящих пакетов, проходящих через данный порт. Если же пакет не содержит тэг VLAN'а, то порт не изменяет такой пакет. Данная функция коммутатора применяется при передаче пакетов от коммутаторов, поддерживающих стандарт 802.1q на устройства, не поддерживающие этот стандарт.

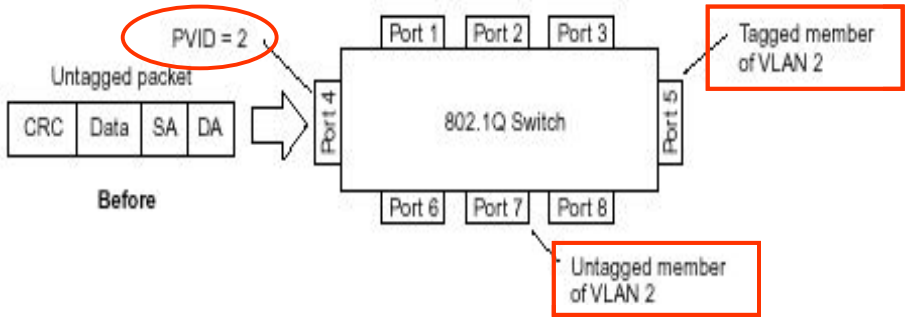



Рисунок 22. Немаркированный пакет.

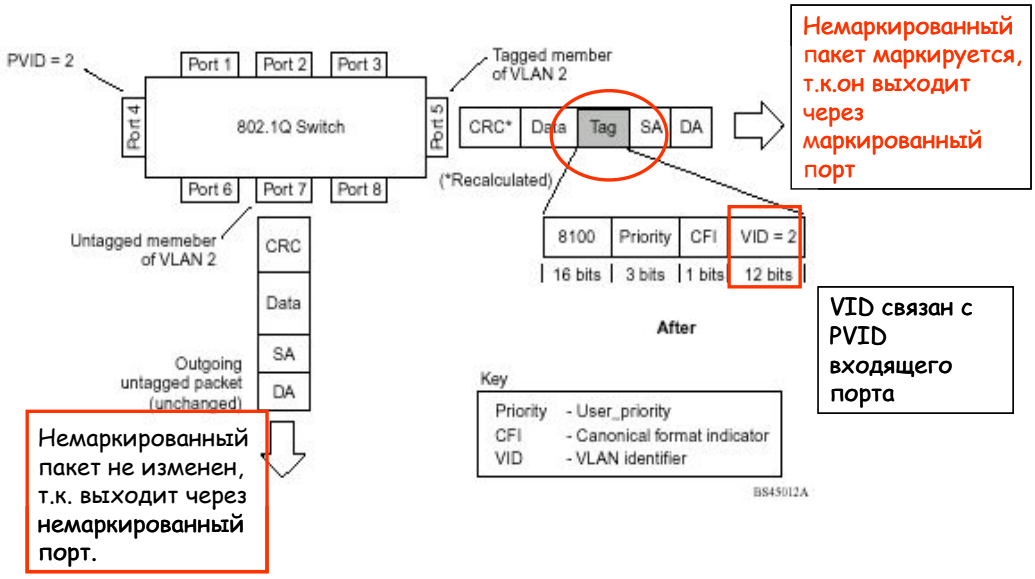



Рисунок 23. Немаркированный пакет, выходящий через маркированный и немаркированный порт.

Для согласования работы устройств, поддерживающих формат кадра 802.1 Q, с теми устройствами, которые не понимают этот формат, разработчики стандарта предложили делить весь трафик в сети на несколько типов.

- **Трафик входного порта** (Ingress Port). Каждый кадр, достигающий коммутируемой сети и идущий либо от маршрутизатора, либо от рабочей станции, имеет определенный порт-источник. На основании его номера коммутатор должен "принять решение" о приеме (или отбрасывании) кадра и передаче его в ту или иную VLAN. Коммутатор проверяет пакет на наличие информации VLAN и на ее основании принимает решение о пересылке пакета.

  Если пакет содержит информацию о VLAN, входной порт сначала определяет, является ли он сам членом данного VLAN. Если нет, то пакет отбрасывается. Если да, то определяется, является ли порт назначения членом данного VLAN. Если оба порта являются членами одного VLAN'а, то пакет пересылается.

  Если пакет не содержит в заголовке информацию VLAN, т.е. является немаркированным пакетом (untagged), то входящий порт добавляет в заголовок пакета метку в соответствии со своим PVID (если он является маркированным портом (tagged)). Затем определяется, принадлежат ли входной порт и порт назначения одному VLAN (имеют одинаковые VID). Если нет, пакет отбрасывается. Если да, то пакет передается.

  Если же входящий порт является немаркированным портом, то перед пересылкой проверяется только, являются ли входной порт и порт назначения членами одной VLAN.

Этот процесс называется *ingress filtering*(входной фильтрацией) и используется для сохранения пропускной способности внутри коммутатора .

- **Трафик выходного порта** (Egress Port). Чтобы попасть в межсетевой маршрутизатор или в оконечную рабочую станцию, кадр должен выйти за пределы коммутатора сети. Коммутатор "решает", какому порту (или портам) нужно передать пакет и есть ли необходимость удалять из него служебную информацию, предусмотренную стандартом 802.1q. Дело в том, что традиционные рабочие станции не всегда воспринимают информацию о VLAN по стандарту 802.1q, но сервер, обслуживающий несколько подсетей с помощью единственного интерфейса, должен ее активно использовать. Если выходной порт сервера подключен к коммутатору, поддерживающему стандарт 802.1q, то следует включить маркировку пакетов на данном порту, чтобы другой коммутатор мог получать информацию о VLAN и на ее основе принимать решения о передаче пакета. Если выходной порт подключен к устройству, не поддерживающему стандарт 802.1q, то тэги должны извлекаться из заголовка пакета, и теперь уже обычный пакет Ethernet может быть принят конечным устройством.

**Поддержка VLAN между 802.1q-совместимыми коммутаторами**

Если имеется несколько коммутаторов, поддерживающих 802.1q и необходимо настроить между ними VLAN, то в таком случае можно использовать маркировку пакетов. Маркировка пакетов добавляет информацию о 802.1q VLAN в заголовок каждого пакета, позволяя другому коммутатору, поддерживающему 802.1q, передавать пакет по назначению. Таким образом, можно использовать возможности стандарта 802.1q и строить сеть на нескольких коммутаторах с поддержкой тэгов, информации о приоритете пакета и др.

Для того, чтобы устройства одной VLAN могли обмениваться данными с устройствами другой VLAN, виртуальные локальные сети необходимо объединить через устройство 3-го уровня, поддерживающее маршрутизацию. Это может быть или отдельный маршрутизатор, или коммутатор, поддерживающий функции 3-го уровня.

## 6.5 Создание VLAN с помощью команд CLI

В таблице приведены команды CLI и их синтаксис, используемые при создании, удалении и управлении виртуальными локальными сетями.

| Команда | Параметры | Описание |
|---|---|---|
| create vlan | <vlan_name 32><br>tag <vlanid><br>advertisement | Создать VLAN |
| delete vlan | <vlan_name 32> | Удалить VLAN |
| config vlan | <vlan_name 32><br>add [tagged\|untagged\|forbidden]<br>delete <portlist><br>advertisement [enable\|disable] | Настроить параметры VLAN |
| config vlan | <vlan_name 32><br>delete <portlist> | Исключить заданные порты из VLAN |
| config vlan | <vlan_name 32> | |
| config gvrp | <portlist><br>all<br>state [enable\|disable]<br>ingress_checking [enable\|disable]<br>acceptable_frame<br>[tagged_only\|accept_all] | Настроить параметры GVRP |
| enable gvrp | | Активизировать GVRP |
| disable gvrp | | Отключить GVRP |
| show vlan | <vlan_name 32> | Показать созданные VLAN и их настройки |
| show gvrp | <portlist> | Показать настройки GVRP |

*Пример 1*. Создание VLAN на коммутаторе.
Создать VLAN с именем v1 на коммутаторе и идентификатором (PVID) равным 2.

```
DES-3226S#create vlan v1 tag 2
Command: create vlan v1 tag 2

Success.
```

*Пример 2*. Удаление VLAN.
Удалить VLAN с именем v1.

```
DES-3226S#delete vlan v1
Command: delete vlan v1

Success.
```

*Пример 3*. Добавить дополнительные порты к ранее сконфигурированному VLAN. Добавить порты с 4 по 8 коммутатора в VLAN v1. Сделать порты маркированными.

```
DES-3226S#config vlan v1 add tagged 4-8
Command: config vlan v1 add tagged 4-8

Success.
```

*Пример 4*. Проверка правильности настройки VLAN на коммутаторе.

```
DES-3226S#show vlan
Command: show vlan

VID :            1          VLAN Name :      default
VLAN TYPE :      static     Advertisement :  Enabled
Member ports :   1-26,  1-26
Static ports :   1-26,  1-26
Untagged ports : 1-25,  1-25
Forbidden ports :
VID :            2          VLAN Name :      v1
VLAN TYPE :      static     Advertisement :  Disabled
Member ports :   26,  26
Static ports :   26,  26
Untagged ports :
Forbidden ports :
Total Entries :  2
```

### 6.6 Асимметричные VLAN

С целью более эффективного использования разделяемых ресурсов, таких как серверы или Интернет- шлюзы, в новом программном обеспечении коммутаторов D-Link реализована поддержка Asymmetric VLAN. Асимметричные виртуальные локальные сети могут быть настроены для того, чтобы позволить серверу (или нескольким серверам) взаимодействовать с разными клиентами через один физический канал связи с коммутатором. Клиенты, при этом, будут полностью изолированы друг от друга. Например, асимметричные VLAN могут быть настроены таким образом, 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.

Основное различие между базовым стандартом 802.1q VLAN или симметричными VLAN и асимметричными VLAN заключается в том, как выполняется отображение адресов. Симметричные VLAN используют отдельные адресные таблицы, и таким образом не существует пересечения адресов между VLAN-ами. Асимметричные VLAN могут использовать одну, общую таблицу адресов. Однако, использование одних и тех же адресов (пересечение по адресам) происходит только в одном направлении. В примере, рассмотренном выше, VLAN, созданная для порта, подключенного к почтовому серверу, имела в своем распоряжении полную таблицу адресов, т.о. любой адрес мог быть отображен на ее порт (PVID).

На использование асимметричных VLAN существуют следующие ограничения:

- Поддержка асимметричных VLAN ограничена автономными коммутаторами.
- Каждый порт должен быть немаркированным.
- GVRP и IGMP Snooping не поддерживаются.

При активизации асимметричных VLAN, уникальный PVID назначается всем портам, создавая отдельную VLAN для каждого порта. Каждый порт при этом, может получать кадры от VLAN по умолчанию. Асимметричные VLAN по умолчанию отключены.

В таблице приведены команды для настройка асимметричных VLAN на коммутаторе с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| enable asymmetric_vlan | | Глобально активизировать асимметричные VLAN.<br>Уникальный PVID назначается всем портам, создавая отдельную VLAN для каждого порта. |
| disable asymmetric_vlan | | Глобально отключить асимметричные VLAN. Asymmetric VLAN отключены по умолчанию. |
| show asymmetric_vlan | | Просмотр статуса Asymmetric VLAN. |

*Пример 1*. Включить асимметричные VLAN.

```
DES-3226S#enable asymmetric_vlan

Command: enable asymmetric_vlan


Success.
```

*Пример 2*. Отключить асимметричные VLAN.

```
DES-3226S#disable asymmetric_vlan

Command: disable asymmetric_vlan


Success.
```

*Пример 3*. Просмотр статуса асимметричных VLAN.

```
DES-3226S# show asymmetric_vlan
Command: show asymmetric_vlan

Asymmetric Vlan : Enabled
```

## 7. Объединение портов и создание высокоскоростных сетевых магистралей

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. Режим резервирования, когда одно из них функционирует, а остальные находятся в "горячем" резерве для замены отказавшего соединения – это протокол Spanning Tree.
2. Режим баланса нагрузки; при этом данные передаются параллельно по всем альтернативным соединениям

Рассмотрим подробно каждый способ.

**Объединение портов (Port Trunking)** - это объединение нескольких физических каналов (*Link Aggregation*) в одну логическую магистраль. Используется для объединения нескольких портов вместе для образования высокоскоростного канала передачи данных и позволяет активно задействовать избыточные альтернативные связи в локальных сетях.

В отличие от протокола STP (Spanning Tree – протокол покрывающего дерева), при агрегировании физических каналов все избыточные связи остаются в рабочем состоянии, а имеющийся трафик распределяется между ними для достижения баланса нагрузки. При отказе одной из линий, входящих в такой логический канал, трафик распределяется между оставшимися линиями.

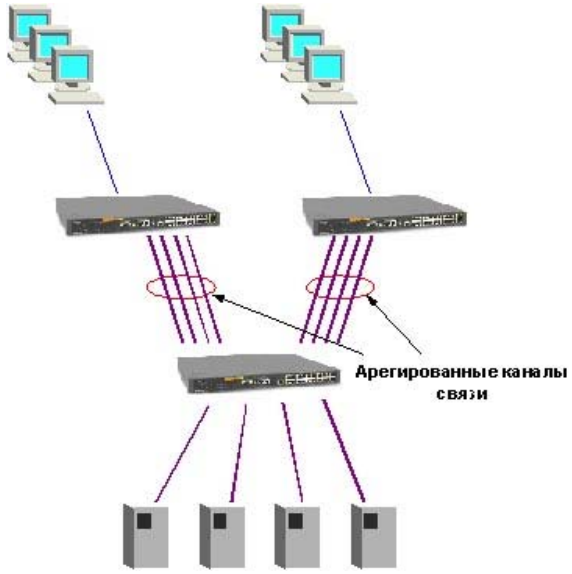



Рисунок 24. Агрегированные каналы связи между коммутаторами.

Включенные в агрегированный канал порты называются членами группы. Один из портов в группе выступает в качестве “связывающего”. Поскольку все члены группы в агрегированном канале должны быть настроены для работы в одинаковом режиме, все изменения настроек, произведенные по отношению к “связывающему” порту, относятся ко всем членам группы. Таким образом, для настройки портов в группе необходимо только настроить “связывающий” 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 протокола OSI. Например, если два или более смежных кадра 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 последовательность их не изменится. Обычно при статическом распределении выбор порта для конкретного сеанса выполняется на основании некоторых признаков поступающих пакетов (на основе выбранного алгоритма агрегирования портов). Как правило, это MAC-адреса источника или назначения, либо оба вместе.

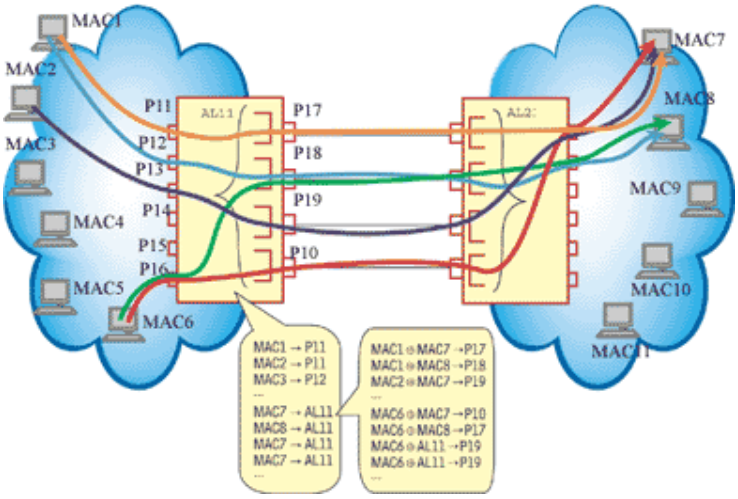



Рисунок 25. Распределение потоков данных по каналам агрегированной линии связи.

Агрегированные линии связи можно организовать с любым другим коммутатором, поддерживающим потоки данных точка-точка по одному порту агрегированного канала.

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

Программное обеспечение коммутаторов D-Link DES-3226S, DES-3326S, DES-3250TG, DGS-3324SR и др. поддерживает два типа агрегирования каналов связи: статическое и динамическое. При статическом агрегировании каналов (установлено по умолчанию), все настройки на коммутаторах выполняются вручную. Динамическое агрегирование каналов основано на спецификации IEEE 802.3ad, которая использует протокол контроля агрегированных линий связи (Link Aggregation Control Protocol) для того, чтобы проверять конфигурацию каналов и направлять пакеты в каждую из физических линий. Кроме этого, протокол LACP описывает механизм добавления и изъятия каналов из единой линии связи. Для этого, при настройке на коммутаторах агрегированного канала связи, соответствующие порты одного коммутатора должны быть сконфигурированы как «активные», а другого коммутатора как «пассивные». «Активные» порты LACP выполняют обработку и рассылку его управляющих кадров. Это позволяет устройствам, поддерживающим LACP, договориться о настройках агрегированного канала и иметь возможность динамически изменять группу портов, т.е. добавлять или исключать из нее порты. «Пассивные» порты обработки управляющих кадров LACP не выполняют.

Стандарт IEEE 802.3ad применим для всех типов Ethernet-каналов, и с его помощью поэтому можно строить даже многогигабитные линии связи, состоящие из нескольких каналов Gigabit Ethernet.

## 7.1 Создание агрегированного канала с помощью команд CLI

В таблице приведены команды, необходимые для объединения портов Ethernet.

| Команда | Параметры | Описание |
|---|---|---|
| create link_aggregation | group_id <value> {type[lacp/static]} | Создать агрегированный канал, динамически или статически |
| delete link_aggregation | group_id <value> | Удалить агрегированный канал |
| config | group_id <value> | Настроить параметры |

| | | |
|---|---|---|
| link_aggregation | master_port <port><br>ports <portlist><br>state [enabled\|disabled] | агрегированного канала |
| config<br>link_aggregation<br>algorithm | mac_source<br>mac_destination<br>mac_source_dest<br>ip_source<br>ip_destination<br>ip_source_dest | Задать алгоритм агрегирования |
| show<br>link_aggregation | group_id <value><br>algorithm | Проверка правильности настроек агрегированного канала |
| config lacp_ports | <portlist> mode [active\|passive] | Настройка портов LACP |
| show lacp_ports | {<portlist>} | Проверка правильности настроек портов LACP |

*Пример 1*. Создание группы агрегированного канала на коммутаторе.

```
DES-3226S#create link_aggregation group_id 1
Command: create link_aggregation group_id 1

Success.
```

*Пример 2*. Удалить ранее созданную группу агрегированного канала.

```
DES-3226S#delete link_aggregation group_id 1
Command: delete link_aggregation group_id 1

Success.
```

*Пример 3*. Настройка созданной группы. Включить порты 5-7, 9 коммутатора в группу агрегированного канала 1, порт 5 сделать «связующим» портом.

```
DES-3226S#config link_aggregation group_id 1 master_port 5 ports 5-7,9
Command: config link_aggregation group_id 1 master_port 5 ports 5-7,9

Success.
```

*Пример 4*. Задать алгоритм агрегирования портов.

Задать алгоритм агрегирования портов, распределяющий трафик по портам агрегированного канала на основе для MAC-адреса источника и приемника.

```
DES-3226S:4#config link_aggregation algorithm
mac_source_dest
Command: config link_aggregation algorithm mac_source_dest

Success.
```

*Пример 5*. Просмотр конфигурации группы агрегированного канала.

```
DES-3226S:4#show link_aggregation
Command: show link_aggregation

Link Aggregation Algorithm = MAC-source-dest
Group ID :      1
Master Port :   5
Member Port :  5-7, 9
Status :            Disabled
Flooding Port : 5
```

*Пример 6*. Создание группы агрегированного канала в соответствии со стандартом 802.3ad.

*Примечание. Для того, чтобы использовать протокол LACP, оба устройства должны поддерживать стандарт IEEE 802.3ad.

```
DES-3226S#create link_aggregation group_id 1 type lacp
Command: create link_aggregation group_id 1 type lacp

Success.
```

*Пример 7*. Настройка «активных» портов LACP на коммутаторе.

```
DES-3226S#config lacp_port 1-12 mode active
Command: config lacp_port 1-12 mode active

Success.
```

*Пример 8*. Просмотр режимов работы портов LACP.

```
DES-3226S#show lacp_ports
Command: show lacp_ports
Port   Activity
------   --------
1        Active
2       Active
3       Active
4       Active
5       Active
6       Active
7       Active
8       Active
9       Active
10     Active
11     Active
12     Active
```

# 8. Spanning Tree Protocol (IEEE 802.1d)

Второй метод, использующийся для повышения отказоустойчивости компьютерной сети, это Spanning Tree Protocol. Разработанный достаточно давно, в 1983 г., он до сих пор остается актуальным. В сетях Ethernet, коммутаторы поддерживают только древовидные связи, т.е. которые не содержат петель. Это означает, что для организации альтернативных каналов требуются особые протоколы и технологии, выходящие за рамки базовых, к которым относится Ethernet.

Алгоритм *Spanning Tree (STA)* позволяет коммутаторам автоматически определять древовидную конфигурацию связей в сети при произвольном соединении портов между собой.

Коммутаторы, поддерживающие протокол STP автоматически создают древовидную конфигурацию связей без петель в компьютерной сети. Такая конфигурация называется покрывающим деревом - Spanning Tree (иногда ее называют остовым деревом). Конфигурация покрывающего дерева строится коммутаторами автоматически с использованием обмена служебными пакетами

Рассмотрим подробно работу протокола STP.

Алгоритм STA требует, чтобы каждому мосту был присвоен идентификатор. *Идентификатор моста*– 8-байтное поле, которое состоит из 2-х частей: 2-байтного приоритета, назначенного администратором и 6 байтного MAC-адреса его блока управления. Каждому порту также назначается уникальный идентификатор в пределах моста, как правило, это его MAC-адрес. Каждому порту моста ставится в соответствие стоимость маршрута, соответствующая затратам на передачу кадра по локальной сети через данный порт.

Процесс вычисления связующего дерева начинается с выбора *корневого моста (root switch),* от которого будет строиться дерево. В качестве корневого моста выбирается коммутатор с наименьшим значением идентификатора. (Первоначально, по умолчанию, все коммутаторы имеют одинаковое значение приоритета, равное 32768. В этом случае, корневой коммутатор определяется по наименьшему MAC-адресу.) Иногда, такой выбор может оказаться далеко не рациональным. Для того чтобы в качестве корневого моста было выбрано определенное устройство (исходя из структуры сети), администратор может повлиять на процесс выборов, присвоив соответствующему коммутатору наименьший идентификатор вручную.

Второй этап работы STP – выбор *корневого порта (root port)* для каждого из остальных коммутаторов сети.

Корневой порт коммутатора – это порт, который имеет по сети кратчайшее расстояние до корневого коммутатора.

Третий шаг работы STP – определение назначенных портов.

Каждый сегмент в коммутируемой сети имеет один *назначенный порт (designated port)*. Этот порт функционирует как единственный порт моста, т.е. принимает пакеты 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 имеет наименьшее расстояние до корневого моста, среди всех портов, подключенных к данному сегменту. Назначенный порт у сегмента может быть только один. У корневого моста все порты являются назначенными, а их расстояние до корня полагается равным нулю. Корневого порта у корневого моста нет.

При построении покрывающего дерева важную роль играет понятие расстояния. По этому критерию выбирается единственный порт, соединяющий каждый коммутатор с корневым коммутатором, и единственный порт, соединяющий каждый сегмент сети с корневым коммутатором. Все остальные порты переводятся в резервное состояние, то есть такое, при котором они не передают обычные кадры данных. При таком выборе активных портов в сети исключаются петли и оставшиеся связи образуют покрывающее дерево.

В качестве расстояния в STA используется метрика *стоимость пути* (Path Cost) – она определяется как суммарное условное время на передачу данных от порта данного коммутатора

до порта корневого коммутатора. *Условное время сегмента* рассчитывается как время передачи одного бита информации через канал с определенной полосой пропускания. В таблице приводятся типичные стоимости пути в соответствии со стандартом IEEE 802.1d:

| Параметр | Скорость канала | Рекомендованное значение | Рекомендованный диапазон | Диапазон |
|---|---|---|---|---|
| Стоимость пути | 4 Мбит/с | 250 | 100-1000 | 1-65535 |
| Стоимость пути | 10 Мбит/с | 100 | 50-600 | 1-65535 |
| Стоимость пути | 16 Мбит/с | 62 | 40-400 | 1-65535 |
| Стоимость пути | 100 Мбит/с | 19 | 10-60 | 1-65535 |
| Стоимость пути | 1 Гбит/с | 4 | 3-10 | 1-65535 |
| Стоимость пути | 10 Гбит/с | 2 | 1-5 | 1-65535 |

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 пакетов, называемых блоками данных протокола моста - *BPDU (Bridge Protocol Data Unit)*.

Пакеты BPDU содержат основную информацию, необходимую для построения топологии сети без петель:

- Идентификатор коммутатора, на основании которого выбирается корневой коммутатор
- Расстояние от коммутатора-источника до корневого коммутатора (стоимость корневого маршрута)
- Идентификатор порта

Пакеты BPDU помещаются в поле данных кадров канального уровня, например, кадров Ethernet.

Коммутаторы обмениваются BPDU через равные интервалы времени (обычно 1-4с). В случае отказа моста (что приводит к изменению топологии) соседние коммутаторы, не получив пакет BPDU в течении заданного времени (Max Age), начинают пересчет связующего дерева.

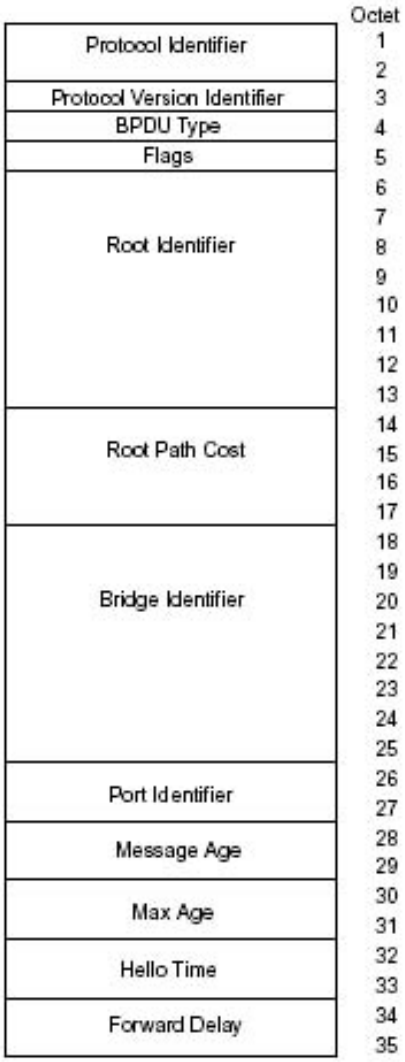



Рисунок 26. Формат BPDU.

Пакет BPDU имеет следующие поля:

- Идентификатор версии протокола STA - 2 байта. Коммутаторы должны поддерживать одну и ту же версию протокола STA, иначе может установиться активная конфигурация с петлями.
- Версия протокола STP – 1 байт.
- Тип BPDU - 1 байт. Существует два типа BPDU - конфигурационный BPDU, то есть заявка на возможность стать корневым коммутатором, на основании которой происходит определение активной конфигурации, и BPDU уведомления о реконфигурации, которое посылается коммутатором, обнаружившим событие, требующее проведения реконфигурации - отказ линии связи, отказ порта, изменение приоритетов коммутатора или портов.
- Флаги - 1 байт. Один бит содержит флаг изменения конфигурации, второй бит - флаг подтверждения изменения конфигурации.
- Идентификатор корневого коммутатора - 8 байтов.
- Расстояние до корня - 2 байта.
- Идентификатор коммутатора - 8 байтов.
- Идентификатор порта - 2 байта.
- Время жизни сообщения - 2 байта. Измеряется в единицах по 0.5 с, служит для выявления устаревших сообщений. Когда пакет BPDU проходит через коммутатор, тот добавляет ко времени жизни пакета время его задержки данным коммутатором.
- Максимальное время жизни сообщения - 2 байта. Если пакет BPDU имеет время жизни, превышающее максимальное, то он игнорируется коммутаторами.
- Интервал hello (время приветствия), через который посылаются пакеты BPDU.

- Задержка смены состояний - 2 байта. Минимальное время перехода портов коммутатора в активное состояние. Такая задержка необходима, чтобы исключить возможность временного возникновения альтернативных маршрутов при неодновременной смене состояний портов во время реконфигурации.

Пакет BPDU уведомления о реконфигурации имеет следующие поля:

- Идентификатор версии протокола STA - 2 байта.
- Версия протокола STP – 1 байт.
- Тип BPDU - 1 байт с установленным флагом реконфигурации топологии.

*Пример работы STP*

Для примера рассмотрены 3 коммутатора, подключенные с образованием петли (рисунок 27). Т.о., в сети могут возникнуть проблемы с зацикливанием пакетов. Например, пусть какой-либо компьютер в сети LAN1 посылает широковещательный пакет. В соответствии с логикой работы коммутаторов, коммутатор А передаст этот пакет во все подключенные к нему сегменты, за исключением того, из которого он пришел. Коммутатор В получит этот пакет и передаст его коммутатору С. Коммутатор С, также получит широковещательный пакет от коммутатора А и передаст его коммутатору В. Тот в свою очередь, вернет его коммутатору А и так далее. Т.е., пакеты могут ходить по сети бесконечно долго, что может привести к нарушению работоспособности сети. В этом примере с помощью STP блокируется соединение между коммутаторами С и В.

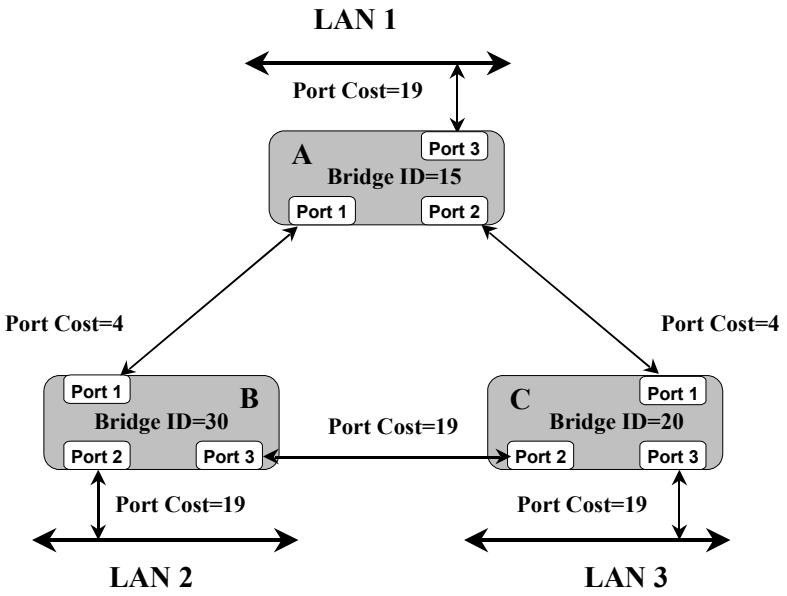



Рисунок 27. Перед применением Spanning Tree.

Итак, после включения питания и загрузки каждый коммутатор начинает считать себя корневым. Когда он генерирует BPDU (через интервал *hello*), он помещает свой идентификатор в поле «идентификатор корневого коммутатора», расстояние до корня устанавливается в 0, а в качестве идентификатора порта указывается идентификатор того порта, через который будет передаваться BPDU.

Как только коммутатор получает BPDU, в котором имеется идентификатор корневого коммутатора, меньше его собственного, он перестает генерировать свои собственные кадры BPDU, и начинает ретранслировать только кадры нового претендента на звание корневого коммутатора. При ретрансляции кадров он наращивает расстояние до корня, указанное в пришедшем BPDU, на условное время сегмента, через который принят данный кадр.

При ретрансляции кадров каждый коммутатор для каждого своего порта запоминает минимальное расстояние до корня. При завершении процедуры установления конфигурации

покрывающего дерева, каждый коммутатор находит свой корневой порт - это порт, который ближе других портов находится по отношению к корню дерева.

Рассмотрим выборы корневых портов коммутаторов на примере рисунка 27.

Когда коммутатор A (корневой мост) посылает BPDU's, они содержат стоимость пути к корневому мосту равную 0. Когда коммутатор B получает эти BPDU, он добавляет стоимость пути Port 1 (4) к стоимости, указанной в полученном BPDU (0). Коммутатор B затем использует значение 4 и посылает BPDU's со стоимостью пути к корню равной 4 через Port 3 и Port 2.

Когда коммутатор C получает BPDU от коммутатора B, он увеличивает стоимость пути к корню до 23 (4 + 19). Однако коммутатор C также получает BPDU от корневого коммутатора A через Port 1. Стоимость пути к корню в этом BPDU равна 0 и коммутатор C увеличивает ее стоимость до 4 (стоимость его Port 1 равна 4). Теперь коммутатор C должен выбрать единственный корневой порт. Коммутатор C выбирает Port 1 в качестве корневого, поскольку его стоимость пути к корню меньше. После этого коммутатор C начинает объявлять стоимость пути до корня равную 4 нижележащим коммутаторам.

Выборы корневого порта коммутатора B происходят аналогично и корневым портом для него становится Port 1 со стоимостью 4.

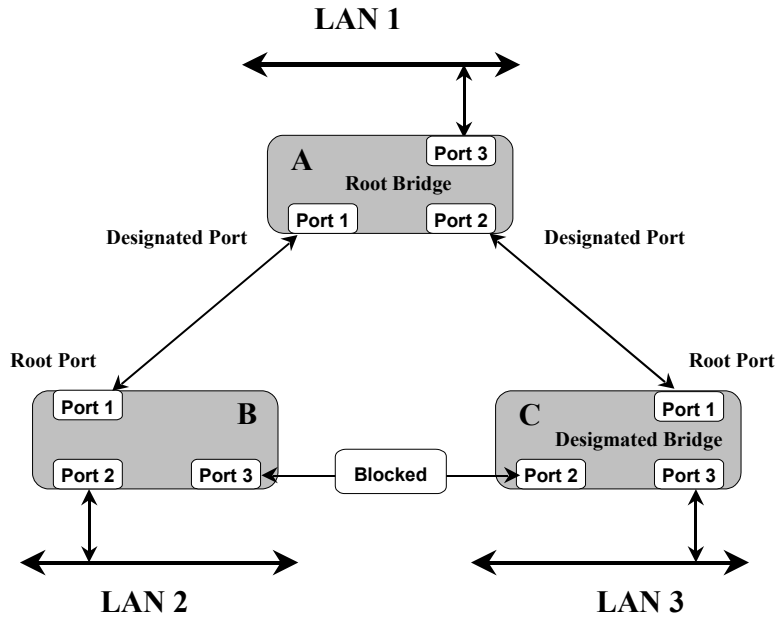



Рисунок 28. После применения Spanning Tree.

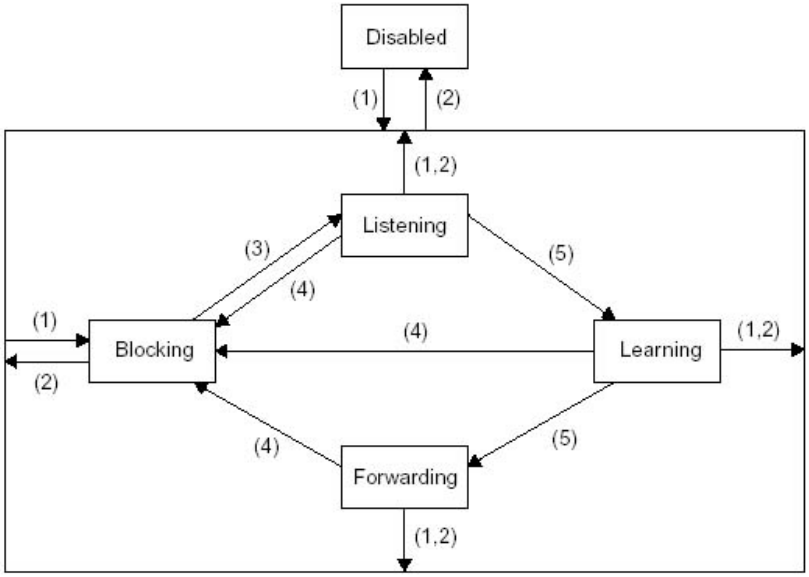
Кроме этого, коммутаторы выбирают для каждого сегмента сети назначенный порт. Для этого они исключают из рассмотрения свой корневой порт, а для всех своих оставшихся портов сравнивают принятые по ним минимальные расстояния до корня с расстоянием до корня своего корневого порта. Если у своего порта это расстояние меньше принятых, то это значит, что он является назначенным портом. Когда имеется несколько портов с одинаковым кратчайшим расстоянием до корневого коммутатора, то для выбора назначенного порта сегмента STP принимает решение на основе последовательного сравнения идентификаторов мостов и идентификаторов портов.

Все порты, кроме назначенных переводятся в заблокированное состояние и на этом построение покрывающего дерева заканчивается.

На коммутаторе B корневым портом является Port 1 (стоимость 4). Поэтому для сегмента коммутатор A – коммутатор B, назначенным портом будет Port 1 коммутатора A. На коммутаторе C корневым портом является Port 1 (стоимость 4). Поэтому для сегмента коммутатор A – коммутатор C, назначенным портом будет Port 2 коммутатора A. В сегменте

коммутатор B – коммутатор C оба порта Port 3 и Port 2 имеют одинаковую стоимость пути, равную 23. В этом случае STP выберет назначенный порт сегмента на основе сравнения идентификаторов мостов. Поскольку идентификатор коммутатора C (20) меньше идентификатора коммутатора B (30), то назначенным портом для этого сегмента станет Port 2 коммутатора C. Port 3 на коммутаторе B заблокируется.

Таким образом, в процессе построения топологии сети каждый порт коммутатора проходит несколько стадий:




1) Порт активен или инициализация порта.
2) Порт отключен администратором или сбой порта.
3) Порт выбран в качестве корневого или назначенного порта.
4) Порт заблокирован.
5) Истек таймер смены состояний.

Рисунок 29. Состояния портов.

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

В процессе нормальной работы корневой коммутатор продолжает генерировать служебные пакеты BPDU, а остальные коммутаторы продолжают их принимать своими

корневыми портами и ретранслировать назначенными. Если по истечении максимального времени жизни сообщения (по умолчанию — 20 с) корневой порт любого коммутатора сети не получит служебный пакет BPDU, то он инициализирует новую процедуру построения покрывающего дерева.

Коммутаторы D-Link также поддерживают протокол Rapid STP (IEEE 802.1w), который обладает лучшим временем сходимости по сравнению с STP (меньше 1 секунды). 802.1w обратно совместим с 802.1d.

### 8.1 Rapid Spanning Tree Protocol (IEEE 802.1w)

Программное обеспечение управляемых коммутаторов D-Link поддерживает две версии протокола Spanning Tree Protocol, Rapid Spanning Tree Protocol (RSTP), как определено в спецификации IEEE 802.1w и версию, совместимую с IEEE 802.1d STP. RSTP может работать с оборудованием, поддерживающим STP, однако все преимущества от его использования будут потеряны.

Протокол IEEE 802.1w Rapid Spanning Tree Protocol (RSTP) является развитием стандарта 802.1d STP. Он был разработан для преодоления отдельных ограничений STP, которые мешали внедрению некоторых новых функций коммутаторов, например, функций 3-его уровня, всё больше и больше применяемых в коммутаторах Ethernet.

Существенным отличием протоколов STP 802.1d и RSTP 802.1w 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.

Таблица. 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) |

Выбор активной топологии завершается присвоением протоколом RSTP определенной роли каждому порту. Роли корневой порт и назначенный порт включают порт в активную топологию. В RSTP существуют 2 роли – альтернативный порт (*Alternate*) и резервный порт (*Backup*), соответствующие состоянию «Заблокирован» в STP и исключающие порт из активной топологии. *Альтернативный порт* предлагает альтернативный основному пути путь в направлении корневого моста.

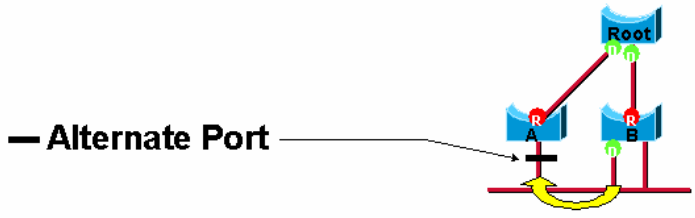



Рисунок 30. Альтернативный порт.

*Резервный порт* предназначен для резервирования пути, предоставляемого выделенным портом в направлении сегментов сети. Резервные порты существуют только в конфигурациях, где есть два или более соединения данного моста с данной сетью (сегментом сети).

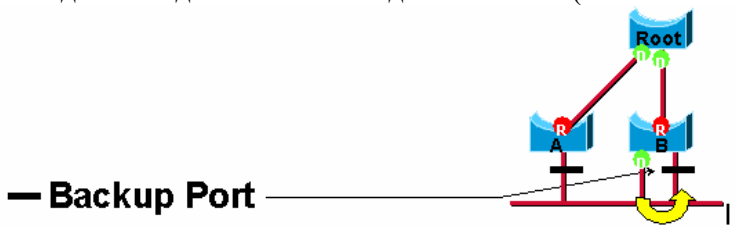



Рисунок 31. Резервный порт.

Процесс вычисления связующего дерева у обоих протоколов одинаков. Однако, при работе RSTP, порт может перейти в состояние продвижения значительно быстрее – он больше не зависит от конфигурации таймеров. Порты больше не должны ждать стабилизации топологии, чтобы перейти в режим продвижения. Для того, чтобы обеспечить быстрый переход в это состояние, протокол RSTP вводит две новые переменные: пограничный порт (edge port) и порт типа «точка-точка» (point-to-point , P2P).

*Пограничным (Edge) портом* объявляется порт, непосредственно подключенный к сегменту, в котором не могут быть созданы петли. Например, порт непосредственно подключен к рабочей станции. Порт, который определен как пограничный, мгновенно переходит в состояние продвижения, минуя состояния прослушивания и обучения. Пограничный порт теряет свой статус и становится обычным портом связующего дерева в том случае, если получит пакет BPDU.

*P2P порт*, обычно используемый для подключения к другим мостам, также способен быстро перейти в состояние продвижения. При работе RSTP все порты, функционирующие в полнодуплексном режиме, рассматриваются как порты P2P, до тех пор, пока не будут переконфигурированы вручную.

### Совместимость 802.1d/802.1w

Протокол RSTP способен взаимодействовать с оборудованием, поддерживающим STP и, если необходимо, может автоматически преобразовывать пакеты BPDU в формат 802.1d. Однако, преимущество быстрой сходимости этого протокола (когда все коммутаторы переходят в состояние пересылки или блокировки и обладают тождественной информацией) теряется. Протокол также предоставляет возможность использования переменной для миграции, в случае обновления программного обеспечения оборудования в сегменте сети для использования RSTP.

## 8.2 Настройка STP с помощью команд CLI

В таблице приведены команды для настройки STP и Rapid STP с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| config stp | maxage <value><br>hellotime <value><br>forwarddelay <value><br>priority <value><br>fdpdu [enable\|disable]<br>txholdcount <1-10><br>version [rstp\|stp] | Настройка временных параметров STP и приоритета моста |
| config stp ports | <portlist><br>cost <value><br>priority <value><br>migrate [yes\|no]<br>edge [true\|false]<br>p2p [true\|false\|auto]<br>state [enable\|disable] | Настройка параметров портов |
| enable stp | | Активизация STP на коммутаторе |
| disable stp | | Деактивизация STP на коммутаторе |

| | | |
|---|---|---|
| show stp | | Просмотр конфигурации STP |
| show stp ports | <portlist> | Просмотр конфигурации портов STP |

*Пример 1*. Активизировать STP (глобально) на коммутаторе.

DES-3226S#enable stp

Command: enable stp

Success.

*Пример 2*. Сконфигурировать STP со следующими значениями для портов:
стоимость пути (path cost) 19, приоритет (priority) 16, состояние (state) enabled для портов 1-5 коммутатора.

DES-3226S#config stp ports 1-5 cost 19 priority 16 state enabled

Command: config stp ports 1-5 cost 19 priority 15 state enabled

Success.

*Пример 3*. Настроить таймеры STP на коммутаторе.
Сконфигурировать STP со следующими значениями таймеров: maxage 18 и hellotime 4.

DES-3226S#config stp maxage 18 hellotime 4

Command: config stp maxage 18 hellotime 4

Success.

*Пример 4*. Запретить STP глобально на коммутаторе.

DES-3226S#disable stp

Command: disable stp

Success.

*Пример 5*. Проверка текущей конфигурации STP на коммутаторе.

DES-3226S#show stp

Command: show stp

Bridge Parameters Settings

STP Status : Enabled

Max Age : 20

Hello Time : 2

Forward Delay : 15

Priority : 32768

STP Version : STP compatible

TX Hold Count : 3

Forwarding BPDU : Enabled

Bridge Current Status

Designated Root Bridge : 00-00-51-43-70-00

Root Priority: 32768

Cost to Root : 200000

Root Port : 10

Last Topology Change : 53sec

Topology Changes Count : 1

Protocol Specification : 3

Max Age : 20

Hello Time : 2

Forward Delay : 15

Hold Time : 3

*Пример 6*. Проверка текущего состояния портов STP .

DES-3226S#show stp ports

Command: show ports

| Port | Designated Bridge | State | Cost | Pri | Edge | P2P | Status | Role |
|---|---|---|---|---|---|---|---|---|
| 1 | N/A | Yes | *200000 | 128 | No | Yes | Disabled | Disabled |
| 2 | N/A | Yes | *200000 | 128 | No | Yes | Disabled | Disabled |
| 3 | N/A | Yes | *200000 | 128 | No | Yes | Disabled | Disabled |
| …… | | | | | | | | |
| 9 | N/A | Yes | *200000 | 128 | No | Yes | Disabled | Disabled |
| 10 | N/A | Yes | *200000 | 128 | No | Yes | Forwarding | NonStp |
| 11 | N/A | Yes | *200000 | 128 | No | Yes | Disabled | Disabled |
| …….. | | | | | | | | |

# 9. Качество сервиса (QoS)

## 9.1 Приоритетная обработка кадров (802.1p)

Построение сетей на основе коммутаторов позволяет использовать приоритезацию трафика, причем делать это независимо от технологии сети. Эта возможность является следствием того, что коммутаторы буферизуют кадры перед их отправкой на другой порт. Коммутатор обычно ведет для каждого входного и выходного порта не одну, а несколько очередей, причем каждая очередь имеет свой приоритет обработки. При этом коммутатор может быть сконфигурирован, например, так, чтобы передавать один низкоприоритетный пакет на каждые 10 высокоприоритетных пакетов.

Поддержка приоритетной обработки может особенно пригодиться для приложений, предъявляющих различные требования к допустимым задержкам кадров и к пропускной способности сети для потока кадров.

Способность сети обеспечивать различные уровни обслуживания, запрашиваемые теми или иными сетевыми приложениями, наряду с проведением контроля за характеристиками производительности – полосой пропускания, задержкой/дрожанием и потерей пакетов – может быть классифицирована по трем различным категориям:

- **Негарантированная доставка данных (best effort service)**. Обеспечение связности узлов сети без гарантии времени и самого факта доставки пакетов в точку

назначения. На самом деле негарантированная доставка не является частью Qos поскольку отсутствует гарантия качества обслуживания и гарантия доставки пакетов.

- **Дифференцированное обслуживание (differentiated service)**. Дифференцированное обслуживание предполагает разделение трафика на классы на основе требований к качеству обслуживания. Каждый класс трафика дифференцируется и обрабатывается сетью в соответствии с заданными для этого класса механизмами QoS (быстрее обрабатывается, выше средняя полоса пропускания, ниже средний уровень потерь). Подобная схема обеспечения качества обслуживания часто называется схемой CoS (Class of Service). Дифференцированное обслуживание само по себе не предполагает обеспечение гарантий предоставляемых услуг. В соответствии с этой схемой трафик распределяется по классам, каждый из которых имеет собственный приоритет. Этот тип обслуживания удобно применять в сетях с интенсивным трафиком. В этом случае важно обеспечить отделение административного трафика сети от всего остального и назначить ему приоритет, позволяющий в любой момент времени быть уверенным в связности узлов сети.
- **Гарантированное обслуживание (guaranteed service)**. Гарантированное обслуживание предполагает резервирование сетевых ресурсов с целью удовлетворения специфических требований к обслуживанию со стороны потоков трафика. В соответствии с гарантированным обслуживанием выполняется предварительное резервирование сетевых ресурсов по всей траектории движения трафика. Например, такие схемы используются в технологиях глобальных сетей Frame Relay и ATM или в протоколе RSVP для сетей TCP/IP. Однако для коммутаторов такого рода протоколов нет, так что гарантий качества обслуживания они пока дать не могут.

Основным вопросом при приоритетной обработке кадров коммутаторами является вопрос назначения кадру приоритета. Так как не все протоколы канального уровня поддерживают поле приоритета кадра, например, у кадров Ethernet оно отсутствует, то коммутатор должен использовать какой-либо дополнительный механизм для связывания кадра с его приоритетом. Наиболее распространенный способ - приписывание приоритета портам коммутатора. При этом способе коммутатор помещает кадр в очередь кадров соответствующего приоритета в зависимости от того, через какой порт поступил кадр в коммутатор. Способ несложный, но недостаточно гибкий - если к порту коммутатора подключен не отдельный узел, а сегмент, то все узлы сегмента получают одинаковый приоритет.

Более гибким является назначение приоритетов кадрам в соответствии со стандартом IEEE 802.1p. Этот стандарт разрабатывался совместно со стандартом 802.1q. В обоих стандартах предусмотрен общий дополнительный заголовок для кадров Ethernet, состоящий из двух байт. В этом дополнительном заголовке, который вставляется перед полем данных кадра, 3 бита используются для указания приоритета кадра. Существует протокол, по которому конечный узел может запросить у коммутатора один из восьми уровней приоритета кадра. Если сетевой адаптер не поддерживает стандарт 802.1p, то коммутатор может назначать приоритеты кадрам на основе порта поступления кадра. Такие помеченные кадры будут обслуживаться в соответствии с их приоритетом всеми коммутаторами сети, а не только тем коммутатором, который непосредственно принял кадр от конечного узла. При передаче кадра сетевому адаптеру, не поддерживающему стандарт 802.1p, дополнительный заголовок должен быть удален.

Коммутаторы обеспечивают дифференцированное обслуживание, поэтому необходима идентификация пакетов, которая позволит отнести их к соответствующему классу трафика CoS, включающему, как правило, пакеты из разных потоков. Указанная задача выполняется путем классификации.

*Классификация пакетов (packet classification)* представляет собой средство, позволяющее отнести пакет к тому или иному классу трафика в зависимости от значений одного или нескольких полей пакета.

В управляемых коммутаторах D-Link используются различные способы классификации пакетов. Ниже перечислены параметры, на основании которых пакет идентифицируется:

- Биты класса приоритета 802.1p
- Поля байта TOS, расположенного в заголовке IP-пакета и поле кода дифференцированной услуги (DSCP)
- Адрес назначения и источника IP-пакета
- Номера портов TCP/UDP.

Поскольку высокоприоритетные пакеты должны обрабатываться раньше низкоприоритетных, в коммутаторах поддерживается несколько очередей приоритетов CoS (например, DES-3226S имеет 4 очереди CoS с разным приоритетом на каждый выходной порт, DGS-3212SR – 8 очередей). Кадры, в соответствии со своим приоритетом, могут быть помещены в разные очереди, и обслуживаться, например, по взвешенному циклическому алгоритму (Weighted Round Robin, WRR).

На рисунке 32 показана схема распределения пакетов с разными приоритетами между очередями CoS. Пакеты с приоритетами P1 и P2 помещаются в очередь Q0, пакеты с приоритетами P0 и P3 помещаются в очередь коммутатора Q1, пакеты с приоритетами P4 и P5 помещаются в очередь коммутатора Q2, пакеты с приоритетами P6 и P7 помещаются в очередь коммутатора Q3.

Для обработки очередей приоритетов могут использоваться различные механизмы обслуживания. В коммутаторах D-Link используются 2 схемы обслуживания очередей: строгая очередь приоритетов (Strict Priority Queuing) и взвешенный циклический алгоритм (Weighted Round Robin).

В первом случае, пакеты, находящиеся в самой приоритетной очереди начинают передаваться первыми. При этом пока более приоритетная очередь не опустеет, пакеты из менее приоритетных очередей передаваться не будут. Второй алгоритм WRR устраняет это ограничение, а также исключает нехватку полосы пропускания для очередей с низким приоритетом. В этом случае для каждой очереди приоритетов задается максимальное количество пакетов, которое может быть передано за один раз и максимальное время ожидания, через которое очередь снова сможет передавать пакеты. Диапазон передаваемых пакетов: от 0 до 255. Диапазон времени оживания: от 0 до 255 (увеличивается на 16мс).

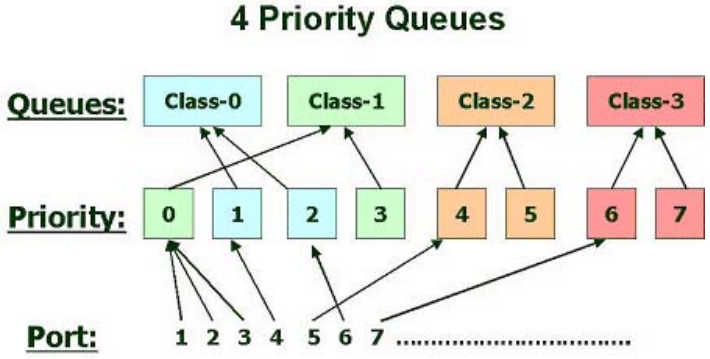



Рисунок 31. Распределение пакетов по очередям приоритетов.

### 9.1.2 Настройка приоритетной обработки кадров с помощью CLI

В таблице приведены команды для настройки приоритетной обработки кадров с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| config scheduling | <class_id 0-3><br>max_packet <value 0-255><br>max_latency <value 0-255> | Настройка расписания обработки пакетов для каждой очереди приоритетов |
| show scheduling | | Просмотр текущего расписания |
| config 802.1p | <priority 0-7> | Используется для привязки |

| | | |
|---|---|---|
| user_priority | <class_id 0-3> | входящего пакета с заданным пользователем приоритетом 802.1p к одной из аппаратных очередей, доступных на коммутаторе. |
| show 802.1p user_priority | | Просмотр текущей схемы привязки |
| config 802.1p default_priority | <portlist><br>all<br><priority 0-7> | Используется для указания того, как привязать входящие пакеты без тега приоритета к одной из аппаратных очередей коммутатора. |
| show 802.1p default_priority | <portlist> | Просмотр приоритетов 802.1p, присвоенных немаркированным входящим пакетам. |

*Пример 1*. Создать расписание обработки очередей приоритетов коммутатора. Для очереди Q0 задать максимальное количество передаваемых пакетов равным 100 и время ожидания – 150

DES-3226S# config scheduling 0 max_packet 100 max_latency 150

Command: config scheduling 0 max_packet 100 max_latency 150

Success.

*Пример 2*. Проверить созданное расписание обработки очередей.

DES-3226S# show scheduling

Command: show scheduling

QOS Output Scheduling

| | MAX. Packets | MAX. Latency |
|---|---|---|
| Class-0 | 50 | 1 |
| Class-1 | 100 | 1 |
| Class-2 | 150 | 1 |
| Class-3 | 200 | 1 |

*Пример 3*. Поместить маркированные пакеты с приоритетом 1 в очередь приоритетов коммутатора Q3.

DES-3226S# config 802.1p user_priority 1 3

Command: config 802.1p user_priority 1 3

Success.

*Пример 4*. Проверить текущую схему привязки маркированных пакетов к очередям коммутатора.

DES-3226S# show 802.1p user_priority

Command: show 802.1p user_priority

COS Class of Traffic

Priority-0 -> <Class-1>

Priority-1 -> <Class-0>

Priority-2 -> <Class-0>

Priority-3 -> <Class-1>

Priority-4 -> <Class-2>

Priority-5 -> <Class-2>

Priority-6 -> <Class-3>

Priority-7 -> <Class-3>

*Пример 5*. Назначить приоритет немаркированному пакету.
Присвоить всем немаркированным пакетам, приходящим на любой из портов коммутатора приоритет 5.

DES-3226S#config 802.1p default_priority all 5

Command: config 802.1p default_priority all 5

Success.

*Пример 6*. Проверить приоритеты, назначаемые немаркированным пакетам.

DES-3226S# show 802.1p default_priority

Command: show 802.1p default_priority

| Port | Priority |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| …….. | |

## 9.2 Контроль полосы пропускания

Контроль полосы пропускания обычно используется для ограничения скорости передачи и приема битов данных для любого порта.

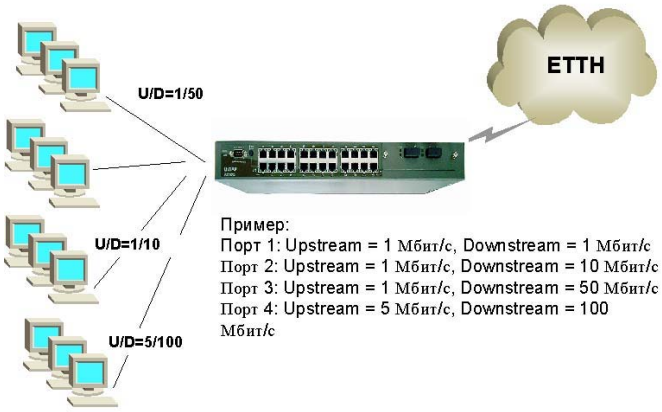



Рисунок 32.

### 9.2.1 Настройка полосы пропускания с помощью команд CLI

В таблице приведены команды для настройки полосы пропускания портов с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| config bandwidth_control | <portlist><br>rx_rate<br>no_limit<br><value 1-1000><br>tx_rate<br>no_limit<br><value 1-1000> | Настройка полосы пропускания на указанном порте |
| show bandwidth_control | <portlist> | Просмотр настроек полосы пропускания для каждого порта |

*Пример 1*. Настроить на портах с 1 по 10 скорость передачи пакетов, равную 10 Мбит/с.

DES-3226S#config bandwidth_control 1-10 tx_rate 10

Command: config bandwidth_control 1-10 tx_rate 10

Success.

*Пример 2*. Проверить настройки полосы пропускания на каждом порте.

DES-3226S:4#show bandwidth_control 1-10

Command: show bandwidth_control 1-10

Bandwidth Control Table

Port RX RATE (Mbit|sec) TX_RATE (Mbit|sec)

---- ---------- ----------

1 no_limit 10

2 no_limit 10

3 no_limit 10

| | | |
|---|---|---|
| 4 | no_limit | 10 |
| 5 | no_limit | 10 |
| 6 | no_limit | 10 |
| 7 | no_limit | 10 |
| 8 | no_limit | 10 |
| 9 | no_limit | 10 |
| 10 | no_limit | 10 |

# 10. Ограничение доступа к сети

## 10.1 Port Security и таблица фильтрации коммутатора

В коммутаторах, помимо стандартной функции динамического построения таблицы MAC-адресов на основе адресов входящих пакетов, реализована функция настройки статической таблицы MAC-адресов. Это позволяет в полной мере контролировать прохождение пакетов через коммутатор, блокируя доступ к сети компьютерам с неизвестными коммутатору MAC-адресами.

Во-первых, можно заблокировать дальнейшее обновление таблицы коммутатора – если конфигурация вашей сети более не изменяется, вы блокируете таблицу MAC-адресов, и т.о. коммутатор будет просто отбрасывать все пакеты, которые поступают с неизвестного адреса.

Во-вторых, можно вручную привязать определенный MAC-адрес к порту коммутатора, и тем самым коммутатор будет постоянно хранить соответствие MAC-адрес-порт, даже при длительной неактивности устройства или при перегрузках сети. Коммутаторы D-Link позволяют создавать статические таблицы MAC-адресов, хранящие до 256 записей.

Коммутаторы имеют возможность настроить таблицу фильтрации MAC-адресов, указав MAC-адрес устройства, и входящие и исходящие пакеты с указанным адресом будут ими отбрасываться.

Таким образом, используя вышеперечисленные функции или их комбинации, можно обеспечить защиту сети от несанкционированного доступа. Например, если привязать MAC-адреса рабочих станций сети (при условии, что структура сети не изменится в течении определенного времени) к портам коммутатора, а затем заблокировать таблицу коммутатора, можно запретить таким образом прохождение пакетов от неизвестных адресов. Данная функция оказывается весьма полезной при построении домовых сетей, сетей провайдера Интернет и локальных сетей с повышенным требованием по безопасности, т.к. исключает доступ незарегистрированных рабочих станций. Подделка MAC-адреса, хотя и возможна, но остается более трудной задачей, чем подделка IP-адреса, следовательно, вы можете обеспечить весьма приемлемый уровень безопасности.

### 10.1.2 Настройка Port Security с помощью CLI

В таблице приведены команды для настройки Port Security с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| config port_security ports | [<portlist>\| all ] {admin_state [enable \| disable] \|max_learning_addr <max_lock_no 0-10> \| lock_address_mode[Permanent\| DeleteOnTimeout\|DeleteOnReset]} | Настройка для портов, перечисленных в списке portlist параметров безопасности |
| clear port_security_entry | vlan_name <vlan_name 32> mac_address <macaddr> port <port> | Удаление настроек безопасности порта, соответствующих указанной VLAN, MAC адресу и порту. |
| Show port_security | {ports <portlist>} | Просмотр текущих настроек безопасности портов |

*Пример 1*. Настроить параметры Port Security для портов 1-5, ограничивающие максимальное количество изучаемых портами MAC-адресов до 5 и позволяющие автоматическое удаление записей из таблицы MAC-адресов с истекшим временем жизни.

```
DES-3226S#config port_security ports 1-5 admin_state enable  max_learning_addr 5 lock_address_mode
DeleteOnTimeout

Command: config port_security ports 1-5 admin_state enable max_learning_addr 5 lock_address_mode DeleteOnTimeout

Stacking port 5 can not be a port-security port
```

*Пример 2*. Проверить конфигурацию Port Security.

```
DES-3226S#show port_security ports 1-24
Command: show port_security ports 1-24

Port#    Admin   State Max.  Learning Addr.  Lock Address Mode
----     ----------- ------------------- -----------------
1        Disabled       1                 DeleteOnReset
2        Disabled       1                 DeleteOnReset
3        Disabled       1                 DeleteOnReset
4        Disabled       1                 DeleteOnReset
5        Disabled       1                 DeleteOnReset
6        Disabled       1                 DeleteOnReset
7        Enabled      10                DeleteOnReset
```

*Пример 3*. Удалить настройки  Port Security для заданных портов.

```
DES-3226S#clear port_security_entry port 15:1
Command: clear port_security_entry port 15:1

Success.
```

### 10.2 Сегментация трафика

Сегментация трафика служит для разграничения доменов на уровне 2.Данная функция позволяет настраивать порты таким образом, чтобы они были изолированы друг от друга, но в то же время имели доступ к разделяемым портам, используемым для подключения серверов и магистрали сети провайдера. Данная функция может быть использована при построении сетей провайдеров.

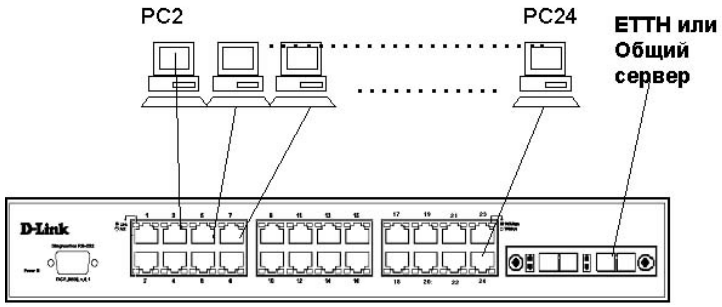



Рисунок 33. Пример использования функции Traffic Segmentation.

Все компьютеры (PC2 - PC24) имеют доступ к порту uplink, но не имеют доступа друг к друг на канальном уровне .

Это решение можно использовать:

1.В проектах ETTH для изоляции портов

2.Для предоставления доступа к общему серверу

### 10.2.1 Настройка Traffic Segmentation с помощью CLI

В таблице приведены команды для настройки Traffic Segmentation с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| config traffic_segmentation | <portlist><br>forward_list<br>null<br><portlist> | Настройка сегментации трафика на коммутаторе |
| show traffic_segmentation | <portlist> | Просмотр настроек сегментации трафика |

*Пример 1*. Настроить порты с 1 по 10 так, чтобы они могли передавать кадры через порты с 11 по 15.

DES-3226S# config traffic_segmentation 1-10 forward_list 11-15

Command: config traffic_segmentation 1-10 forward_list 11-15

Success.

*Пример 2*. Просмотреть текущую таблицу сегментации трафика, настроенную на коммутаторе.

DES-3226S#show traffic_segmentation

Command: show traffic_segmentation

Traffic Segmentation Table

Port Forward Portlist

---- ----------

1 9-15

2 9-15

……..

11 1-26

12 1-26

13 1-26

……

## 10.3 Протокол IEEE 802.1x

Протокол IEEE 802.1x определяет доступ на основе модели Клиент/Сервер и протокол аутентификации, который не позволяет неавторизованным устройствам подключаться к локальной сети через порты коммутатора. Сервер аутентификации (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, управление доступом протокола 802.1x позволит только трафику протокола Extensible Authentication Protocol over LAN (EAPOL) проходить через порт, к которому подключен клиент. После успешной аутентификации, обычный трафик может передаваться через порт.

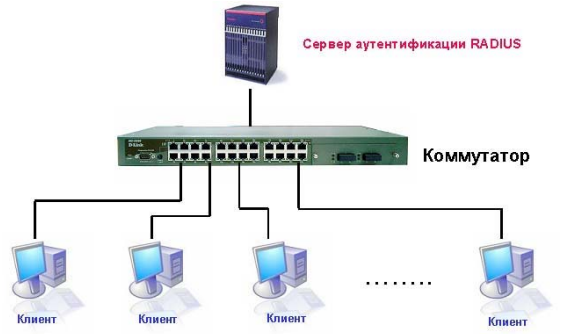



Рисунок 34.

**Роли устройств**

При 802.1x аутентификации на основе портов, устройства в сети выполняют определенные роли.

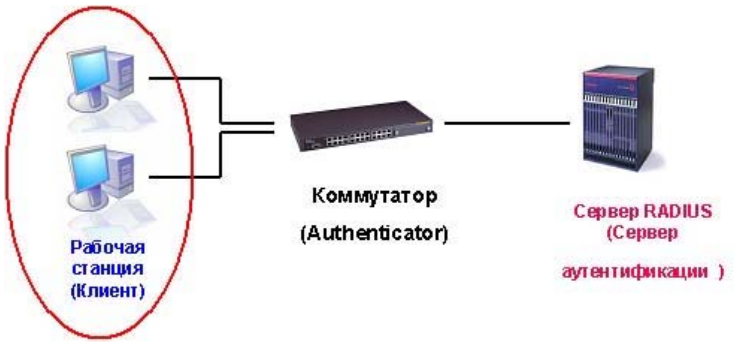



Рисунок 35.

*Клиент* – 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.

Рисунок 35.

*Сервер аутентификации* – выполняет фактическую аутентификацию клиента. Сервер аутентификации проверяет подлинность клиента и информирует коммутатор предоставлять или нет клиенту доступ к локальной сети. *RADIUS* работает в модели клиент/сервер, в которой информация об аутентификации передается между сервером и клиентами *RADIUS*. Т.к. коммутатор работает как прокси, сервис аутентификации прозрачен для клиента.


Рисунок 36.

*Коммутатор* (также называется *аутентификатор (authenticator)*) – управляет физическим доступом к сети, основываясь на статусе аутентификации клиента. Коммутатор работает как посредник между клиентом и сервером аутентификации, получая запрос на проверку подлинности от клиента, проверяя данную информацию при помощи сервера аутентификации, и пересылая ответ клиенту. Коммутатор включает клиент RADIUS, который отвечает за инкапсуляцию и деинкапсуляцию кадров EAP и взаимодействие с сервером аутентификации.

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 рисунке показана схема обмена, инициируемая клиентом, использующая метод аутентификации One-Time-Password (OTP) сервером RADIUS.

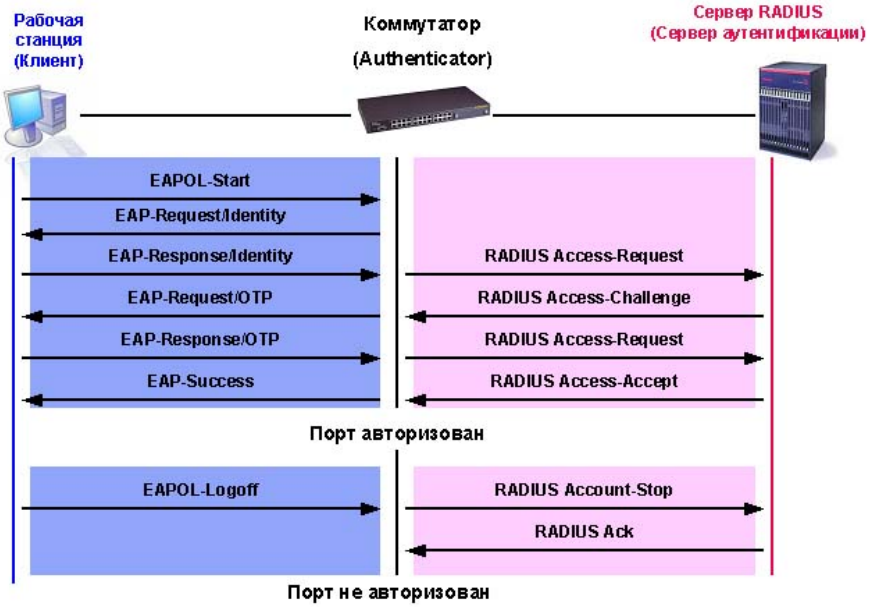



Рисунок 37.

Аутентификация 802.1x может быть выполнена как на основе MAC-адресов, так и на основе портов:

- **При аутентификации 802.1x на основе MAC-адресов** сервер проверяет не только имя пользователя/пароль, но и максимальное количество MAC-адресов, доступных для работы. Если предел достигнут, то он блокирует новый MAC-адрес.
- **При аутентификации 802.1x на основе портов,** после того, как порт был авторизован, любой пользователь, подключенный к порту, может получить доступ к локальной сети.

### 10.3.1 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 протокола 802.1x. Когда клиент аутентифицирован, порт переходит в *авторизованное* состояние, позволяя передачу любого трафика от него.

Возможны 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.

В другом случае, когда клиент с поддержкой 802.1x подключается к порту, на котором не запущен протокол 802.1x, клиент 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 посылать кадры.

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, в доступе к сети будет отказано из-за ошибок аутентификации.

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.

### 10.3.2 Ограничения аутентификации 802.1x на основе портов

Протокол 802.1x не поддерживает следующие типы портов:

- Агрегированные каналы (Trunk port)
- Порты EtherChannel — перед тем, как настраивать 802.1x на порте, необходимо удалить его из интерфейса EtherChannel.

### 10.3.3 Настройка 802.1x с помощью CLI

**На рабочей станции**: необходимо установить клиентское ПО для 802.1x, если оно отсутствует (клиент 802.1x встроен в ОС Window XP).

**Сервер Radius**: Windows NT или Windows 2000 Server Radius Server Service.

**На коммутаторе**:

- активировать 802.1x на устройстве
- настроить 802.1x на портах
- настроить параметры для сервера Radius

В таблице приведены команды для настройки 802.1x с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| enable 802.1x | | Активизировать сервер 802.1x на коммутаторе |
| disable 802.1x | | Запретить сервер 802.1x на коммутаторе |
| show 802.1x | [auth_state \| auth_configuration] {ports <portlist>} | Просмотр текущей конфигурации сервера 802.1x на коммутаторе |
| config 802.1x capability | ports <portlist>\|all<br>authenticator<br>none | Настройка 802.1x на портах и определение их роли: аутентификатор или не аутентификатор. |
| config 802.1x auth_parameter | ports <portlist>\|all<br>default<br>direction [both\|in]<br>port_control<br>[force_unauth\|auto\|force_auth]<br>quiet_period <sec 0-65535><br>tx_period <sec 1-65535><br>supp_timeout <sec 1-65535><br>server_timeout <sec 1-65535><br>max_req <value 1-10><br>reauth_period <sec 1-65535><br>enable_reauth [enable\|disable] | Настройка параметров аутентификации 802.1x для указанного списка портов. Default - возврат всех портов из указанного списка к настройкам 802.1x по умолчанию. |
| config 802.1x auth_mode | [port_based \|mac_based] | Настройка режима аутентификации 802.1x: на основе портов или на основе MAC-адресов. |
| config 802.1x init | config 802.1x init [port_based ports [<portlist\|all>] \|mac_based ports [<portlist>\|all]<br>{mac_address <macaddr>}] | Мгновенная инициализация функции 802.1x на портах из указанного списка или для указанных MAC-адресов. |
| config 802.1x reauth | [port_based ports [<portlist\|all>] \|mac_based<br>ports [<portlist>\|all] {mac_address <macaddr>}] | Повторная аутентификация на коммутаторе уже аутентифицированных устройств на основе MAC- |

| | | |
|---|---|---|
| | | адресов или номеров портов. |
| config radius add | <server_index 1-3><br><server_ip><br>key <passwd 32><br>default<br>auth_port <udp_port_number><br>acct_port <udp_port_number> | Настройка параметров коммутатора для работы с сервером RADIUS. |
| config radius delete | <server_index 1-3> | Удаление ранее созданной конфигурации для работы с сервером RADIUS. |
| config radius | <server_index 1-3><br>ipaddress <server_ip><br>key <passwd 32><br>auth_port <udp_port_number><br>acct_port <udp_port_number> | Изменение параметров коммутатора для работы с сервером RADIUS. |
| show radius | | Просмотр текущей конфигурации RADIUS на коммутаторе. |

*Пример1*. Активировать 802.1x на устройстве.

DES-3226S#enable 802.1x

Command: enable 802.1x

Success.

*Пример 2*. Просмотр состояния аутентификации 802.1x на портах коммутатора.

DES-3226S#show 802.1x auth_state ports 1-5

Command: show 802.1x auth_state ports 1-5

| Port | Auth PAE State | Backend State | Port Status |
|---|---|---|---|
| 1 | ForceAuth | Success | Authorized |
| 2 | ForceAuth | Success | Authorized |
| 3 | ForceAuth | Success | Authorized |
| 4 | ForceAuth | Success | Authorized |
| 5 | ForceAuth | Success | Authorized |

*Пример 3*. Просмотр текущей конфигурации 802.1x.

DES-3226S#show 802.1x auth_configuration ports 1

Command: show 802.1x auth_configuration ports 1

802.1X : Enabled

Authentication Mode : Port_based

Authentication Protocol : Radius_Eap

Port number : 15:1

Capability : None

AdminCrlDir : Both

OpenCrlDir : Both

Port Control : Auto

QuietPeriod : 60 sec

TxPeriod : 30 sec

SuppTimeout : 30 sec

ServerTimeout : 30 sec

MaxReq : 2 times

ReAuthPeriod : 3600 sec

ReAuthenticate : Disabled

*Пример 4*. Настроить 802.1x на портах с 1 по 10.

DES-3226S#config 802.1x capability ports 1 – 10 authenticator

Command: config 802.1x capability ports 1-10 authenticator

Success.

*Пример 5*. Настроить режим аутентификации 802.1x на основе портов.

DES-3226S#config 802.1x auth_mode port_based

Command: config 802.1x auth_mode port_based

Success.

*Пример 6*. Инициализировать повторную аутентификацию 802.1x для портов 1-18.

DES-3226S#config 802.1x reauth mac_based ports 1-18

Command: config 802.1x reauth mac_based ports 1-18

Success.

*Пример 7*. Настроить коммуникационные параметры для сервера Radius:

DES-3226S#config radius add 1 10.48.74.121 key dlink default

Command: config radius add 1 10.48.74.121 key dlink default

Success.

## 10.4 Access Control Lists (ACL)

Списки управления доступом (Access Control Lists) обеспечивают ограничение прохождения трафика через коммутатор. Профили доступа указывают коммутатору, какие виды пакетов принимать, а какие – отвергать. Прием пакетов или отказ в приеме основывается на определенных признаках, таких как адрес источника, адрес приемника или адрес порта.

Профиль управления доступом дает возможность управлять трафиком и просматривать определенные пакеты, применяя списки доступа (ACL) на всех интерфейсах коммутатора.

В коммутаторах D-Link существует два основных типа профилей управления доступом: Ethernet и IP. Фильтрация в этих типах профилей может выполняться на основе MAC -адресов источника и приемника, VLAN, IP-адресов, номеров портов.

Профили доступа работают последовательно, в порядке возрастания их номеров (Profile ID). Пакет проверяется на соответствие условиям, указанным в профилях доступа, начиная с

первого профиля. Если профиль подходит, пакет или принимается или отбрасывается и дальше не проверяется. Если не один профиль не подходит, применяется политика по умолчанию, разрешающая прохождение всего трафика.

### 10.4.1 Создание профилей доступа (с использованием Web-интерфейса)

Процесс создание профиля доступа делится на 2 основные части:

1) Создание маски профиля доступа: указывается какую часть или части кадра будет проверять коммутатор, например MAC адрес источника или IP адрес назначения.
2) Создание правил профиля доступа: вводится условие, которое коммутатор будет использовать для определения действий над кадром (принять или отбросить).

***Шаг 1*:** Создание маски профиля (Access Profile Mask)

1. Зайдите на Web-интерфейс управления коммутатором. Выберите пункт Advanced Setup/ Access Profile Mask Setting.

2. Щелкните на кнопке *New* на странице таблицы настройки масок профилей Access Profile Mask Setting. Появится новое меню. Используйте его для создания профиля доступа и укажите, какие условия использовать для проверки кадра. Как только профиль будет создан, к профилю можно будет применить правила.

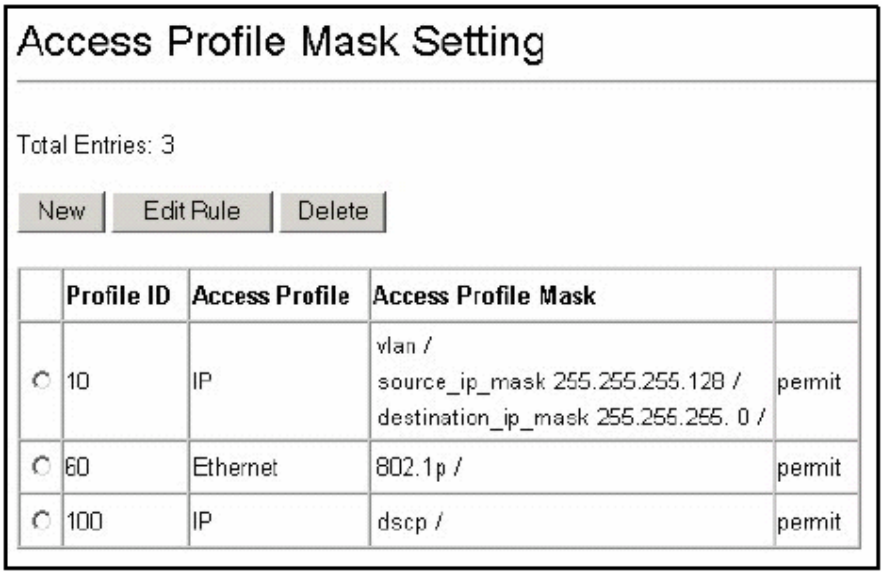



Рисунок 38. Таблица настроек масок профилей


Рисунок 39. Профили доступа на уровне MAC

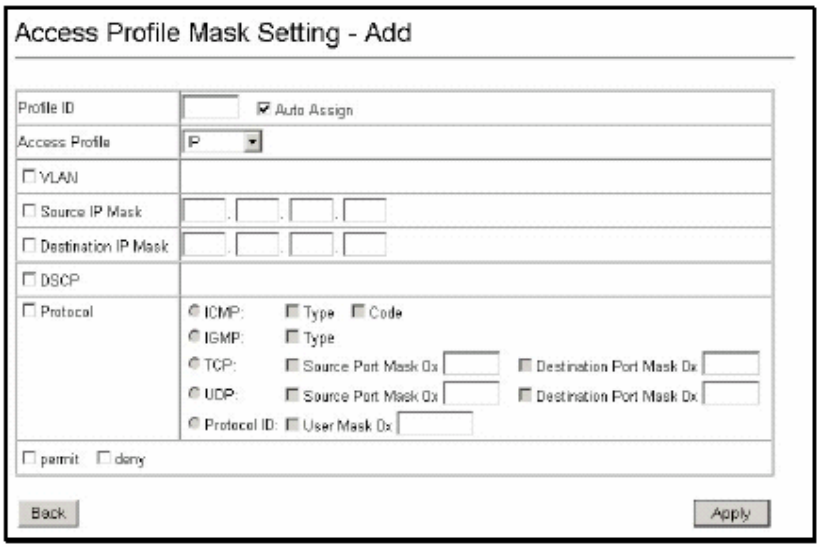



Рисунок 40. Профили доступа на уровне IP

3. Задайте следующие параметры маски профиля доступа:

1) **Идентификатор профиля (Profile ID)**: Наберите уникальный идентификационный номер для профиля или разрешите установить этот номер автоматически, выбрав опцию Auto Assign. Это значение может быть в диапазоне от 1 до 255.

2) **Профиль доступа (Access Profile**): Выберите профиль Ethernet или IP. Вид меню изменится в соответствии с требованиями для выбранного типа профиля (см. рисунки выше). Используйте Ethernet, для того, чтобы коммутатор исследовал часть заголовка 2-го уровня каждого пакета. Используйте IP, для того, чтобы коммутатор исследовал IP адрес в заголовке каждого кадра.

3) **VLAN**: Выберите эту опцию, для того, чтобы коммутатор исследовал поле VLAN заголовка каждого пакета и использовал его в качестве критерия или части критерия для принятия решения о передаче пакетов.

Для профиля Ethernet:

4) **Маска адреса источника MAC (Source MAC Mask)**: Маска адреса источника MAC – введите маску MAC адреса для MAC адреса источника.

5) **Маска адреса назначения MAC (Destination MAC Mask)**: Маска адреса назначения MAC – введите маску MAC адреса для MAC адреса назначения.

6) **802.1p**: Выберите эту опцию, для того, чтобы коммутатор исследовал значение приоритета 802.1p заголовка каждого пакета и использовал его в качестве критерия или части критерия для принятия решения о передаче пакетов.

7) **Ethernet Type**: Выберите эту опцию для того, чтобы коммутатор исследовал значение типа Ethernet в заголовке каждого кадра.

Для профиля IP:

4) **Маска адреса источника IP (Source IP Mask):** Маска адреса источника IP - введите маску IP адреса для IP адреса источника.

5) **Маска адреса назначения IP (Destination IP Mask):** Маска адреса назначения IP - введите маску IP адреса для IP адреса назначения.

6) **DSCP**: Выберите эту опцию, для того, чтобы коммутатор исследовал DiffServ Code Point (DSCP) поле каждого пакета и использовал его в качестве критерия или части критерия для принятия решения о передаче пакетов.

7) **Protocol**: Выберите эту опцию для того, чтобы коммутатор исследовал определенные поля соответствующих протоколов (ICMP, IGMP, TCP, UDP) в заголовке каждого кадра.

Для протоколов TCP и UDP в качестве критерия указываются номера портов приложений. Можно использовать либо номер порта источника, либо номер порта приемника, либо оба порта вместе.

В поле *Source Port Mask Ox* укажите маску порта TCP/UDP для порта источника в шестнадцатеричном виде (hex 0x0-0xffff).

В поле *Destination Port Mask Ox* укажите маску порта TCP/UDP для порта приемника в шестнадцатеричном виде (hex 0x0-0xffff).

8) **Permit/Deny** (для всех профилей).

*Permit* –указывает на то, что пакет, который соответствует профилю будет принят и передан коммутатором.

*Deny* –указывает на то, что пакет, который соответствует профилю будет отброшен коммутатором.

***Шаг 2*:** Создание правила для маски профиля доступа.

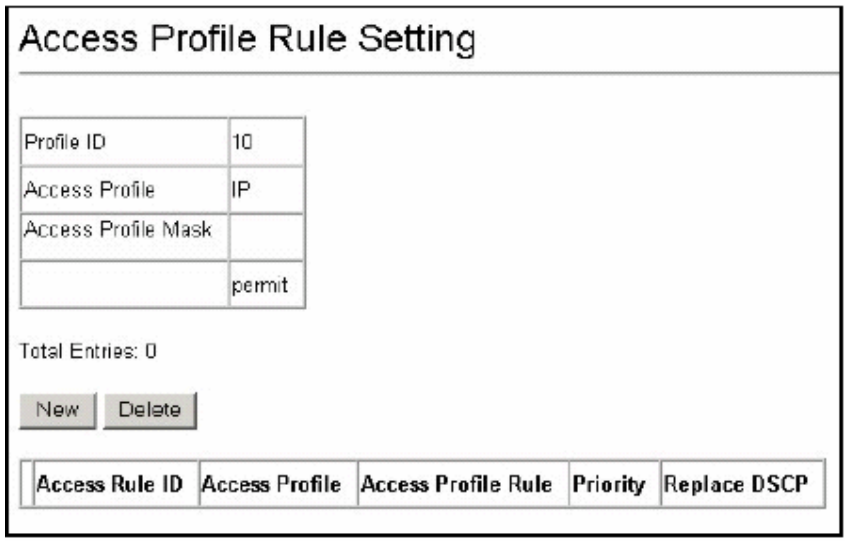



Рисунок 41. Установка правил профиля доступа

1. Выберите нужный профиль доступа в таблице настроек масок профилей и нажмите кнопку *Edit Rule*.

2.Создайте новое правило для профиля доступа, щелкнув на кнопке *New*. Удалить, ранее созданное правило, можно, выделив его и нажав кнопку *Delete*.

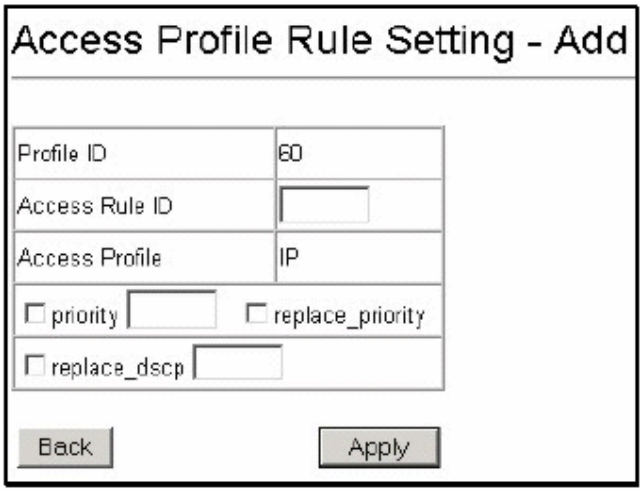



Рисунок 42. Создание правила для профиля доступа

3. Введите значения в соответствие с ранее заданной маской профиля.

В случае необходимости, при совпадении профиля, значения тега для 802.1p может быть заменено новым, меняющим приоритет пакета. Для этого надо выбрать опцию priority и ввести нужное значение в соседнем поле. Самый низший приоритет имеет значение 0, наивысший –7.

### 10.4.2 Алгоритм создания профиля доступа

1) Проанализируйте задачи фильтрации и определите, какой профиль доступа использовать: Ethernet или IP.

2) Определитесь со стратегией и запишите ее.

3) Основываясь на стратегии, определите, какие маски профиля доступа Access Profile Mask нужны и создайте их

4) Добавьте правила Access Profile Rule связанные с маской.

### 10.4.3 Настройка Access Control Lists (ACL) с помощью CLI

В таблице приведены команды для настройки ACL с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| create access_profile | ethernet<br>vlan<br>source_mac <macmask><br>destination_mac <macmask><br>802.1p<br>ethernet_type<br>ip<br>vlan<br>source_ip_mask <netmask><br>destination_ip_mask <netmask><br>dscp<br>icmp<br>type<br>code<br>igmp<br>type<br>tcp<br>src_port_mask <hex 0x0-0xffff><br>dst_port_mask <hex 0x0-0xffff><br>udp<br>udp src_port_mask <hex 0x0-0xffff><br>dst_port_mask <hex 0x0-0xffff><br>protocol_id<br>user_mask <hex 0x0-0xffffffff><br>permit<br>deny<br>profile_id <value 1-255>} | Создание профиля доступа на коммутаторе и определение того, какую часть заголовка каждого входящего кадра будет проверять коммутатор. Маска определяет те значения, которые коммутатор будет проверять в указанных полях заголовка кадра. |
| delete access_profile | Profile_id <value 1-255> | Удалить ранее созданный профиль доступа. |
| config access_profile profile_id <value 1-255> | access_profile<br>profile_id <value 1-255><br>add access_id <value 1-255><br>ethernet<br>vlan <vlan_name 32><br>source_mac <macaddr><br>destination_mac <macaddr><br>802.1p <value 0-7><br>ethernet_type <hex 0x0-0xffff><br>ip<br>vlan <vlan_name 32><br>source_ip <ipaddr><br>destination_ip <ipaddr><br>dscp <value> | Конфигурирование профиля доступа и определение значений, на основе которых коммутатор отфильтрует пакет или передаст по месту назначения. Маска, введенная в команде create access_profile, укажет коммутатору то место в заголовке пакета, которое необходимо проверить, чтобы принять решение о фильтрации. |

| | | |
|---|---|---|
| | icmp<br>type <value 0-255><br>code <value 0-255><br>igmp<br>type <value 0-255><br>tcp<br>src_port <value 0-65535><br>dst_port <value 0-65535><br><br>udp<br>src_port <value 0-65535><br>dst_port <value 0-65535><br>protocol_id <value 0-255><br>user_define <hex 0x0-0xffffffff><br>priority <value 0-7><br>replace_priority<br>replace_dscp <value 0-63><br>delete <value 1-255> | |

* *Примечание*: В коммутаторах существуют ограничения на максимальное количество профилей доступа и правил, определенных для них. Так, например, коммутатор DES-3226S может поддерживать максимально 10 профилей доступа, содержащих максимум 50 правил (50 правил – суммарное количество правил для всех 10 определенных профилей).

**Примеры профилей доступа**

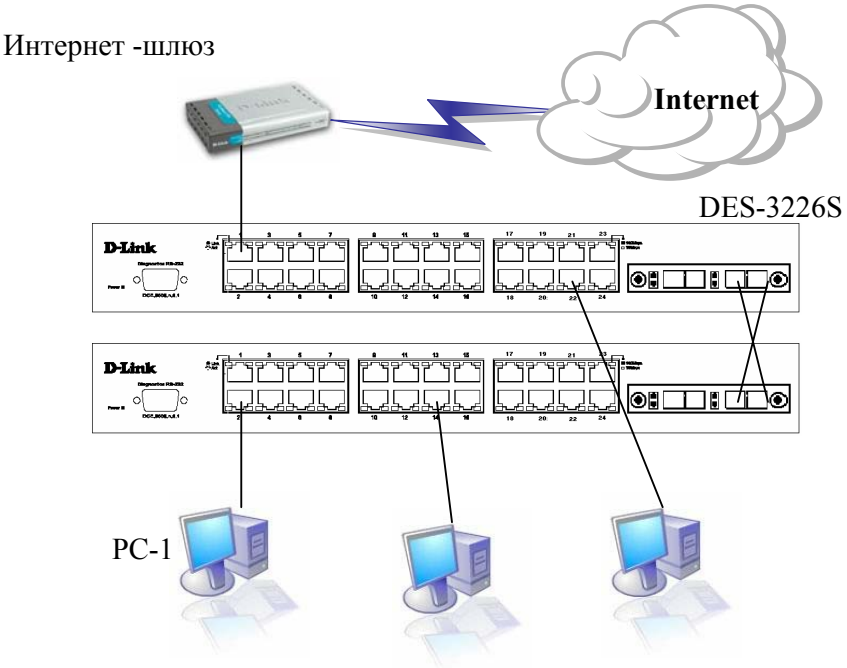



Рисунок 43.

*Пример 1*.

В этом примере профиль доступа разрешает доступ в Интернет только рабочим станциям с заданными MAC-адресами. Трафик других станций блокируется.

Интернет-шлюз:
IP=10.254.254.251/8
MAC: 00-50-BA-00-00-19

PC1: Разрешен доступ в Интернет
IP: 10.1.1.1/8,
MAC: 0050ba6b18c8
Gateway=10.254.254.251

Остальные компьютеры (Запрещен доступ в Интернет)
IP: 10.x.x.x/8

Определим правила и создадим профили доступа (с помощью CLI).

Правила:
*Если* **MAC-адрес назначения = адрес шлюза** и **MAC-адрес источника = адрес разрешенного компьютера**, то *разрешить* (можно ввести несколько таких правил для других компьютеров, которым разрешен доступ)
*Если* **MAC-адрес назначения = адрес шлюза**, то *запретить*.
*Иначе*, *разрешить* все остальное по умолчанию

Команды:
#Разрешить компьютеру с MAC 0050ba6b18c8 получать доступ в Интернет

DES-3226S# create access_profile ethernet source_mac FF-FF-FF-FF-FF-FF destination_mac FF-FF-FF-FF-FF-FF permit profile_id 10

DES-3226S# config access_profile profile_id 10 add access_id 11 ethernet source_mac 00-50-ba-6b-18-c8 destination_mac 00-50-ba-00-00-19

# Запретить остальным компьютерам доступ к Интернет

DES-3226S#create access_profile ethernet destination_mac FF-FF-FF-FF-FF-FF deny profile_id 20

DES-3226S# config access_profile profile_id 20 add access_id 21 ethernet destination_mac 00-50-ba-00-00-19

*Пример 2*.

В этом примере профиль доступа запрещает доступ в Интернет рабочей станции с заданным MAC-адресом. Трафик других станций разрешен.

Интернет-шлюз:
IP=10.254.254.251/8
MAC: 00-50-BA-00-00-19

PC1: Запрещен доступ в Интернет

IP: 10.1.1.1/8,
MAC: 0050ba6b18c8
Gateway=10.254.254.251

Остальные компьютеры (Разрешен доступ в Интернет)
IP: 10.x.x.x/8

Правила:
*Если* **MAC-адрес назначения = адрес шлюза** и **MAC-адрес источника = адрес запрещенного компьютера**, то *запретить* (можно ввести несколько таких правил для других компьютеров, которым запрещен доступ)
*Иначе*, *разрешить* все остальное по умолчанию

Команды:
# Запретить компьютеру с MAC 0050ba6b18c8  доступ в Интернет.

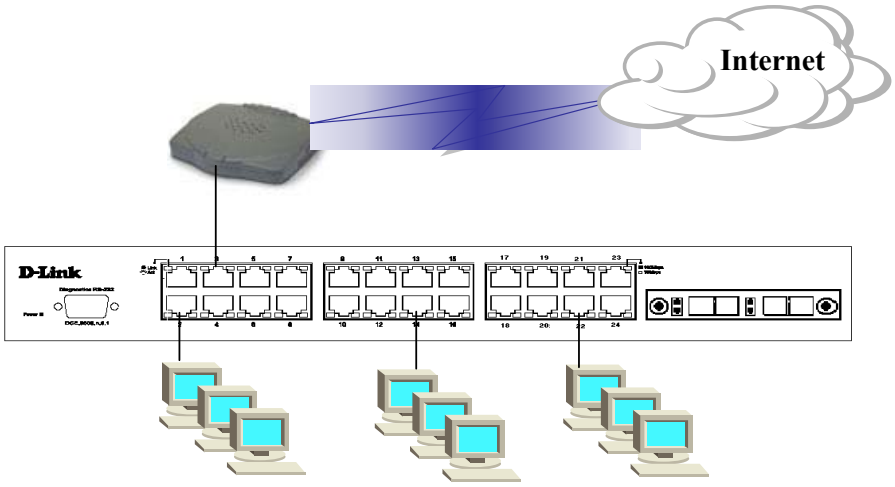
DES-3226S#create access_profile ethernet source_mac FF-FF-FF-FF-FF-FF destination_mac FF-FF-FF-FF-FF-FF deny profile_id 10

DES-3226S#config access_profile profile_id 10 add access_id 11 ethernet source_mac 00-50-ba-6b-18-c8 destination_mac 00-50-ba-00-00-19

# Разрешить остальным компьютерам  доступ к Интернет
(применяется команда по умолчанию, разрешающая весь трафик)

*Пример 3.*

В этом примере профиль доступа разрешает доступ в Интернет только рабочим станциям с заданными IP-адресами.  Трафик других станций блокируется.




.1 - .63

Другие компьютеры

Рисунок 44.

Интернет-шлюз:
IP= 192.168.1.254/32

Сеть: 192.168.1.x

Доступ в Интернет разрешен рабочим станциям с адресами:
192.168.1.1 - 192.168.1.63
Остальным компьютерам доступ в Интернет запрещен

Правила:
1. *Если* **DestIP=192.168.1.254/32** и **SrcIP=192.168.1.1/26**, то доступ *разрешен*
2. *Если* **DestIP=192.168.1.254/32** и **SrcIP=192.168.1.1/24**, то доступ *запрещен*
3. *Иначе*, по умолчанию *разрешить* доступ *всем*

Команды:
# Разрешить доступ компьютерам с адресами 192.168.1.1÷ 192.168.1.63 на шлюз 192.168.1.254

DES-3226S#create access_profile ip destination_ip_mask 255.255.255.255 source_ip_mask 255.255.255.192 permit profile_id 10

DES-3226S#config access_profile profile_id 10 add access_id 11 ip destination_ip 192.168.1.254 source_ip 192.168.1.1

# Запретить доступ компьютерам с адресами 192.168.1.1÷192.168.1.253 на шлюз 192.168.1.254

DES-3226S#create access_profile ip destination_ip_mask 255.255.255.255 source_ip_mask 255.255.255.0 deny profile_id 20

DES-3226S# config access_profile profile_id 20 add access_id 21 ip destination_ip 192.168.1.254 soruce_ip 192.168.1.1

# Иначе, по умолчанию разрешить доступ

*Пример4* (профиль доступа для коммутатора 3-го уровня).

Разрешить доступ только в одну сеть из других подсетей. Подсеть 1(192.168.1.x) может быть доступной из подсети 2, подсети 3, подсети 4. Подсеть 2, подсеть 3, подсеть 4 не имеют доступ друг к другу.

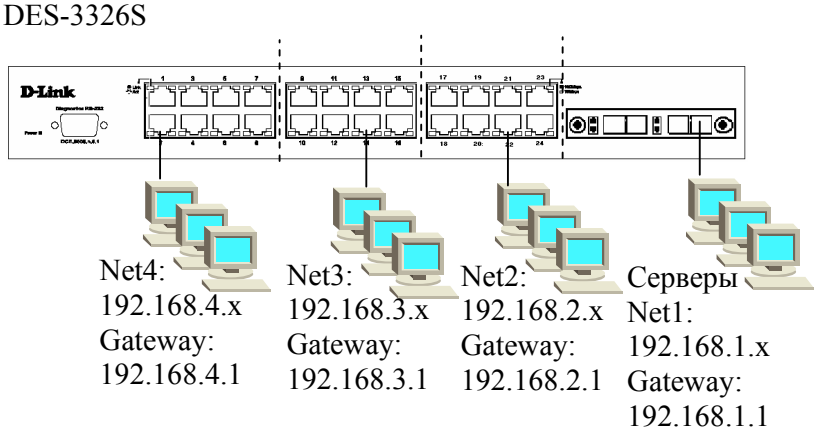



Рисунок 45.

Правила:
1. *Если* **Dest. IP=192.168.1.x**, то *разрешить* доступ
2. *Если* **Src. IP=192.168.1.x**, то *разрешить* доступ
3. *Если* **DestIP=192.168.2.x** и **SrcIP=192.168.2.x**, то *разрешить* доступ
4. *Если* **DestIP=192.168.3.x** и **SrcIP=192.168.3.**x, то *разрешить* доступ
5. *Если* **DestIP=192.168.4.**x и **SrcIP=192.168.4.x**, то *разрешить* доступ
6. *Запретить все* остальное

Команды:
# Разрешить доступ только к подсети 192.168.1.x из других подсетей
DES-3226S#create access_profile ip destination_ip_mask 255.255.255.0 permit profile_id 10

DES-3226S#config access_profile profile_id 10 add access_id 11 ip destination_ip 192.168.1.2

# Разрешить доступ только из подсети 192.168.1.x в другие подсети
DES-3226S#create access_profile ip source_ip_mask 255.255.255.0 permit profile_id 20

DES-3226S#config access_profile profile_id 20 add access_id 21 ip source_ip 192.168.1.2

# Разрешить доступ внутри подсетей 192.168.2.x, 192.168.3.x и 192.168.2.x.

```
DES-3226S#create access_profile ip source_ip_mask 255.255.255.0 destination_ip_mask 255.255.255.0 permit profile_id
30
DES-3226S#config access_profile profile_id 30 add access_id 31 ip source_ip 192.168.2.2 destination_ip 192.168.2.2
DES-3226S#config access_profile profile_id 30 add access_id 32 ip source_ip 192.168.3.2 destination_ip 192.168.3.2
DES-3226S#config access_profile profile_id 30 add access_id 33 ip source_ip 192.168.4.2 destination_ip 192.168.4.2
```

#### здесь можно добавить другие сети, при необходимости

# Запретить все остальное.

```
DES-3226S#create access_profile ip source_ip_mask 0.0.0.0 deny profile_id 40
DES-3226S#config access_profile profile_id 40 add access_id 41 ip source_ip 0.0.0.0
```

*Пример 5*. Посмотреть все сконфигурированные на коммутаторе профили доступа можно с помощью команды **show access_profile**.

```
DES-3226S# show access_profile
Access Profile Table
Access Profile ID:1                    Mode : Deny
                                       TYPE : IP
=====================================================
MASK  Option  Source IP MASK
                    255.255.255.0
------------------- ---------------------
Access ID
------------------- ---------------------
1                   10.42.73.0
```

## 11. Многоадресная рассылка

*Многоадресная рассылка (Multicast)* – это технология экономии полосы пропускания, которая сокращает трафик за счет доставки одного потока информации сразу тысячам корпоративных или частных абонентов. Преимущества многоадресной рассылки используют такие приложения как видеоконференции, корпоративная связь, дистанционное обучение.

Суть многоадресной рассылки заключается в том, что она позволяет нескольким получателям принимать сообщения без передачи сообщений каждому узлу широковещательного домена.

Многоадресная рассылка предполагает отправку сообщений или данных 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. После этого пакеты многоадресной рассылки IP, содержащие групповой адрес в поле назначения заголовка, будут поступать на этот узел и обрабатываться.

### 11.1 Адресация многоадресной рассылки

Групповые адреса определяют произвольную группу IP-узлов, присоединившихся к этой группе и желающих получать адресованный ей трафик.

Назначением групповых адресов управляет IANA (Internet Assigned Numbers Authority, Агентство по выделению имен и уникальных параметров протоколов Интернет). Оно выделило для групповой IP-адресации старые адреса класса D. Это означает, что область многоадресной рассылки охватывает адреса с 224.0.0.0 до 239.255.255.255.

IANA зарезервировало область многоадресной рассылки IP 224.0.0.0-224.0.0.255 для сетевых протоколов сегментов локальных сетей. Пакеты с такими адресами никогда не выходят за пределы локальной сети. Вторая группа адресов в диапазоне 224.0.1.0-224.0.1.255 – это глобальные адреса, которые могут использоваться для многоадресной передачи данных в Интернет.

### 11.2 MAC-адреса групповой рассылки

Как правило, рабочие станции локальной сети получают и обрабатывают пакеты только в случае совпадения MAC-адреса назначения пакета с их собственным MAC-адресом или если MAC-адрес – широковещательный. Существует несколько средств для того, чтобы несколько узлов многоадресной группы могли получить один и тот же пакет.

Многоадресная рассылка IP формирует MAC-адрес на основе адреса групповой рассылки IP третьего уровня. Кадр MAC имеет стандартный префикс из 24 битов. Этот префикс –01-00-05 – используется для всех адресов многоадресной рассылки Ethernet. При это оставшиеся 24 бита используются для создания MAC-адреса многоадресной рассылки. Когда создается MAC-адрес, 25-й бит (или бит высокого порядка) приравнивается к 0, и затем последние 23 бита IP-адреса преобразуются в остальные 23 бита MAC-адреса. Это проиллюстрировано на рисунке.

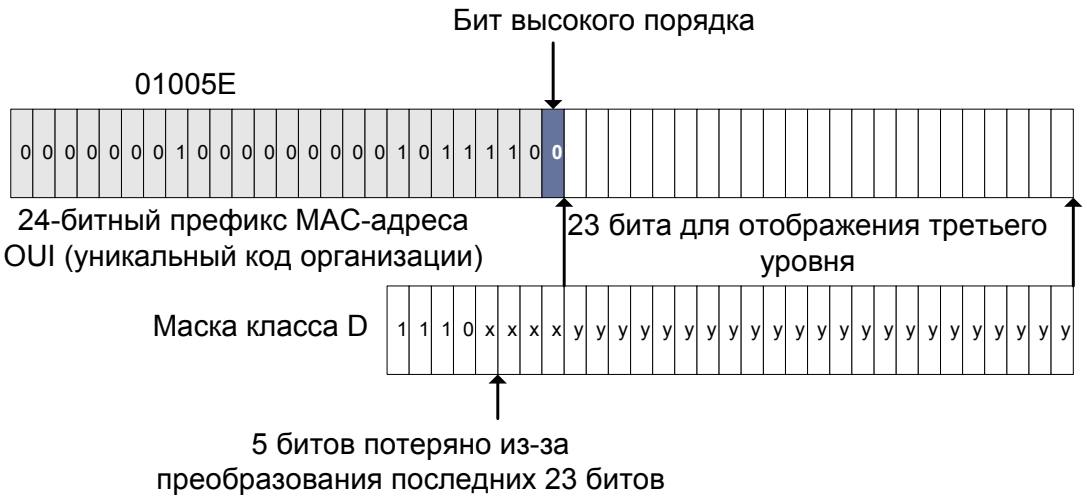



Рисунок 46. Преобразование группового IP-адреса в адрес Ethernet.

Поскольку при преобразовании теряются 5 битов 1-го октета IP-адреса, получившийся адрес не является уникальным. На самом деле одному MAC-адресу соответствует 32 группы многоадресной рассылки.

### 11.3 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** (Internet Group Management Protocol, межсетевой протокол управления группами) 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 получают IGMP-сообщения и периодически посылают запросы, чтобы определить, какие группы активны или неактивны в данной сети.

**Протокол IGMP v1**

В версии 1 протокола IGMP существуют два типа IGMP-сообщений:

- Запрос о принадлежности к группе;
- Ответ о принадлежности к группе.

Узлы отсылают IGMP-ответы, которые соответствуют определенной многоадресной группе, чтобы подтвердить свое желание присоединиться к этой группе. Маршрутизатор периодически отправляет IGMP-запрос, чтобы убедиться, что хотя бы один узел в подсети еще намерен получать трафик, предназначенный для этой группы. При отсутствии ответа на три последовательных IGMP-запроса, маршрутизатор отключает группу и прекращает передавать адресованный ей трафик.

**Протокол IGMP v2**

В версии 2 протокола IGMP существуют четыре типа IGMP-сообщений:

- Запрос о принадлежности к группе;
- Ответ о принадлежности к группе по версии 1;
- Ответ о принадлежности к группе по версии 2;
- Покинуть группу.

В основном работа IGMP 2 не отличается от IGMP 1. Разница заключается в наличие сообщений о выходе из группы. Теперь узлы сами могут сообщить локальному многоадресному маршрутизатору о намерении покинуть группу. В ответ маршрутизатор отсылает группе специальный запрос, чтобы определить, остались ли в ней еще узлы, желающие получать данный трафик. Если ответа не поступит, маршрутизатор отключает группу и прекращает передачу трафика. Это может значительно сократить задержки, связанные с прекращением членства в группе, по сравнению с IGMP 1. Нежелательный и ненужный трафик может быть прекращен гораздо быстрее.


Рисунок 47. Принцип работы протокола IGMP. Первый рисунок показывает процесс подписки на группу, второй – выход из группы.

### 11.4 Управление многоадресной рассылкой на 2 уровне

Стандартное поведение коммутатора 2-ого уровня заключается в передаче всего многоадресного трафика на каждый порт, принадлежащий локальной сети-приемнику на данном коммутаторе. Это связано с тем, что коммутатор не находит записи об MAC-адресе групповой рассылки в своей таблице коммутации, и поэтому рассылает пакеты через все порты. Это противоречит основному назначению коммутатора, которое заключается в ограничении

трафика и доставке его только тем портам, для которых такие данные действительно предназначены.

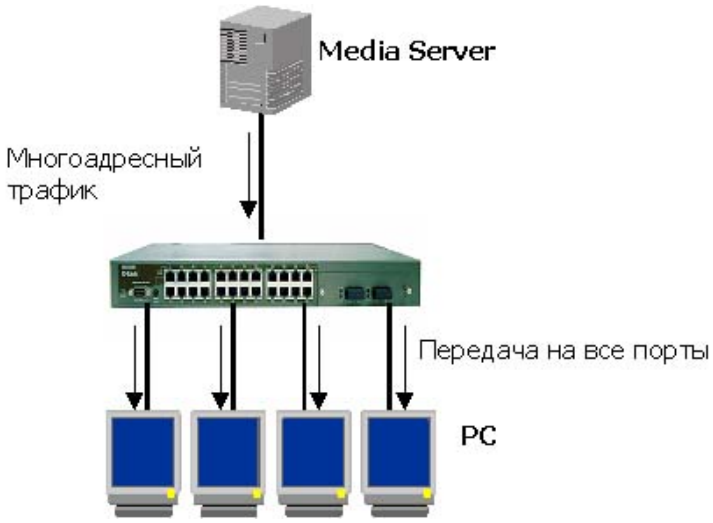



Рисунок 48. Передача многоадресного трафика без поддержки управления им на коммутаторе.

Управление многоадресной рассылкой на коммутаторе может быть выполнено несколькими способами:

Виртуальные локальные сети VLAN могут определять соответствующие границы многоадресной группы. Этот подход прост, однако он не поддерживает динамическое добавление или исключение членов из группы.

Второй метод, который поддерживается коммутаторами D-Link – **IGMP-прослушивание (IGMP-snooping)**. IGMP-прослушивание – это проверки или прослушивание локальной сети на наличие в IGMP-пакетах, передаваемых между узлом и маршрутизатором, некоторой информации 3-его уровня. Когда коммутатор получает IGMP-отчет узла для многоадресной группы, он заносит номер порта узла в запись своей ассоциированной многоадресной таблицы. Когда коммутатор получает IGMP-сообщение о выходе узла из группы, он удаляет номер порта этого узла из записи таблицы.

Поскольку управляющие IGMP-сообщения передаются в виде многоадресных пакетов, они неотличимы от многоадресных данных 2-ого уровня. Коммутатор, на котором осуществляется IGMP-прослушивание, проверяет все многоадресные пакеты и ищет среди них те, которые содержат управляющую информацию. IGMP-прослушивание сильно загружает центральный процессор и может снизить производительность коммутатора. Поэтому в коммутаторах обычно используются специализированные микросхемы, которые проверяют IGMP-сообщения на аппаратном уровне.

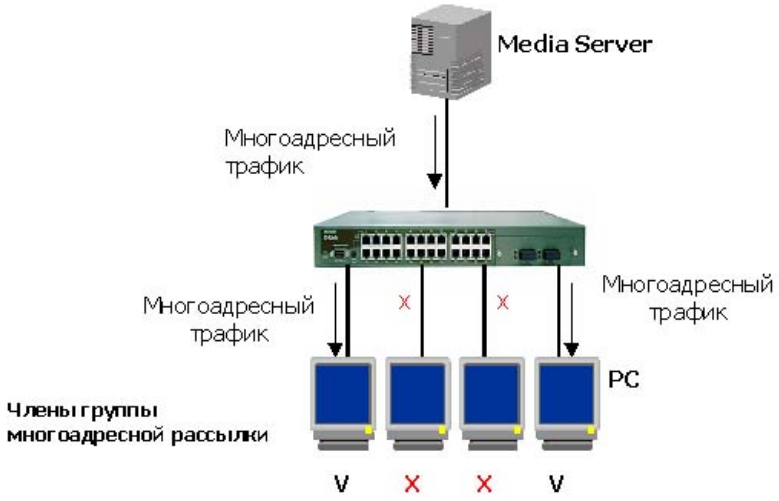



Рисунок 49. Передача многоадресного трафика с поддержкой IGMP-snooping.

### 11.5 Настройка IGMP- snooping с помощью CLI

В таблице приведены команды для настройки IGMP-snooping с помощью CLI.

| Команда | Параметры | Описание |
|---|---|---|
| config igmp_snooping | <vlan_name 32><br>all<br>host_timeout <sec 1-16711450><br>router_timeout <sec 1-16711450><br>leave_timer <sec 1-16711450><br>robustness_variable <value 1-255><br>last_member_query_interval <sec 1-25><br>state [enable\|disable] | Настройка IGMP-snooping на коммутаторе. |
| config igmp_snooping querier | <vlan_name 32><br>all<br>query_interval <sec 1-65535><br>max_response_time <sec 1-25><br>robustness_variable <value 1-255><br>last_member_query_interval <sec 1-25><br>state [enable\|disable] | Настройка времени в секундах между передачами основной очереди, максимальное время в секундах ожидания отчета от члена группы и разрешенное количество потерь пакетов в подсети. |
| config router_ports | <vlan_name 32> [add\|delete]<br><portlist> | Настройка указанных портов для подключения к многоадресному маршрутизатору. Это будет гарантировать, что все пакеты достигнут многоадресного маршрутизатора несмотря на протокол и т.п. |
| enable igmp snooping | forward_mcrouter_only | Активизировать IGMP-snooping на коммутаторе. **forward_mcrouter_only** указывает, что коммутатор будет направлять весь мультикастовый трафик только многоадресному маршрутизатору с поддержкой мультикастов. В противном случае, коммутатор будет направлять трафик любому IP маршрутизатору. |

| | | |
|---|---|---|
| disable igmp snooping | | Отключить IGMP-snooping на коммутаторе. |
| show igmp snooping | vlan <vlan_name 32> | Просмотр текущего состояния IGMP-snooping на коммутаторе. |
| show igmp snooping group | vlan <vlan_name 32> | Просмотр текущей конфигурации IGMP-snooping группы на коммутаторе. |
| show router ports | vlan <vlan_name 32><br>static<br>dynamic | Просмотр текущей конфигурации портов маршрутизатора на коммутаторе. |

*Пример 1*. Настроить IGMP-snooping.

DES-3226S#config igmp_snooping default host_timeout 250 state enable

Command: config igmp_snooping default host_timeout 250 state enable

Success.

Параметр **host_timeout <sec>** - Задаёт максимальное время, в течение которого узел может быть членом многоадресной группы без получения коммутатором отчёта узла о нахождении в группе. Время по умолчанию 260 сек.

*Пример 2*. Настроить IGMP-snooping querier на коммутаторе.

DES-3226S#config igmp_snooping querier default query_interval 125 state enable

Command: config igmp_snooping querier default query_interval 125 state enable

Success.

*Пример 3*. Включить IGMP-snooping на коммутаторе.

DES-3226S#enable igmp_snooping

Command: enable igmp_snooping

Success.

*Пример 4*. Настроить статические порты для подключения к маршрутизатору.

DES-3226S#config router_ports default add 1-10

Command: config router_ports default add 1-10

Success.

*Пример 5*. Проверить текущее состояние IGMP-snooping на коммутаторе.

DES-3226S#show igmp_snooping

Command: show igmp_snooping

IGMP Snooping Global State : Disabled

Multicast router Only : Disabled
VLAN Name : default
Query Interval : 125
Max Response Time : 10
Robustness Value : 2
Last Member Query Interval : 1
Host Timeout : 260
Route Timeout : 260
Leave Timer : 2
Querier State : Disabled
Querier Router Behavior : Non-Querier
State : Disabled

VLAN Name : vlan2
Query Interval : 125
Max Response Time : 10
Robustness Value : 2
Last Member Query Interval : 1
Host Timeout : 260
Route Timeout : 260
Leave Timer : 2
Querier State : Disabled
Querier Router Behavior : Non-Querier
State : Disabled

Total Entries: 2

Литература:


1. Руководство пользователя коммутатора D-Link DES-3226S.
2. Материалы для тренингов D-Link.
3. Компьютерные сети. Принципы, технологии, протоколы. В.Г. Олифер, Н.А. Олифер. – СПб:Питер, 2000.
4. Руководство по технологиям объединенных сетей, 3-е издание.: Пер. с англ.– М.:Издательский дом «Вильямс», 2002.
5. Качество обслуживания в сетях IP.: Вегешна Шринивас. Пер. с англ.– М.:Издательский дом «Вильямс», 2003.
6. ЭНЦИКЛОПЕДИЯ СЕТЕВОГО ОБОРУДОВАНИЯ. http://www.routers.ru

**ПРИЛОЖЕНИЕ А. Синтаксис команд.**

Следующие символы используются для описания ввода команд и ожидаемых значений и аргументов. Оперативная справка содержится в CLI и доступна через интерфейс консоли, используя аналогичный синтаксис.

*Примечание:* Все команды чувствительны к регистру. Перед вводом команды проверьте, что отключен Caps Lock или другие нежелательные функции, которые изменят регистр текста.

| **<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, одно из которых должно быть введено. |
| **{ фигурные скобки }** | |
| Назначение | Содержит необязательное значение или набор необязательных аргументов. |

**ПРИЛОЖЕНИЕ В. Глоссарий**

# 1

**10Base-F.** Реализация стандарта IEEE 802.3 Ethernet с использованием оптического кабеля.

**10Base-T.** Спецификация IEEE 802.3i для сетей Ethernet с использованием неэкранированного кабеля на основе скрученных пар ("витая пара").

**100Base-T**. Спецификация IEEE 802.3us для сетей Ethernet со скоростью передачи 100 Мбит/сек на основе неэкранированного кабеля на основе скрученных пар ("витая пара").

**100Base-TX.** Часть спецификации IEEE 802.3u Ethernet для скорости 100 Мбит/с с использованием 2-пар неэкранированного медного кабеля категории 5.

**100Base-FX.** Часть спецификации IEEE 802.3u Ethernet для скорости 100 Мбит/с с использованием оптических кабелей и стандарта FDDI TP-PMD для PMD (физическая среда).

**10Gbase-ER.** Оптический интерфейс 10G Ethernet, обеспечивающий передачу сигналов со скоростью 10 Гбит/с по одномодовому кабелю протяженностью около 40 км (25 миль) при длине волны оптического излучения 1550 нм.

**10Gbase-LR.** Оптический интерфейс 10G Ethernet, обеспечивающий передачу сигналов со скоростью 10 Гбит/с по одномодовому кабелю протяженностью более 10 км (6 миль) при длине волны оптического излучения 1310 нм.

**10Gbase-SR.** Оптический интерфейс 10G Ethernet, обеспечивающий передачу сигналов со скоростью 10 Гбит/с по многомодовому кабелю протяженностью до 300 метров (990 футов) при длине волны оптического излучения 850 нм.

# A

**Access method.** Метод доступа. Набор правил, обеспечивающих арбитраж доступа к среде передачи. Примерами методов доступа являются CSMA/CD (Ethernet) и передача маркера (Token Ring).

**Address**. Адрес. Уникальный идентификатор, присваиваемый сети или сетевому устройству для того, чтобы другие сети и устройства могли распознать его при обмене информацией.

**Address mask.** Адресная маска. Битовая маска, используемая для выбора битов из адреса Internet для адресации подсети. Маска имеет размер 32 бита и выделяет сетевую часть адреса Inetrnet и один или несколько битов локальной части адреса. Иногда называется маской подсети.

**Address resolution** (разрешение адреса). Используется для преобразования адресов сетевого уровня (Network Layer) в обусловленные средой (media-specific) адреса. См. также ARP.

**Agent** . Агент. Применительно к SNMP термин агент означает управляющую систему.
В модели клиент-сервер - часть системы, выполняющая подготовку информации и обмен ею между клиентской и серверной частью.

**Aggregate link.** Агрегированный канал. Это объединение нескольких физических каналов в одну логическую магистраль.

**Application Layer.** Уровень приложений. Верхний уровень модели OSI, обеспечивающий такие коммуникационные услуги, как электронная почта и перенос файлов.

**ARP: Address Resolution Protocol.** Протокол разрешения адресов. Протокол Internet, используемый для динамического преобразования адресов Internet в физические (аппаратные) адреса устройств локальной сети. В общем случае ARP требует передачи широковещательных сообщений всем узлам, на которое отвечает узел с соответствующим запросу IP-адресом.

# B

**Backbone**. Магистраль, часть сети, по которой передается основной трафик и которая является чаще всего источником и приемником других сетей.

**Backplane**. Объединительная плата. Физическое соединение между интерфейсным процессором или платой, шинами данных и шинами распределения питания системного блока устройства.

**Bandwidth**. Полоса пропускания, диапазон между самой высокой и самой низкой частотой, доступной для передачи сетевых сигналов. Диапазон частот измеряется в герцах (Гц).

**Bridge.** Мост. Устройство, соединяющее две или несколько физических сетей и передающее пакеты из одной сети в другую. Мосты работают на канальном уровне OSI модели.

**Bridge Protocol Data Unit (BPDU).** Модуль данных мостового протокола. Пакет приветствия протокола связующего дерева (Spanning –Tree Protocol), который посылается через заданные интервалы времени для обмена информацией между мостами сети.

**Broadcast.** Широковещание. Система доставки пакетов, при которой копия каждого пакета передается всем узлам, подключенным к сети. Примером широковещательной сети является Ethernet.

**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

**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.

**CLI.** Command Line Interface , интерфейс командной строки. Позволяет пользователю взаимодействовать с операционной системой 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**. Коллизионный домен. Часть сети Ethernet, все узлы которой распознают коллизию независимо от того, в какой части сети эта коллизия возникла.

**CoS.** Class of Service, класс обслуживания. Характеристика, позволяющая определить, как протокол верхнего уровня использует протокол нижнего уровня для обработки его сообщений. Другое название ToS.

**Crossover.** Перекрестное соединение. Соединение (внешнее или внутреннее) передатчика на одном конце коммуникационного канала с приемником на другом его конце.

**CSMA/CD.** Carrier sense multiple access/collision detection - множественный доступ к среде с обнаружением конфликтов и распознаванием несущей. Механизм доступа, при котором устройства, готовые для передачи данных, сначала проверяют наличие частоты. Если ее нет в течении некоторого заданного промежутка времени, то устройства могут приступать к передаче данных. При одновременной передаче со стороны двух устройств возникает коллизия, которая может быть обнаружена всеми вызвавшими ее устройствами. Такая коллизия, в свою очередь, задерживает повторную передачу данных этими устройствами на некоторое произвольное время. CSMA/CD –доступ используется в Ethernet и IEEE 802.3.

**Cut-through packet switching**. Сквозная коммутация пакетов. Способ коммутации, при котором данные проходят через коммутатор таким образом, что ведущий край пакета покидает коммутатор на выходном порте еще до того, как закончится прием пакета на входном порте. Устройство со сквозной коммутацией пакетов считывает, обрабатывает и передает пакеты сразу после определения адреса приемника и выходного порта. Этот способ также называется оперативной коммутацией пакетов.

# D

**Data Link Layer** . Канальный уровень. Уровень 2 в модели OSI, который обеспечивает надежную передачу данных по физическому соединению. Канальный уровень отвечает за физическую адресацию, сетевую топологию, дисциплину линии связи, уведомления об ошибках, упорядоченную доставку кадров и управление потоком. IEEE делит этот уровень на два подуровня: MAC и LLC.

**Designated bridge**. Назначенный мост. Мост, через который проходит самый дешевый маршрут для передачи кадра из сегмента к корневому мосту.

**DHCP.** Dynamic Host Configuration Protocol, протокол динамической конфигурации узла. Обеспечивает механизм динамического распределения и повторного использования освобождаемых IP-адресов.

# E

**EIA RS-232**. Стандарт EIA для 25-контактного последовательного интерфейса, используемого для соединения ПК или терминалов (DTE) с коммуникационным оборудованием (DCE) типа модемов.

**EMI.** Electromagnetic interference, электромагнитная интерференция. Взаимное наложение электромагнитных сигналов, из-за которых может нарушиться целостность сигналов и увеличиться частота ошибок в каналах передачи данны.

**Encapsulation**. Инкапсуляция. Метод, используемый многоуровневыми протоколами, в которых уровни добавляют заголовки в модуль данных протокола (protocol data unit - PDU ) из вышележащего. В терминах Internet - пакет содержит заголовок физического уровня, за

которым следует заголовок сетевого уровня (IP), а за ним - заголовок транспортного уровня (TCP), за которым располагаются данные прикладных протоколов.

**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/сек) и Gigabit Ethernet (1000 Мбит/с).

**Ethernet address**. Адрес Ethernet. 48-битовое значение, являющееся уникальным идентификатором устройства (порта Ethernet) в сети. Обычно записывается 12 шестнадцатиричными цифрами.

# F

**Fault management**. Контроль сбоев. Одна из пяти определенных ISO областей управления сетями. Основной задачей этой области сетевого управления является детектирование, изоляция и корректировка сбойных фрагментов сети.

**Fault tolerance.** Устойчивость к сбоям. Способность программы или системы корректно работать при возникновении сбоев. Устойчивые к сбоям системы создаются для обеспечения работы при отключении питания, повреждении дисков, серьезных ошибках пользователей и т.п.

**Fiber optic cable**. Оптический кабель. Кабель, содержащий одно или несколько оптических волокон и предназначенный для передачи данных.

**Filtering.** Фильтрация. Процесс проверки пакетов данных в сети и определения адресатов для принятия решения о дальнейшей пересылке (данная ЛВС, удаленная ЛВС) 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 за передачей данных между двумя точками сети и позволяющие избегать потери данных в результате переполнения приемных буферов.

**Forwarding table.** Таблица пересылки. Таблица, содержащая идентификаторы и адреса, а также пределы рассылки адресов.

**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.

**Full duplex.** Дуплексная передача. Одновременная передача данных между станцией-отправителем и станцией- получателем.

# G

**GUI**. 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.** Полудуплексный режим. Способность канала в каждый момент времени только передавать или принимать информацию. Прием и передача, таким образом, должны выполняться поочередно.

# I

**IEEE.** 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. Подкомитет 802 является частью технического комитета по компьютерным коммуникациям (Technical Committee for Computer Communications), основанного в 1980 году для обеспечения совместимости оборудования и программ различных фирм. Членами IEEE являются ANSI и ISO.

**IEEE 802**. Комитет IEEE 802. Один из комитетов IEEE, ответственный за разработку стандартов для локальных и городских сетей. Наибольшее распространение получили стандарты Ethernet, Token Ring, Wireless LAN.

**IEEE 802.3.** Спецификация IEEE для локальных сетей CSMA/CD.

**IGMP.** Internet Group Management Protocol, межсетевой протокол управления группами. Протокол, используемый IP-узлами для уведомления смежных широковещательных маршрутизаторов об их участии в широковещательных группах.

**IP.** Internet Protocol, IP- протокол. Часть стека протоколов TCP/IP, определенного в RFC 791 . 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.

**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.

# L

**LAN.** 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.

**LLC.** Logical Link Control, подуровень управления логическим соединением. Высший из двух подуровней канального уровня, определенный IEEE. Управляет обработкой ошибок, потоками, кадрированием, а также адресацией MAC-подуровня. Наиболее распространенным LLC-протоколом является IEEE 802.2. Существуют варианты IEEE 802.2 с подтверждением и без подтверждения.

# M

**MAC.** Media Access Control, управление доступом к передающей среде. Низший из двух подуровней канального уровня, определенный IEEE. MAC-подуровень управляет доступом к совместно используемым носителям.

**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 address learning**. Изучение MAC-адресов. Служба самообучающегося моста, в котором хранится MAC-адрес источника для каждого полученного пакета. Затем эти адреса используются для передачи следующих пакетов только через те мостовые интерфейсы, где расположены эти адреса. Пакеты с нераспознанным адресом передаются через все мостовые интерфейсы. Эта схема позволяет уменьшить трафик через присоединенные локальные сети. Служба изучения MAC-адресов определена в стандарте IEEE 802.1.

**MIB.** Management Information Base, база управляющей информации. База данных, где хранится информация для управления сетью, которая используется и поддерживается протоколом сетевого управления SNMP. Значение MIB-объекта может быть изменено или извлечено с помощью команд SNMP и сетевой системы управления (например, D-Link D -View) с GUI-интерфейсом. MIB-объекты образуют древовидную структуру с открытыми (стандартными) и закрытыми (частными) ветвями.

**MTU.** Maximum Transmission Unit, модуль передачи максимального размера. Максимальный размер (в байтах) пакета данных, который можно передать через данный интерфейс.

**Multicast.** Многоадресная рассылка. Режим копирования одиночных пакетов и их передачи заданному подмножеству сетевых адресов. Эти адреса задаются в поле адреса приемника (Destination address field).

**Multicast address.** Групповой адрес. Общий адрес, который относится к некоторой группе нескольких сетевых устройств.

**Multicast group.** Многоадресная группа. Динамически определенная группа IP-узлов, идентифицируемая одним групповым IP-адресом.

**Multicast router**. Многоадресный маршрутизатор. Маршрутизатор, используемый для передачи IGMP-запросов для присоединенных локальных сетей. Узлы, принадлежащие к многоадресной группе, отвечают на запрос посылкой IGMP-отчетов с обозначением тех широковещательных групп, к которым они относятся. Многоадресный маршрутизатор отвечает за передачу дейтаграмм от данной группы ко всем сетям, которые содержат членов этой группы.

# N

**Network .** Сеть.

1. Соединение группы узлов (компьютеров или других устройств).
2. Группа точек, узлов или станций, соединенных коммуникационными каналами и набор оборудования, обеспечивающего соединение станций и передачу между ними информации.

**Network Address**. Сетевой адрес. Адрес сетевого уровня, который относится к логическому, а не к физическому сетевому устройству. Он также называется протокольным адресом (protocol address).

**Network Layer**. Сетевой уровень. Уровень 3 модели OSI, отвечающий за маршрутизацию, переключение и доступ к подсетям через всю среду OSI.

**Node.** Узел. Точка присоединения к сети, устройство, подключенное к сети.

**NVRAM.** NonVolatile RAM, энергонезависимое ОЗУ. Оперативное запоминающее устройство, содержимое которого сохраняется при отключении питания.

# 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. Блок информации помечается на уровне 3 (сетевой) модели OSI.

**PDU.** Protocol Data Unit, модуль данных протокола. Термин OSI для пакетов данных.

**Physical Layer.** Физический уровень. Уровень 1 модели OSI. Определяет электрические, механические, процедурные и функциональные спецификации для создания, поддержки и разрыва физического соединения между конечными системами.

**Ping.** Packet INternet Groper, проверка доступности адресата. Эхо-сообщение ICMP и ответ на него. Инструмент, используемый для проверки доступности адресата в IP-сетях.

**Port density**. Плотность портов. Количество портов на шасси.

**Port security**. Безопасность портов. Функция, применяемая в коммутаторах для обеспечения безопасности.

**Proxy ARP**. 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 хосту.

# Q

**QoS**. Quality of Service, качество обслуживания. Показатель эффективности системы передачи данных, который отражает качество передачи.

# R

**RADIUS**. Remote Authentication Dial-In User Service, служба аутентификации удаленных пользователей. Предварительный стандарт IETF, обеспечивающий аутентификацию, проверку полномочий и другие операции при доступе в сеть удаленных пользователей по коммутируемым линиям.

**Redundancy.** Избыточность. Дублирование устройств, сервисов и соединений. В случае неисправности позволяет избыточным устройствам, службам и соединениям выполнять функции неисправных.

**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. Соотношение ожидаемых и полученных по каналу вспомогательных элементов сетевых служб. Чем выше это соотношение, тем надежнее линия.

**RMON.** Remote MONitoring, удаленный мониторинг. Спецификация MIB-агента, описанная в RFC 1271, которая определяет функции удаленного мониторинга сетевых устройств. Спецификация RMON предоставляет многочисленные возможности для мониторинга, определения неисправностей и отчетности.
Модуль удаленного мониторинга, позволяющий собирать информацию об устройстве и управлять им через сеть.

**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.

# 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.

**Session Layer.** Сеансовый уровень. Уровень 5 модели OSI, обеспечивающий способы ведения управляющего диалога между системами.

**SNMP.** Simple Network Management Protocol, простой протокол управления сетью. Протокол управления сетью, используемый почти исключительно в сетях TCP/IP. SNMP предоставляет средства контроля и управления сетевыми устройствами, конфигурациями, производительностью и безопасностью, а также сбора статистической информации.

**SOHO.** Small Office, Home Office, малый и домашний офис. Сетевые комплексы и технологии доступа для офисов не имеющих прямого подключения к крупным корпоративным сетям.

**Spanning Tree**. Связующее дерево. Нециклическая часть сетевой топологии.

**Spanning Tree Algorithm**. Алгоритм построения связующего дерева. Алгоритм, используемый протоколом связующего дерева для построения связующего дерева. Иногда применяется аббревиатура STA.

**Spanning Tree Protocol**. Протокол связующего дерева. Мостовой протокол, использующий алгоритм связующего дерева и позволяющий самообучающемуся мосту динамически обрабатывать петли в сетевой топологии путем создания связующего дерева. Мосты обнаруживают петли путем обмена сообщениями BPDU с другими мостами и ликвидируют их посредством блокирования выбранных мостовых интерфейсов.

**Store and forward packet switching**. Коммутация пакетов 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 пакетов без буферизации (cut-through packet switching).

**Switch.** Коммутатор. Сетевое устройство, которое фильтрует, пересылает и направляет кадры в зависимости от их адреса приемника. Коммутатор работает на канальном уровне модели OSI.

**Switch LAN.** Коммутируемая сеть. Локальная сеть с коммутаторами.

# T

**Tag.** Тег. Идентификационная информация, в том числе и номер.

**TCP.** Transmission Control Protocol, протокол управления передачей. Ориентированный на соединение протокол транспортного уровня, обеспечивающий надежную дуплексную передачу данных. TCP входит в набор протоколов TCP/IP.

**TCP/IP.** Transmission Control Protocol/Internet Protocol , протокол управления передачей/ Internet –протокол. Общее название набора протоколов, разработанных министерством обороны США в 1970-е гг. для всемирного сетевого комплекса.

**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.

**TFTP.** Trivial File Transfer Protocol, простейший протокол передачи файлов. Упрощенная версия протокола FTP, который позволяет компьютерам обмениваться файлами по сети.

**Throughput.** Пропускная способность. Объем информации, поступающей и, возможно, проходящей через определенный участок сети в определенный момент времени.

**Traffic segmentation.** Сегментация трафика. Функция, используемая в коммутаторах для разграничения доменов на уровне 2.

**Transport Layer.** Транспортный уровень. Уровень 4 модели OSI, отвечающий за надежную передачу данных между конечными системами.

**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. Основная магистраль состоит из нескольких магистралей.

# U

**UDP.** User Datagram Protocol, протокол дейтаграмм пользователя. Протокол транспортного уровня, не требующий подтверждения соединения. Входит в набор TCP/IP. UDP обеспечивает обмен дейтаграммами без подтверждения и гарантий доставки.

# V

**VLAN.** 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.