**Концентратор (Хаб)**

Работает на физическом уровне. (Layer 1).

Хаб получает биты на одном порту и пересылает их во все другие порты.

**Типы хабов:**

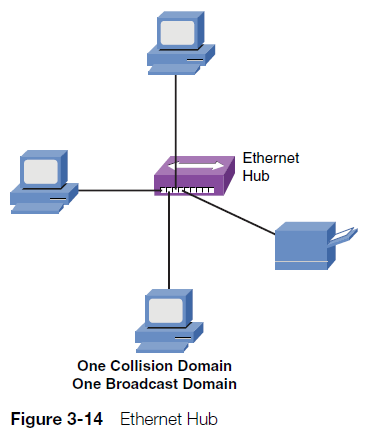
* **Пассивный**. Не изменяет (электрически) полученные биты.
* **Активный**. Восстанавливает входящие биты и посылает во все порты.
* **Умный**. (smart) Под “умным” хабом обычно подразумевается активный хаб поддерживающий дополнительные функции, например Simple Netwrok Management Protocol (SNMP).

Главный недостаток: все порты образут **домен коллизий**.

Домен колизий - это область сетив которой может выполняться только одна операция обмена данными в опеределенный момент времени.

Второй недостаток: неэффективное использование пропускной способности (данные шлются во все порты).

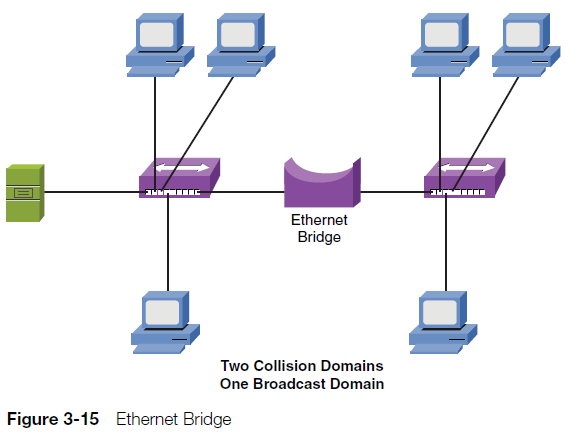
Беспроводная точка доступа - почти как хаб. Так же образует домент колизий.



**Мосты**

Layer 2 устройство.

Мосты делят домены коллизий. Но это один широковещательный домен.



**Коммутаторы**

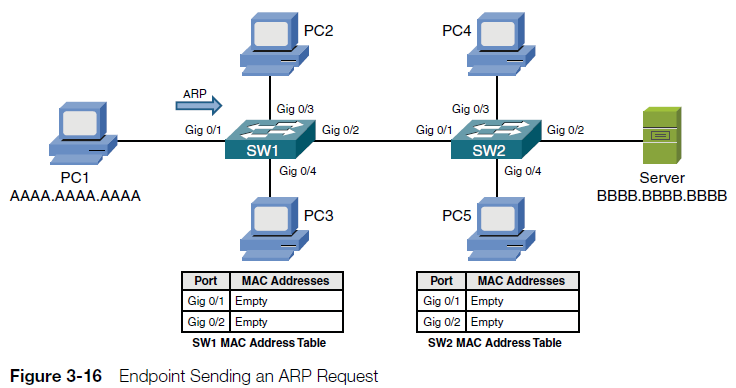
1. У коммутатора есть таблица MAC адресов
2. Когда коммутатор не знает кто где - он обучается - работает как концентратор - шлет пакет во все порты

Пример того, как заполняется таблица MAC адресов:

рассмотрим как PC1 подключается к Telnet серверу. PC1 и Server находятся в одной подсети.

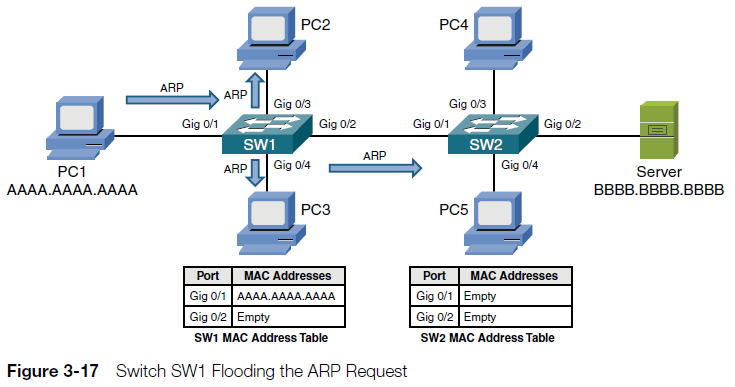
Перед тем как послать запрос на открытие Telnet сессии, PC1 надо знать IP адрес (Layer 3 адрес) и MAC адрес (Layer 2) адрес сервера. IP адрес как правило известен, или его можно узнать с помощью DNS. В нашем примере IP адрес сервера известнен. Остается узнать MAC адрес.

PC1 не содержит MAC адрес сервера в своем кэше (ARP cache), поэтому он посылает address resolution protocol (ARP) запрос.



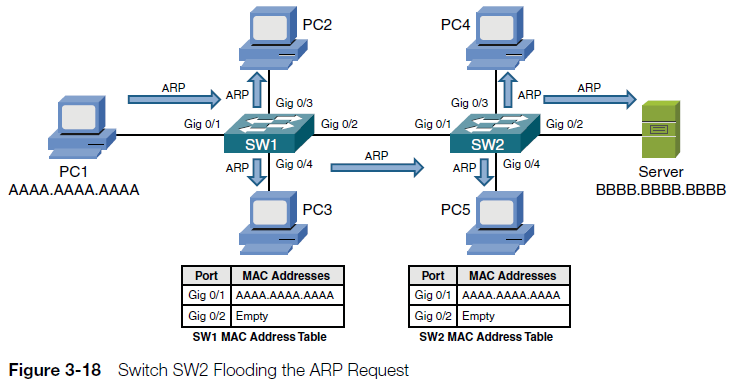
Когда коммутатор SW1 получает ARP запрос от PC1 он записывает в таблицу MAC адресов - для порта Gigabit 0/1, MAC адрес: AAAA.AAAA.AAAA.AAAA.

Так как ARP запрос - широковещательный: SW1 рассылает его во все другие порты.

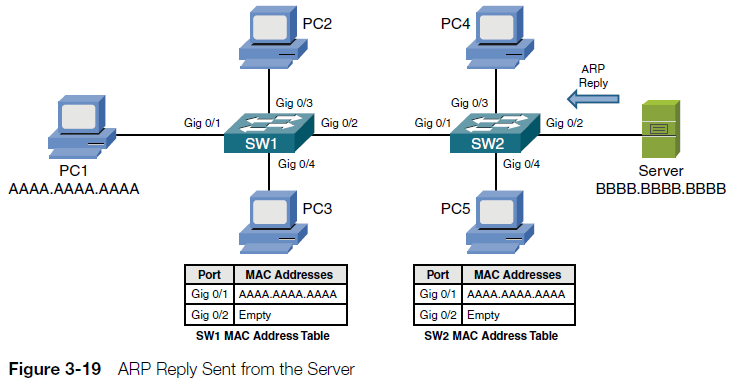


Когда SW2 получает ARP-запрос на своем Gig0/1 порту, он добавляет адрес источника ARP-запроса в свою таблицу MAC адресов.

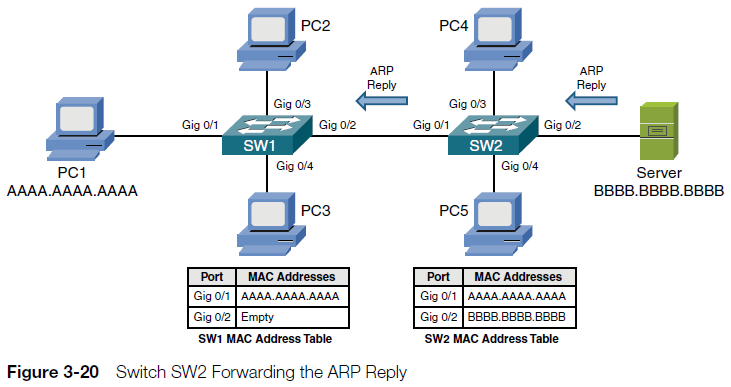
SW2 рассылает ARP-запрос дальше во все другие порты.



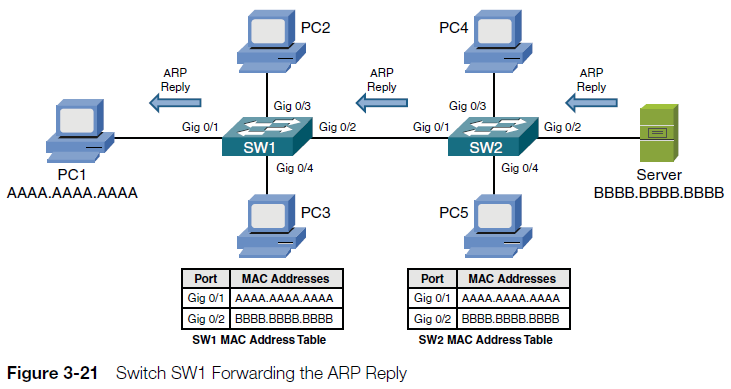
Сервер получает ARP-запрос и отвечает ARP-ответом. ARP-ответ в отличии от ARP-запроса не широковещательный, его получатель: AAAA.AAAA.AAAA.AAAA.



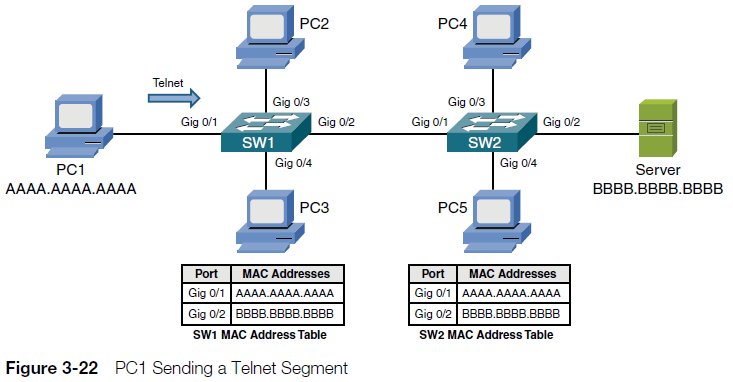
SW2 получив ARP-ответ от сервера добавляет его MAC адрес в свою таблицу MAC адресов. Далее зная что получатель ARP-ответа с MAC-адресом АААА.АААА.АААА.АААА находится на порту Gig 0/1, посылает ARP-ответ только на этот порт.



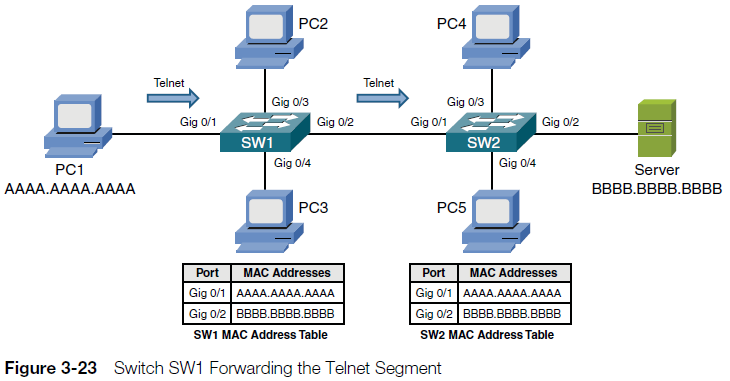
SW1 получает на Gig 0/2 ARP-ответ и добавляет в свою таблицу MAC адресов строчку - о том что адрес BBBB.BBBB.BBBB.BBBB находится на порту Gig 0/2. Далее SW1 передает ARP-запрос на порт Gig 0/1.



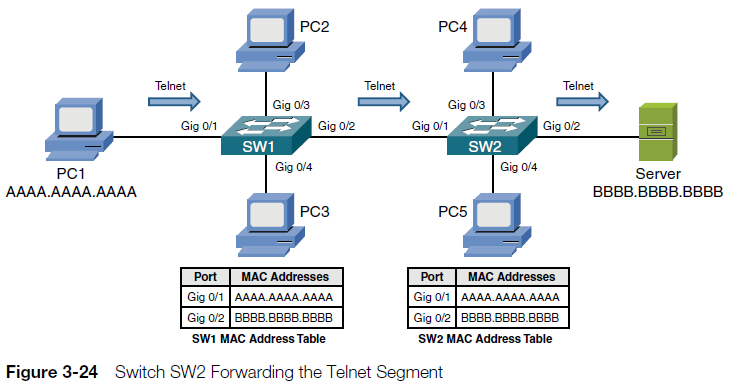
После получения ARP-ответа, PC1 узнает MAC адрес сервера. Теперь PC1 может собрать Telnet сегмент и передать его серверу.



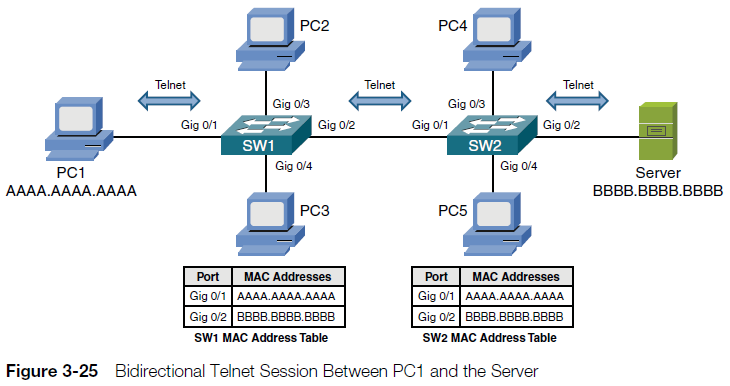
SW1 находит MAC-адрес сервера BBBB.BBBB.BBBB.BBBB в своей таблице MAC адресов и передает Telnet сегмент далее в порт Gig 0/2.



SW2 так же знает адрес получателя Telnet сегмента, и направляет сегмент на порт Gig 0/2.

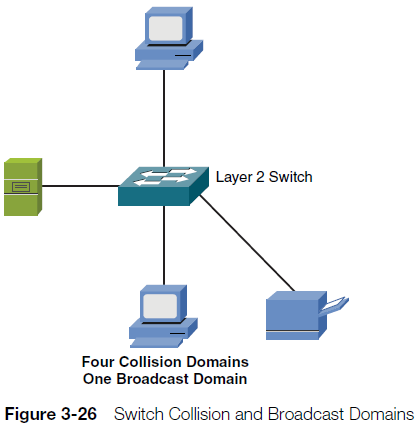


В итоге сервер отвечает PC1 и Telnet сессия устанавливается.



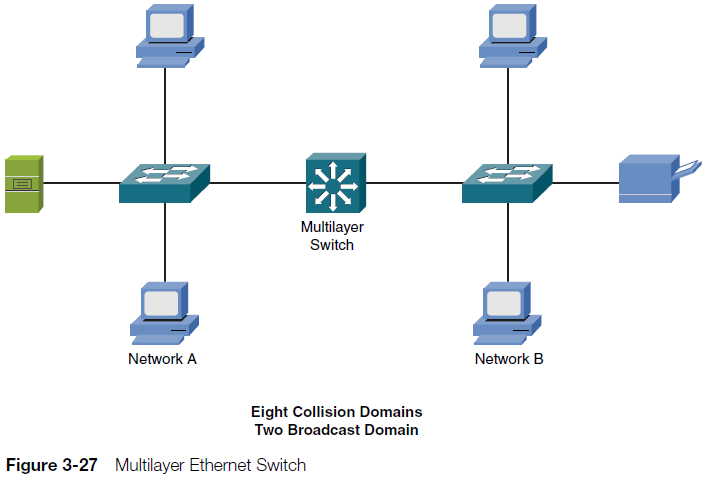
Т.к. PC1 изучил MAC адрес сервера с помощью ранее посланного arp-запроса и сохранил его в локальном arp кэше, передача последующих Telnet сегментов не потребует больше arp-запросов.

Записи в arp-кэше PC1 имеют таймаут, и если они не будут использоваться некоторое время, они будут удаляться. Таким образом компьютеру потребуется снова выполнить arp-запрос. Посылка ARP-запроса, добавляет дополнительное количество времени к общей задержке.



**Многоуровневый коммутатор**

* Может работать как маршрутизатор, и пересылать данные на основе IP адреса получателя.
* В некоторой литературе называется Layer 3 Switch
* Каждый порт - отдельный домен коллизий

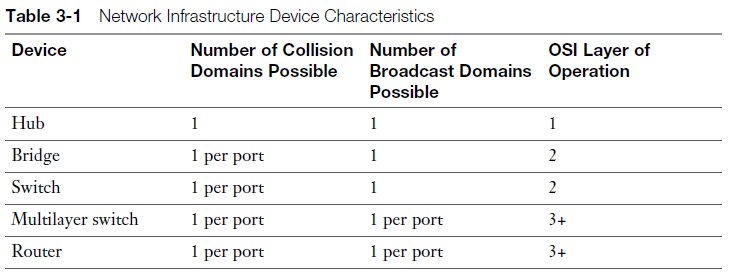


**Маршрутизатор**

* Layer 3 устройство
* пересылает данные на основе логических сетевых адресов (IP адрес)
* имеет возможность работать с уровнями выше сетевого - например QoS.
* Отличие маршрутизатора от многоуровнего коммутатора - маршрутизатор поддерживает больше интерфейсов разного типа; более многофункционален.
* Например, если вам нужно устройство Layer 3 для подключения к Провайдеру Интернет Услуг (Internet Service Provider) через последовательный порт, вам будет легче найти модуль серийного порта для вашего маршрутизатора.

**Активное сетевое оборудование. Итоги**

Таблица 3-1 суммирует характеристики активного сетевого оборудования



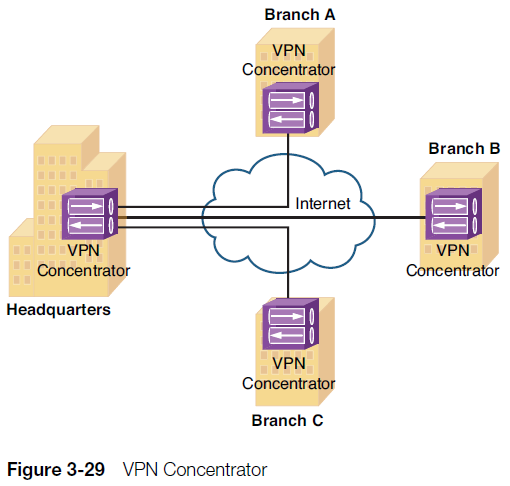
**СПЕЦИАЛИЗИРОВАННОЕ СЕТЕВОЕ ОБОРУДОВАНИЕ**

Хотя инфраструктурное сетевое оборудование (коммутаторы, мрашрутизаторы) строит ядро сети, для добавления большего количества сервисов конечным пользователям, многие сети содержат специализированное оборудование: vpn-концентраторы, межсетевые экраны, dns-серверы, dhcp-серверы, proxy-серверы, caching engines, content switches.

**VPN концентраторы**

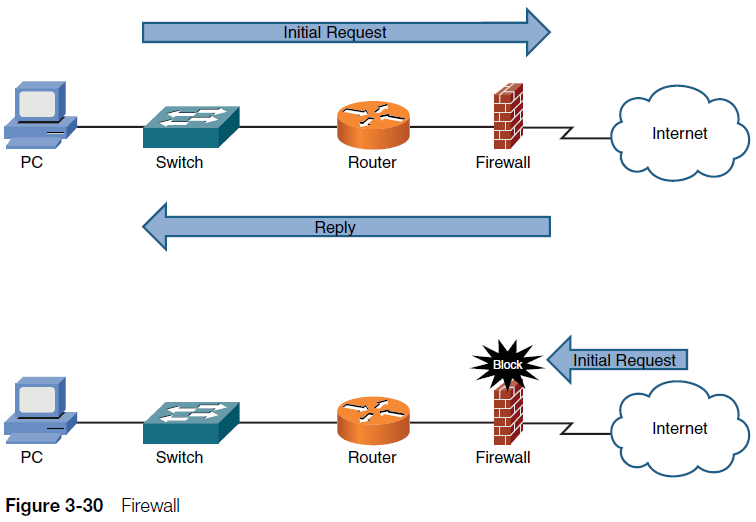
Организациям имеющим множество распределенных филиалов, часто надо защитить соединения между филиалами. Одна из возможностей сделать это: купить множество WAN соединений соединяющих все сайты. Иногда, дешевле создать защищенные соединения через небезопасную сеть, такую как Интернет. Этот защищенный туннель называется VPN (virtual private network). В зависимости от используемой технологии VPN, устройства организующие VPN могут требовать выполнения “тяжелых” расчетов. Например, рассмотрим главный офис компании с 100 VPN соединений к 100 удаленным сайтам. Устройство в главном офисе обрабатывает, занимается шифрованием и аутентификация каждого туннеля - результат высокие нагрузки на устройство.

Некоторые модели маршрутизаторов могут выполнять функции VPN концентратор, но можно использовать и отдельное устройство.



**Межсетевые экраны (Firewalls)**

Устройство сетевой безопасности.



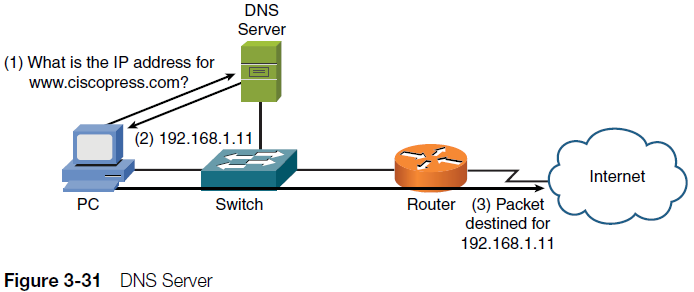
**DNS-сервера**

Domain Name System (DNS) позволяет решать задачу преобразованию доменного имени в соответствующий IP адрес.

Т.к. маршутизаторы (или многоуровневые коммутаторы) продвигают данные на основе IP адресов - IP пакету нужно содержать IP адрес, а не доменное имя.

Пример.

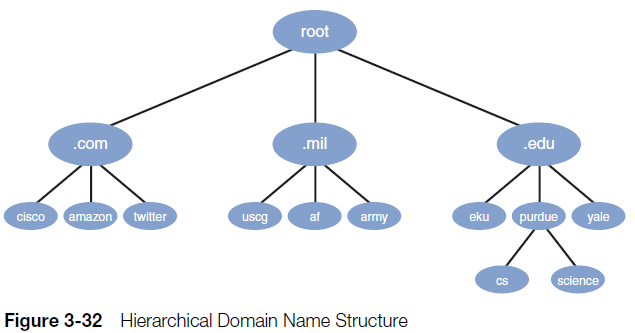
Пользователь хочет попасть на сайт [www.ciscopress.com](http://www.ciscopress.com) - это fully-qualified domain name (FQDN). Вводит имя сайта в браузере. Но браузер не может отправить HTTP-запрос - ему нужно знать IP адрес. Чтобы узнать IP адрес, клиент посылает запрос DNS-серверу. DNS-сервер содержит базу данных - в которой записаны соответствия FQDN и IP адресов. Так же DNS-сервер содержит адреса других серверов, чтобы работать с ними в случае если в его базе (кэше) нет таких записей. DNS-сервер разрешает имя и возвращает ответ. Клиент формирует HTTP запрос и отправляет его в сеть.



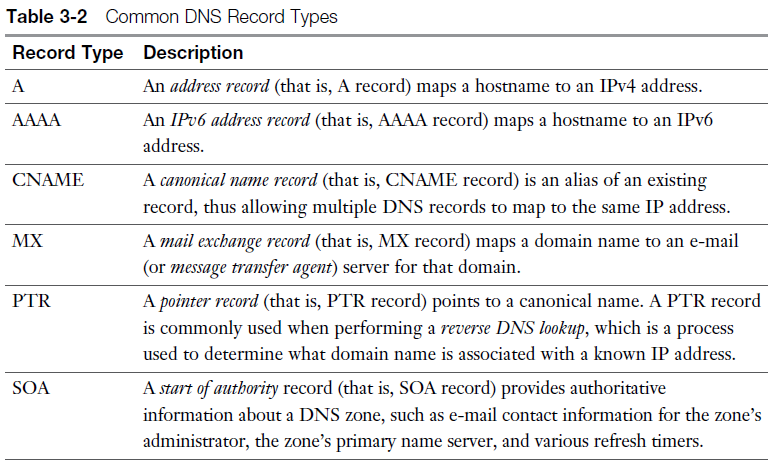
FQDN - это серия строк разделенных точками. Правая строка (последняя часть) представляет корневой домен.

Пример корневых доменов: .com, .ru, .edu.

DNS-сервера нижних уровней могут ссылаться на DNS-сервера верхних уровней - чтобы разрешить нелокальные FQDN.



БД DNS-сервера кроме полных доменных имен так же содержит ресурсные записи.



DNS-записи могут становится невалидными - когда вы меняете IP адрес, но запись в БД DNS-сервера не меняется.

Для решения этой проблемы есть dynamic DNS (DDNS).

Другой вариант DNS - Extension Mechanisms for DNS (EDNS).

**DHCP-сервер**

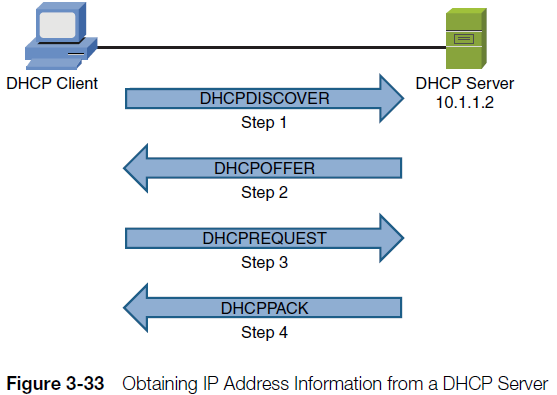
Вопрос: Как устройство может получить свой первоначальный IP адрес?

* первый вариант - настроить вручную (это занимает много времени, подверждено ошибкам)
* второй вариант - динамическое назначение .

Dynamic Host Configuration Protocol (DHCP)

Назначается не только IP адрес - а целый набор параметров (маска сети, шлюз по умолчанию, dns-сервер).

Пример: алгоритм работы DHCP

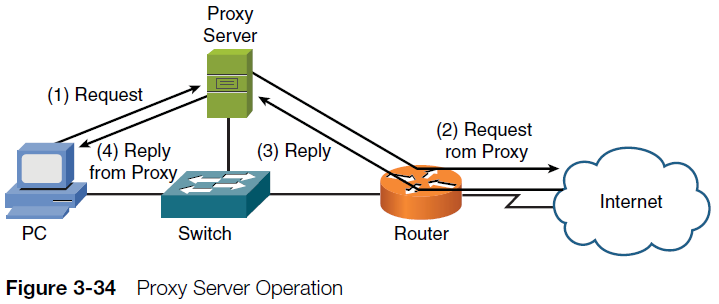


1. Когда DHCP клиент загрузился у него нет никаких настроек. Клиент посылает широковещательный запрос в поисках DHCP-сервера.
2. Когда DHCP-сервер получает DHCPDISCOVER запрос он отвечает unicast сообщением - DHCPOFFER. В этом сообщении содержатся настройки которые предлагает сервер. Т.к. в сети может быть несколько DHCP-серверов, клиент чаще всего выбирает первые DHCPOFFER который он получит.
3. Клиент посылает unicast DHCPREQUEST, с просьбой выдать ему настройку.
4. Сервер подтверждает выдачу настроек сообщением DHCPACK. Это сообщение содержит параметры IP конфигурации.

* Так как применяется широковещательный DHCPDISCOVER - в подсети куда он был послан может не оказаться DHCP-сервера. Для решения таких проблем выделяется DHCP-relay агент. Он пересылает запрос в другую сеть DHCP-серверу.
* DHCP-сервер может быть настроен для раздачи адресов принадлежащих разным сетям. Сервер определяет нужноую подсеть по DHCP-запросу и выделяет адрес из нужного пула адресов. Один такой пул адресов называется областью (scope).
* Когда клиент получил настройки, они выдаются на определенное время аренды. Оно называется - lease time.
* Некоторым устройствам в сети (например, серверам нужны постоянные адреса) для этого на DHCP сервере делается привязка конкретного MAC адреса к IP адресу.

**Proxy Servers**

Некоторые клиенты настроены так, что при обращении к Интернет их пакеты пересылаются прокси-серверу. Прокси-сервер получает запрос клиента, и от имени клиента прокси посылает запрос в интернет. Когда прокси-сервер получает ответ из Интернета, он пересылает его клиенту.



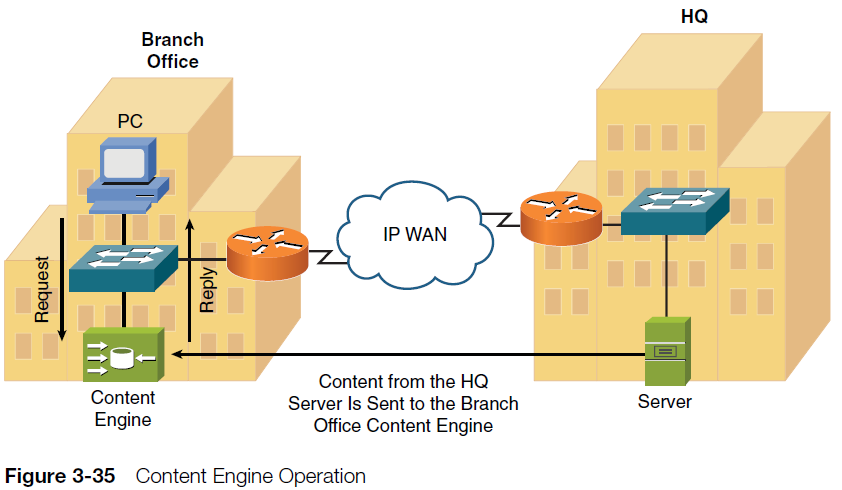
Вопрос: Зачем так делать, какие плюсы ?

1. Безопасность. Так как все запросы идут через прокси сервер, IP адреса внутри доверенной сети скрыты для Интернета.
2. Кэширование. Сохраняет пропускную способность.
3. Фильтрация контента. Ограничивает доступ клиентов к определенным URL (социальные сети, мессенджеры, игры и т.д.).

**Content Engines**

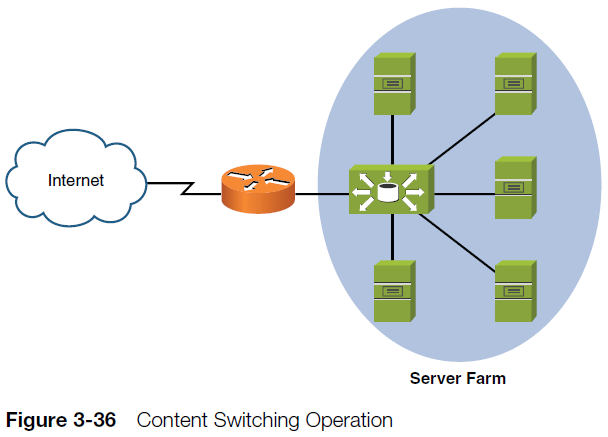
Кешировать контент может прокси сервер, но в некоторых сетях используются выделенные приборы для выполнения кеширования контента - их называют - caching engines, content engines

В примере ниже, в офисе филиала используется content engine, который кеширует контент с сервера главного офиса. Это значительно разгружает WAN-link.



**Content Switches**

Рассмотри серверную ферму представленную на рисунке 3-36. Для компаний с большим присутствием в Интернет (поисковые системы, интернет-магазины, социальные сети) мало одного сервера, т.к. он будет перегружен запросами из Интернет. Чтобы разгрузить один сервер испоьзуется балансировщик нагрузки распределяющий входящие запросы по различным серверам в серверной ферме, каждый из которых поддерживает идентичную копию данных и приложений.



* Главное достоинство балансировщика нагрузки - это масштабируемость. По мере увеличения нагрузки - новые сервера могут добавлять к группе серверов через балансировщик нагрузки.
* Если серверу нужно обслуживание, он может сообщить балансировщику нагрузки о выходе из “оборота”. Затем, после обслуживания, может снова быть добавленым к группе серверов.

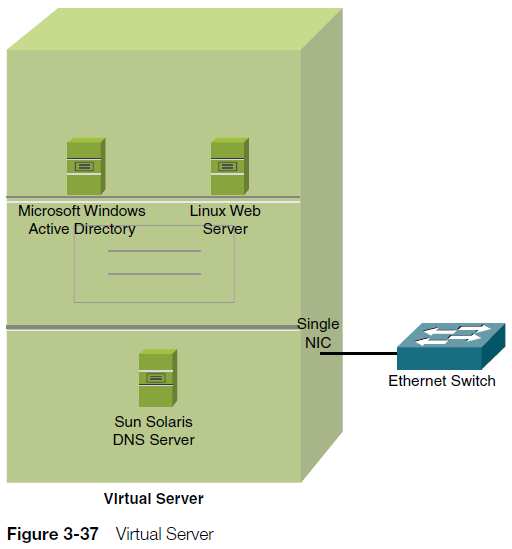
**ВИРТУАЛЬНЫЕ СЕТЕВЫЕ УСТРОЙСТВА**

Изменяется парадигма. Зачастую теперь организациям не нужен свой дата-центр (с большой системой кондиционирования и т.п.) соерджащий множество физических серверов каждый из которых предоставляет свой сервис (например e-mail, dns-server, MS Active Directory).

**Виртуальные сервера**

Мощи современных серверов высокого класса достаточно, чтобы один физический сервер выполнят задачи множества независимых серверов. С появлением виртуализации, несколько серверов (на которых могут работать разные ОС) могут быть запущены как экземпляры виртуального сервера на одном физическом устройстве.

Например один сервер высокого класса может содержать экземпляр Microsoft Windows Server с Active Directody, экземпляр Lunix в роли корпоративного веб-сервера, экземпляр Sun Solaris UNIX сервера с ролью DNS-сервера. В примере сервер имеет один сетевой интерфейс, но многие виртуальные серверные платформы имеют несколько сетевых интерфейсов. Наличие нескольких сетевых интерфейсов обеспечивает повышенную пропускную способность и балансировку нагрузки.

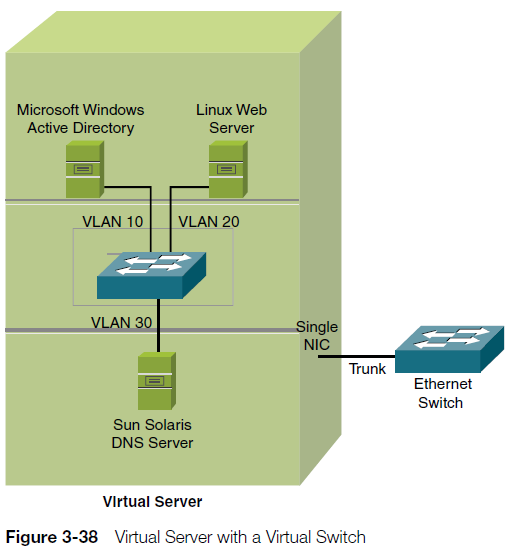


**Виртуальные коммутаторы**

В передыдущем примере все виртуальные сервера находятся в одной подсети это может иметь свои негативные последствия по QoS и безопасности. Если бы сервера были на отдельных физических машинах, они могли быть подключены к разным порта коммутатора, и назначены в разные VLAN, и каждый сервер был бы в своем широковещательном домене.

Некоторые платформы виртуализации поддерживают кроме виртуальных серверов, виртуальные коммутаторы.

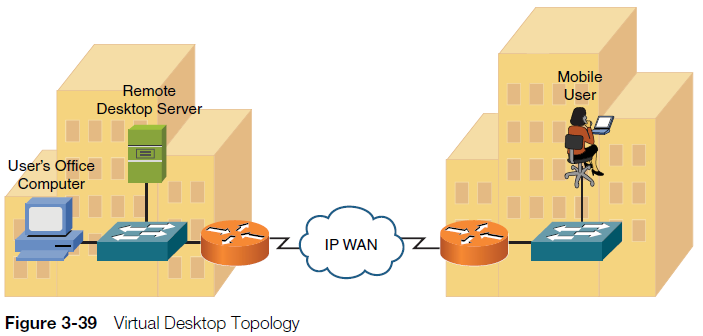
Пример с виртуальным коммутатором. Сервер делит логически инстансы на VLAN’ы, и весь тегирированный трафик путешествует по транковому порту подсоединенному к физическому коммутатору.



**Виртуальные рабочие столы**

Еще одной технологие виртуализации являются виртуальные рабочие столы. Сегодня пользователи мобильны и им нужен доступ к данным своего офисного компьютера из множества различных мест.

Например пользователь может быть в аэропорту используя свой смартфон, ему нужен доступ к документу с которым он работал на компьютере в офисе. При использовании виртуального рабочего стола, данные пользователя находятся в центре обработки данных, а не на жестком диске рабочего компьютера. Поэтому, предоставляя учетные данные, безопасное соединение может быть установлено между централизованным хранилищем пользовательских данных и устройством пользователя.



**Другие решения виртуализации**

Все описанные ранее технологии виртуализации находятся на корпоративном месте (т.е. on-site), но некоторые провайдеры предлагают off-site технологии. В частности если клиент поставщика услуг не хочет\не может содержать свой дата-центр, технологии виртуализации располагаются в дата-центре провайдера услуг, и клиенту выставляется счет на основе шаблонов использования.

Такие сервисы называются NaaS - Network as a Service, подразумевая что сеть может быть предоставлена как услуга со стороны провайдера услуг (например как телефонная сеть, или подключение через ISP к интернету). ! Не путать с application servoce provider который предоставляет ПО как услугу: Software as a Service (SaaS).

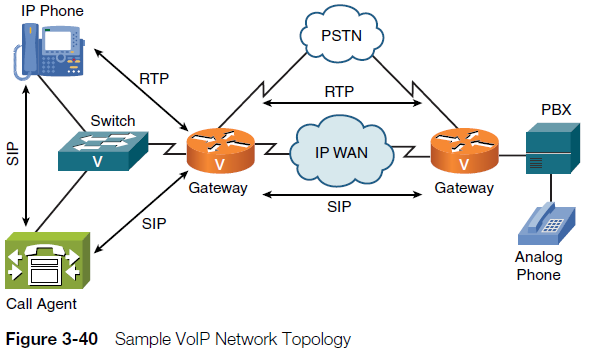
Так же как и аутсорсинг функций сетей передачи данных с NaaS, копроративная телефония так же может быть на оутсорсе. Многие компании эксплуатируют свои собственные телефонные системы PBX (Private Branch Echange). На смену этому приходят virtual PBX. Такие решения чаще всего используют Voice over IP (VoIP) решения - где голос помещается в пакеты данных и передается по сети передачи данных. Обычно провайдер услуг предоставляет шлюзы IP-телефонии для конвертации из телефонных систем в свои virtual PBX.

**VoIP ПРОТОКОЛЫ и КОМПОНЕНТЫ**

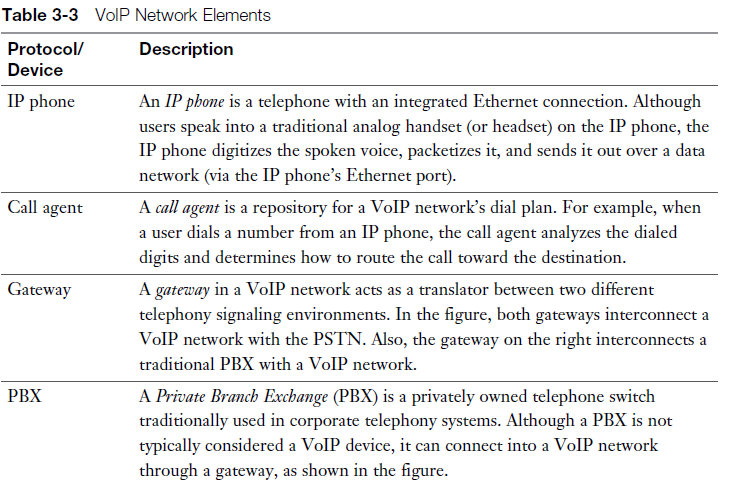
“Голос по IP” (VoIP) оцифровывает устную речь в пакеты и передает эти пакеты через сети передачи данных.

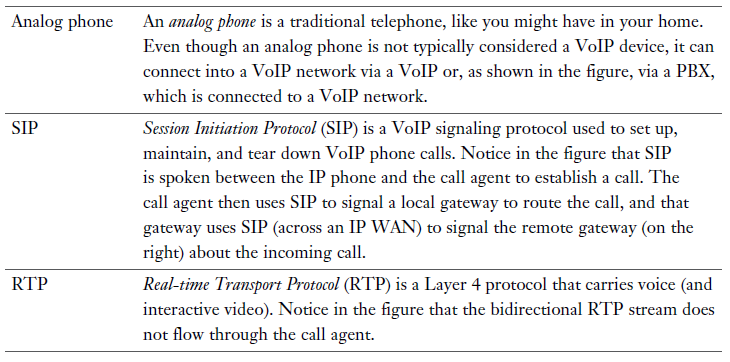
Это позволяет голосу, данным и видео быть переданными по одной среде передачи данных. VoIP дает не только уменьшение эксплуатационных расходов, а также дополнительные сервисы - видеоконференцсвязь, ПО для определения доступности и еще много того чег онет в традиционной телефонии.

Ниже представлен пример простой VoIP сети.



Ниже представлена таблица с описанием основных VoIP протоколов и устройств.





**Выводы**

* Рассмотрели различные среды передачи данных включая коаксиальный кабель, экранированую и неэкранированную витую пару, волоконную-оптику, и беспроводные технологии
* Рассмотрели различное инфраструктурное оборудование сетей: концентраторы, мосты, коммутаторы, многоуровневые коммутаторы и маршрутизаторы.
* Рассмотрели специализированное сетевое оборудование: VPN-концентраторы, межсетевые экраны, DNS-сервера, DHCP-сервера, прокси-сервера, content engines, балансировщики нагрузки.
* Рассмотреле технологии виртуализации: виртуальный сервер, виртуальный коммутатор, виртуальный рабочий стол, виртуальный PBX.
* Рассмотрели VoIP технологии и описали основные протоколы и аппаратное обеспечение необходимое для построение VoIP сети.

**Ключевые термины (знать\понимать):**

coaxial cable, twisted-pair cable, shielded twisted-pair cable, unshielded twisted-pair cable, electromagnetic interference (EMI), plenum, multimode fiber (MMF), singlemode fiber (SMF), 66 block, 110 block, hub, switch, router, multilayer switch, firewall, Domain Name System (DNS) server, Dynamic Host Configuration Protocol (DHCP), proxy server, content engine, content switch, virtual server, virtual switch, virtual desktop, on-site, off-site, Network as a Service (NaaS), virtual PBX, Session Initiation Protocol (SIP), Real-time Transport Protocol (RTP)