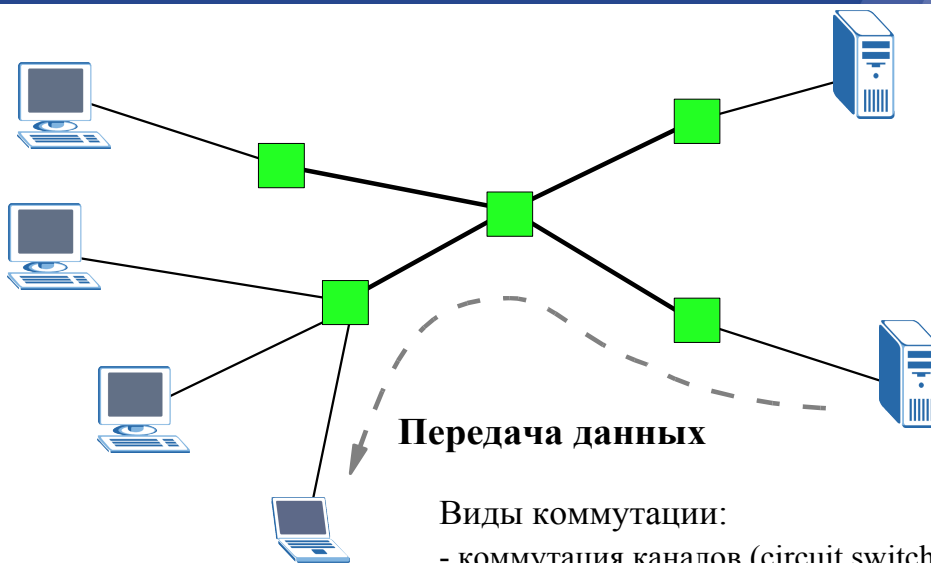


Сетевые протоколы и алгоритмы, используемые в оборудовании **ZyXEL**

Учебный курс
ZC00R
v3.04

ZyXEL

Компьютерные сети



Виды коммутации:

- коммутация каналов (circuit switching);
- коммутация сообщений (message switching);
- коммутация пакетов (packet switching).

ZyXEL

Коммуникационные сети должны обеспечивать связь своих абонентов между собой. Абонентами могут вступать ЭВМ, сегменты локальных сетей, факс-аппараты или телефонные собеседники. Как правило, в сетях общего доступа невозможно предоставить каждой паре абонентов собственную физическую линию связи, которой они могли бы монопольно «владеть» и использовать в любое время. Поэтому в сети всегда применяется какой-либо способ коммутации абонентов, который обеспечивает разделение имеющихся физических каналов между несколькими сеансами связи и между абонентами сети.

Существуют три принципиально различные схемы коммутации абонентов в сетях:

1. коммутация каналов (circuit switching);
2. коммутация сообщений (message switching);
3. коммутация пакетов (packet switching).

Коммутация каналов - процесс организации последовательности каналов, соединяющих пары абонентских или административных систем друг с другом. В результате коммуникации последовательность каналов соединяется в единый канал, проходящий через всю коммуникационную сеть.

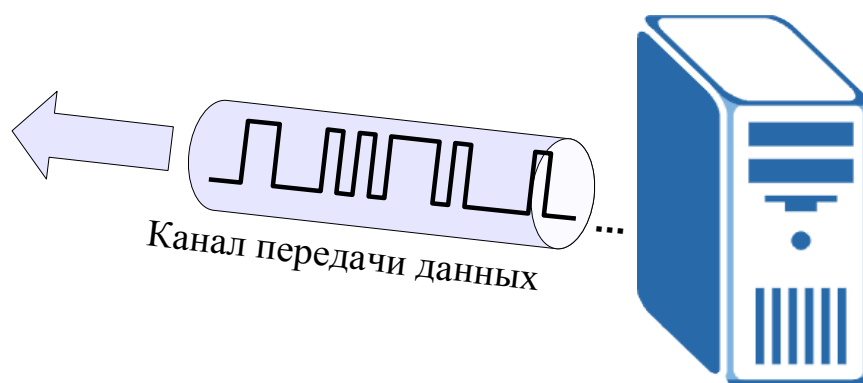
Коммутация сообщений - коммутация, обеспечивающая передачу через сеть сообщений с промежуточными этапами сборки, хранения и разборки в узлах коммутации.

Каждое промежуточное устройство принимает сообщение, локально его сохраняет и отправляет его при освобождении канала связи.

Коммутация пакетов - технология доставки сообщений, при которой данные, разбитые на отдельные пакеты, могут пересылаться из исходного пункта в пункт назначения разными маршрутами. Конкретный маршрут выбирается передающим и принимающим компьютерами, исходя из наличия соединения и объем трафика.

На данный момент наибольшее распространение получили сети с **пакетной коммутацией**.

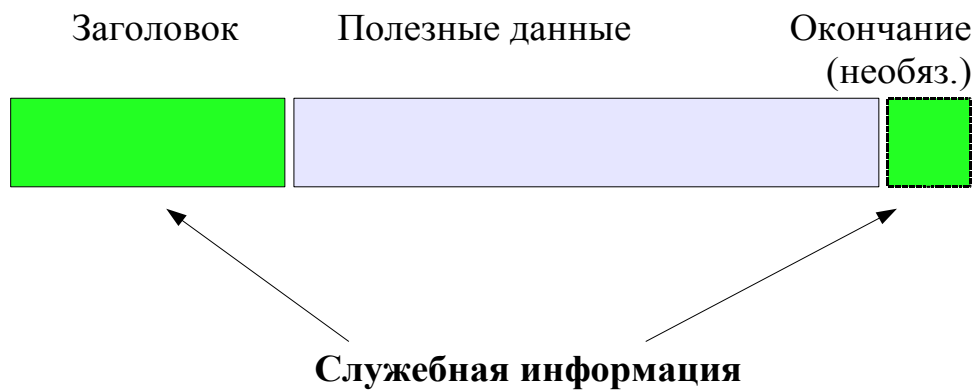
Передача данных



ZyXEL

Все виды данных в компьютерном мире представляют собой бинарный код. Каждый метод кодирования устанавливает по какому правилу данные будут представлены в виде сигналов. В случае электрических линий передачи каждому набору бит будет приведён в соответствие определённый набор уровней электрического сигнала. Некоторые методы кодирования обладают свойством самосинхронизации, что позволяет упростить процесс декодирования. Так же имеет большое значение то, какое количество данных может быть передано по физическому каналу

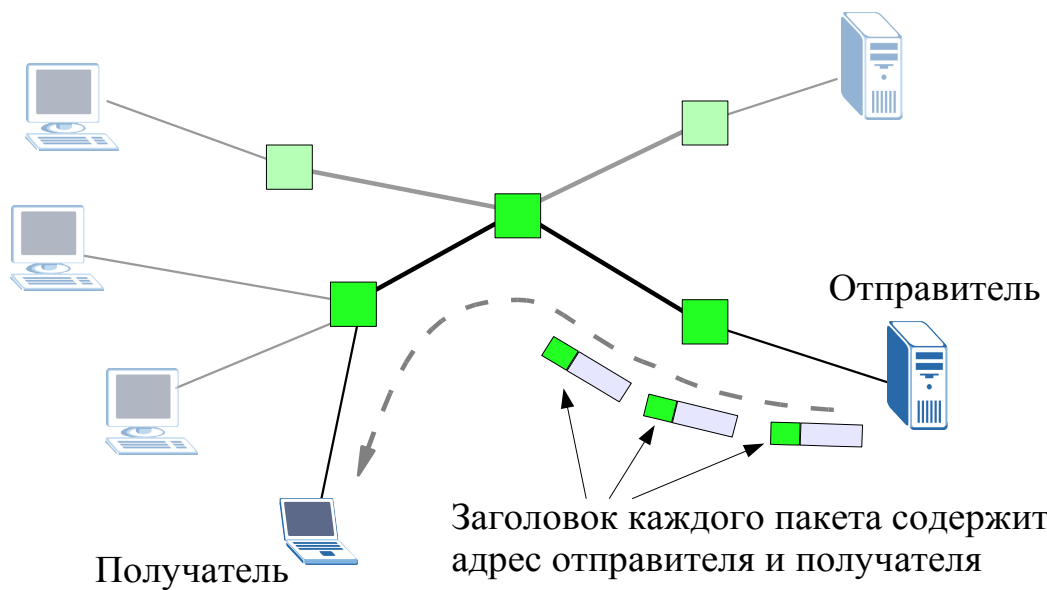
Пакеты



Сетевой протокол – набор правил, позволяющий осуществлять соединение и обмен данными между двумя включенными в сеть компьютерами. Протоколом описывается в том числе и формат пакетов.

ZyXEL

Адресация

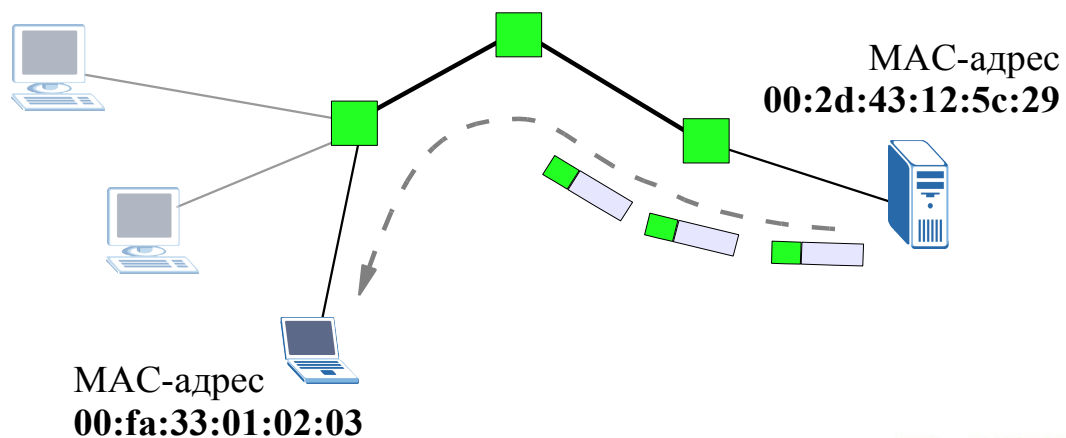


ZyXEL

Сети Ethernet

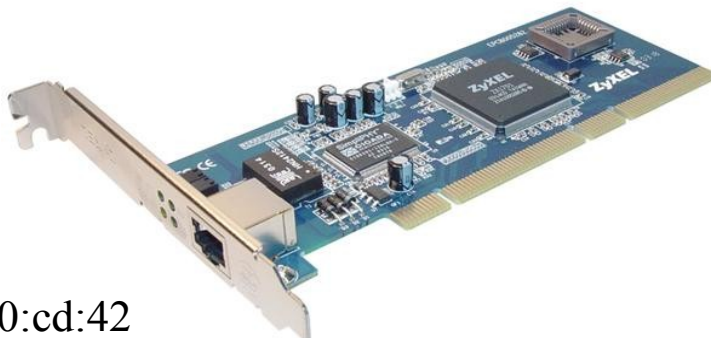
Ethernet – самая популярная технология локальных сетей. Отличается простотой и низкой стоимостью.

Адреса узлов Ethernet называются **MAC-адресами**.



Адаптер Ethernet

Каждое устройство Ethernet получает уникальный MAC-адрес на заводе. MAC-адрес имеет длину 6 байт, первые три из которых – идентификатор производителя.



MAC-адрес:

00:13:49:00:cd:42



Идентификатор производителя

ZyXEL

Кадр Ethernet

MAC-адрес получателя	MAC-адрес отправителя	Тип*	Полезные данные	FCS**
6 байт	6 байт	2 байта	от 46 до 1500 байт	4 байта

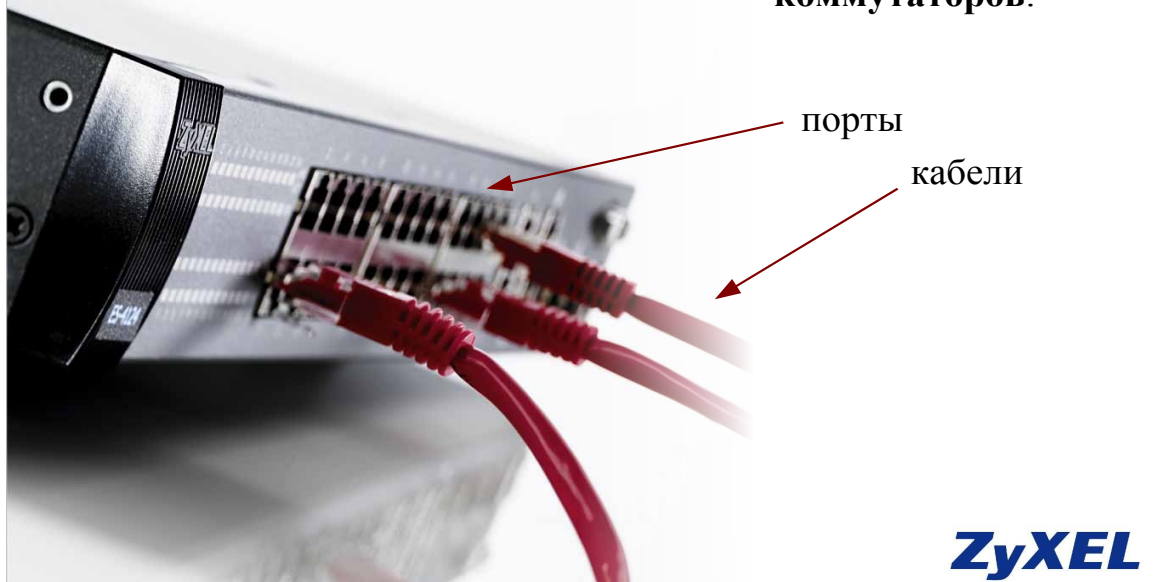
* тип содержит идентификатор, описывающий, что именно передается в теле пакета.

** FCS – Frame Checksum – контрольная сумма, позволяющая проверить, что данные не были повреждены при передаче.

ZyXEL

Сетевое оборудование

Задача сетевого оборудования – доставить кадр получателю. Современные сети Ethernet строятся на основе **коммутаторов**.



Коммутаторы Ethernet

Коммутаторы соединяются с оконечными узлами и между собой.

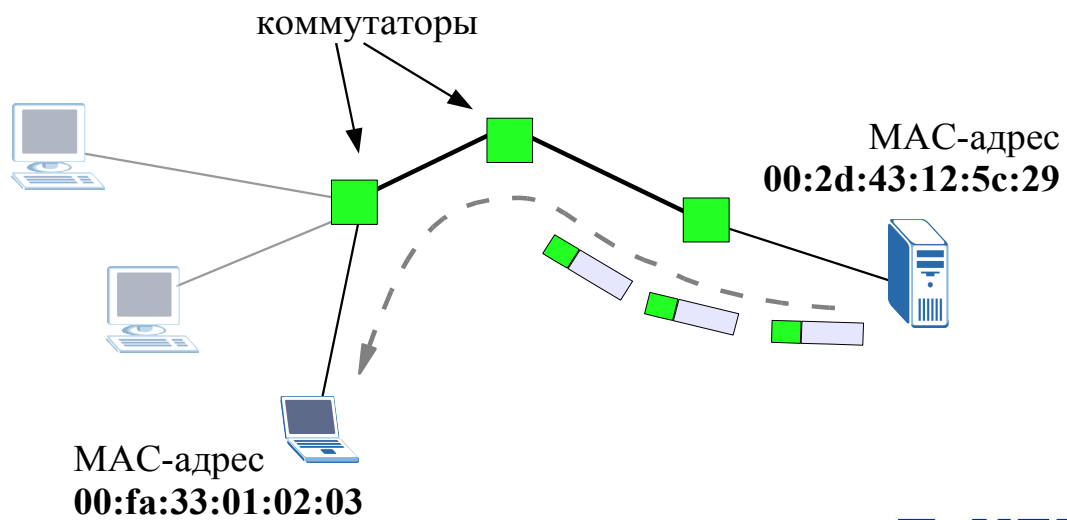
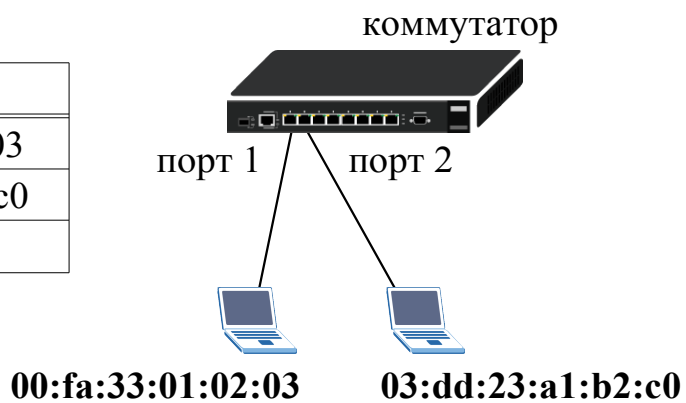


Таблица фильтрации

Откуда коммутатор знает, к какому порту подключен тот или иной получатель?

Таблица фильтрации:

Порт	MAC-адрес
1	00:fa:33:01:02:03
2	03:dd:23:a1:b2:c0
...	...



ZyXEL

Типы рассылок

MAC-адрес
получателя



- Одноадресная рассылка (Unicast)
- Групповая рассылка (Multicast)
восьмой бит MAC-адреса установлен в единицу
- Широковещательная рассылка – всем узлам (Broadcast)
MAC-адрес имеет вид **ff:ff:ff:ff:ff:ff**, все биты установлены в единицу

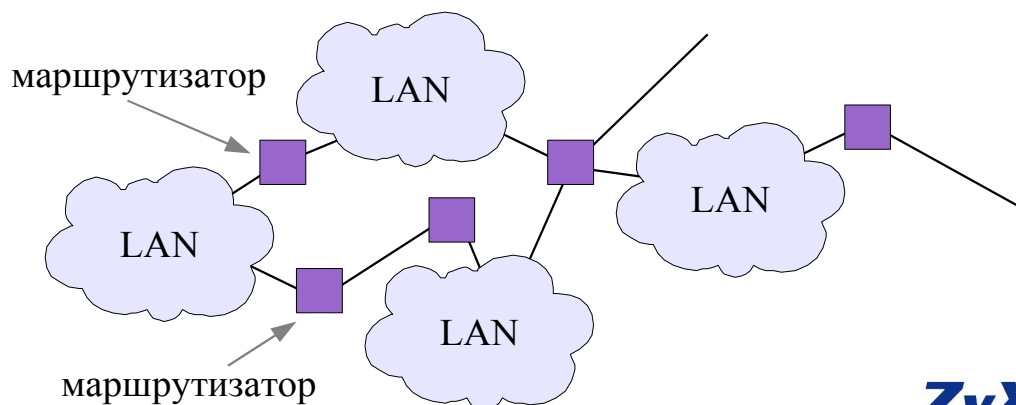
ZyXEL

Глобальные сети

Почему Ethernet не годится для построения глобальных сетей?

Потому что каждый коммутатор должен содержать таблицу *всех* MAC-адресов сети, а в глобальной сети миллионы узлов.

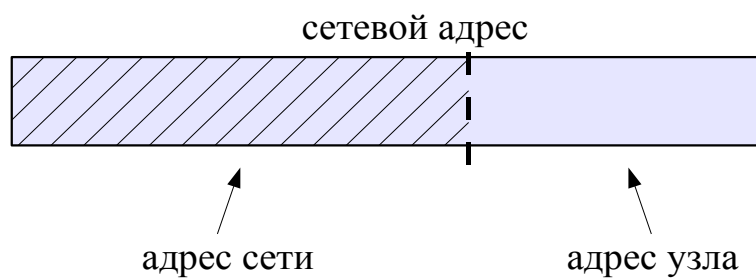
Глобальная сеть строится объединением локальных сетей с помощью *маршрутизаторов*.



ZyXEL

Сетевые адреса

Помимо локального MAC-адреса каждому узлу сети назначается сетевой адрес, который состоит из двух частей: адреса *сети* и адреса *узла*.



ZyXEL

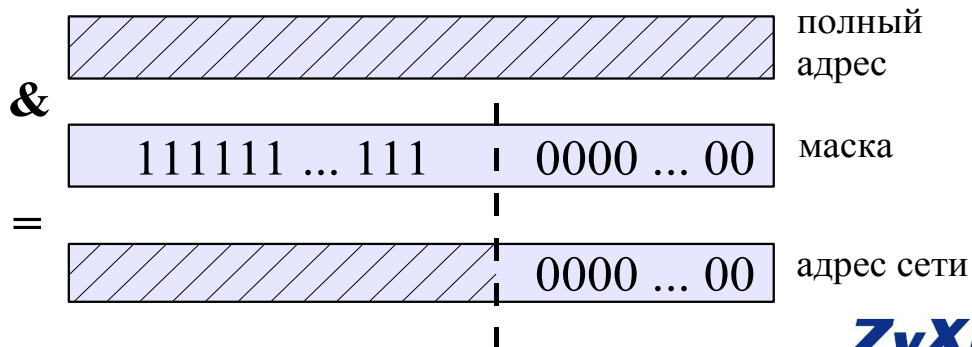
IP-адрес и маска

IP – Internet Protocol, самый распространенный сетевой протокол.

IP адрес имеет длину 4 байта и записывается так:

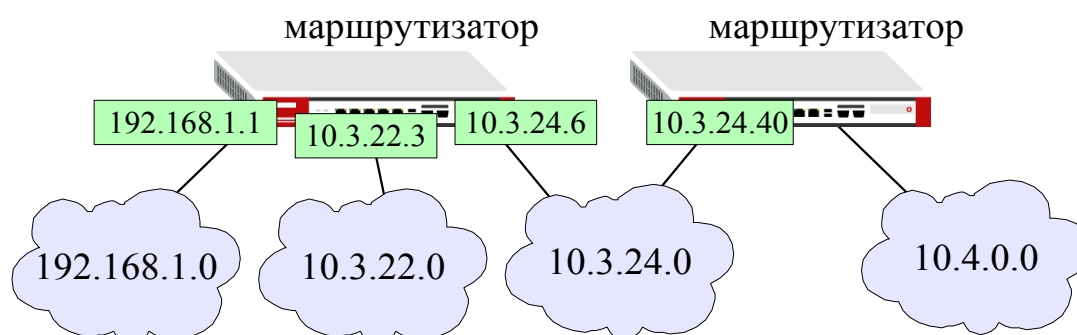
192 . 168 . 31 . 5

Какая часть IP-адреса отвечает за сеть, а какая за узел, определяется битовой *маской*.



ZyXEL

Таблица маршрутизации



Сеть	маска	выходной интерфейс	следующий шлюз
192.168.1.0	24	eth0	—
10.3.22.0	24	eth1	—
10.3.24.0	24	eth2	—
10.4.0.0	16	eth2	10.3.24.40

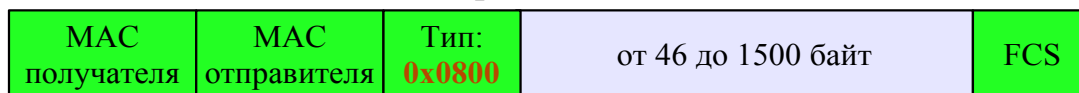
ZyXEL

Инкапсуляция

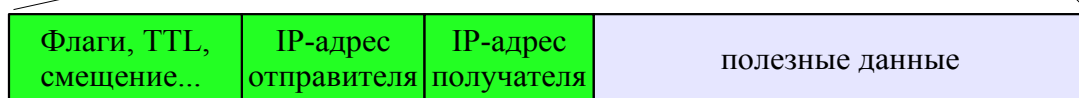
Для передачи IP-пакета по локальной сети, он вкладывается внутрь кадра Ethernet. Передача одного пакета в качестве «полезных данных» другого пакета называется *инкапсуляцией*.

Тип **0x0800** указывает на то, что передается IP.

Кадр Ethernet

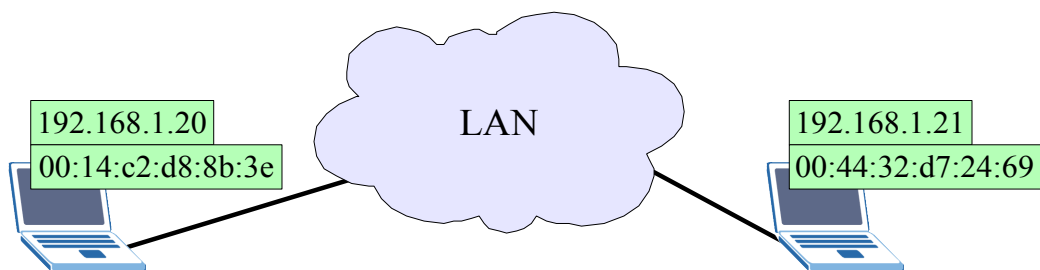


Пакет IP



ZyXEL

ARP



хочет отправить
пакет на адрес
192.168.1.21

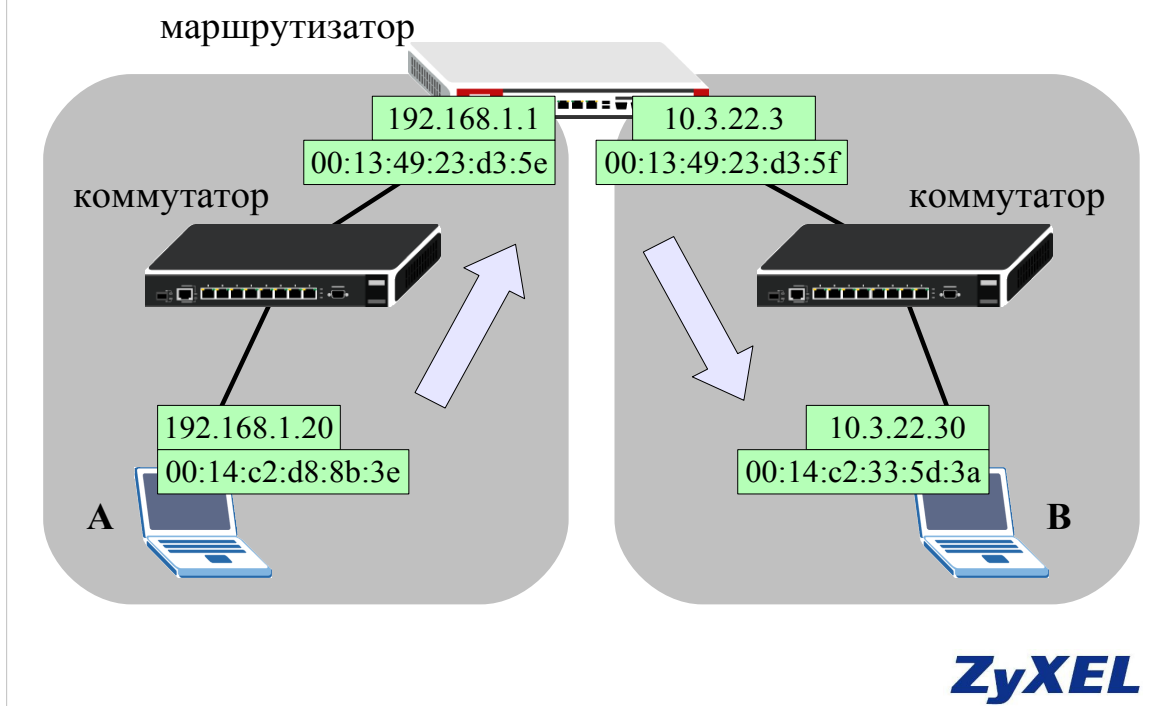
(*всем*) «У кого IP-адрес 192.168.1.21?»

«У меня IP-адрес 192.168.1.21 и
MAC-адрес 00:44:32:d7:24:69 »

передача

ZyXEL

Передача через шлюз



MAC-адреса меняются при передаче пакета из одной локальной сети в другую, а IP-адреса остаются прежними. Именно по ним маршрутизаторы определяют, куда передавать пакет. Сами маршрутизаторы имеют несколько IP- и MAC-адресов, в зависимости от количества поддерживаемых локальных сетей.

Алгоритм передачи пакета в составной сети с узла А на узел В состоит из следующих шагов:

- Узел А имеет следующие настройки: свой IP-адрес, маску и IP-адрес шлюза по умолчанию. Например, узел А на слайде имеет IP-адрес 192.168.1.20, маску 255.255.255.0 и IP-адрес шлюза 192.168.1.1. Обратите внимание, что при наложении маски на адрес А и адрес шлюза получается одна и та же подсеть: 192.168.1.0.
- Узел А накладывает маску на IP-адрес узла В, которому он хочет отправить пакет. Если получается та же подсеть – 192.168.1.0, узел А использует ARP и доставляет пакет напрямую, по локальной сети.
- Если при наложении маски получается другая подсеть, узел А с помощью ARP определяет MAC-адрес шлюза, и отправляет туда пакет.
- Получив пакет, шлюз видит по IP-адресу назначения, что пакет предназначен не ему, и начинает искать в таблице маршрутизации подходящую запись. В нашем примере, он находит подсеть 10.3.22.0, определяет с помощью ARP MAC-адрес узла с IP-адресом 10.3.22.30, и доставляет пакет по назначению.

TTL

В заголовке IP-пакета содержится значение TTL – Time to Live, которое уменьшается на единицу при прохождении маршрутизатора. Когда оно уменьшится до нуля, пакет будет удален. TTL помогает избавить сеть от «заблудившихся» пакетов, ходящих по кругу.

TTL занимает 8 байт.

Пакет IP

	TTL		IP-адрес отправителя	IP-адрес получателя	полезные данные
--	-----	--	-------------------------	------------------------	-----------------

ZyXEL

Длина заголовка и пакета

Заголовок и тело IP-пакета могут быть переменной длины, поэтому в заголовке содержится поле длины заголовка (IHL), и длины пакета (Total length).

Пакет IP

IHL	Total length	IP-адрес отправителя	IP-адрес получателя	полезные данные
-----	--------------	----------------------	---------------------	-----------------

ZyXEL

Фрагментация

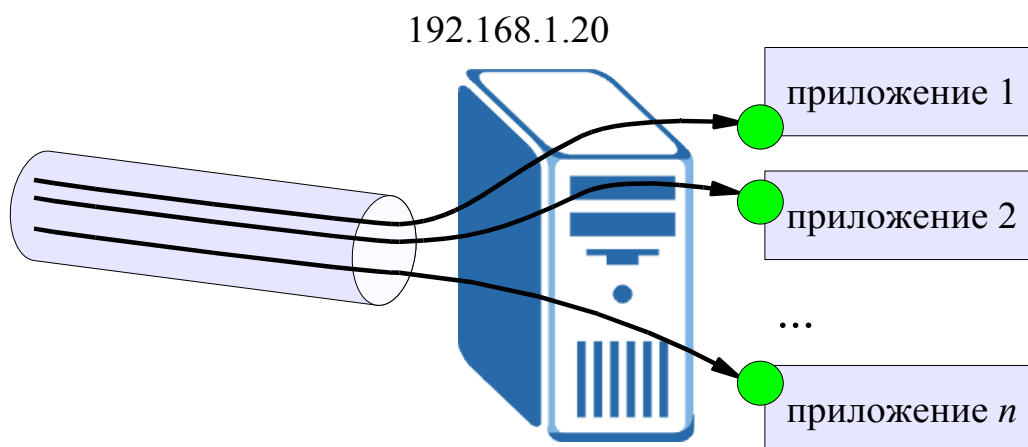
Длина IP-пакета может составлять до 65535 байт. Если стандарт локальной сети (например, Ethernet) не позволяет передавать такие большие блоки данных, маршрутизатор разобьет пакет на фрагменты.

Каждый фрагмент имеет свой заголовок IP, в котором указан номер фрагмента (Identification) и смещение фрагмента от начала пакета (Fragment offset).

Каждый сетевой интерфейс (порт) маршрутизатора имеет параметр MTU – Maximum Transmit Unit, в котором указан максимальный размер IP-пакета, поддерживаемый подключенной локальной сетью.

ZyXEL

Адресация приложений



На одном IP-адресе может работать несколько независимых приложений, поэтому внутри IP вкладывается еще один протокол, отвечающий за сеанс связи между приложениями.

ZyXEL

Сеансовый протокол UDP

Кадр Ethernet

MAC получателя	MAC отправителя	Тип: 0x0800	от 46 до 1500 байт	FCS
-------------------	--------------------	-----------------------	--------------------	-----

Пакет IP

	Protocol		IP-адрес отправителя	IP-адрес получателя	полезные данные
--	----------	--	-------------------------	------------------------	-----------------

Пакет UDP

порт отправителя	порт получателя	длина	контр. сумма	полезные данные
---------------------	--------------------	-------	-----------------	-----------------

порты идентифицируют взаимодействующие приложения в диапазоне от 0 до 65535.

ZyXEL

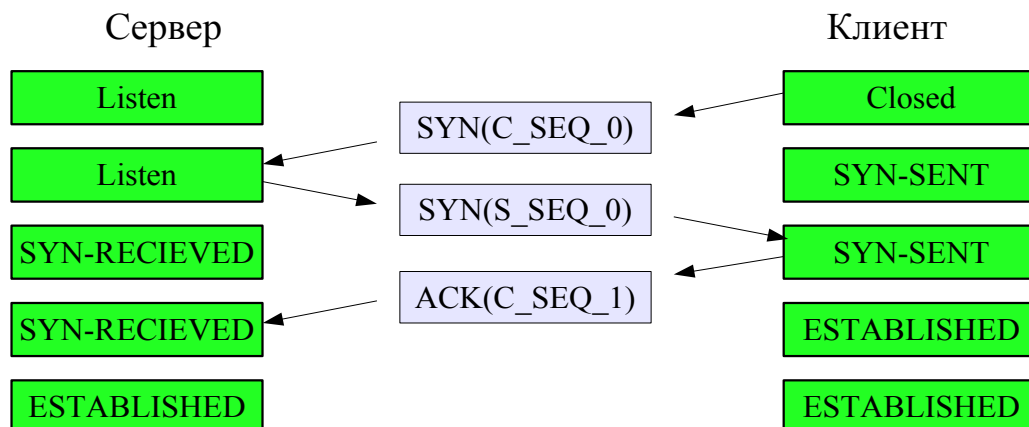
Для взаимодействия сетевых приложений протокол UDP использует 16-ти битные порты, которые могут принимать значения от 0 до 65535. Порт 0 является зарезервированным, но может использоваться как порт источника, если приложение не ожидает ответных данных.

Порты с 1 по 1023 являются системными и фиксированными, во многих ОС привязка к ним требует повышенных привилегий приложения.

Порты с 1024 по 49151 – зарегистрированные.

Порты с 49152 по 65535 – свободно используемые и временные. Используются клиентскими приложениями для связи с серверами.

Сеансовый протокол TCP



ZyXEL

TCP (англ. Transmission Control Protocol — протокол управления передачей) — один из основных сетевых протоколов Internet, предназначенный для управления передачей данных в сетях и подсетях TCP/IP.

Выполняет функции протокола транспортного уровня упрощённой модели OSI. IP-идентификатор — 6.

TCP — это транспортный механизм, предоставляющий поток данных, с предварительной установкой соединения, за счёт этого дающий уверенность в безошибочности получаемых данных, осуществляет повторный запрос данных в случае потери пакетов и устраняет дублирование при получении двух копий одного пакета. В отличие от UDP, TCP гарантирует, что приложение получит данные точно в такой же последовательности, в какой они были отправлены, и без потерь.

Установление связи клиент-сервер осуществляется в три этапа:

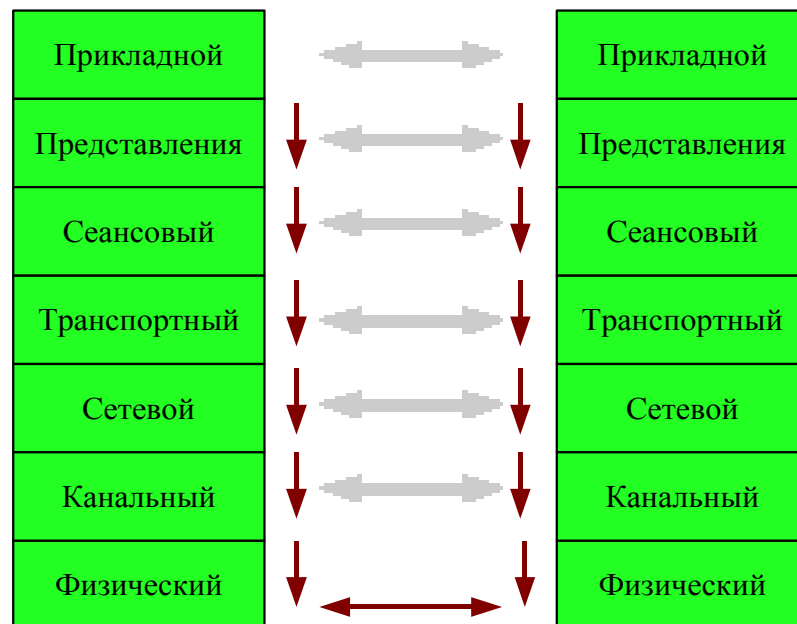
1. Клиент посылает SYN-сегмент с указанием номера порта сервера, который предлагается использовать для организации канала связи (active open).
2. Сервер откликается, посылая свой SYN-сегмент, содержащий идентификатор (ISN - initial sequence number). Начальное значение ISN не равно нулю. Процедура называется *passive open*.
3. Клиент отправляет подтверждение получения syn-сегмента от сервера с идентификатором равным ISN (сервера)+1.

Стандартная процедура установления связи представлена на слайде (под словом “стандартная” подразумевается отсутствие каких-либо отклонений от штатного режима, например, одновременного открывания соединения со стороны сервера и клиента). Если же соединение одновременно инициируется клиентом и сервером, в конечном итоге будет создан только один виртуальный канал.

Префикс S на слайде указывает на сервер, а C — на клиента. Параметры в скобках обозначают относительные значения ISN. После установления соединения $ISN(S) = s_seq_1$, а $ISN(C) = c_seq_1$.

Каждое соединение должно иметь свой неповторимый код ISN. Для реализации режима соединения прикладная программа на одном конце канала устанавливается в режим пассивного доступа ("passive open"), а операционная система на другом конце ставится в режим активного доступа ("active open"). Протокол TCP предполагает реализацию 11 состояний (established, closed, listen, syn_sent, syn_received и т.д.; см. также RFC-793), переход между которыми строго регламентирован.

Модель OSI



ZyXEL

Сетевая модель ВОС (модель взаимосвязи открытых систем — англ. Open Systems Interconnection Reference Model-OSI) — абстрактная модель для сетевых коммуникаций и разработки сетевых протоколов. Представляет уровневый подход к сети. Каждый уровень обслуживает свою часть процесса взаимодействия. Благодаря такой структуре совместная работа сетевого оборудования и программного обеспечения становится гораздо проще и понятнее. Модель состоит из 7 уровней, каждый из которых выполняет определенную функцию.

Прикладной уровень (Application layer)

Верхний (7-й) уровень модели, обеспечивает взаимодействие сети и пользователя. Уровень разрешает приложениям пользователя доступ к сетевым службам, таким как обработчик запросов к базам данных, доступ к файлам, пересылке электронной почты. Также отвечает за передачу служебной информации, предоставляет приложениям информацию об ошибках и формирует запросы к уровню представления.

Уровень представления (Presentation layer)

Этот уровень отвечает за преобразование протоколов и кодирование/декодирование данных. Запросы приложений, полученные с уровня приложений, он преобразует в формат для передачи по сети, а полученные из сети данные преобразует в формат, понятный приложениям. На этом уровне может осуществляться сжатие/распаковка или кодирование/декодирование данных, а также перенаправление запросов другому сетевому ресурсу, если они не могут быть обработаны локально.

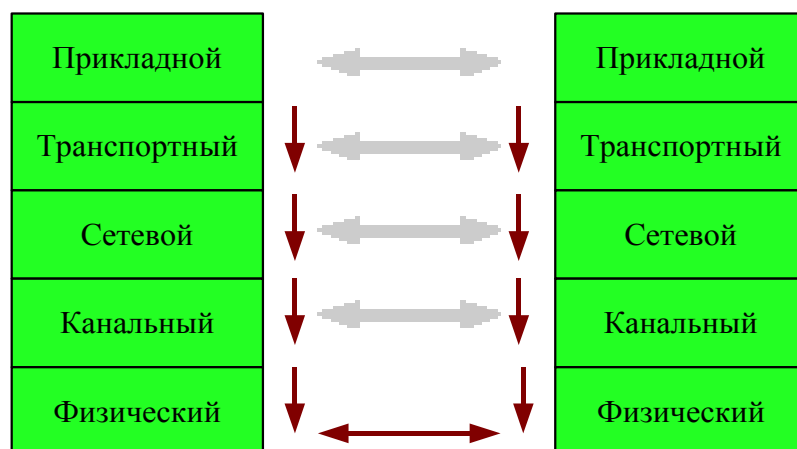
Сеансовый уровень (Session layer)

Отвечает за поддержание сеанса связи, позволяя приложениям взаимодействовать между собой длительное время. Уровень управляет созданием/завершением сеанса, обменом информацией, синхронизацией задач, определением права на передачу данных и поддержанием сеанса в периоды неактивности приложений. Синхронизация передачи обеспечивается помещением в поток данных контрольных точек, начиная с которых возобновляется процесс при нарушении взаимодействия

Транспортный уровень (Transport layer)

4-й уровень модели, предназначен для доставки данных без ошибок, потерь и дублирования в той последовательности, как они были переданы. При этом неважно, какие данные передаются, откуда и куда, то есть он предоставляет сам механизм передачи. Блоки данных он разделяет на фрагменты, размер которых зависит от протокола, короткие объединяет в один, а длинные разбивает. Протоколы этого уровня предназначены для взаимодействия типа точка-точка.

Модель OSI стека TCP/IP



ZyXEL

Сетевой уровень (Network layer)

3-й уровень сетевой модели ВОС, предназначен для определения пути передачи данных. Отвечает за трансляцию логических адресов и имён в физические, определение кратчайших маршрутов, коммутацию и маршрутизацию, отслеживание неполадок и заторов в сети. На этом уровне работает такое сетевое устройство, как маршрутизатор.

Канальный уровень (Data Link layer)

Этот уровень предназначен для обеспечения взаимодействия сетей на физическом уровне и контроля за ошибками, которые могут возникнуть. Полученные с физического уровня данные он упаковывает в кадры данных, проверяет на целостность, если нужно исправляет ошибки и отправляет на сетевой уровень. Канальный уровень может взаимодействовать с одним или несколькими физическими уровнями, контролируя и управляя этим взаимодействием. Спецификация IEEE 802 разделяет этот уровень на 2 подуровня — MAC (Media Access Control) регулирует доступ к разделяемой физической среде, LLC (Logical Link Control) обеспечивает обслуживание сетевого уровня. На этом уровне работают коммутаторы, мосты.

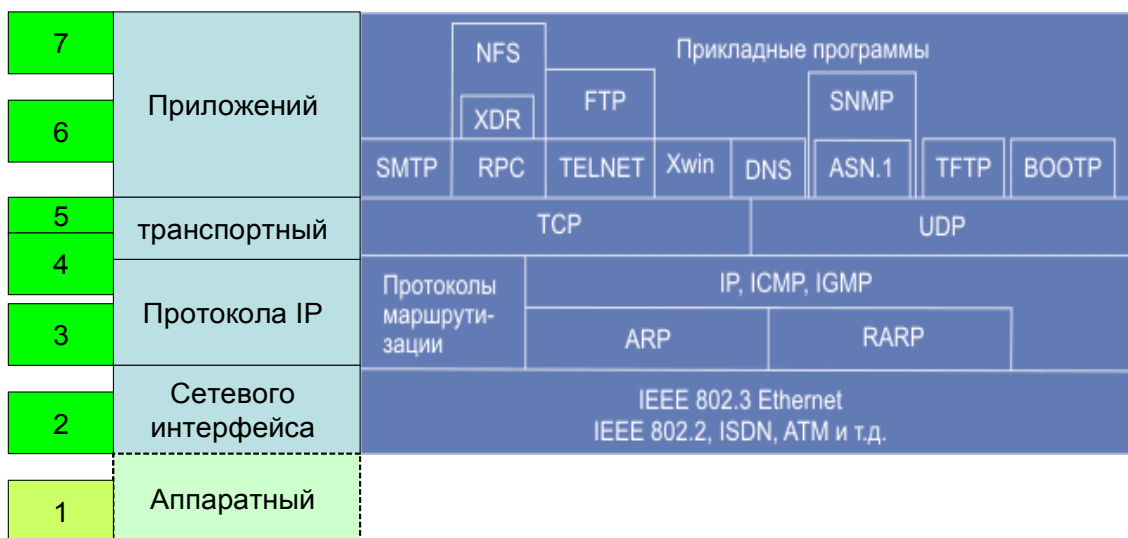
В программировании этот уровень представляет драйвер сетевой платы, в операционных системах имеется программный интерфейс взаимодействия канального и сетевого уровня между собой, это не новый уровень, а просто реализация модели для конкретной ОС. Примеры таких интерфейсов: ODI, NDIS

Физический уровень (Physical layer)

Самый нижний уровень модели, предназначен непосредственно для передачи потока данных. Осуществляет передачу электрических или оптических сигналов в кабель и соответственно их приём и преобразование в биты данных в соответствии с методами кодирования цифровых сигналов. Другими словами, осуществляет интерфейс между сетевым носителем и сетевым устройством. На этом уровне работают концентраторы, повторители (ретрансляторы) сигнала и медиа конверторы. Функции физического уровня реализуются на всех устройствах, подключенных к сети. Со стороны компьютера функции физического уровня выполняются сетевым адаптером или последовательным портом.

В стеке TCP/IP верхние 3 уровня (прикладной, представительный и сеансовый) модели OSI объединяют в один — прикладной. Поскольку в таком стеке не предусматривается унифицированный протокол передачи данных, функции по определению типа данных передаются приложению.

Стек TCP/IP

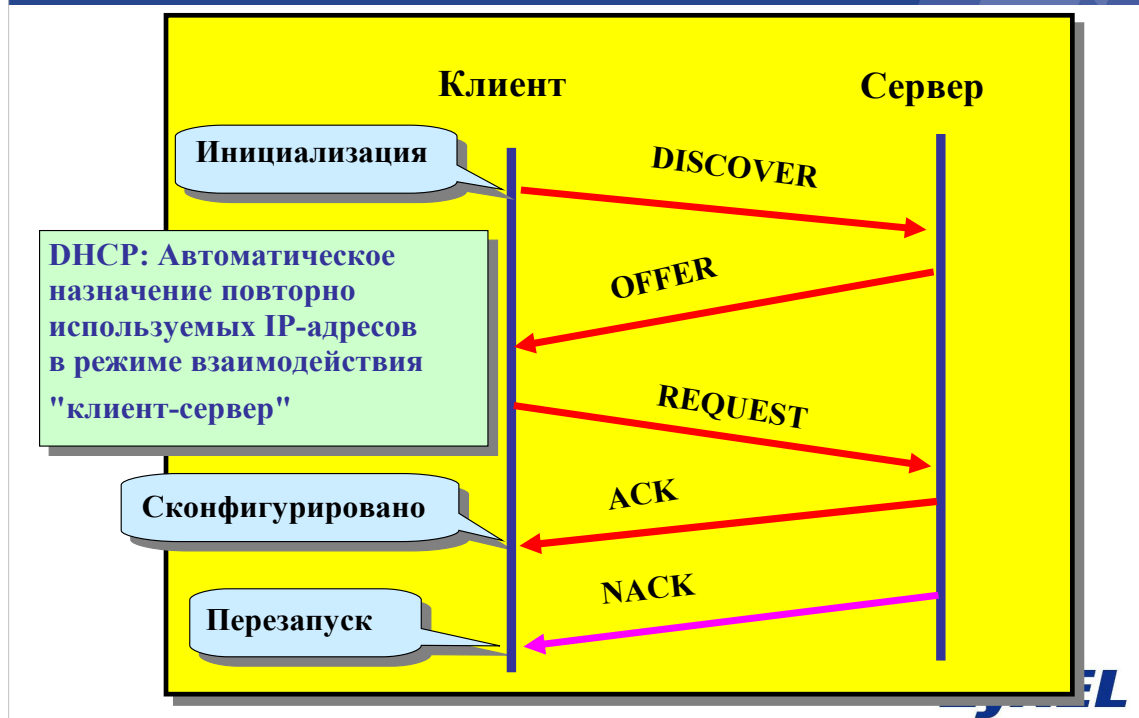


ZyXEL

DHCP

ZyXEL

Функционирование DHCP



DHCP (Dynamic Host Configuration Protocol/Протокол динамического выбора конфигурации хост-машины) построен по модели "клиент-сервер". Функция динамического назначения позволяет автоматически повторно использовать адрес, если он больше не требуется клиенту, которому он был назначен.

Взаимодействие "клиент-сервер"

1. Клиент рассылает в своей физической локальной подсети сообщение "DISCOVER" (ОБНАРУЖЕНИЕ), содержащее предполагаемые значения IP-адреса и продолжительности его аренды. Если в данной подсети DHCP-сервер отсутствует, сообщение может быть передано на серверы DHCP, находящиеся в другой физической подсети, ретранслирующими агентами протокола DHCP/BOOTP.
2. В ответ каждый сервер посылает сообщение "OFFER" (ПРЕДЛОЖЕНИЕ), содержащее доступный IP-адрес.
3. Клиент получает одно или несколько сообщений "OFFER" от серверов и выбирает один сервер для запроса параметров конфигурации, а затем посылает сообщение "REQUEST" (ЗАПРОС). Это сообщение должно включать ИД сервера, показывающий, какой сервер выбран.
4. Сервер получает сообщение "REQUEST" от клиента. Сервер фиксирует привязку к клиенту и посылает ответное сообщение "ACK" (ПОЛОЖИТЕЛЬНАЯ КВИТАНЦИЯ), содержащее параметры конфигурации для клиента.
5. Клиент получает сообщение "ACK", а затем выполняет конфигурирование.
6. Если к моменту поступления сообщения REQUEST предложенный IP-адрес уже был отправлен другому клиенту, сервер отвечает сообщением NACK.

DHCP

- Оборудование компании ZyXEL может выступать в качестве
 - DHCP Server – все оборудование поддерживающее работу на 3-ем уровне
 - DHCP Relay – практически все коммутационное оборудование ZyXEL
 - DHCP Relay Option 82 – DSLAM, Switch

ZyXEL

Коммутационное оборудование ZyXEL поддерживает следующие три опции работы с DHCP

1. Выступать в качестве DHCP Server для локальной сети.

Поддерживается всем оборудованием, работающем на третьем уровне, а именно – CPE xDSL с режимом router, ZyWALL, Routers, Dimension 4 Series и т.д.

2. Выступать в качестве DHCP Relay, т.е. перенаправлять запросы пользователей на предопределенные DHCP сервера, которых может быть два или три в зависимости от типа коммутационного оборудования. Для клиентов DHCP Сервером назначается устройство ZyXEL, которое в свою очередь перенаправляет запрос на DHCP Сервера, получает от них ответ и направляет их клиенту. Поддерживается почти всем коммуникационным оборудованием ZyXEL.

3. DHCP Relay option 82. Данная функция будет рассмотрена позже. Поддерживается DSLAM и управляемыми коммутаторами.

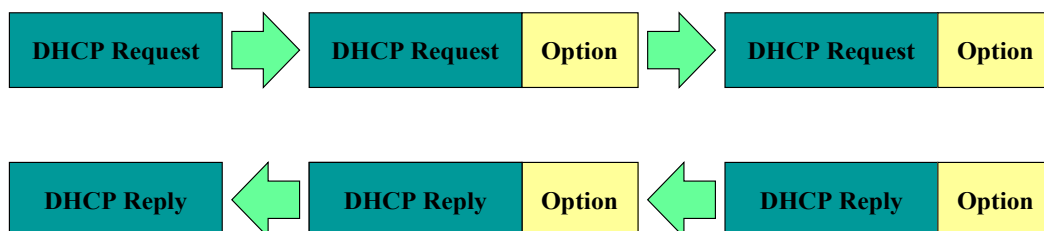
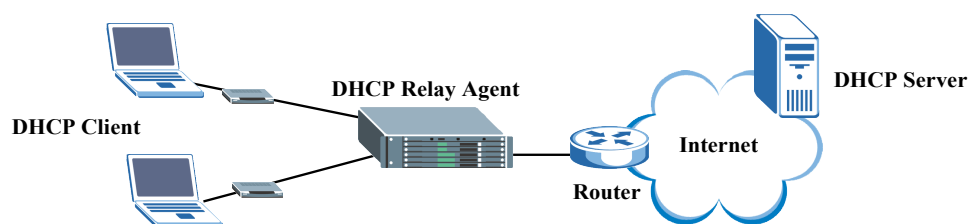
DHCP Relay Agent Option 82

- Relay Agent Information Option (Option 82) – это дополнительная информация, которая работает между DHCP relay agent и DHCP Server
- Данная информация используется DHCP server для предоставления различных сервисов DHCP клиентам
- Не все DHCP server поддерживают это поле.
- Определено в RFC 3046

ZyXEL

- DHCP Relay Agent используется для перенаправления данных между DHCP Клиентом и Сервером.
- Relay Agent Information Option (82) – опциональная информация, которая используется DHCP relay агентом и сервером.
- Данная информация используется DHCP server для идентификации пользователя, от которого поступил текущий запрос. Например, можно назначать IP адреса из различных подсетей различным пользователям в зависимости от точки их подключения в коммутатор.
- Это достаточно новая функция и не все DHCP server поддерживают ее.
- Данная функция определена в RFC3046.

Процесс

**ZyXEL**

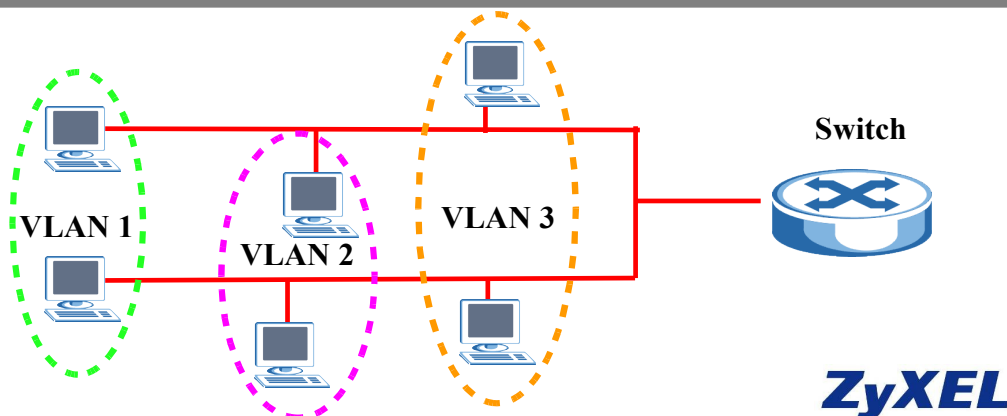
Port Based VLAN

ZyXEL

Что такое VLAN

•Virtual LAN

- Разделение физической ЛВС на несколько логических подсетей
- Изолирование каждого порта для увеличения безопасности
- Изолирование широковещательного трафика



Виртуальная локальная сеть (VLAN) представляет собой логический домен циркулярной рассылки, который может охватывать множество физических локальных сетевых сегментов. За каждым портом коммутатора может быть закреплена конкретная VLAN, которая может быть логически сегментирована в соответствии с ее функциями и задачами.

Порты одной VLAN имеют общий домен циркулярной рассылки. Порты, относящиеся к различным VLAN, не могут осуществлять циркулярную рассылку.

Безопасность рабочей группы и сети

Можно повысить уровень безопасности путем сегментирования сети на отдельные домены циркулярной рассылки. Кроме того, можно регулировать размер и структуру домена путем регулирования размера и структуры VLAN.

Контроль активности трафика

VLAN позволяют группировать порты коммутатора таким образом, чтобы трафик ограничивался только членами той или иной группы. Данная функция ограничивает циркулярную, одноадресную и многоадресную рассылку (лавинная адресация) только портами, включенными в конкретную VLAN. VLAN делают возможным эффективное разделение трафика, тем самым, обеспечивая более высокую пропускную способность.

Типы VLAN

- **Типы VLAN**

- VLAN на базе порта, нет стандарта
 - На базе только одного коммутатора
- VLAN на базе MAC, нет стандарта
 - Не популярная реализация
- VLAN на базе признака (tag-based), IEEE 802.1q
 - Может быть между несколькими коммутаторами.
Ethernet MTU 1522 байт (Обычный MTU=1518 байт)

ZyXEL

Существует три типа виртуальной локальной сети: **VLAN на базе порта** позволяет создавать VLAN из различных портов одного моста. **VLAN на базе MAC** позволяет объединять в сегмент MAC-адреса хост-машин, а **VLAN на базе признака** позволяет создавать VLAN по какому-либо признаку. Признак записывается после MAC адреса источника в кадре Ethernet, что позволяет идентифицировать VLAN.

Виртуальная локальная сеть на базе порта

VLAN на базе порта требует для каждого порта определить его выходной порт

- Выходной порт разрешен для входного
- VLAN на базе порта управляет только исходящим трафиком
- Чтобы сделать VLAN для 2-х портов:
 - > Определить соответствующий выходной порт для каждого порта

ЗУМ

VLAN на базе порта требует, чтобы для каждого входного порта были определены доступные выходные порты. Кадры Ethernet пересылаются в соответствии с этими правилами.

Пример настройки VLAN на базе порта

Port Based VLAN Setup

Setting Wizard: All connected Apply

Port isolation
All connected

Incoming

	1	2	3	4	5	6	7	8	9	10	11	12	
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4
5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7
8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	8
9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10
11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	11
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12
CPU	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CPU

Outgoing

	1	2	3	4	5	6	7	8	9	10	11	12	
1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	1
2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	2
3	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3
4	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	4
5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	5
6	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	6
7	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	7
8	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	8
9	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	9
10	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	10
11	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	11
12	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	12
CPU	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	CPU

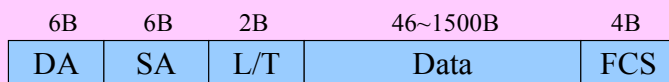
ZyXEL

VLAN 802.1Q

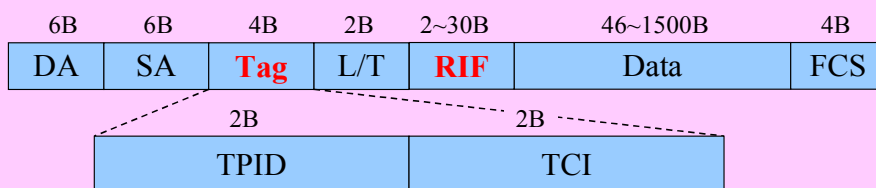
ZyXEL

Tag-based VLAN (802.1Q VLAN)

802.3/Ethernet Frame



802.3/Ethernet Tagged Frame



RIF : Routing Information Field (optional)

ZyXEL

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

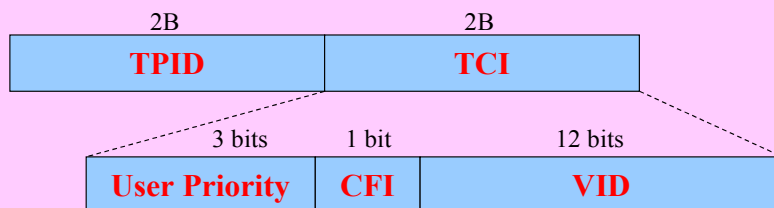
Стандарт IEEE 802.1p специфицирует метод указания приоритета кадра, основанный на использовании новых полей, определенных в стандарте IEEE 802.1Q.

К кадру Ethernet добавляются 4 байта. TPID (Tag Protocol Identifier) 2 байта, которые содержат информацию о принадлежности кадра Ethernet к VLAN и TCI (Tag Control Information), рассмотренный далее. Добавление четырех байтов к максимальному размеру кадра Ethernet ведет к возникновению проблем в работе многих коммутаторов, обрабатывающих кадры Ethernet аппаратно. Чтобы избежать их, группы по стандартизации предложили сократить на два байта максимальный размер полезной нагрузки в кадре.

Спецификация IEEE 802.1p, создаваемая в рамках процесса стандартизации 802.1Q, определяет метод передачи информации о приоритете сетевого трафика. Стандарт 802.1p специфицирует алгоритм изменения порядка расположения пакетов в очередях, с помощью которого обеспечивается своевременная доставка чувствительного к временным задержкам трафика.

Структура тэга

Tag Header :



- **TPID** : Признак идентификатора протокола, 802.1Q TPID = 81-00
- **Priority** : 8 уровней приоритета используются в 802.1P для QoS
- **CFI** : Canonical Format Indicator, равен 0 для Ethernet
- **VID** : VLAN ID, 4096 VLAN используются в 802.1Q для идентификации VLAN

ZyXEL

TPID : TPID идентифицирует принадлежность данного кадра к стандарту 802.1Q и имеет значение 8100 в 16с/с. Если кадр имеет идентификатор EtherType равный 8100, то этот кадр несет нагрузку IEEE 802.1Q / 802.1P.

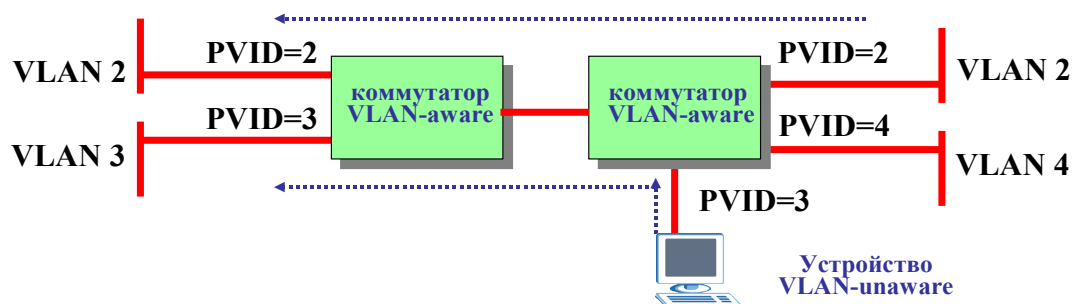
TCI содержит три поля.

Priority : Первые три бита TCI используются под приоритет. Возможно восемь значений (2^3) приоритета. IEEE 802.1P работает именно с этими 3 битами приоритета.

CFI : Canonical Format Indicator однобитовый флаг, который всегда равен 0 для кадров Ethernet. CFI для идентификации, если в поле данные находятся данные других стандартов, не Ethernet, например Token Ring. В этом случае этот бит будет равен 1. Если кадр был получен с Ethernet порта и CFI равен 1, то этот кадр должен быть перенаправлен на untagged порт.

VID : VLAN ID идентификатор VLAN, который и используется в стандарте 802.1Q. Это поле состоит из 12 бит и позволяет закодировать 4096 (2^{12}) VLANов. Из 4096 возможных значений, VID равное 0 и 4095 (FFF) зарезервированы, поэтому максимальное количество VLAN, которые работают в сети равно 4,094. VID = 0 определяет, что данный кадр не несет информации о VLAN, а несет только информацию о приоритете. VID = 4095 в оборудовании ZyXEL используется для внутренней коммутации (например, в DSLAM)

802.1Q VLAN



- 802.1Q Tag VLAN

- Каждый VLAN имеет свой уникальный VID
- Каждый член VLAN может взаимодействовать с другим членом

- VLAN-aware

- Устройство, которое может распознавать и поддерживать VLAN на базе признака.

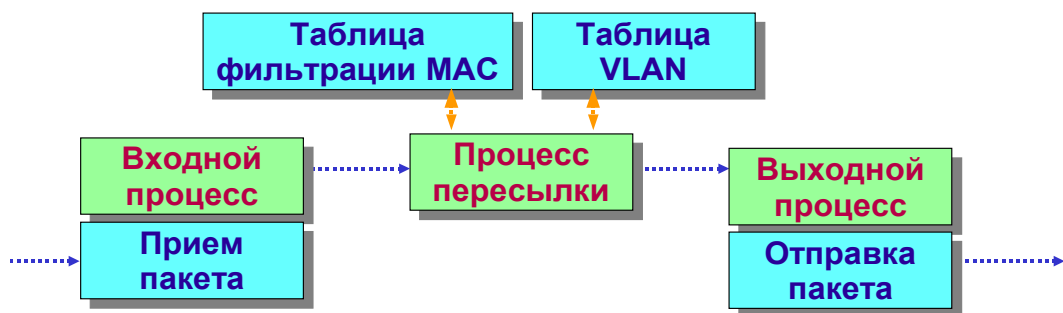
- VLAN-unaware

- Устройство, которое не может распознать VLAN на базе признака

ZyXEL

Порты с одинаковыми VID могут взаимодействовать друг с другом. Коммутатору необходимо знать, какие устройства поддерживают VLAN, а какие нет. Это позволяет коммутатору убрать дополнительную информацию о VLAN при передаче кадра не поддерживающему данный стандарт устройству.

Процессы 802.1Q



- **Входной процесс:**
 - Ставит маркер. Если кадр с маркером, то он без изменений направляется в процесс пересылки, если без маркера, то на него ставится маркер согласно входному правилу.
- **Процесс пересылки (перенаправления)**
 - Принимает решения о фильтрации или пересылке пакета в порт назначения согласно таблицам VLAN и MAC.
- **Выходной процесс:**
 - Определяет, оставлять ли признак VLAN в кадре. Если известно, что к порту подключено устройство VLAN-Unaware, то тег снимается.

ZyXEL

Входное правило



- Коммутатор с поддержкой VLAN может принимать кадры как с признаком, так и без
- Для кадров с признаком
 - Передача без изменений
- Для кадров без признака
 - Добавление PVID в кадр
 - Передача кадра с признаком
- PVID
 - Port VLAN ID по умолчанию для входящих кадров без признаков

ZyXEL

На этапе применения Входного правила определяется наличие признака в кадре и классификация кадра по VLAN. На каждом порту определено свое Входное правило. Если Входное правило определяет прием только кадров с признаками, то другие кадры будут отбрасываться. Если в правиле определено принимать все кадры, то и кадры с признаком, и без признака будут приниматься и обрабатываться:

- При приеме кадра с признаком он сразу направляется на процесс пересылки
- При приеме кадра без признака, **происходит добавление признака со значением PVID** в кадр. Каждый физический порт имеет VID по умолчанию PVID (Port VID). PVID добавляется в кадры без признака и приоритетные кадры с признаком (кадры с VID=0), которые поступают на порт.

После выполнения входного правила, все кадры имеют 4 дополнительных байта, в которых записана информация VLAN, и отправляются на процесс пересылки.

Процесс пересылки

Таблица Mac-Vlan

Mac Address	Vlan	Port	Tag	Aging
00:A0:C5:11:11:11	1	1	tag	0
00:A0:C5:22:22:22	2	2	untag	20
00:A0:C5:33:33:33	3	3	tag	50
00:A0:C5:44:44:44	3	4	untag	infinite

- Решение о пересылке принимается на основе таблиц
 - MAC адрес назначения принятого кадра
 - VID принятого кадра
- База фильтрации содержит
 - Таблицу Mac-Vlan: MAC адрес, Vlan, Порт, Время жизни записи
 - Таблицу VLAN: SVLAN + DVLAN

ZyXEL

На этапе процесса пересылки принимается решение о пересылке полученного кадра на основе Базы фильтрации. Решение о выборе выходного порта определяется на базе таблицы **Mac Vlan**

Если необходимо отправлять кадры с признаком на определенный порт, то этот порт должен быть выходным портом для этого VID.

База фильтрации хранит информацию по VLAN, используемую для коммутации кадров. Она содержит статическую таблицу (**Static VLAN** или Таблица SVLAN) и динамическую таблицу (**Dynamic VLAN** или таблицу DVLAN). Таблица SVLAN ведется вручную администратором. Таблица DVLAN динамически пополняется при помощи протокола GVRP, и не может изменяться администратором.

Таблица SVLAN

• Статическая информация по VLAN

- Fixed : Порт всегда член данного VLAN
- Forbidden : Порту запрещено регистрироваться как члену данного VLAN
- Normal : Как покажет регистрация по протоколу GVRP VLAN

• Статические данные

VID3 : Port 3(Fixed)
VID4 : Port 2(Forbidden), Port 3(Fixed)
VID5 : Port 1(Normal)

• Таблица SVLAN

VID	Port	Ad Control	Tag
3	3	Fixed	Tag
4	2	Fixed	UnTag
5	1	Normal	Tag
6	1	Normal	UnTag

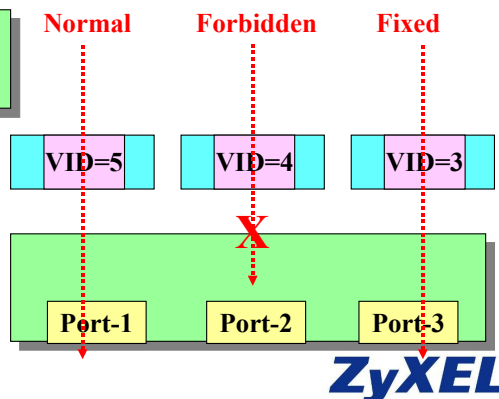


Таблица SVLAN базы фильтрации содержит следующую информацию:

b. **VID** : VLAN ID

c. **Port** : Порт коммутатора

d. **Ad Control** : Управление, которое бывает трех типов: **forbidden**, **fixed** и **normal**.

- **Forbidden**: Данный порт не может быть членом данного VLAN.
- **Fixed**: Данный порт постоянно является участником этого VLAN.
- **Normal**: Данный тип означает, что вхождение порта в определенные VLAN определяются таблицей DVLAN на базе протокола GVRP.

e. **Egress Tag Control (Контроль признака на выходе)**: Эта информация используется для выходного правила. Если установлено tag, то выходной кадр с признаком, если установлено UnTag, то выходной кадр без признака. Если Ad Control = Forbidden, то Egress Tag Control = none. **Egress**

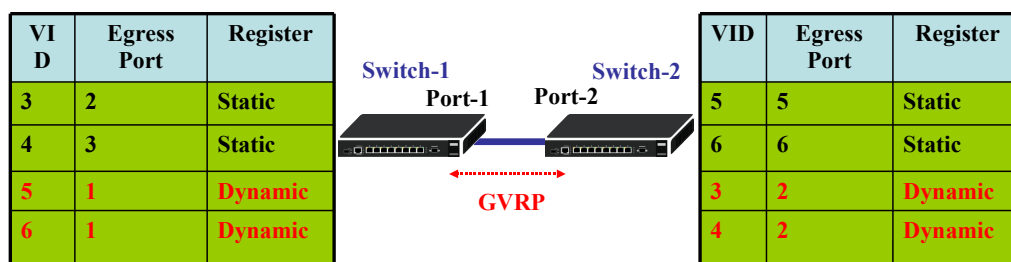
Регистрация VLAN

• Два варианта регистрации VLAN

- Статически: Выходной порт статически добавляется администратором
- Динамически: Выходной порт автоматически добавляется из данных протокола GVRP

• Функции GVRP

- Динамическое создание VLAN между соседними VLAN-aware устройствами
- VLAN автоматически регистрируются и удаляются

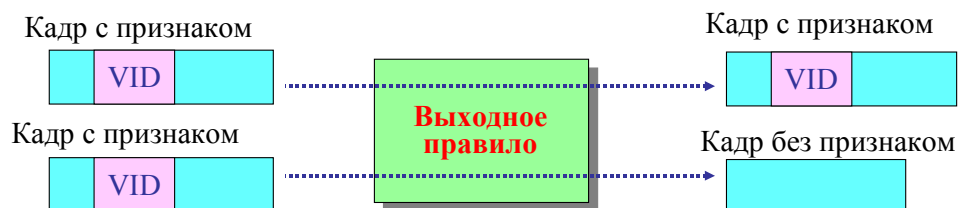


Static VLAN добавляются вручную администратором и, как правило, работают на одном коммутаторе (сети). Dynamic VLAN автоматически добавляются из данных протокола GVRP, и не могут быть изменены и добавлены администратором.

GVRP значит **GARP VLAN Registration Protocol**, GVRP – протокол регистрации портов членов VLAN в сети.

Например, имеется два 802.1Q VLAN коммутатора на базе признаков – Switch-1 и Switch-2. Каждый коммутатор содержит SVLAN таблицу, заполненную администратором и пустую DVLAN таблицу. После включения GVRP коммутаторы обмениваются информацией и VLAN таблица пополняется соответствующими записями.

Выходное правило



- Соответствует последнему полю Базы Фильтрации
- Для VLAN-aware устройств на приеме
 - Отсылаемые кадры должны быть с признаком
- Для VLAN-unaware устройств на приеме
 - Отсылаемые кадры должны быть без признака

ZyXEL

Выходное правило решает должна ли быть информация о признаке VLAN в отправляемом кадре или нет. Решение принимается на основе информации поля **Egress Tag Control (Контроль признака на выходе)** из базы фильтрации.

802.1p Priority

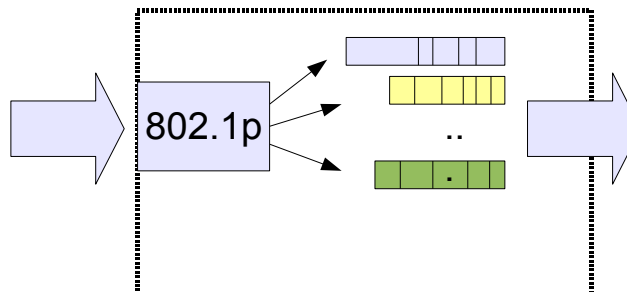
ZyXEL

802.1p

получение
пакетов



выходной
порт



отправка
пакетов

- 8 очередей на каждом порту
- отображение **802.1p** на номер очереди
- каждая очередь имеет вес
- алгоритмы обработки очередей:
SPQ, WRR, WFQ

ZyXEL

IEEE 802.1P определяет до 8 различных типов трафика путем добавления признака в кадр MAC уровня. В соответствии с **Priority bit** в тегированном кадре IEEE 802.1Q, коммутатор классифицирует кадр в различные очереди.

Рабочий процесс IEEE 802.1P аналогичен процессу IEEE 802.1Q и состоит из трех этапов **Ingress Process (Входной процесс)**, **Forwarding Process (Процесс пересылки)** и **Egress Process (Выходной процесс)**.

При поступлении кадра в коммутатор с включенной поддержкой 802.1P, вначале Ingress Process классифицирует полученный кадр. Затем кадр передается Forwarding Process. После Forwarding Process он поступает на Egress Process и затем отправляет получателю.

Ingress Process :

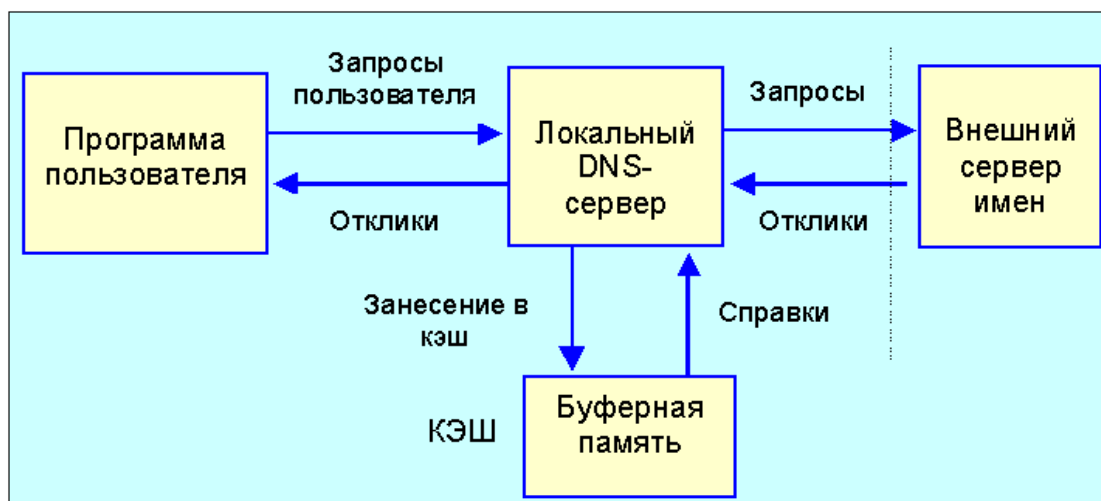
- Кадры без признака не несут никакой информации о VID и Priority. При поступлении такого кадра, **Ingress Process добавляет поле признака в Ethernet кадр с значениями PVID и Default Priority.**
- При получении кадра с признаком он уже содержит информацию о VID и Priority. Ingress Process пропускает такой кадр без изменения на Forwarding Process.

После Ingress Process все кадры имеют 4 байтовое поле признака и перенаправляются на Forwarding Process.

DNS и DynDNS

ZyXEL

Схема функционирования DNS



ZyXEL

DNS (Domain Name System) - это распределенная база данных, поддерживающая иерархическую систему имен для идентификации узлов в сети Internet. Служба DNS предназначена для автоматического поиска IP-адреса по известному символьному имени узла. Спецификация DNS определяется многими стандартами, а именно

RFC 1034 - Domain Names - Concepts and Facilities Изложение основных концепций и функциональных особенностей DNS.

RFC 1035 - Domain Names - Implementation and Specification Описывает механизм реализации DNS и соответствующие спецификации протоколов.

RFC 1996 - A Mechanism for Prompt Notification of Zone Changes (DNS NOTIFY) Описание использования кода операции NOTIFY для DNS с целью контроля оповещения подчиненных серверов о том, что данные на основном сервере были изменены.

RFC 2065 - Domain Name Security Extension Спецификация расширений для DNS, поддерживающих использование криптографических цифровых подписей наряду с соответствующими клиентами DNS.

RFC 2136 - Dynamic Updates in the Domain Name System (DNS UPDATE) Изложение стандартов функционирования DDNS (рассмотрен далее)

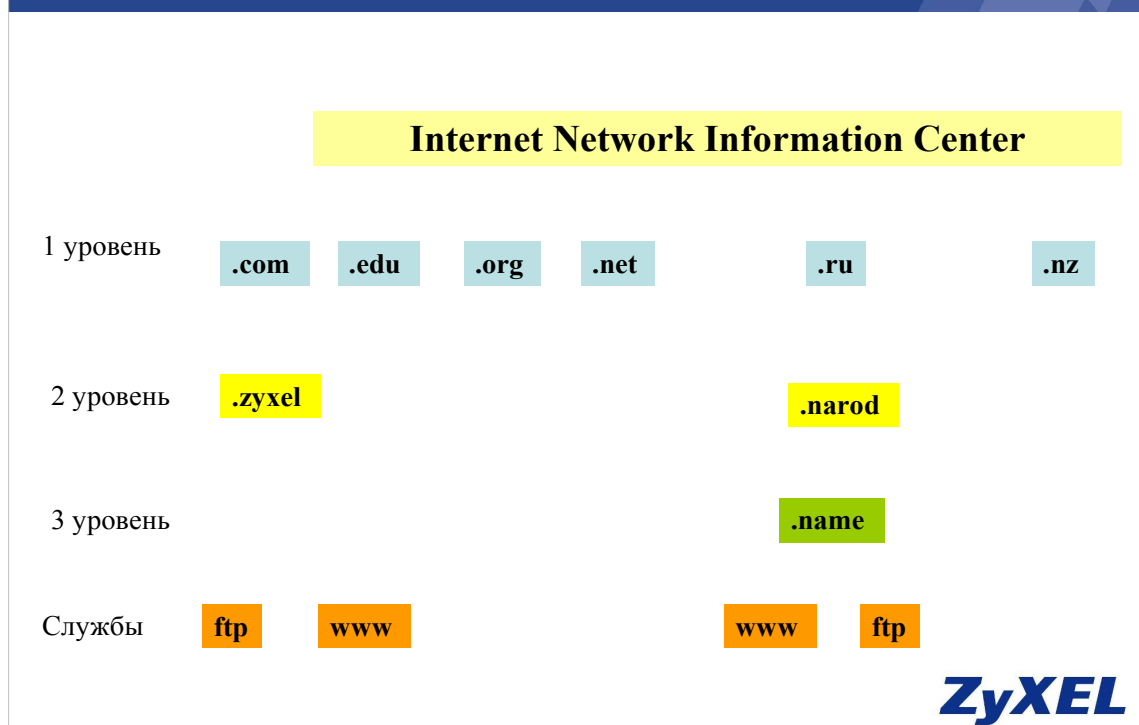
RFC 2137 - Secure Domain Name System Dynamic Update Опирающееся на стандарт 2136 описание практики реализации защиты и использования цифровых подписей DNSSEC с целью защиты и ограничения обновлений.

RFC 2168 - Resolution of Uniform Resource Identifiers Using the Domain Name System Определение экспериментального протокола для использования новой записи о ресурсах DNS - NAPTR (Naming Authority Pointer) - с целью реализации отображения частей URI на имена доменов.

Протокол DNS является служебным протоколом прикладного уровня. Этот протокол несимметричен - в нем определены DNS-серверы и DNS-клиенты. DNS-серверы хранят часть распределенной базы данных о соответствии символьных имен и IP-адресов. Эта база данных распределена по административным доменам сети Internet. Клиенты сервера DNS знают IP-адрес сервера DNS своего административного домена и по протоколу IP передают запрос, в котором сообщают известное символьное имя и просят вернуть соответствующий ему IP-адрес.

Если данные о запрошенном соответствии хранятся в базе данного DNS-сервера, то он сразу посылает ответ клиенту, если же нет - то он посылает запрос DNS-серверу другого домена, который может сам обработать запрос, либо передать его другому DNS-серверу. Все DNS-серверы соединены иерархически, в соответствии с иерархией доменов сети Internet. Клиент опрашивает эти серверы имен, пока не найдет нужные отображения. Этот процесс ускоряется из-за того, что серверы имен постоянно кэшируют информацию, предоставляемую по запросам. Клиентские компьютеры могут использовать в своей работе IP-адреса нескольких DNS-серверов, для повышения надежности своей работы.

Структура Имен



База данных DNS имеет структуру дерева, называемого доменным пространством имен, в котором каждый домен (узел дерева) имеет имя и может содержать поддомены. Имя домена идентифицирует его положение в этой базе данных по отношению к родительскому домену, причем точки в имени отделяют части, соответствующие узлам домена.

Корень базы данных DNS управляется центром Internet Network Information Center. Домены верхнего уровня (1-ого уровня) назначаются для каждой страны, а также на организационной основе. Имена этих доменов должны следовать международному стандарту ISO 3166. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры.

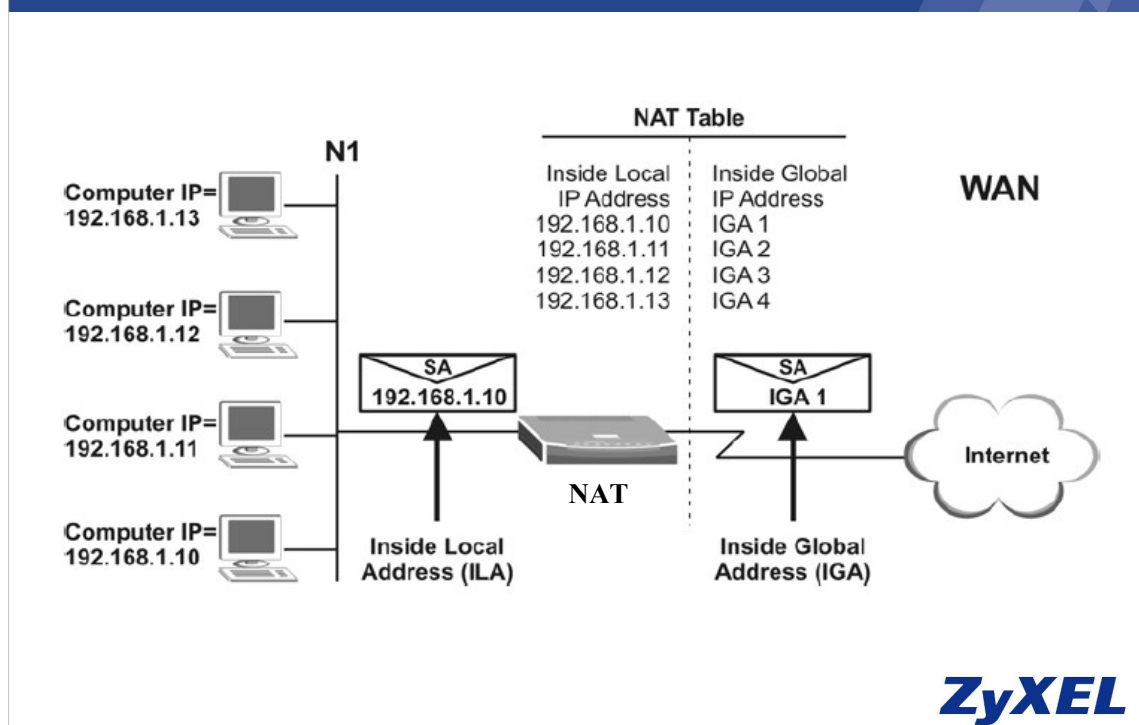
Корневые серверы хранят информацию об именах и адресах всех серверов доменов второго уровня. Существует два вида запросов: рекурсивные и итеративные. Первый вид предполагает получение клиентом IP-адреса, а второй - адреса сервера, который может сообщить адрес. Первый вид медленнее, но дает сразу IP-адрес, второй эффективнее - в вашем сервере копится информация об адресах серверов имен. com - коммерческие организации (например - zyxel.com);

Каждый домен DNS администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям. Каждый домен имеет уникальное имя, а каждый из поддоменов имеет уникальное имя внутри своего домена. Имя домена может содержать до 63 символов. Каждый хост в сети Internet однозначно определяется своим *полным доменным именем (fully qualified domain name, FQDN)*, которое включает имена всех доменов по направлению от хоста к корню.

NAT и SUA

ZyXEL

Функционирование NAT



NAT – Network Address Translation.

Каждый пакет имеет два адреса – адрес источника и адрес назначения. Для исходящих пакетов, внутренний локальный адрес (ILA) является адресом источника в локальной сети, а внутренний глобальный адрес (IGA) - адресом источника в глобальной сети. Для входящих пакетов, ILA – это адрес назначения в локальной сети, а IGA - адрес назначения в глобальной сети. NAT преобразовывает частные (локальные) IP-адреса в уникальные глобальные, необходимые для связи с хостами других сетей. NAT заменяет исходный IP-адрес источника (и номера портов источника TCP или UDP для отображения много-к-одному и много-ко-многим с перегрузкой) в каждом пакете и пересылает его в Интернет. NAT сохраняет исходные адреса и номера портов, чтобы можно было восстановить исходные значения во входных ответных пакетах.

Применение NAT

- Большинство сетей используют NAT для трансляции внутренних IP адресов во внешние
- В целях повышения безопасности применяют NAT в интеграции с Firewall и IDS (Intrusion detection system) или IDP (Intrusion Detection and Prevention)

ZyXEL

На сегодняшний день в большинстве локальных сетей, имеющих доступ в сеть Internet, используются функции NAT. Это связано с тем, что количество IP адресов, используемых в сети Internet несоизмеримо мало с количеством хостов, которые работают в сети Internet. Именно для решения этой проблемы и был разработан NAT. В сети достаточно иметь несколько IP адресов, которые будут иметь сервера NAT для обеспечения локальным пользователям доступ во внешнюю сеть. Каждый сервер NAT теоретически может поддерживать до 65536 соединений с внешней сетью. Исходя из этого параметра, как максимально возможного, и из реального количества одновременно поддерживаемых сеансов каждого NAT сервера выбирается и создается соответствующая структура локальной сети.

Со временем оказалось, что иметь несколько централизованных точек доступа во внешнюю сеть с внутренних локальных машин очень удобно с точки зрения безопасности. Внутренние машины используют внутреннюю адресацию, которая не видна из сети Internet. Таким образом доступ извне возможен только через IP адреса NAT серверов. Администрировать их с точки зрения безопасности гораздо удобнее и надежнее, нежели множество пользователей, которые имеют общедоступные IP адреса.

Сейчас достаточно популярны реализации т.н. «гибридных» шлюзов безопасности, которые включают в себя и NAT, и DNS, и IDS (или более новую реализацию IDP), firewall, защиту от DoS атак, и другие необходимые сервисы. Они могут быть как интегрированы на одном сервере, так и разделены физически на разных хостах.

Адреса локальных сетей

- **10.0.0.0 - 10.255.255.255**
- **172.16.0.0 - 172.31.0.0**
- **192.168.0.0 - 192.168.255.255**



Независимо от технических возможностей, адреса внутренней сети не следует выбирать случайным образом. Специально для этих целей существуют зарезервированные адреса. Эти адреса не присвоены и никогда не будут присвоены какому-либо хосту, непосредственно соединенному с Интернет.

Зарезервированными являются следующие адреса:

От **10.0.0.0** до **10.255.255.255**, маска 255.0.0.0 (класс А)

От **172.16.0.0** до **172.31.0.0**, маска 255.255.0.0 (класс В)

От **192.168.0.0** до **192.168.255.255**, маска 255.255.255.0 (класс С)

Типы трансляции адресов

- Один к одному
- Много к одному
- Много ко многим с перегрузкой
- Много ко многим без перегрузки
- Сервер



NAT поддерживает пять типов отображения IP/портов. Используются следующие типы:

1. **Один-к-одному:** В режиме один-к-одному, NAT ZyXEL преобразовывает один локальный IP-адрес в один глобальный IP-адрес.
2. **Много-к-одному:** В режиме много-к-одному, NAT ZyXEL преобразовывает несколько локальных IP-адресов в один глобальный IP-адрес. Это эквивалент SUA (например, PAT – преобразование адресов портов), функциональной возможности получения счета одиночного пользователя ZyXEL, которая поддерживалась в предыдущих моделях маршрутизаторов корпорации ZyXEL (опция **SUA Only (Только SUA)** в современных маршрутизаторах).
3. **Много-ко-многим с перегрузкой:** В режиме много-ко-многим с перегрузкой, NAT ZyXEL преобразовывает несколько локальных IP-адресов в общие глобальные IP-адреса.
4. **Много-ко-многим без перегрузки:** В режиме много-ко-многим без перегрузки, NAT ZyXEL преобразовывает каждый локальный IP-адрес в уникальный глобальный IP-адрес.
5. **Сервер** Этот тип позволяет указать внутренние серверы различных служб, закрытые NAT, к которым предоставляется доступ для внешних пользователей, хотя настоятельно рекомендуется вместо этого для данных серверов использовать порт DMZ оборудования ZyXEL. Данная функция работает следующим образом: вы определяете определенные сервера внутри вашей сети и настраиваете перенаправление всего трафика, который приходит из внешней сети на определенный адрес по определенному порту на внутренний сервер. При этом номер порта не изменяется.

Стоит отметить, что Один ко одному и много ко многим без перегрузки не изменяют номера портов при трансляции.

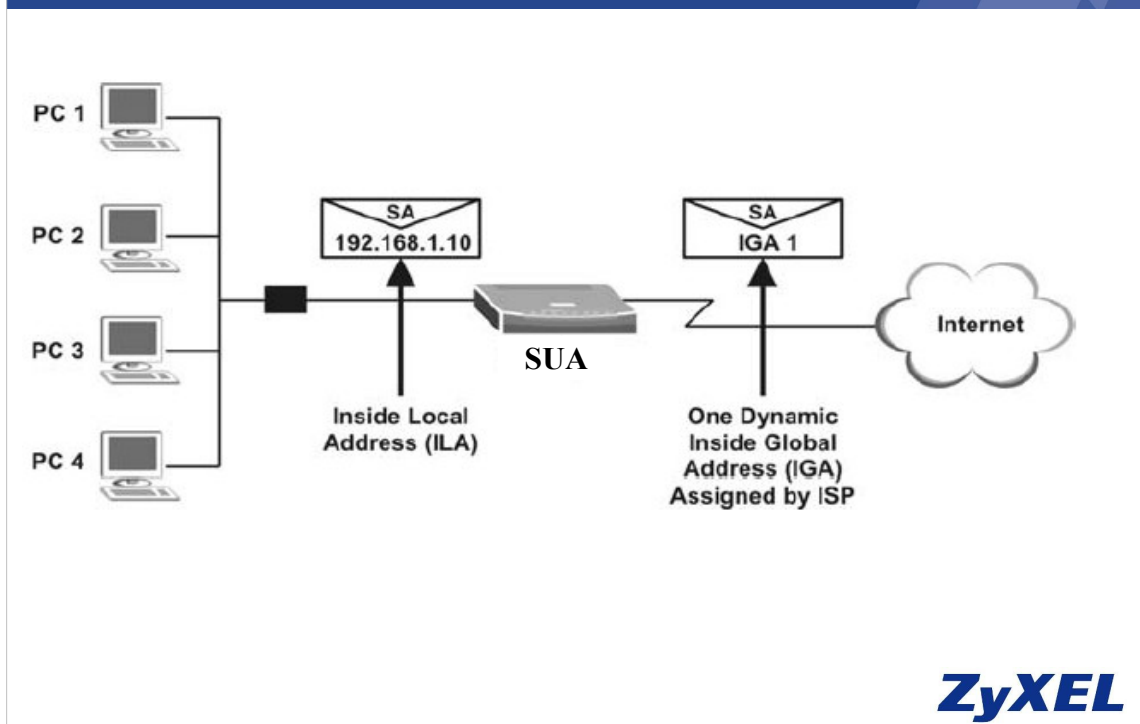
SUA – Single User Account

- Используется когда есть только один внешний IP адрес
- Аналогичен режиму многие к одному NAT (режим все к одному)



При включении функции SUA устройство ZyXEL переходит в режим NAT сервера, при этом трансляция осуществляется в режиме все к одному – все внутренние IP адреса транслируются в один и только один внешний IP адрес. Возможна настройка общедоступных внутренних серверов по номеру порта внешнего IP адреса аналогично режиму Сервер NAT.

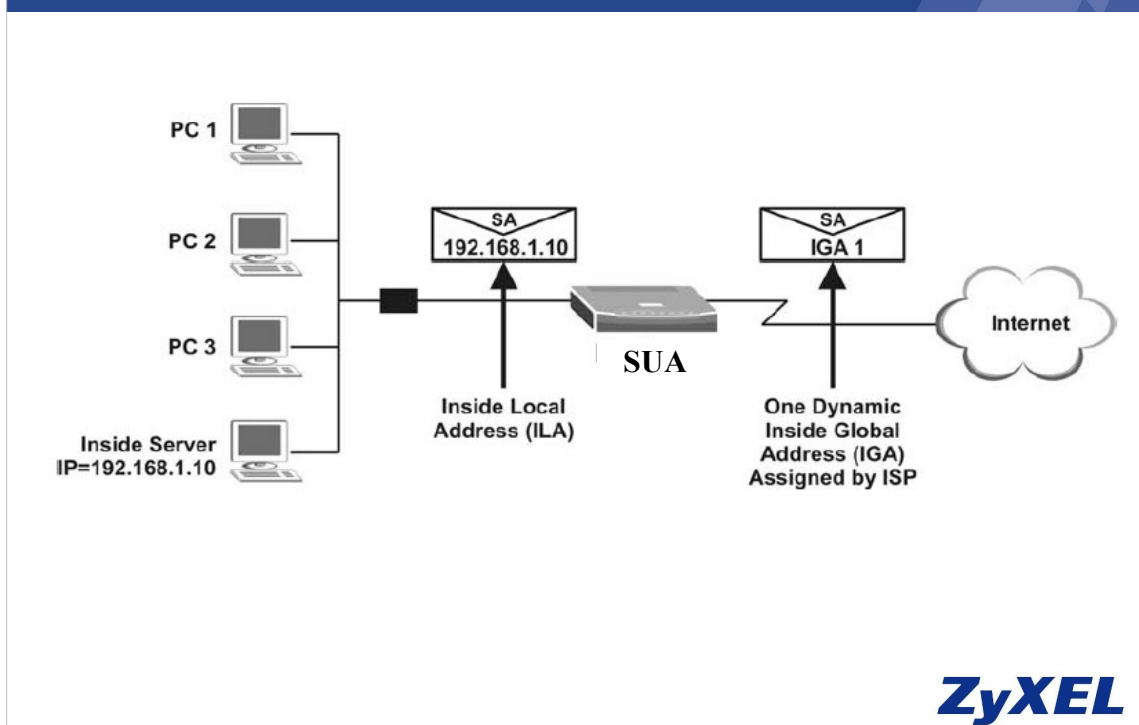
Пример 1. Только доступ в Internet



В этом примере доступа в Интернет вам необходимо только одно правило, в котором все внутренние локальные адреса (ILA) преобразовываются в один динамический внутренний глобальный адрес (IGA), назначенный Интернет-провайдером.

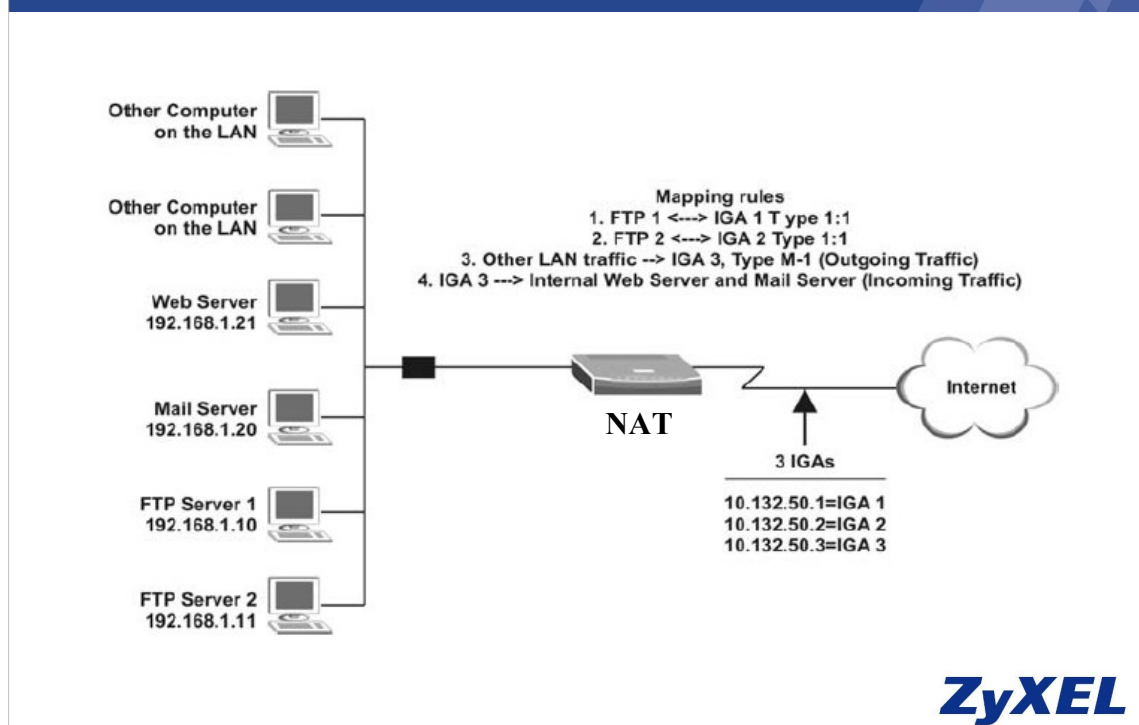
Удобнее всего использовать режим SUA.

Пример 2. Доступ в Internet с внутренним сервером.



Настройки текущего примера аналогичны предыдущему примеру, за исключением того, что вы отображаете все внешние запросы на внешний IP адрес на определенный сервер локальной сети. При этом номер порта при отображении не меняется. Все дальнейшие действия, такие как установка соединения и т.п., отслеживаются SUA сервером и корректно отображаются во внутренней таблице.

Пример 3: Несколько IP-адресов с внутренними серверами



В данном примере существует три 3 IGA, предоставленных Интернет-провайдером. Существует несколько отделов, но два из них имеют собственные серверы FTP. Все отделы подключены к одному маршрутизатору. В примере резервируется по одному IGA для каждого отдела с сервером FTP и все отделы используют оставшийся IGA. Сопоставьте серверы FTP первым двум IGA, а остальной трафик ЛВС перенаправьте на оставшийся IGA. Назначьте третий IGA для внутреннего web-сервера и почтового сервера. Необходимо следующим образом сконфигурировать четыре правила, два двунаправленных и два однонаправленных.

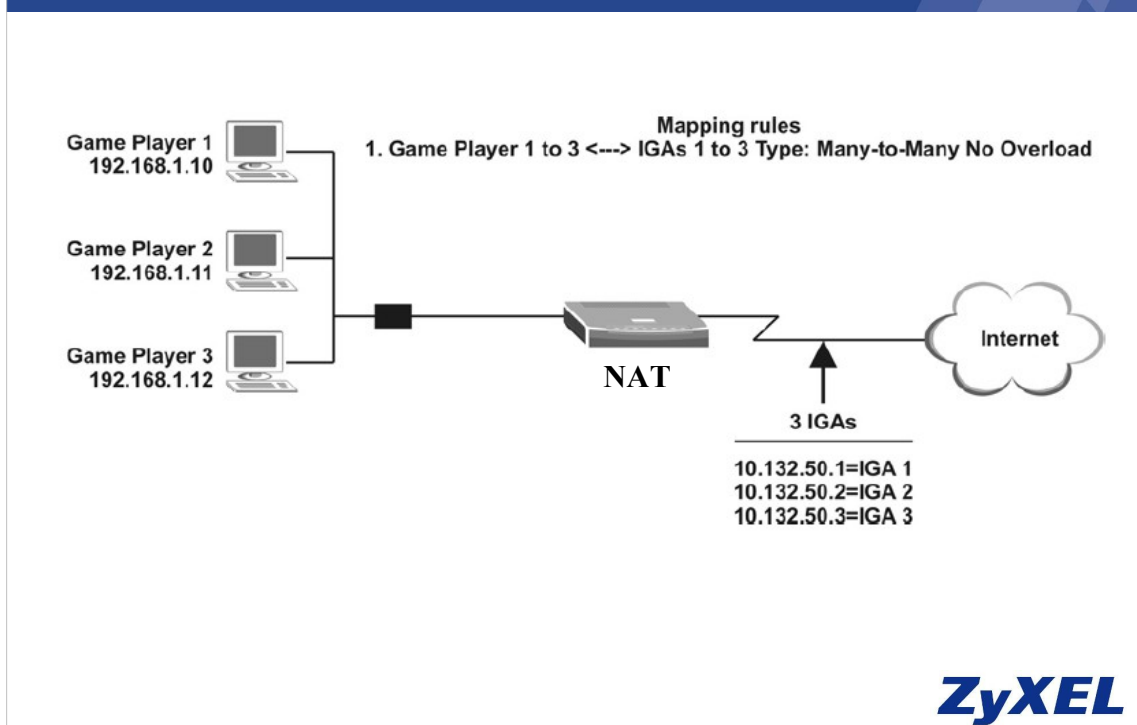
Rule 1. Сопоставьте первый IGA первому внутреннему серверу FTP для FTP-трафика в обоих направлениях (отображение **1: 1**, выдаются и локальный и глобальный IP-адрес).

Rule 2. Сопоставьте второй IGA второму внутреннему серверу FTP для FTP-трафика в обоих направлениях (отображение **1: 1**, выдаются и локальный и глобальный IP-адрес).

Rule 3. Сопоставьте другой выходной трафик ЛВС IGA3 (отображение **Много: 1**).

Rule 4. Также сопоставьте третий IGA web-серверу и почтовому серверу локальной сети. Тип **Server (Сервер)** позволяет предоставить множество серверов различных типов другим компьютерам, закрытым NAT в локальной сети.

Пример 4: Прикладные программы, не поддерживающие NAT



Некоторые приложения не поддерживают отображения NAT при помощи трансляции адресов портов TCP или UDP. В этом случае лучше использовать отображение **много-ко-многим без перегрузки**, поскольку номера портов *не* изменяются при использовании отображения NAT типа **много-ко-многим без перегрузки** (и **один-к-одному**).

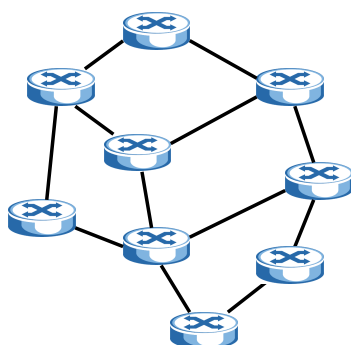
Другие приложения, такие как некоторые игровые программы или протокол IPSec в транспортном режиме, не поддерживают NAT, поскольку они включают информацию об адресации в потоке данных. Такие приложения не будут работать через NAT, даже при использовании типов отображения один-к-одному и много-ко-многим без перегрузки.

Протокол покрывающего дерева (STP/RSTP)

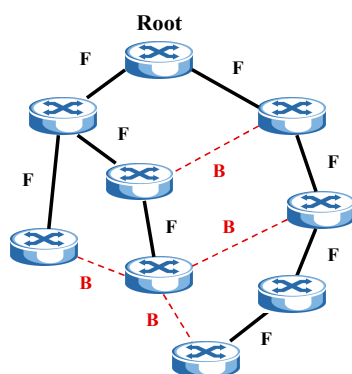
ZyXEL

Структура ЛВС

Первоначальная



После STP

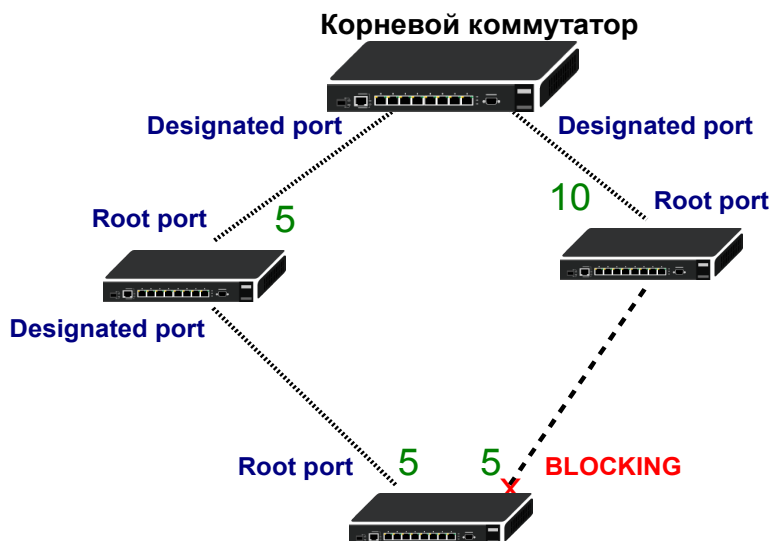


F : Forward

B : Block

ZyXEL

Spanning Tree Protocol (RSTP)



Протокол покрывающего дерева, Spanning Tree Protocol (STP) служит для отключения тех портов коммутатора, которые приводят к появлению «циклов» в сети, т.к. невозможно построить корректную таблицу фильтрации MAC-адресов в сети с циклами. Кроме того, циклы приводят к возникновению широковещательных штормов.

Сначала из всех коммутаторов выбирается «корневой», и затем происходит отключение портов с наибольшей «стоимостью» передачи кадра до корневого моста. Протокол STP требует определенных настроек параметров коммутатора и его портов.

В устройствах ZyXEL реализована быстрая версия STP – 802.1w RSTP (Rapid STP), вошедшая в стандарт коммутатора 802.1d от 2004 года.

Алгоритм работы STP начинается с выбора корневого коммутатора. Все коммутаторы рассылают друг другу пакет Hello с указанным в нем приоритетом. Корневым становится устройство с наименьшим значением приоритета или (если таких несколько) с наименьшим MAC-адресом.

Когда корневой коммутатор будет определен, он начнет рассылать пакеты BPDU (Bridge Protocol Data Unit) со всех своих портов. В BPDU содержится информация о «стоимости» пути. BPDU, исходящие от корневого коммутатора, имеют нулевую стоимость пути. Каждый коммутатор, получая BPDU, прибавляет туда стоимость пути, приписанную к порту, на который BPDU был получен, и передает BPDU с новым значением стоимости во все порта, за исключением того, откуда был получен этот BPDU пакет.

Если BPDU был получен на два и более портов коммутатора, то коммутатор делает вывод, что в сети имеется цикл – и отключает порт, стоимость пути к которому оказалась выше. Таким образом, у каждого не корневого коммутатора в сети окажется один порт, на который он получает BPDU с наименьшей стоимостью (называется **корневой порт** – **root port**), и несколько неотключенных портов, по которым пакеты идут вниз по сети (**назначенные порты** – **designated ports**). У корневого коммутатора все порты являются назначенными, и ни один не отключен.

В протоколе RSTP каждый порт может быть в одном из трех состояний:

- **Discarding** – порт не включен в активную топологию, не вносит записей в таблицу MAC-адресов;
- **Learning** – на порту включен режим обучения по BPDU-пакетам;
- **Forward** – порт включен в активную топологию.

В RSTP портам могут быть присвоены роли: root, designated (о них уже говорилось выше), а также alternate и backup. Порт получает роль **alternate**, если другой порт получил более выгодный BPDU от соседнего коммутатора. Порт получает роль **backup**, если на него пришел BPDU, отправленный другим портом этого же коммутатора.

STP vs RSTP

STP и RSTP предназначены для одного и того же – для построения дерева.

Время реакции STP на изменение топологии составляет $2 * (\text{forward_delay} + \text{max_age})$, а RSTP – менее секунды.

RSTP совместим с STP, но при этом его преимущества теряются.

ZyXEL

Отличия STP (802.1d до 2004 года) от RSTP (802.1w) заключаются в следующем:

Состояния портов: состояния портов Disabled, Listening и Blocking алгоритма STP объединены в RSTP в одно – Discarding. Добавлены новые роли портов: **backup** – резервный для назначенного – присутствует, когда несколько портов моста подключены к одной LAN; **alternate** – альтернативный корневому – избыточный путь к корневому коммутатору. В активную топологию эти новые роли не входят. Активную топологию формируют порты, находящиеся в состоянии forward. Быстрая смена топологии происходит за счет быстрого включения alternate-портов и лавинообразного процесса handshake.

Формат сообщения о конфигурации: к битовым флагам Topology Change (TC) и Topology Change Ack (TCA) добавлены еще шесть – два отвечают за proposal/agreement механизма handshake, остальные четыре отвечают за роль и состояние порта, сгенерировавшего сообщение. В поле типа и версии протокола передается число 2 (в STP передавался 0).

Сообщения keep-alive. Каждый коммутатор в RSTP независимо генерирует собственные сообщения keep-alive через интервал Hello.

Быстрое обновление таблицы фильтрации. В STP, когда мост обнаруживал изменение топологии, он отправлял пакет TCN (Topology change notification) на корневой мост, оставлял старые записи в таблице фильтрации (принудительно уменьшив их возраст), а пакеты TC имел право рассылать только корневой мост. Все это приводило к медленной реакции сети на изменения топологии. В RSTP мосты сразу стирают ненужные записи из таблицы фильтрации, а также имеют право рассылать собственные кадры TC.

Быстрый переход из Discarding в Forward. В STP, из опасения получить циклы, порты переводились из Blocking в Forward только по прошествии некоторого интервала времени (несколько секунд), что приводило к полной потере связи. В RSTP корневой порт может перейти в состояние forward без всяких дополнительных действий, а назначенный порт для перехода в forward использует механизм handshake, получая подтверждение от соседнего коммутатора.

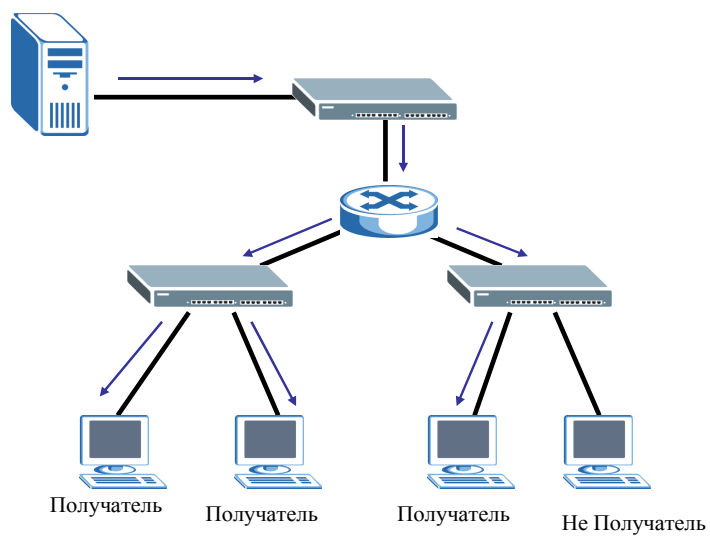
Изменение топологии. Когда коммутатор RSTP обнаруживает изменение топологии, он запускает таймер, вдвое превышающий Hello, и затем посылает на корневой и назначенные порты сообщение с флагом TC. Коммутатор, получивший TC, сразу же очищает свою таблицу MAC, оставляя там только записи для порта, с которого был получен TC, затем он в свою очередь запускает таймер и передает TC.

RSTP совместим с STP, однако его основное преимущество – быстрая реакция на изменения топологии – теряется при взаимодействии со старыми STP-коммутаторами.

Многоадресная рассылка

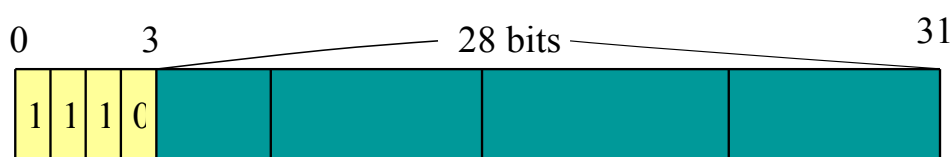
ZyXEL

Multicast

**ZyXEL**

Multicast IP Адрес

- IP класса D
- Один IP идентифицирует группу



224.0.0.0 ~ 239.255.255.255

ZyXEL

Multicast – протокол сетевого уровня. Стандарт определяет использовать IP адреса класса D. Каждый IP адрес идентифицирует группу пользователей. Диапазон IP адресов: с 224.0.0.0 по 239.255.255.255.

Предопределенные Группы

- Некоторые Multicast IP зарезервированы IANA
- Распространенные зарезервированные IP
 - 224.0.0.1 : Все узлы данной подсети
 - 224.0.0.2 : Все маршрутизаторы данной подсети
 - 224.0.0.9 : маршрутизаторы RIP-2
 - 224.0.1.1 : NTP(Network Time Protocol)

* IANA - Internet Assigned Numbers Authority

ZyXEL

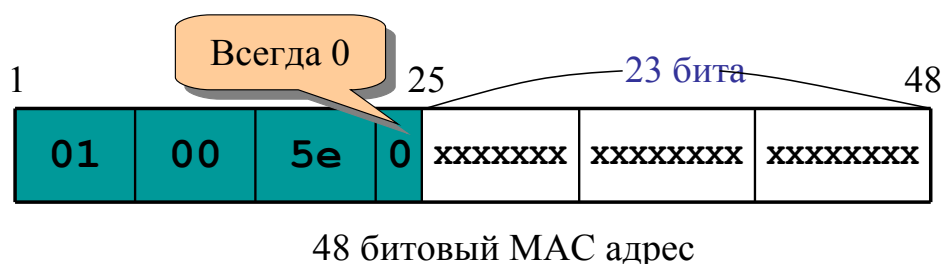
IANA - Internet Assigned Numbers Authority

<http://www.iana.org/assignments/multicast-addresses>

В IP диапазоне класса D существуют зарезервированные IP адреса, которые находятся в диапазоне между 224.0.0.0 и 224.0.0.255. Они используются в протоколах маршрутизации, протоколах управления, и т.д. Маршрутизаторы с поддержкой Multicast не должны отправлять трафик с такими IP адресами дальше в сеть если используется передача не определенных различными стандартами трафика в не зависимости от значения TTL.

Multicast IP и MAC адрес

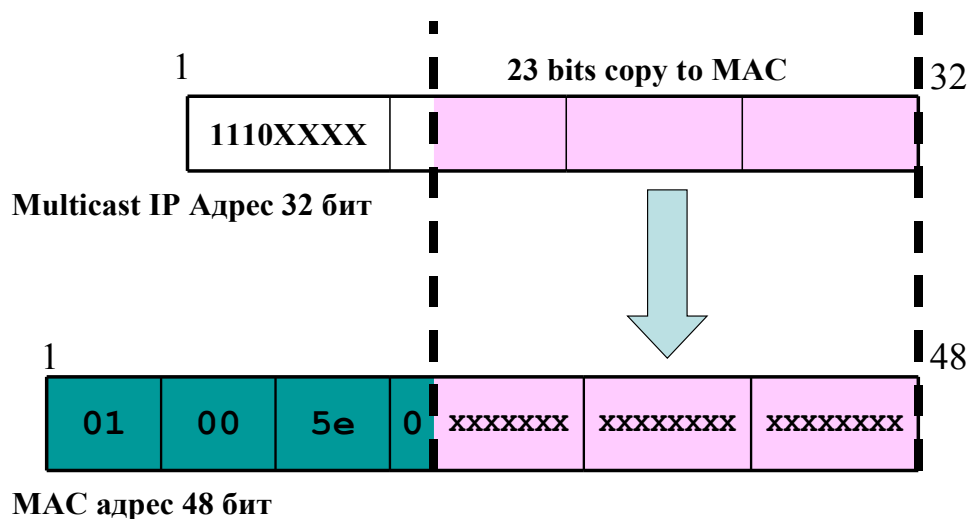
- Префикс 01:00:5e для Multicast кадров
- Поле информации в MAC адресе: младшие 23 бита
- Диапазон 01:00:5e:00:00:00 ~ 01:00:5e:7f:ff:ff



ZyXEL

- В сетях Ethernet, каждый пакет должен иметь MAC адрес для правильной доставки на втором уровне. Так как IP Multicast не имеет никакого отношения к физическим адресам уровня MAC, то необходим определенный механизм назначения MAC адреса в соответствии с IP Multicast адресом. Для решения этой проблемы IANA выделили определенный диапазон MAC адресов для адресации Multicast трафика.
- MAC адрес имеет 48 бит. IANA зарезервировали младшие 23 бита для адресации multicast IP. Префикс 01:00:5e и 25 бит всегда установлен в 0. Так что диапазон Multicast MAC адресов с 01:00:5e:00:00:00 по 01:00:5e:7f:ff:ff.

Зависимость MAC от Multicast IP



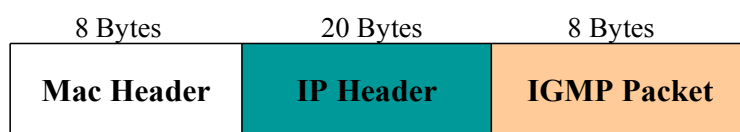
* 32 (2^5) multicast IP адреса будут иметь один и тот же multicast MAC адрес.

ZyXEL

- IANA зарезервировала младшие 28 бит IP адреса и младшие 23 бита MAC адреса для Multicast сообщений. MAC адрес использует младшие 23 бита IP адреса путем копирования из IP в MAC адрес.
- Так как используется только младшие 23 бита IP адреса для точной идентификации трафика на MAC уровне, существует вероятность, что $32(2^5)$ multicast IP адреса будут иметь один и тот же multicast MAC адрес.
- Если в сети существует подписчики на разные IP Multicast услуги, но при этом в этих IP адресах младшие 23 бита совпадают, то все пользователи будут получать как трафик одной подписки, так и трафик другой подписки на канальном уровне. Определение правильной подписки произойдет на сетевом уровне при сравнении IP адресов.

IGMP

- Internet Group Management Protocol
- Автоматический контроль и регулирование multicast потока по LAN.
- Работает на Сетевом уровне
- IGMP v1 – RFC 1112
- IGMP v2 – RFC 2236
- IGMP v3 – RFC 3376

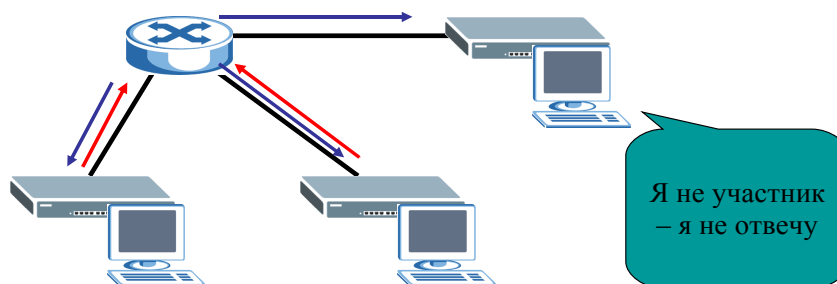


ZyXEL

- IGMP - Internet Group Management Protocol (межсетевой протокол управления группами).
- Используется для контроля и управления multicast трафиком через локальную сеть в автоматическом режиме. IGMP работает только между маршрутизатором и клиентским хостом.
- Это протокол третьего уровня
- Сейчас существует три версии IGMP. IGMP version 1 определен в RFC1112, version 2 определен в RFC2236 и version 3 определен в RFC3376
- Заголовок IGMP находится после заголовка IP в пакете. Он 8 байт.

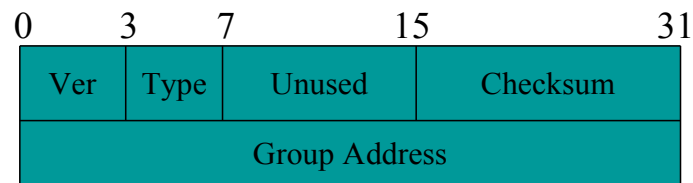
Тип сообщения IGMP

- Query message
- Report message



ZyXEL

Формат IGMP v1



- Version = 1
- Type :
 - 1 = Query
 - 2 = Report
- Group address :
 - Адрес Multicast Group

ZyXEL

Как работает IGMP v1?

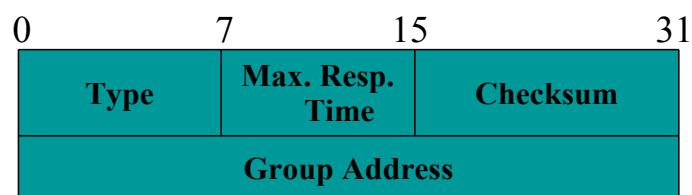
- Присоединение к группе: отправка **Report** на групповой адрес не дожидаясь **Query**
- Multicast router отправляет **General Query** для распознавания членов (224.0.0.1)
- Выход из группы без каких-либо сообщений

ZyXEL

Отличительные особенности IGMP version 1.

- Для присоединения к группе необходимо отправить только report message на групповой номер для того, чтобы стать членом группы.
- Multicast router отправляет General query message по адресу 224.0.0.1. Данный запрос используется для определения наличия хотя бы одного члена группы в подсети.
- Логика работы члена группы в подсети:
Когда хост получает query, он запускает таймер обратного отсчета для каждой multicast group в которую он входит. Счетчик устанавливается какому-то произвольному числу из допустимого значения. В IGMPv1, максимум равен 10 секундам. Счетчик (таймер) устанавливается в значение из диапазона 0-10 секунд.
Когда счетчик обнулится – истечет время – хост отправит membership report в группу, которая ассоциирована с этим таймером. Данный report говорит о том, что данный хост все еще является членом этой группы. Однако, если хост получит membership report перед тем как таймер соответствующей группы обнулится, хост сбросит таймер и подавит свой собственный report.
- Когда член группы хочет выйти, он просто перестает отправлять какие-либо сообщения в сети по протоколу IGMP v.1.

Формат пакета IGMP v2



- Type :
 - 0x11 = Query
 - 0x12 = Version 1 Report
 - 0x16 = Version 2 Report
 - 0x17 = Leave Report
- Maximum Response Time :
 - Максимальное время реагирования

ZyXEL

- В Version 2 нет поля version, которое теперь объединено с полем type. Существует четыре типа. 11 – запрос (query). 12 - version 1 report. 16 - version 2 report. 17 - leave report (сообщение о покидании группы).
- Maximum Response Time
По умолчанию = 10 seconds. Значимое поле только в Membership Query. Определяет максимальное время отправки ответа. Единица значения - 1/10 секунды.
- Group address – адрес multicast группы.

Как работает IGMP v2?

- Присоединение к группе отправкой **Report** на групповой адрес не дожидаясь **Query** (как и в IGMP v1)
- Router с наименьшим IP адресом - “elected” querier (запрашивающий router)
- Multicast router отправляет **General Query** для обнаружения членов
- Уход из группы по **Leave message** на 224.0.0.2
- Multicast Router отправляет **Group Specific Query** для обнаружения существует ли члены текущей группы

ZyXEL

- Для присоединения к группе необходимо отправить только report message на групповой номер для того, чтобы стать членом группы. (Аналогично IGMP v.1)
- Если в сети существует несколько маршрутизаторов, то маршрутизатор с наименьшим IP адресом выбирается в качестве отправителя запросов. Другие маршрутизаторы не будут отправлять запросы клиентам.
- Логика работы члена группы в подсети такая же как и для IGMP v.1 за исключением того, что временной диапазон зависит от значения maximum response time
- Когда член группы хочет выйти, он отправляет сообщение о выходе на адрес 224.0.0.2 (все маршрутизаторы этой подсети).
- Когда multicast router получает сообщение о выходе члена из группы, он отправляет query запрос на текущую группу для обнаружения наличия хотя бы одного члена данной группы. Запрос отправляется только для данной группы. По сравнению с general query, данный запрос не нагружает сеть ответами членов всех групп.

Как работает IGMP v3?

- Использование новых мультикаст-сервисов - **SSM**.
- Оптимизация использования полосы пропускания: получатель запрашивает получение трафика только от известных источников.
- Улучшенная безопасность: отсутствие DoS-атак от неизвестных источников.

ZyXEL

В IGMP версии 3 была добавлена поддержка фильтрации адресов (**source filtering**). С помощью этого механизма узел может сообщить, с каких адресов он хочет получать пакеты, а с каких нет. В протоколе IGMP v3 используются следующие типы IGMP-сообщений:

- запрос о составе группы протокола IGMP версии 3;
- отчет о составе группы протокола IGMP версии 3.

При использовании протокола IGMPv3 получатели сообщают о принадлежности к группе многоадресной рассылки, используя приведенные ниже режимы.

Режим **INCLUDE**. В этом режиме получатель сообщает о членстве в группе узлов и приводит список адресов источников (список **INCLUDE**), от которых желательно получение данных.

Режим **EXCLUDE**. В этом режиме получатель сообщает о членстве в группе многоадресной рассылки и предоставляет список адресов источников (список **EXCLUDE**), от которых получение потоков данных нежелательно. При этом узел будет получать потоки данных только от тех источников, IP-адреса которых не перечислены в списке **EXCLUDE**. Для получения данных от всех источников узлы используют режим принадлежности к группе с пустым списком **EXCLUDE**.

Протокол IGMPv3 поддерживает многоадресную рассылку, специфичную для конкретного источника "**Source Specific Multicast**" (**SSM**).

Если два приложения с различными источниками и получателями используют один и тот же групповой адрес многоадресной рассылки IP, то получатели будут получать данные от обоих приложений-отправителей.

В сети многоадресной рассылки, использующей протокол **SSM**, ближайший к получателю маршрутизатор увидит запрос от принимающего приложения на подключение к конкретному источнику многоадресной рассылки. Приложение получателя может сообщить о своем намерении присоединиться к конкретному источнику путем использования режима **INCLUDE** в протоколе IGMPv3.

Способность протокола **SSM** явным образом включать и исключать источники обеспечивает некоторый уровень безопасности. Потоки данных от источника к получателю, не указанные в списке **INCLUDE**, не будут отправляться пользователям, не заинтересованным в них.

Управление

ZyXEL

Обзор

- Через консоль
- По telnet
- По WEB
- По FTP
- По SSH, HTTPS, SFTP
- По TFTP
- По Bootp/TFTP
- Через SNMP

ZyXEL

Консоль

- Все оборудование ZyXEL имеет или выведенный на корпус консольный порт, или скрытый на печатной плате.



Через консоль возможно получить доступ к SMT управлению, которое может содержать как только командную строку, так еще и систему меню.

Используется для локального управления, обновления программного обеспечения по Bootp/TFTP в ОС ATMOS.

Пароль по умолчанию – 1234. Наиболее безопасный способ управления оборудованием.

Telnet

- Аналогичен управлению по консоли только подключение осуществляется удаленно через Ethernet.

ZyXEL

Данный способ управления оборудованием аналогичен по функционалу управлению по консоли, однако управление возможно удаленное через Ethernet. Недостаток – данные, в том числе и пароль, передаются в открытом виде, сложно применима если надо сменить какую-либо информацию, связанную с IP адресацией.

WEB

- Наиболее удобный способ.
- Используется встроенный WEB сервер.
- Возможности по настройке такие же или выше чем при настройке через командную строку

ZyXEL

В данном случае оборудование ZyXEL выступает в качестве WEB сервера. Пароль аналогичен паролю для консоли и telnet. В некоторых версиях ПО необходимо ввести имя пользователя – admin.

Наиболее удобный способ начального конфигурирования. Возможности ограничены и практически отсутствуют средства отладки и поиска неисправностей.

FTP

- Оборудование ZyXEL выступает в качестве FTP сервера.
- Используется только для доступа к файлу (-ам) конфигурации и внутреннего ПО.



В данном случае оборудование ZyXEL выступает в качестве FTP сервера. Пароль аналогичен паролю для консоли и telnet. В некоторых версиях ПО необходимо ввести имя пользователя – admin.

Наиболее удобный способ обновления и сохранения конфигурации. Может быть использован для специализированных программ, которые по расписанию сохраняют конфигурацию с оборудования и автоматически обновляют ПО при выходе новой версии.

SSH, HTTPS, SFTP

- Аналогично telnet, web и ftp, только все данные передаются в зашифрованном виде и использованием сертификатов.
- Данная функция присуща оборудованию обеспечения безопасности – ZYWALL, и оборудованию, работающему с сертификатами

ZyXEL

Аналогично telnet, Web, FTP, но данные передаются в зашифрованном виде. Используются сертификаты. Поддерживается ZyWALL, которые работают с сертификатами.

TFTP

- Аналогично FTP по функциональным характеристикам, только оборудование ZyXEL выступает в качестве клиента, а сервером является ПК, с которого идет управление.

ZyXEL

BOOTP/TFTP

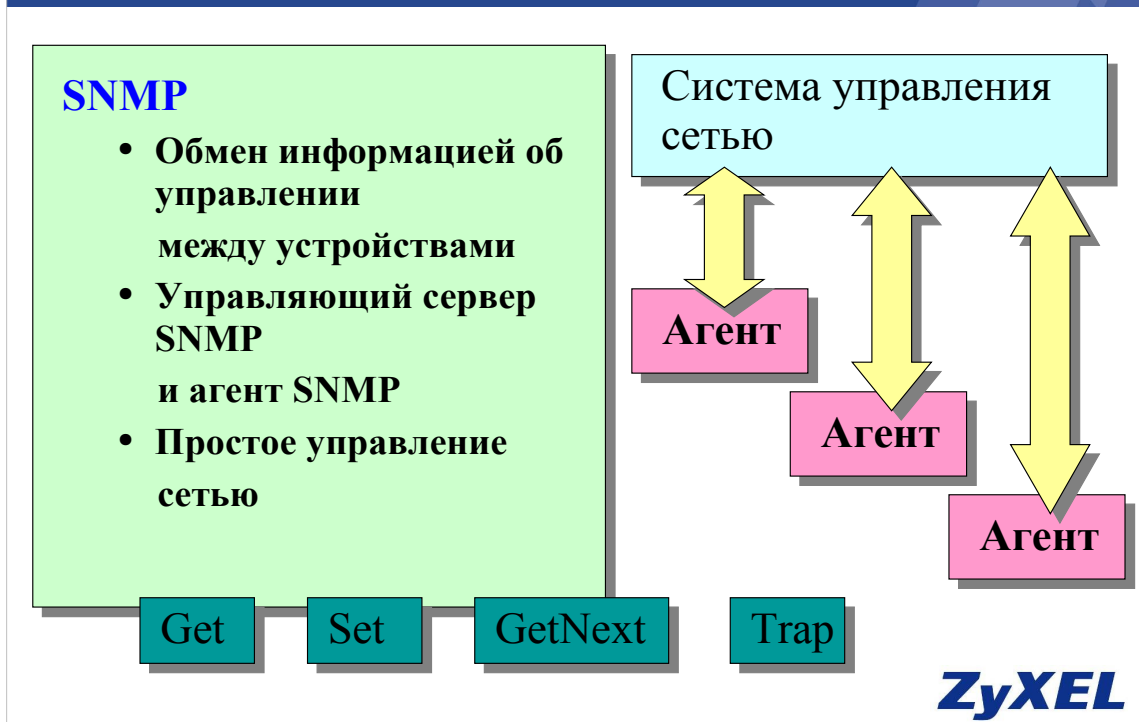
- Используется в ОС ATMOS (ADSL и G.SHDSL DSLAM модули – AAM1008, SAM1008, ALC-1024, SLC1024)
- Используется как механизм восстановления ПО после сбоя
- Более подробно рассмотрен в соответствующем курсе.



Используется в ОС ATMOS (ADSL и G.SHDSL DSLAM модули – AAM1008, SAM1008, ALC-1024, SLC1024)

Используется как механизм восстановления ПО после сбоя. Требуется подключения как по консоли, так и по Ethernet.

SNMP



SNMP (Simple Network Management Protocol/Простой протокол управления сетью) представляет собой протокол прикладного уровня, используемый для обмена информацией об управлении между сетевыми устройствами (напр., маршрутизаторами).

Благодаря SNMP сетевым администраторам гораздо проще управлять производительностью сети, обнаруживать и решать проблемы сети. SNMP принадлежит к стеку протоколов TCP/IP и использует UDP для обмена сообщениями между клиентом управления и агентом, находящимся в узле сети.

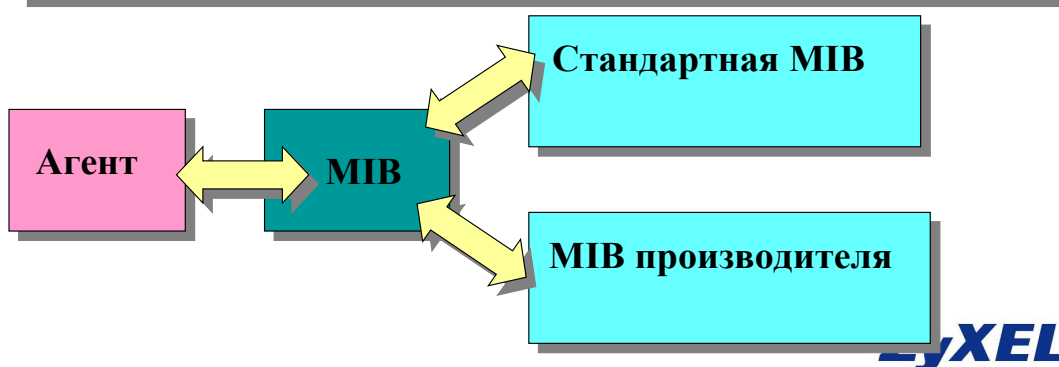
Командами SNMP являются: **Get**, **GetNext**, **Set** и **Trap**. С помощью этих команд можно управлять переменными, такими, как счетчики статистики, статус порта узла и т.д., существующие в узлах сети. Все функции управления SNMP реализуются с помощью этих простых команд.

Для SNMP пароль и логин не имеет никакого отношения к паролю для управления устройством (к паролю для консоли, telnet и т.д.)

Базы управляющей информации SNMP

ZyXEL поддерживает MIB II (База управляющей информации)

- Набор переменных, поддерживаемых каждым устройством
- Стандартная MIB
- MIB производителя



Набор переменных, поддерживаемых каждым узлом, носит название **Базы управляющей информации (MIB)**. MIB состоит из нескольких частей, включая **Стандартную MIB**, определяемую как часть SNMP, и **MIB производителя**, которая предназначена для управления аппаратным обеспечением конкретного производителя.

Текущая стандартная MIB, MIB II, определяется в документе RFC 1213 и содержит 171 объект. Эти объекты группируются по протоколу (включая TCP, IP, UDP, SNMP) и другим категориям, таким, как 'система' и 'интерфейс'.

Протокол SNMP представляет собой простой протокол, работающий по схеме "запрос-ответ". Ниже описываются четыре операции SNMP.

1. Get (Получить)

Позволяет системе управления сетью (Network Management System – NMS) извлекать значения объектов.

2. GetNext (Получить следующее)

Позволяет NMS извлекать следующее значение объекта из таблицы или списка внутри агента. В версии 1 SNMP, если NMS хочет извлечь все элементы таблицы из агента, она инициирует сначала команду 'Get', а затем серию команд 'GetNext'.

3. Set (Установить)

Позволяет NMS установить значения для объектов внутри агента.

4. Trap (Прерывание)

Используется агентом для информирования NMS о произошедших событиях.

Часть оборудования поддерживает только стандартные MIB, а часть оборудования поддерживает и MIB производителя, через которые возможно полное управление устройством как в визуальном, так и в командном режиме.