Глава 10.
Уровень приложений

Основы сетевых технологий

Cisco | Networking Academy®
Mind Wide Open™

Содержание

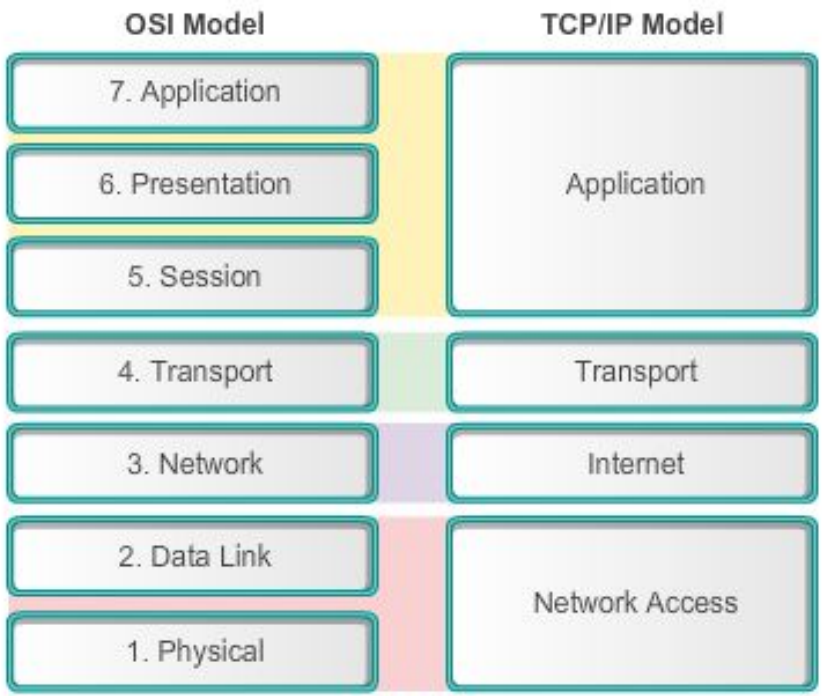
- Протоколы уровня приложений
- Сервисы уровня приложений
  - •DNS
  - • HTTP
  - • Сервис e-mail и протоколы smtp/pop
  - • FTP
  - • DHCP
- Сообщение которое сможет прочесть каждый

Пересмотр моделей OSI и TCP/IP

Уровень приложений является самым верхним уровнем в моделях OSI и TCP/IP.

Функциональные возможности протоколов уровня приложений TCP/IP охватывают примерно три верхних уровня модели OSI: уровень приложения, уровень представления и сеансовый уровень.

Уровни 5, 6 и 7 модели OSI используются в качестве опорных для разработчиков и поставщиков прикладного программного обеспечения, чтобы создавать продукты, которым требуется доступ к сети (например, веб-браузеры).



Уровень приложений

Уровень приложений ближе всех находится к конечному пользователю.

На этом уровне обеспечивается взаимодействие между приложениями, используемыми для обмена данными, и базовой сетью, по которой передаются сообщения.

Протоколы уровня приложений используются для обмена данными между программами, исполняемыми на узле-источнике и узле-получателе.

Уровень представления

На уровне представления задействованы три основные функции:

- кодирование и преобразование данных уровня приложений;
- сжатие данных;
- шифрование данных для передачи и их расшифровка после получения по адресу назначения.

На уровне представления форматируются данные уровня приложений и устанавливаются стандарты форматов файлов

Сеансовый уровень

Сеансовый уровень

- Функции сеансового уровня создают и обеспечивают диалоги между исходными и конечными приложениями
- Сеансовый уровень обрабатывает обмен данными для запуска диалогов, поддержания их активности и перезапуска сеансов

Протоколы уровня приложений TCP/IP

Протоколы прикладного уровня TCP/IP определяют форматы и управляют данными, необходимыми для многих распространённых функций обмена данными через Интернет.

Протоколы уровня приложений TCP/IP

- **Протокол преобразования имён интернет-доменов** (DNS): используется для преобразования интернет-доменов в IP-адреса
- **Telnet**: протокол эмуляции терминала; используется для предоставления удалённого доступа к серверам и сетевым устройствам
- **Протокол загрузки** (BOOTP): предшественник протокола DHCP; сетевой протокол, используемый для получения данных IP-адреса во время загрузки
- **Протокол динамической конфигурации узла** (DHCP): используется для назначения узлу IP-адреса, маски подсети, шлюза по умолчанию и DNS-сервера
- **Протокол передачи гипертекста** (HTTP): используется для передачи файлов, составляющих веб-страницы в Интернете

Протоколы уровня приложений TCP/IP

- **Протокол передачи файлов** (FTP): используется для интерактивной передачи файлов между системами
- **Простой протокол передачи файлов** (TFTP): используется для активной передачи файлов без установления соединения
- **Простой протокол передачи эл. почты** (SMTP): используется для передачи сообщений и вложений электронной почты
- **Почтовый протокол** (POP): используется почтовыми клиентами для получения электронной почты с удалённого сервера
- **Протокол доступа к сообщениям в Интернете** (IMAP): ещё один протокол получения электронной почты

Одноранговые сети

В P2P-сети два компьютера (или более двух) подключаются между собой по сети и могут открывать доступ к своим ресурсам (например, к принтерам и файлам) без использования выделенного сервера.

Каждое подключённое к сети оконечное устройство (также называемое одноранговым) может выполнять функции как сервера, так и клиента.

Один компьютер может играть роль сервера для одной операции, одновременно выступая в роли клиента для других операций.

Функции клиента и сервера устанавливаются по запросу.

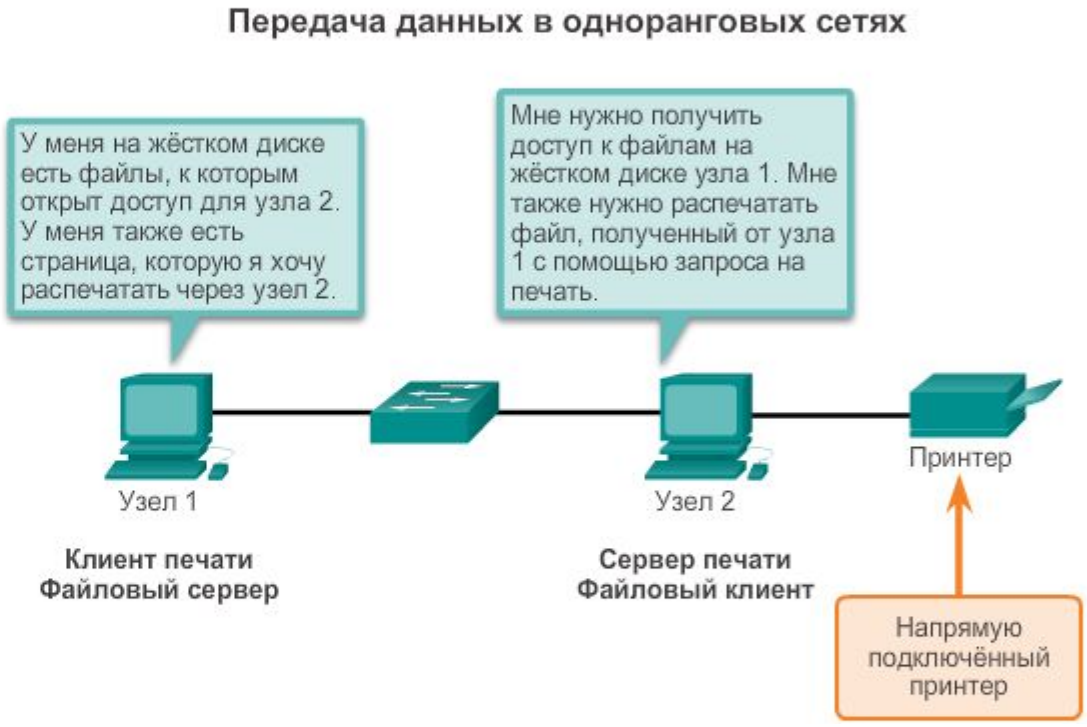
Сетевая модель P2P состоит двух частей: P2P-сетей и P2P-приложений. Обе части имеют сходные функции, но принцип их работы отличается

Одноранговые сети

Примером может служить простая домашняя сеть с двумя компьютерами, как показано на рисунке.

В этом примере к Узлу2 принтер подключён напрямую через USB, а к принтеру открыт общий доступ по сети, чтобы Узел1 мог его использовать.

Оба устройства считаются равными в рамках обмена данными



Роли клиента и сервера устанавливаются на время запроса.

Одноранговые приложения

В некоторых P2P-приложениях используется гибридная система, где общий доступ к ресурсам децентрализован, а индексы, указывающие на местоположения ресурсов, хранятся в центральном каталоге.

В гибридной системе каждый узел обращается к серверу индексации, чтобы получить местоположение ресурса, который хранится на другом узле.

Сервер индексации также может помогать узлам подключаться друг к другу, но после установки соединения узлы обмениваются данными без дополнительного обращения к серверу.

P2P-сети

В P2P-сети использование ресурсов в сети децентрализовано.

Большинство современных операционных систем поддерживают открытие общего доступа к файлам и принтерам без дополнительного серверного программного обеспечения.

Недостатком одноранговых сетей является сложность применения политики безопасности и доступа.

Учётные записи пользователей и права доступа должны отдельно настраиваться на каждом устройстве.

Модель типа «клиент-сервер»

В модели типа «клиент-сервер» устройство, запрашивающее информацию, называется клиентом, а устройство, которое отвечает на данный запрос, — сервером.

Клиент начинает обмен данными, отправляя запрос на получение данных с сервера, который в ответ отправляет один или несколько потоков данных клиенту.

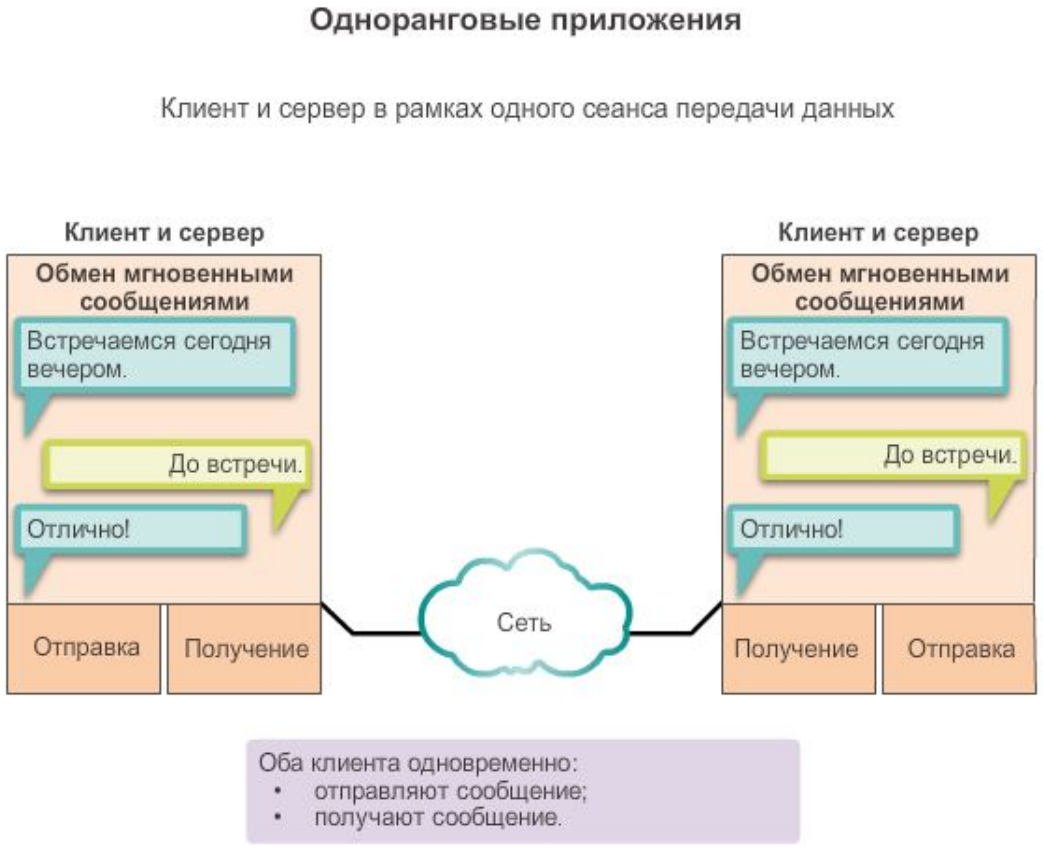
Протоколы уровня приложений описывают формат запросов и ответов между клиентами и серверами.

В дополнение к фактической передаче данных для этого обмена данными также может потребоваться аутентификация пользователей и идентификация передаваемых файлов данных.

Одноранговые приложения

- Одноранговое приложение (P2P) позволяет устройству выступать в роли как клиента, так и сервера в пределах одного сеанса связи, как показано на рисунке.

- И клиент, и сервер могут инициировать обмен данными и считаются равноправными в рамках процесса обмена данными



Типичные приложения P2P

С помощью приложений P2P все компьютеры в сети, где функционирует приложение, могут выступать в роли клиента или сервера для других компьютеров в сети, где функционирует это приложение

К типичным P2P-приложениям относятся:

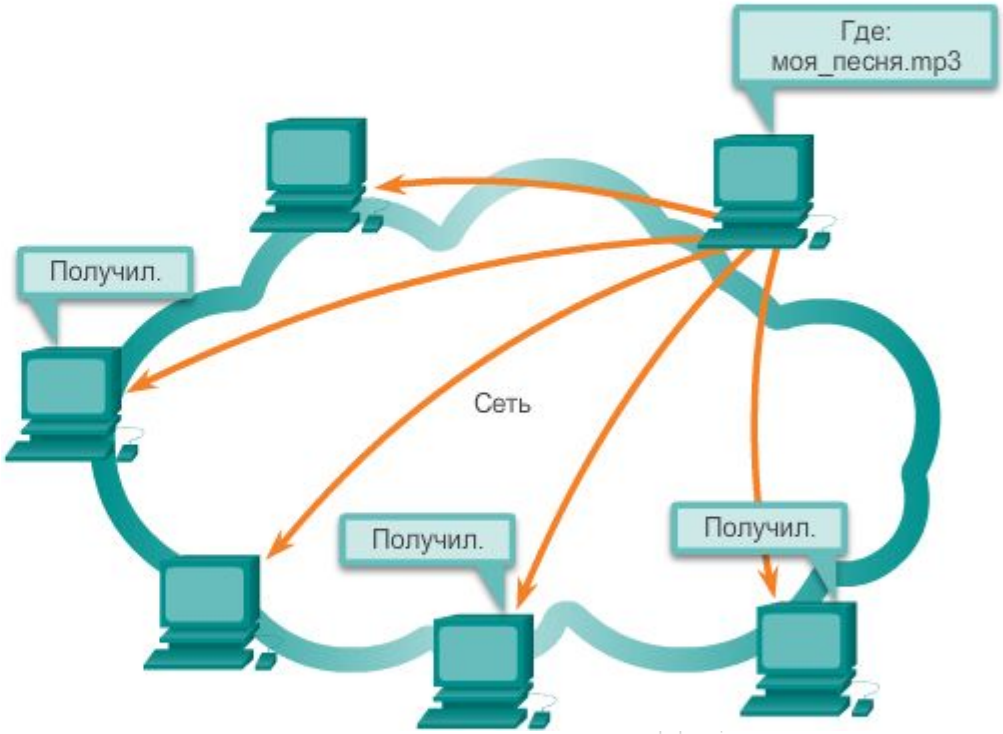
- eDonkey
- eMule
- Shareaza
- BitTorrent
- Bitcoin и многие другие

- Некоторые приложения 2P2 разработаны на основе протокола Gnutella, который позволяет пользователям обмениваться файлами, хранящимися на жёстком диске

Типичные приложения P2P

Как показано на рисунке, клиентское программное обеспечение, совместимое с протоколом Gnutella, позволяет пользователям подключаться к сервисам Gnutella через Интернет, а также находить и использовать ресурсы, доступ к которым был открыт другими узлами Gnutella.

Для доступа к сети Gnutella существует множество клиентских приложений, в том числе: Gnucleus, BearShare, Morpheus, LimeWire, WinMX и XoloX.



Модель типа «клиент-сервер»

Пример сети типа «клиент-сервер» является сервис электронной почты для отправки, получения и хранения сообщений электронной почты. клиенту.

Почтовый клиент на домашнем компьютере отправляет запрос серверу электронной почты на получение списка новых сообщений.

Сервер отвечает, отправляя запрошенное сообщение клиенту.

Модель типа «клиент-сервер»

Клиент может передавать файл на сервер для хранения.

Как показано на рисунке:

- передача данных от клиента к серверу называется отправкой

- в направлении от сервера к клиенту — загрузкой.

Протокол передачи гипертекста или язык разметки

После ввода в адресной строке веб-адреса или унифицированного указателя ресурса (URL-адрес) веб-браузер устанавливает соединение по протоколу HTTP с веб-сервисом, запущенным на сервере.

Пример

URL: http://www.cisco.com/index.html

Протокол передачи гипертекста или язык разметки

Пример

URL: http://www.cisco.com/index.html

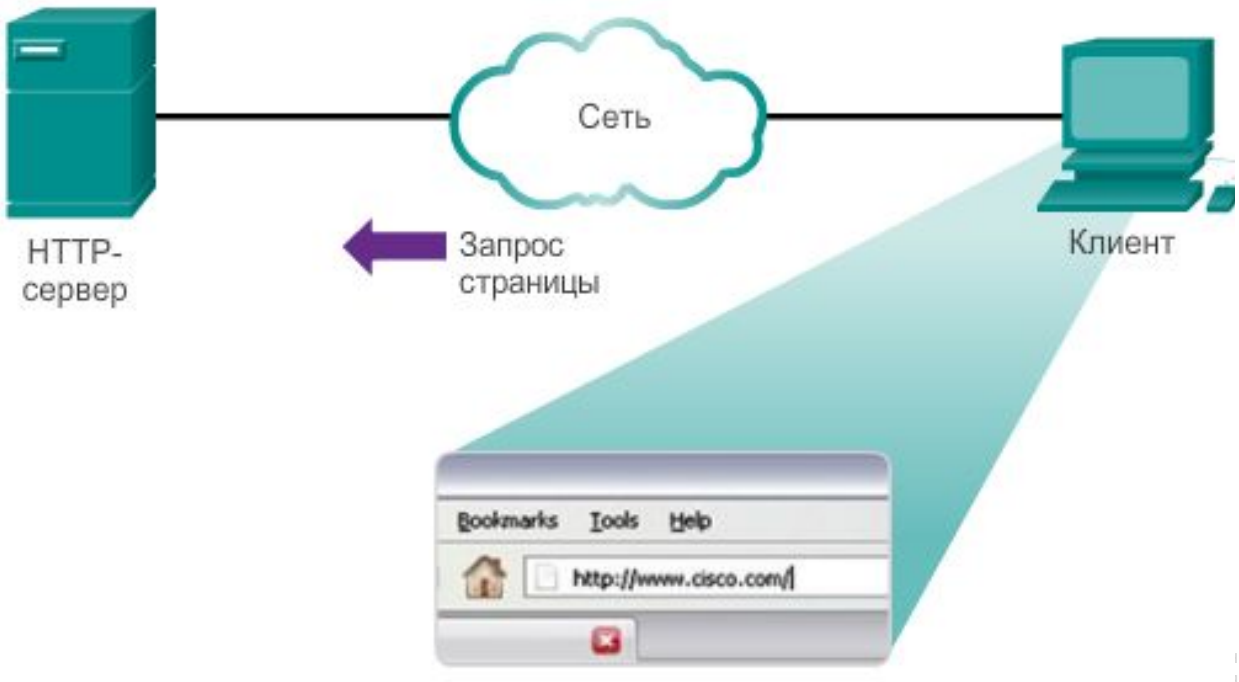
На первом шаге браузер интерпретирует три части URL-адреса:

1. http (протокол или схема);
2. www.cisco.com (имя сервера);
3. index.html (название конкретного запрашиваемого файла).

Протокол передачи гипертекста или язык разметки

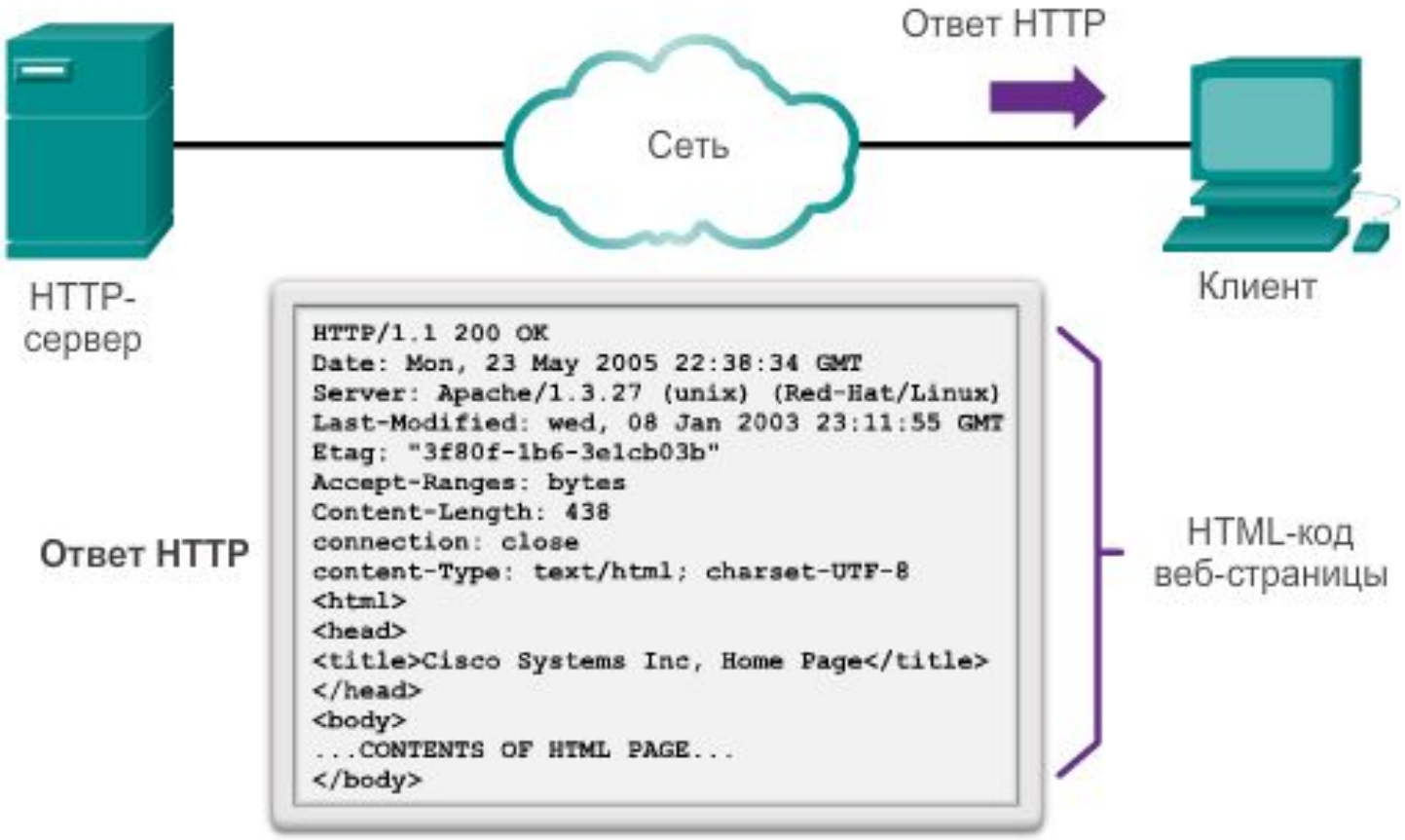
На шаге 2 браузер проверяет имя сервера www.cisco.com, чтобы преобразовать его в числовой адрес, по которому устанавливается подключение к серверу.

Согласно требованиями HTTP-протокола браузер отправляет GET-запрос серверу и запрашивает файл index.html.



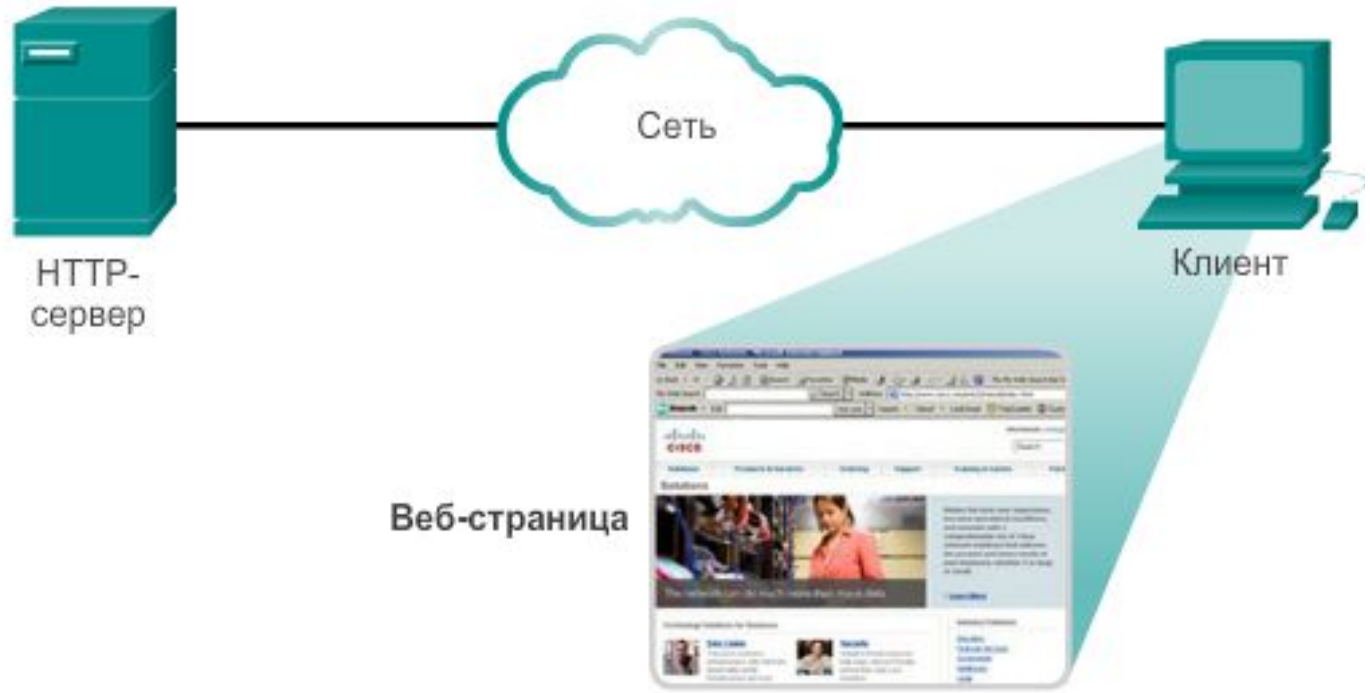
Протокол передачи гипертекста или язык разметки

На третьем шаге сервер отправляет браузеру HTML-код этой веб-страницы.



Протокол передачи гипертекста или язык разметки

На 4 шаге браузер декодирует HTML-код и форматирует страницу в окне браузера.



HTTP и HTTPS

Разработаны для публикации и получения HTML-страниц

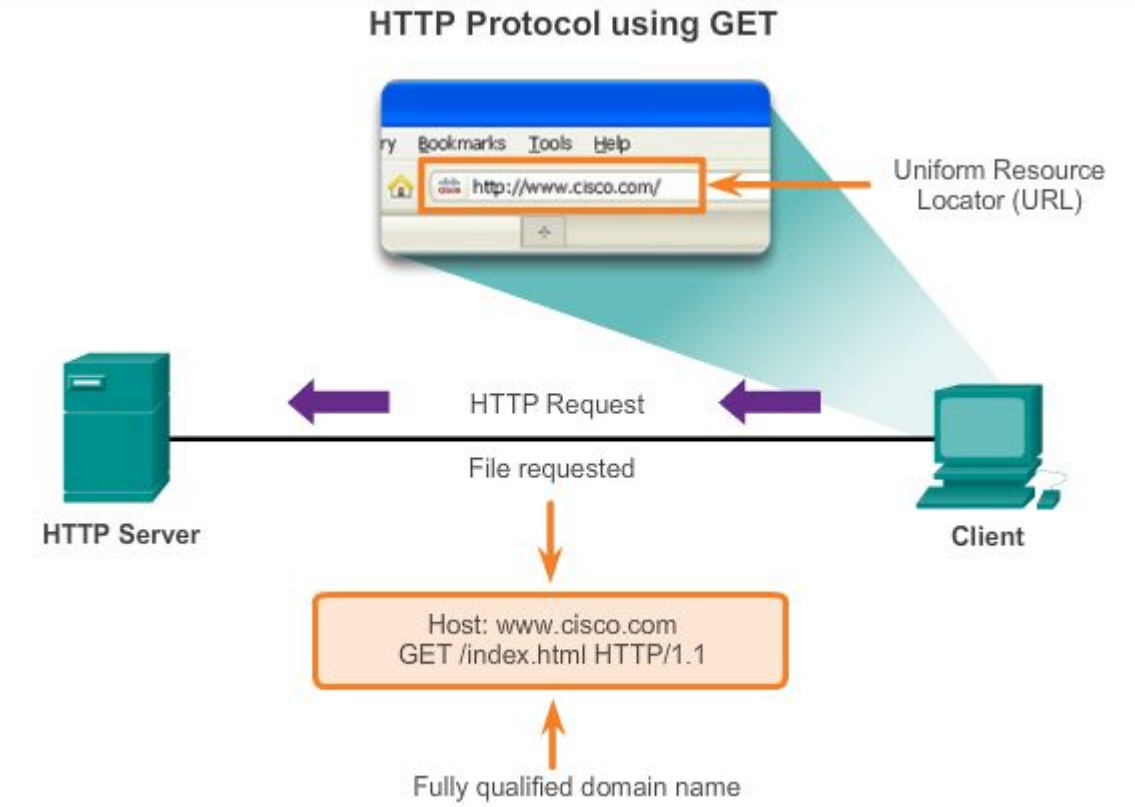
Используются для передачи данных

Определяют протокол «запрос-отклик»

Протокол HTTP основан на механизме «запрос-отклик».

Когда клиент (обычно веб-браузер) отправляет запрос веб-серверу, протокол HTTP определяет типы сообщений, используемые для этого взаимодействия.

HTTP и HTTPS



Три стандартных типа сообщений: GET, POST и PUT

- GET — запрос клиента на предоставление данных
- POST и PUT используются для отправки сообщений, передающих данные на веб-сервер

HTTP и HTTPS

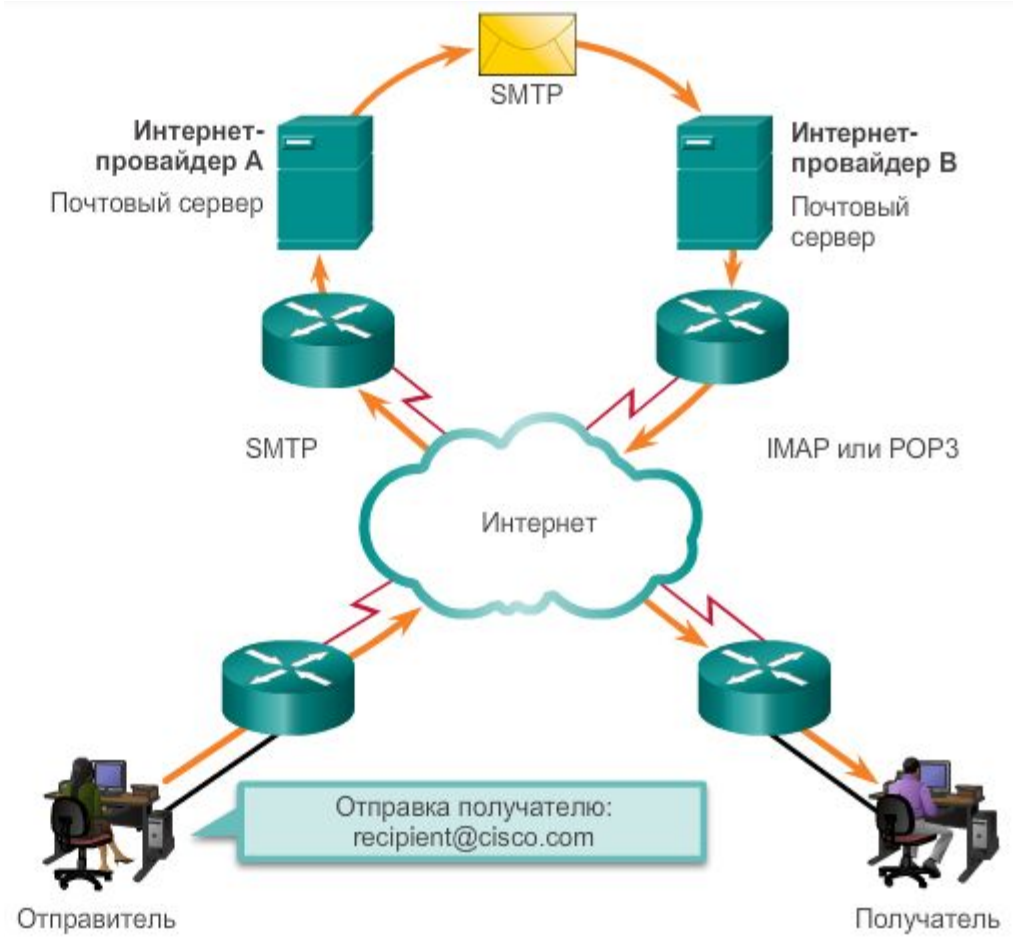
Протокол HTTP не является безопасным, сообщения запросов передаются серверу открытым текстом, который может быть перехвачен и прочитан.

Для защищённого двустороннего обмена данными с веб-серверами в Интернете используется протокол HTTPS.

HTTPS позволяет использовать аутентификацию и шифрование для защиты данных, пересылаемых между клиентом и сервером.

В протоколах HTTPS и HTTP процессы «клиент запрашивает — сервер отвечает» аналогичны, но поток данных шифруется посредством SSL перед началом передачи по сети.

SMTP, POP и IMAP



SMTP, POP и IMAP

Электронная почта — это набор средств для доставки, хранения и поиска электронных сообщений в сети.

Сообщения электронной почты хранятся на серверах электронной почты в базах данных.

Клиенты электронной почты для отправки и получения сообщений обращаются к серверам электронной почты.

Серверы электронной почты взаимодействуют с другими серверами электронной почты для обмена сообщениями между доменами.

Почтовый клиент не соединяется непосредственно с другим почтовым клиентом для отправки сообщения.

Оба клиента должны доверить транспортировку сообщений серверу электронной почты.

SMTP, POP и IMAP

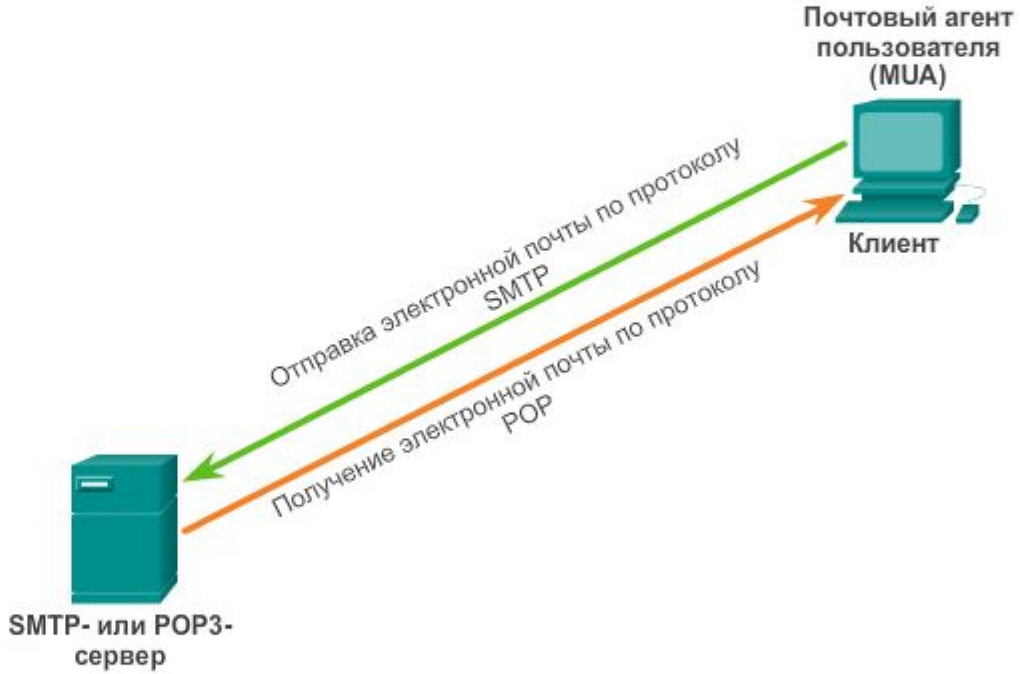
Для работы с электронной почтой используются три отдельных протокола: SMTP, POP и IMAP.

SMTP: отправка сообщений с клиента или с сервера

POP: получение сообщение от почтового сервера

IMAP: протокол доступа к сообщениям в Интернете

Почтовый клиент предоставляет функции обоих протоколов в рамках одного приложения



SMTP, POP и IMAP

Протокол SMTP используется для надёжной и эффективной передачи электронной почты.

Для нормальной работы SMTP-приложения требуется, чтобы почтовые сообщения были правильно отформатированы

Процессы SMTP должны быть запущены на клиенте и на сервере

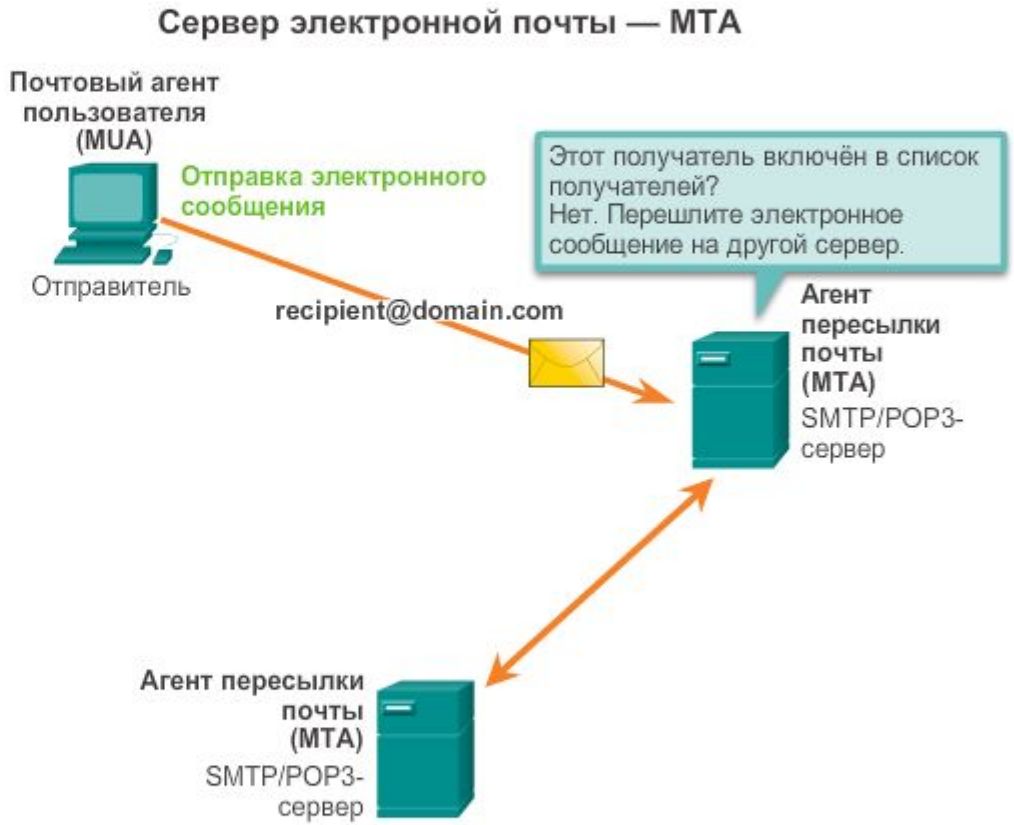
В заголовке сообщения должны быть указаны адреса электронной почты получателя и отправителя в правильном формате

Используется порт 25

SMTP, POP и IMAP

Когда клиент отправляет сообщение электронной почты, процесс SMTP-клиента подключается к процессу SMTP-сервера по известному 25.

Установив соединение, клиент пытается отправить по нему сообщение электронной почты серверу.



SMTP, POP и IMAP

Когда сервер получает сообщение, он помещает его в очередь сообщений локальной учётной записи или пересылает другому серверу, выполнив такой же процесс установки SMTP-соединения.

Целевой сервер электронной почты в момент доставки сообщения может оказаться недоступен или перегружен.

На этот случай в SMTP предусмотрено временное хранение сообщений с последующей повторной отправкой.

Периодически сервер проверяет очередь сообщений и пытается отправить их повторно.

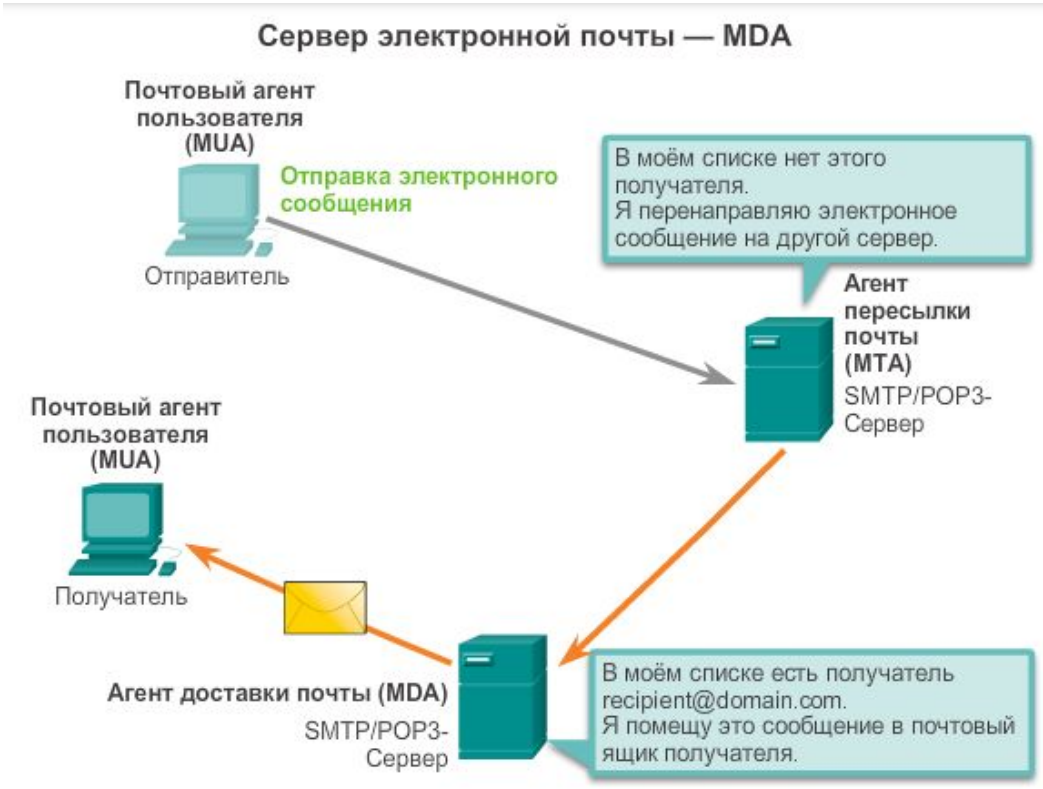
Если сообщение не удаётся доставить в течение установленного времени, оно возвращается отправителю с уведомлением о невозможности доставки.

SMTP, POP и IMAP

Протокол POP позволяет рабочим станциям получать сообщения электронной почты с серверов электронной почты.

При использовании протокола POP сообщения загружаются клиентом с сервера и удаляются на сервере.

Сетевой сервис POP на сервере пассивно ожидает запросов подключения клиентов к TCP-порту 110.



Служба доменных имён

Протокол DNS служит для преобразования читаемых имён, используемых для ссылки на сетевые ресурсы.

Протокол DNS определяет автоматизированный сервис, который сопоставляет имена ресурсов с соответствующими числовыми сетевыми адресами.

В этом протоколе описывается формат для запросов, ответов и самих данных.

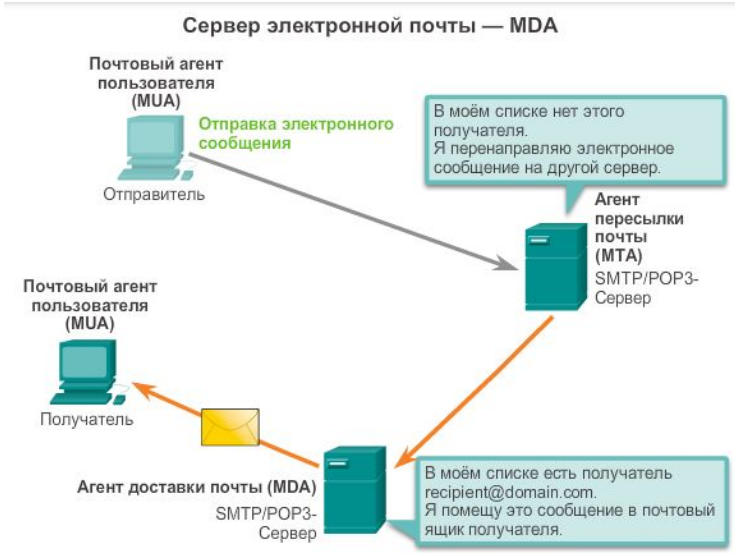
При обмене данными по протоколу DNS используется единый формат, который называется сообщением.

Такой формат сообщения используется для всех типов запросов клиента и ответов сервера, сообщений об ошибках и передачи записей ресурсов между серверами.

SMTP, POP и IMAP

Принцип работы протокола POP:

- Для использования этого сетевого сервиса клиент запрашивает TCP-соединение с сервером.
- После установки соединения сервер POP посылает приветствие.
- Затем клиент и сервер POP обмениваются командами и откликами, пока подключение не будет закрыто или прервано.



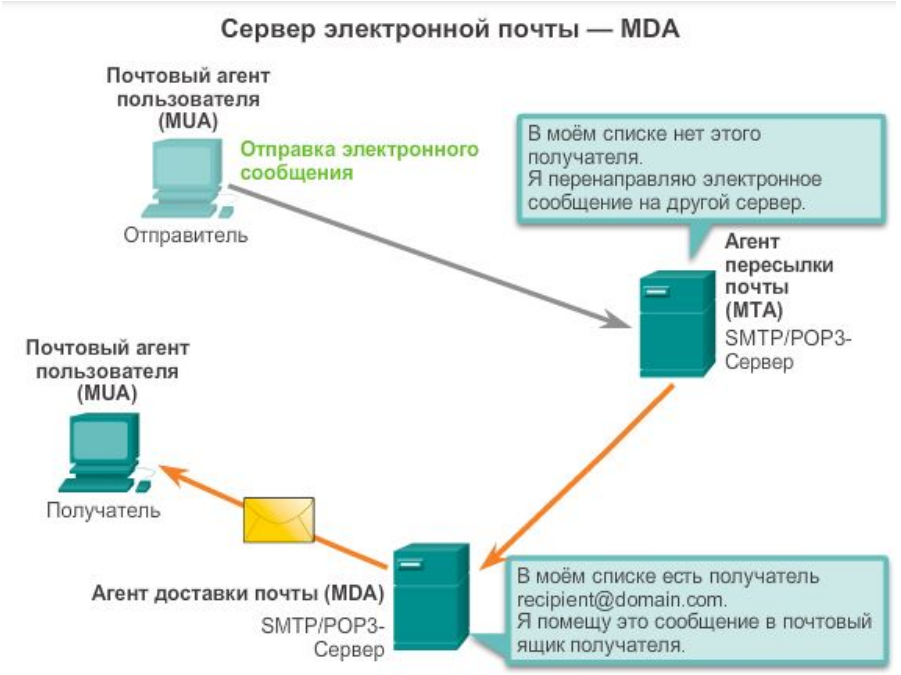
IMAP

При подключении пользователя к серверу IMAP в клиентское приложение загружаются только копии сообщений.

Исходные сообщения остаются на сервере до тех пор, пока они не будут удалены вручную.

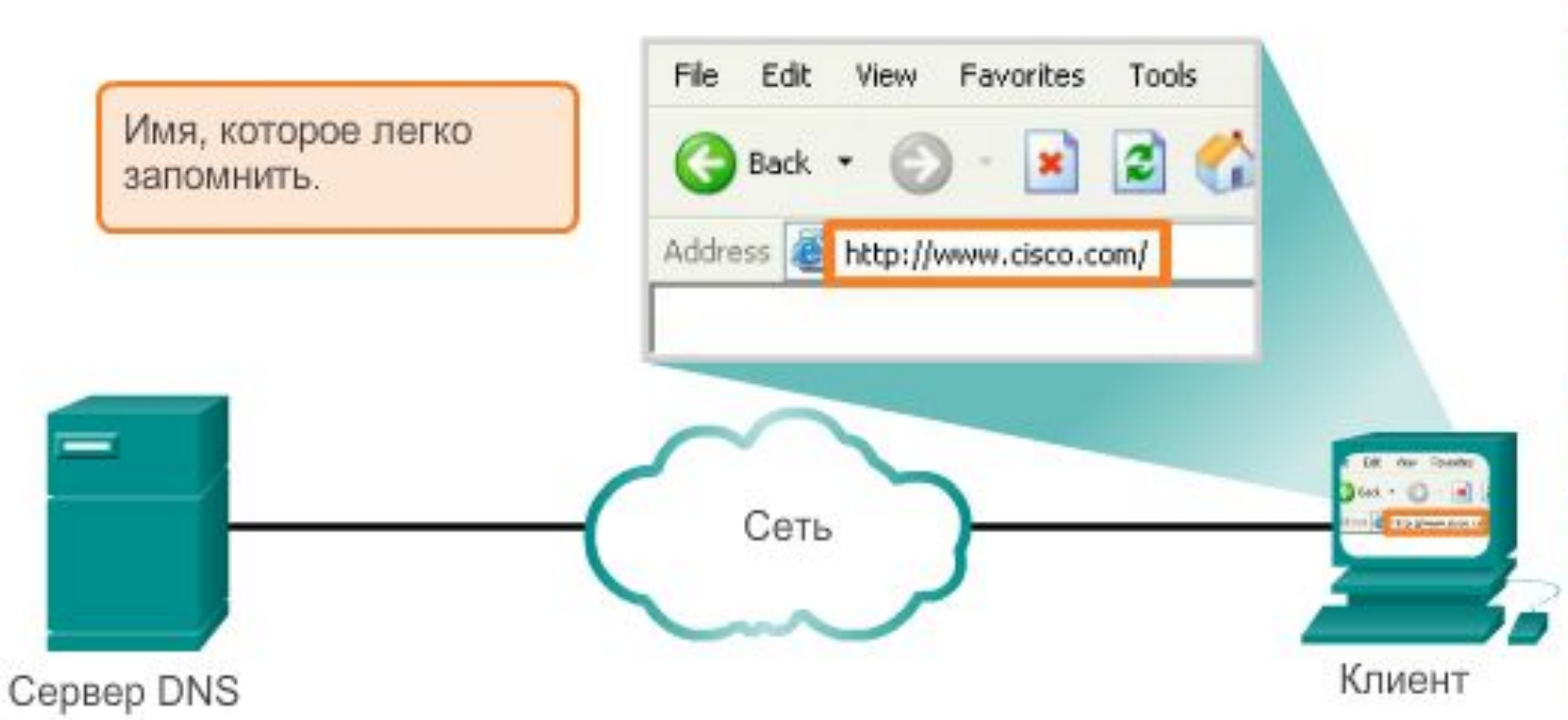
Пользователи просматривают копии сообщений в клиентах электронной почты.

Если пользователь решает удалить сообщение, оно синхронно удаляется из клиента и с сервера.



Служба доменных имён

Принцип работы DNS - Шаг 1

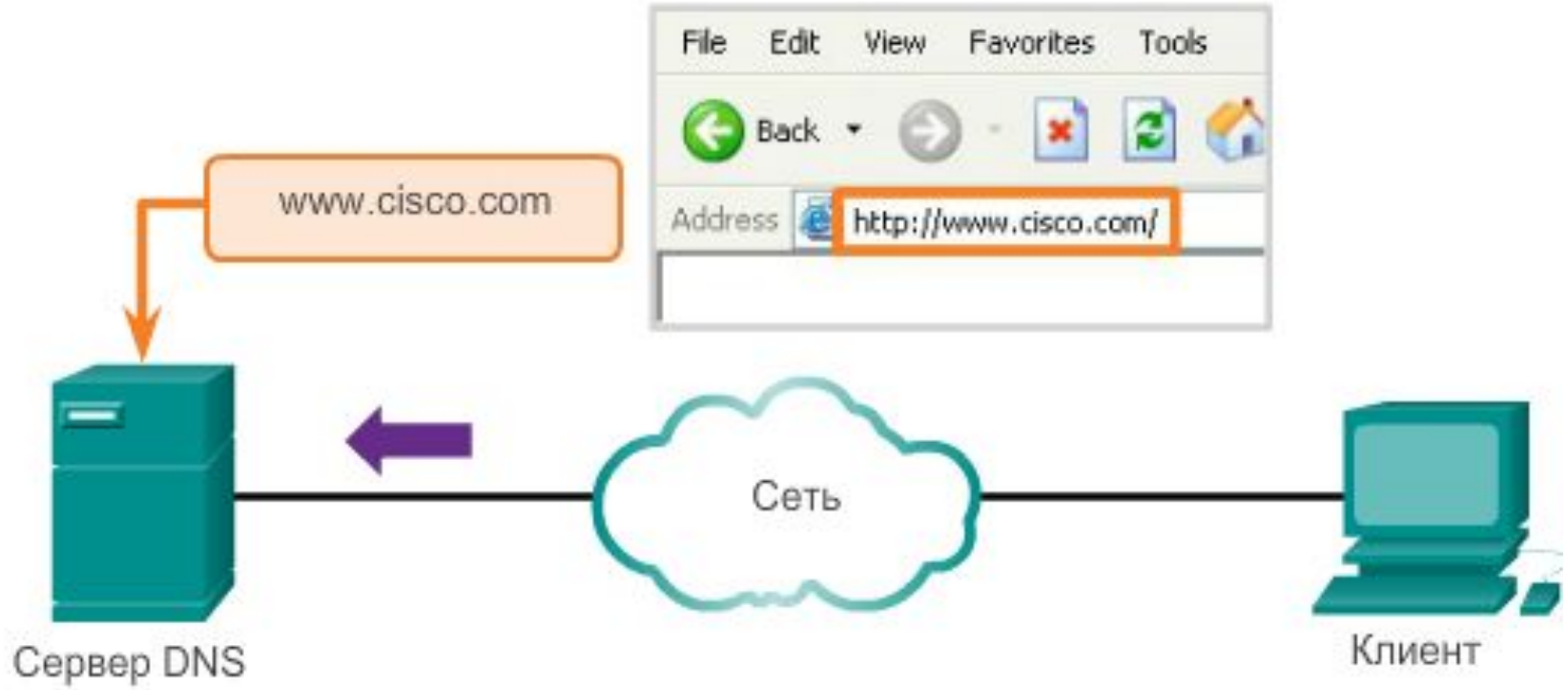


Служба доменных имён

Принцип работы DNS - Шаг 2

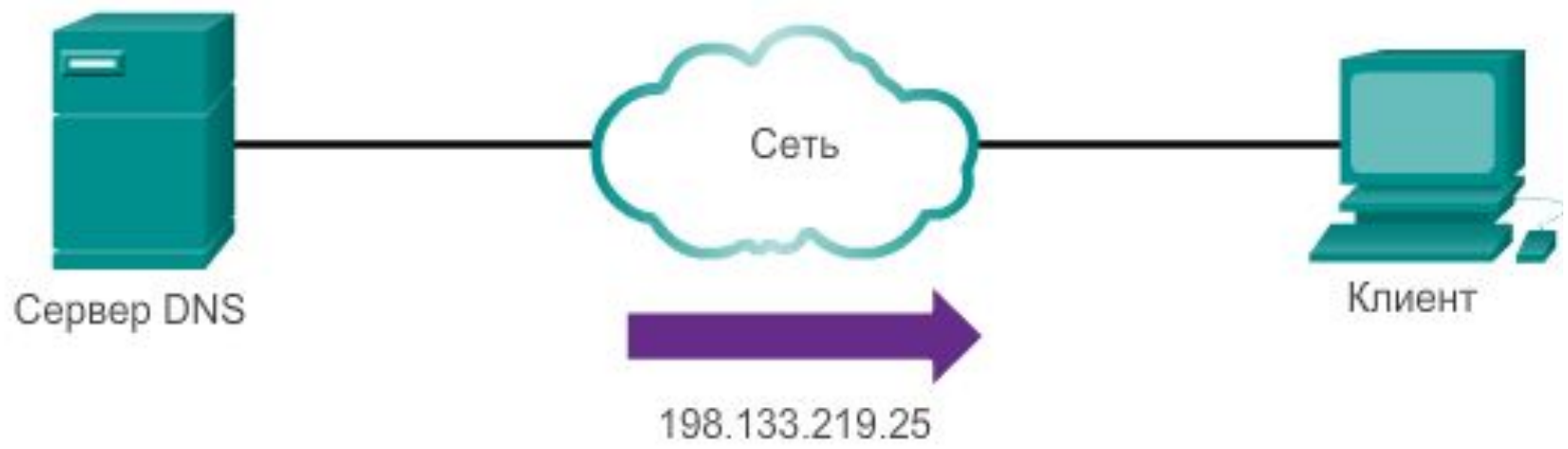
Служба доменных имён

Принцип работы DNS - Шаг 3



Служба доменных имён

Принцип работы DNS - Шаг 4



Служба доменных имён

Принцип работы DNS - Шаг 5

Формат сообщений DNS

- DNS-сервер обеспечивает разрешение имён с помощью программы для поддержки сервера имён доменов (BIND)
- На DNS-сервер хранятся различные типы записей ресурсов, используемые для преобразования имён
- Они содержат имя, адрес и тип записи
- Типы записей:
  - **A** — адрес оконечного устройства
  - **NS** — доверенный сервер имён
  - **CNAME** — каноническое имя псевдонима; используется в том случае, когда для нескольких служб существует один сетевой адрес, но для каждой службы используется отдельная запись в DNS
  - **MX** — запись обмена сообщениями; сопоставляет доменное имя со списком серверов обмена сообщениями

Формат сообщений DNS

Когда клиент выполняет запрос, процесс BIND сервера сначала ищет это имя в своих записях, чтобы разрешить его.

Если имя не удалось разрешить по локальным записям, сервер обращается к другим серверам для разрешения имени

Сервер временно хранит числовой адрес, соответствующий имени в кэш-памяти

Команда Windows ipconfig /displaydns отображает все кэшируемые DNS

Иерархия DNS

Примеры доменов верхнего уровня:

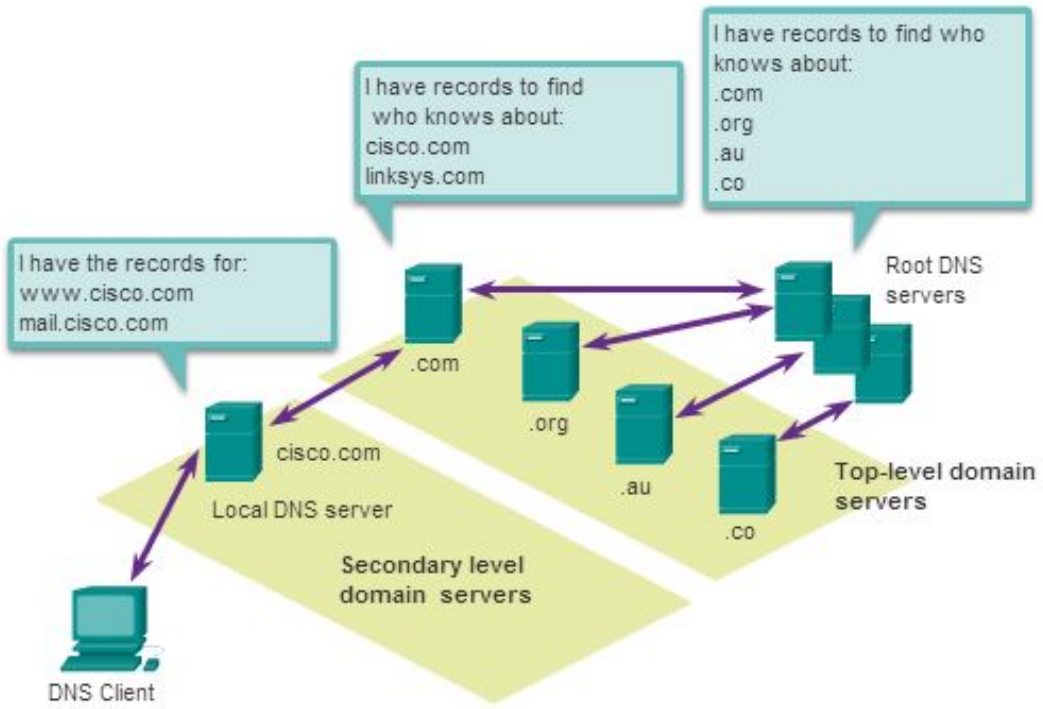
.au — Австралия

.co — Колумбия

.com — коммерческое или промышленное предприятие

.jp — Япония

.org — некоммерческая организация

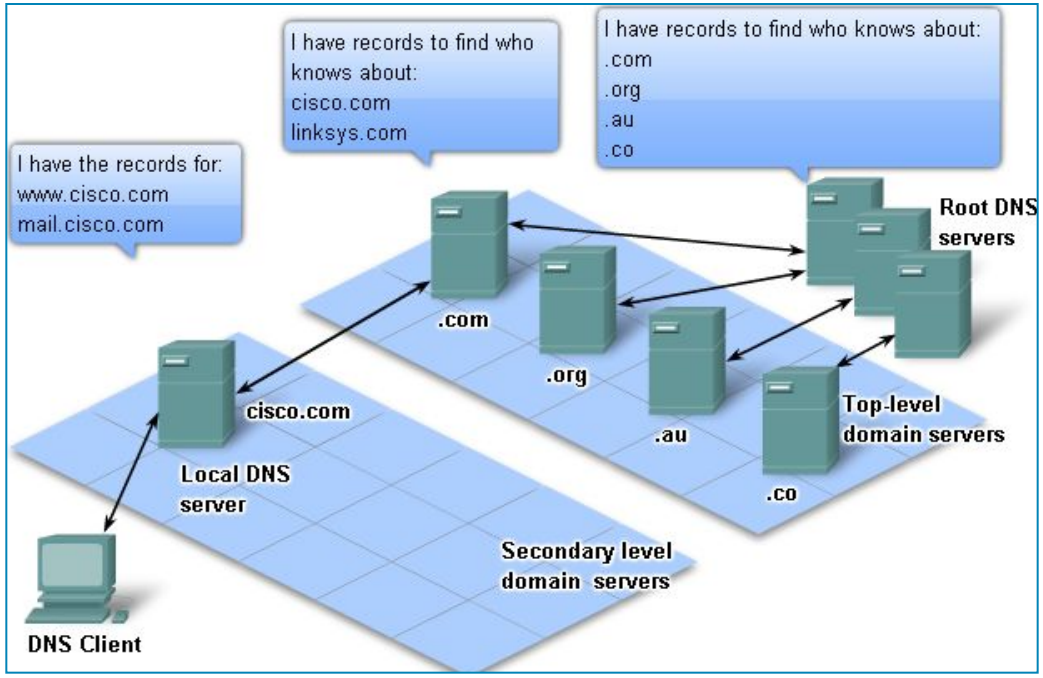


A hierarchy of DNS servers contains the resource records that match names with addresses.

Пример работы DNS

Предположим, мы набрали в браузере адрес cisco.netacad.net.

Браузер спрашивает у сервера DNS: «какой IP-адрес у cisco.netacad.net»?

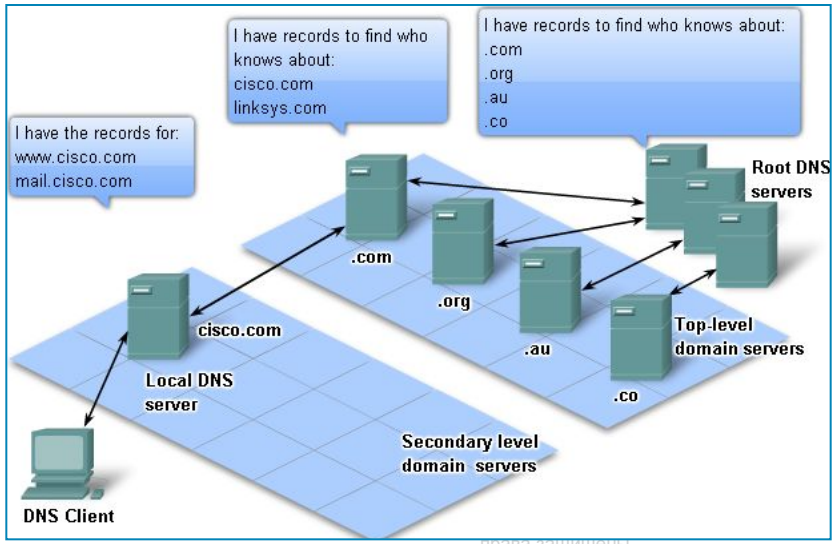


Пример работы DNS

При этом сервер DNS может ничего не знать не только о запрошенном имени, но даже обо всём домене netacad.net.

В этом случае имеет место *рекурсия*: сервер обращается к *корневому серверу* — например, 198.41.0.4.

Этот сервер сообщает — «У меня нет информации о данном адресе, но я знаю, что 204.74.112.1 поддерживает доменную зону net.»



Пример работы DNS

Тогда сервер DNS направляет свой запрос к 204.74.112.1, но тот отвечает «У меня нет информации о данном сервере, но я знаю, что 207.142.131.234 поддерживает доменную зону netacad.net.»

Наконец, тот же запрос отправляется к третьему DNS-серверу (который является авторитетным сервером для зоны netacad.net), и получает ответ — IP-адрес, который и возвращает клиенту — браузеру.

Запрос на определение имени обычно не идёт дальше *кеша DNS*, который помнит (ограниченное время) ответы на запросы, проходившие через него ранее.

Пример работы DNS

В данном случае при разрешении имени, то есть в процессе поиска IP по имени:

- браузер отправил известному ему DNS-серверу т.н. *рекурсивный запрос* — в ответ на такой тип запроса сервер обязан вернуть «готовый результат», то есть IP-адрес, либо сообщить об ошибке;
- а сам DNS-сервер, получивший запрос от клиента, последовательно отправлял *итеративные запросы*, на которые получал от других DNS-серверов уточняющие ответы, пока не получил авторитетный ответ от сервера, ответственного за запрошенную зону

Утилита nslookup

ОС позволяют использовать утилиту, которая называется «nslookup».

Утилита **nslookup** предназначена для выполнения запросов на разрешение имен в IP-адреса к DNS-серверам.

Утилита достаточно сложна и содержит свой собственный командный интерпретатор.

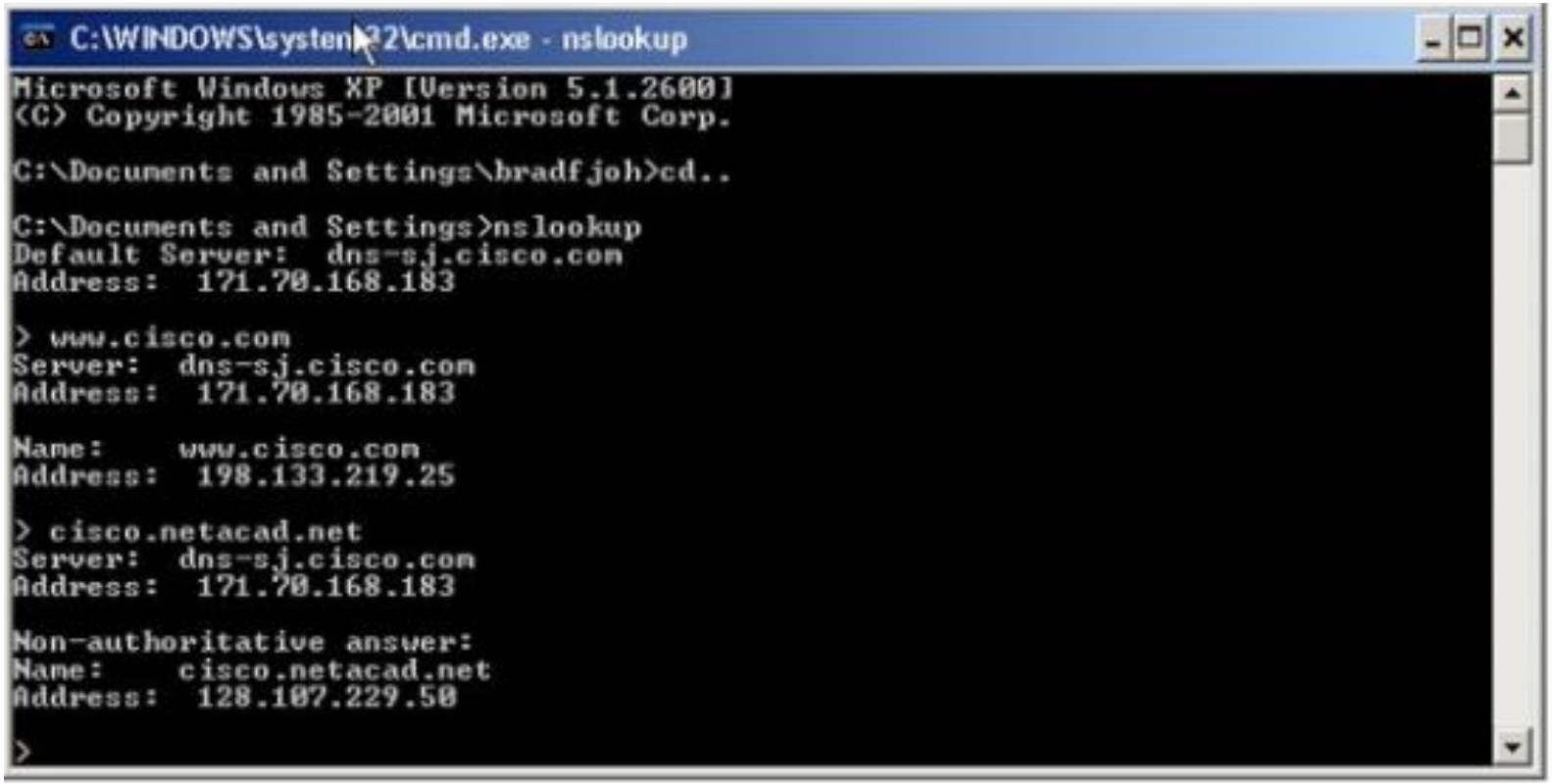
В простейшем случае утилита **nslookup** имеет следующий синтаксис:
nslookup [хост [сервер]]

где хост DNS-имя хоста, которое должно быть преобразовано в IP-адрес

сервер - Адрес DNS-сервера, который будет использоваться для разрешения имени.

Утилита nslookup

Например, при вводе команды **nslookup dns-sjk.cisco.com** утилита выдает информацию о хосте представленная на рисунке.

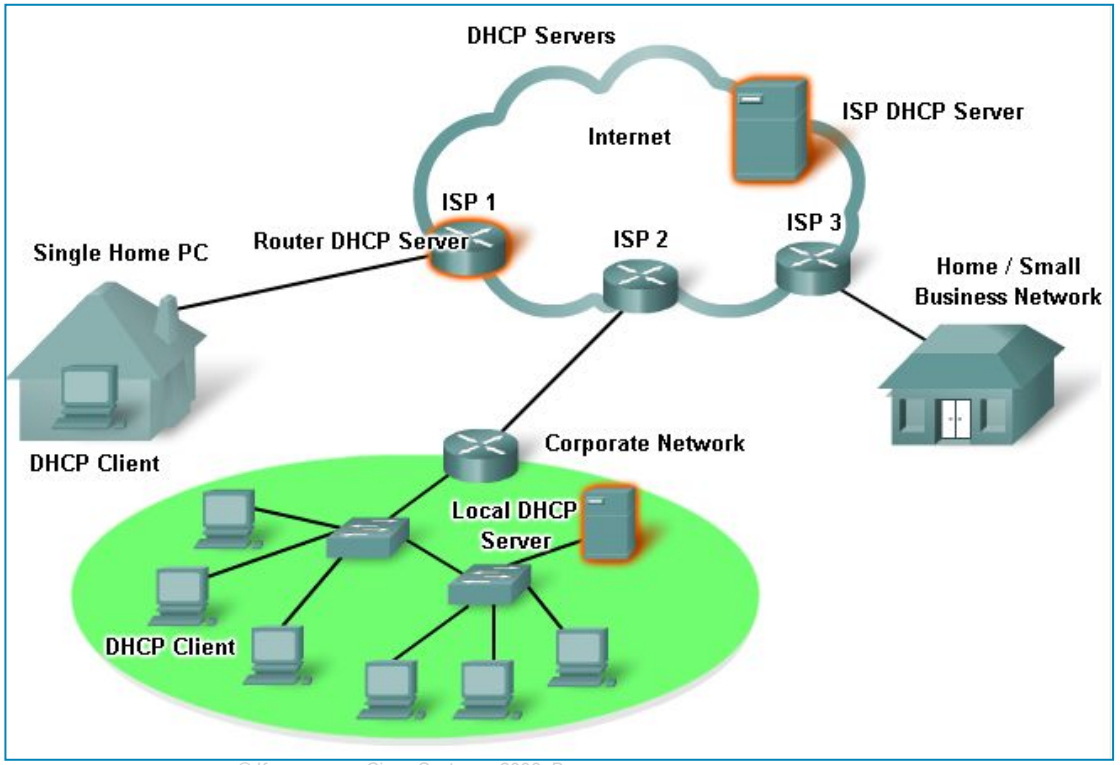


DHCP

Служба DHCP позволяет устройствам в сети получать IP-адреса и другую информацию с сервера DHCP.

Эта служба автоматизирует выделение IP-адресов, масок подсети, шлюза и других параметров IP-сети.

Перечислите все протоколы позволяющий автоматизировать настройку IP адресов на устройствах?



DHCP

DHCP разработан на базе Bootstrap Protocol (BOOTP), системы для автоматического получения информации о конфигурации BOOTP-клиентом от BOOTP-сервера при начальной загрузке.

DHCP построен по схеме клиент-сервер, где DHCP-сервер выделяет сетевые адреса и доставляет конфигурационные параметры динамически конфигурируемым ЭВМ.

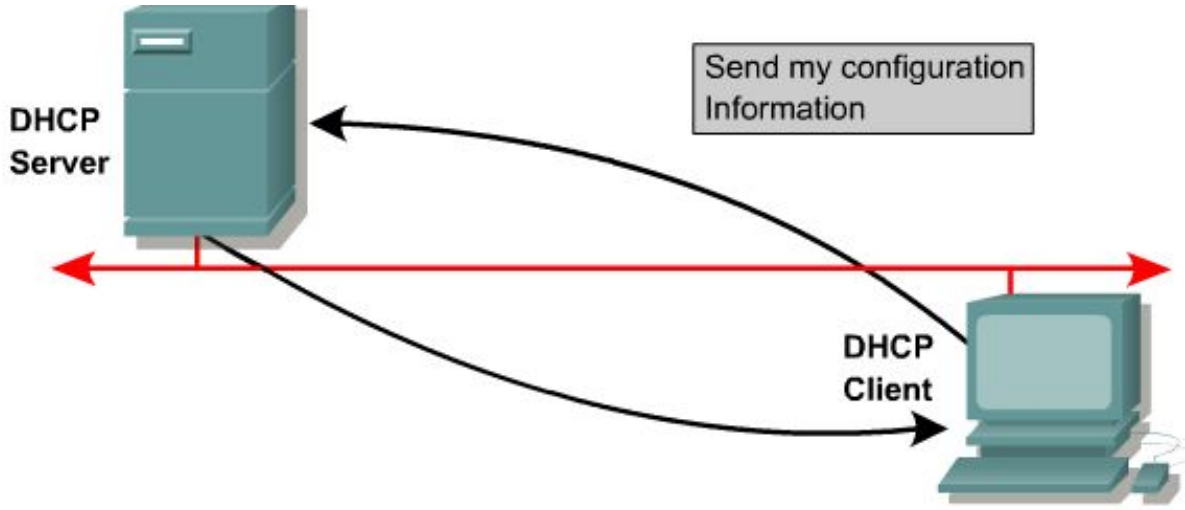
DHCP (Dynamic Host Configuration Protocol - протокол динамической конфигурации узлов) поддерживает:

- безопасную, надежную и простую конфигурацию сети TCP/IP
- препятствует возникновению конфликтов адресов
- помогает сохранять использование IP-адресов клиентов.

Механизмы назначения IP адресов в DHCP

DHCP позволяет назначать IP-адреса тремя способами:

- **Automatic Allocation** - при автоматическом назначении сервер DHCP присваивает запрашивающему клиенту DHCP постоянный IP-адрес




```
Here is Your Configuration:
• IP Address: 192.204.18.7
• Subnet Mask: 255.255.255.0
• Default Routers: 192.204.18.1, 192.204.18.3
• DNS Servers: 192.204.18.8, 192.204.18.9
• Lease Time: 5 days
```

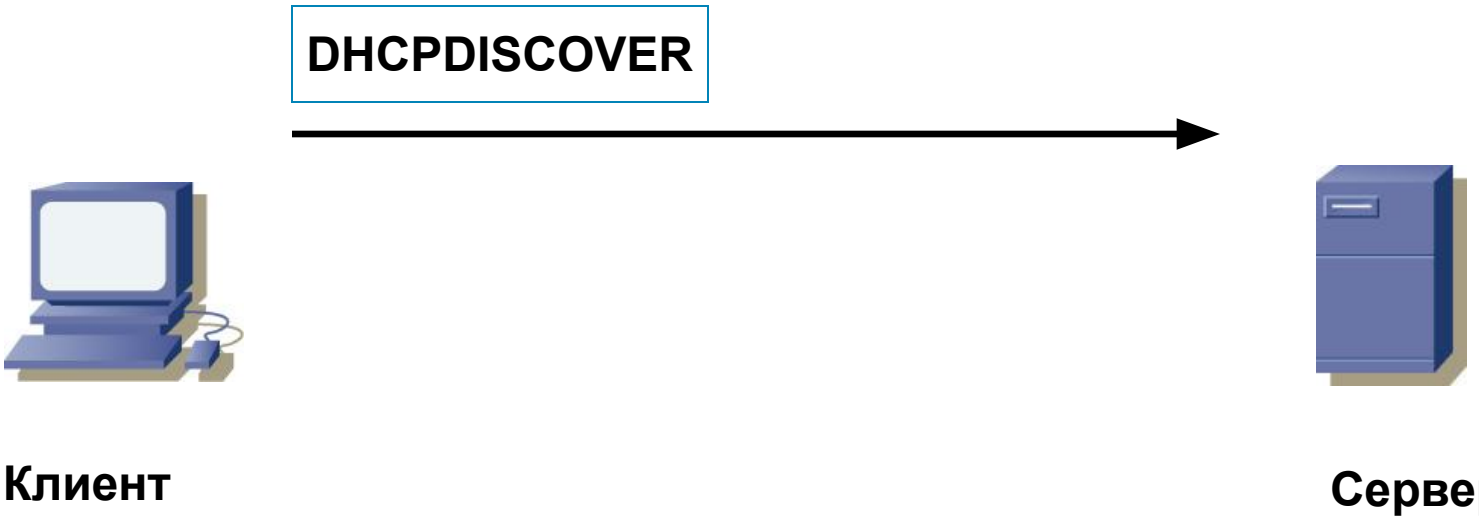
Механизмы назначения IP адресов в DHCP

- **Manual Allocation** - при назначении вручную IP-адреса выбираются администратором сети, а сервер DHCP извещает клиента о назначении.
- **Dynamic Allocation** - при динамическом назначении сервер DHCP выделяет IP-адрес на ограниченный период времени ("время аренды") или до отказа клиента от адреса - в зависимости от того, что произойдет раньше.

Динамическое назначение полезно в ситуации, когда компьютеры подключаются к сети время от времени.

При отключении от сети IP-адрес становится клиенту ненужным; он извещает об этом сервер, так что тот может переназначить адрес нуждающемуся в нем узлу.

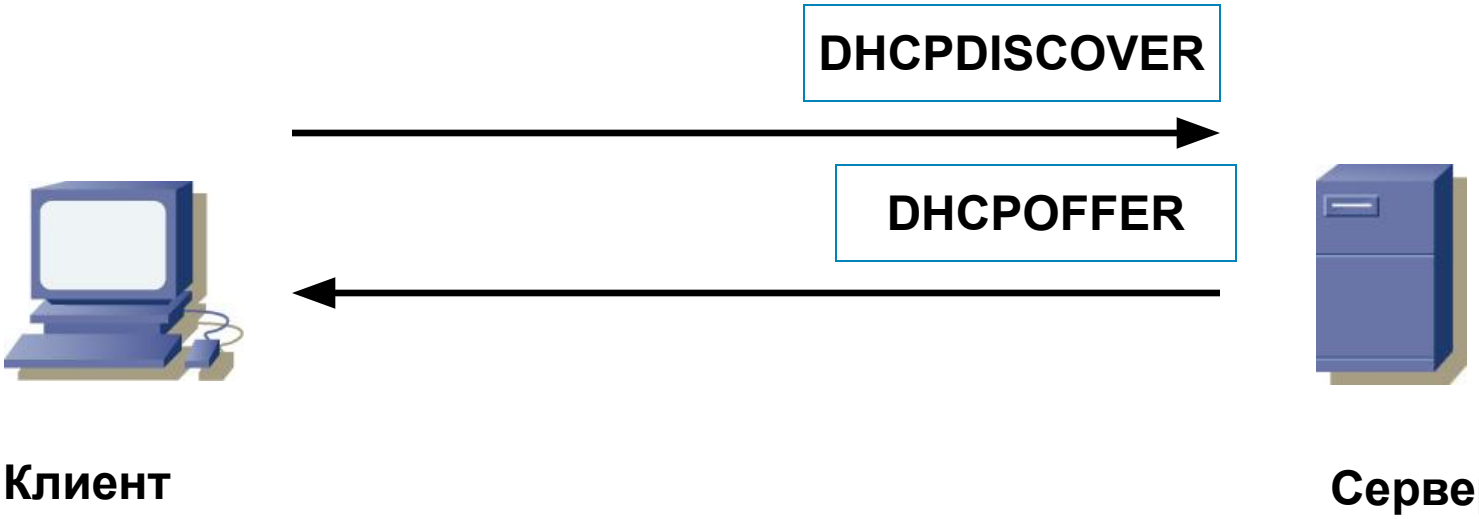
Взаимодействие клиента и сервера при выделении сетевого адреса



Клиент широковещательно пересылает сообщение DHCPDISCOVER по сети в поисках сервера.

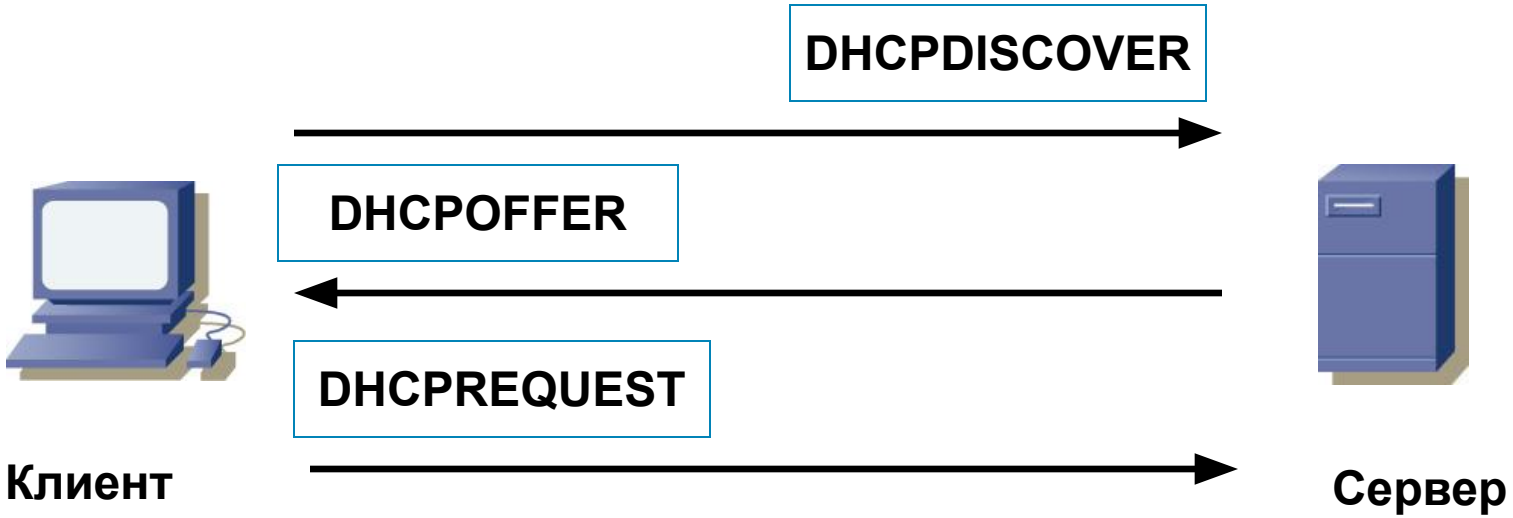
Сообщение DHCPDISCOVER может включать опции, которые предлагают значения для сетевого адреса и длительности его использования.

Взаимодействие клиента и сервера при выделении сетевого адреса



Все активные DHCP серверы посылают в ответ широковещательное сообщение - пакет DHCPOFFER, содержащий предлагаемый IP-адрес и "время аренды" (срок, в течение которого клиент может пользоваться адресом).

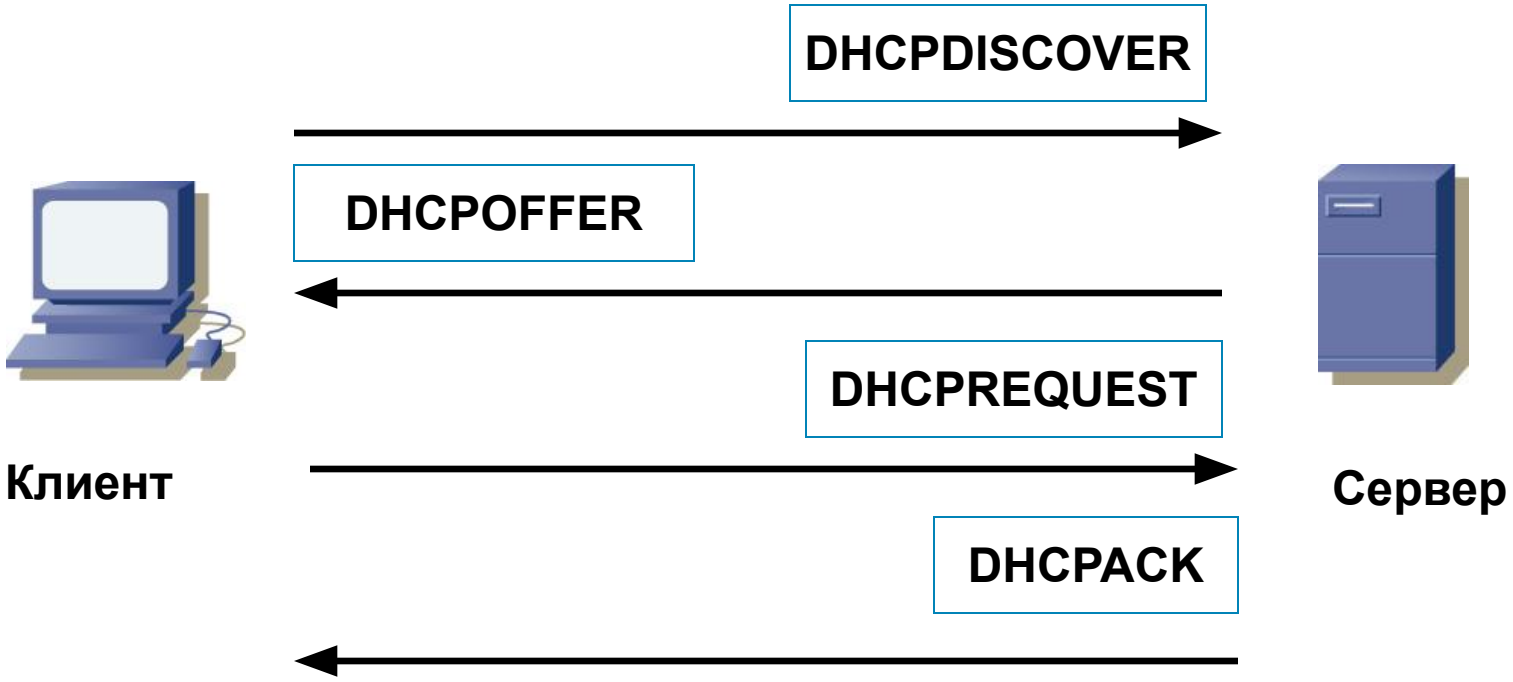
Взаимодействие клиента и сервера при выделении сетевого адреса



Клиент выбирает адрес из полученных пакетов DHCPOFFER. (Выбор клиента зависит от его назначения - например, он может выбрать адрес с наибольшим временем аренды).

Вслед за тем клиент посылает пакет DHCPREQUEST с адресом выбранного сервера.

Взаимодействие клиента и сервера при выделении сетевого адреса



Выбранный сервер посылает подтверждение (DHCPACK), и процесс согласования завершается. Пакет DHCPACK содержит оговоренные адрес и время аренды.

Взаимодействие клиента и сервера при выделении сетевого адреса

Сервер помечает выделенный адрес как занятый - до окончания срока аренды этот адрес не может быть присвоен другому клиенту.

Клиенту осталось только сконфигурировать себя в соответствии с назначенным адресом и можно приступать к работе в сети.

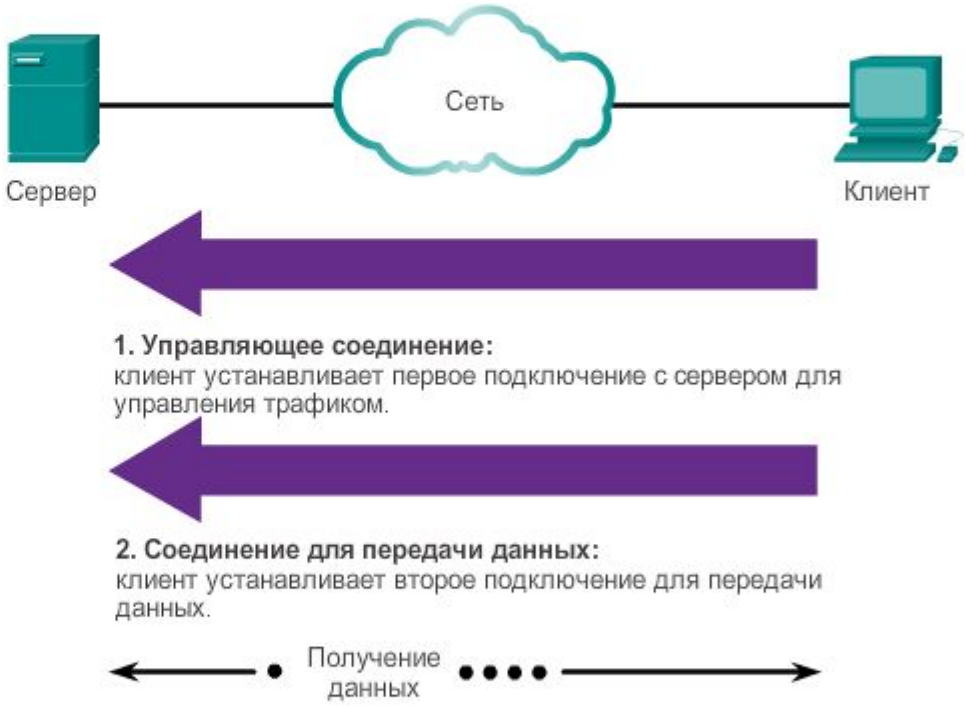
Отметим, клиент должен выбрать одно из предложений и послать в ответ пакет DHCPREQUEST с идентификатором выбранного сервера.

Другие серверы просматривают пакет DHCPREQUEST и заключают на основе идентификатора сервера, что их предложение было отвергнуто.

Протокол передачи файлов (FTP)

- FTP был разработан для передачи данных между клиентом и сервером.
- FTP-клиент — это приложение, которое запускается на компьютере, а также отправляет и принимает данные с сервера, на котором запущена служба FTP.

Для передачи данных по FTP требуется два соединения между клиентом и сервером: одно для команд и ответов, другое — для фактической передачи файлов.



Протокол передачи файлов (FTP)

Клиент устанавливает первое соединение с сервером для управления трафиком, который состоит из команд клиента и ответов сервера.

Затем клиент устанавливает второе соединение с сервером для непосредственной передачи данных.

Это подключение создаётся для каждой передачи данных.

Данные могут передаваться в любом направлении.

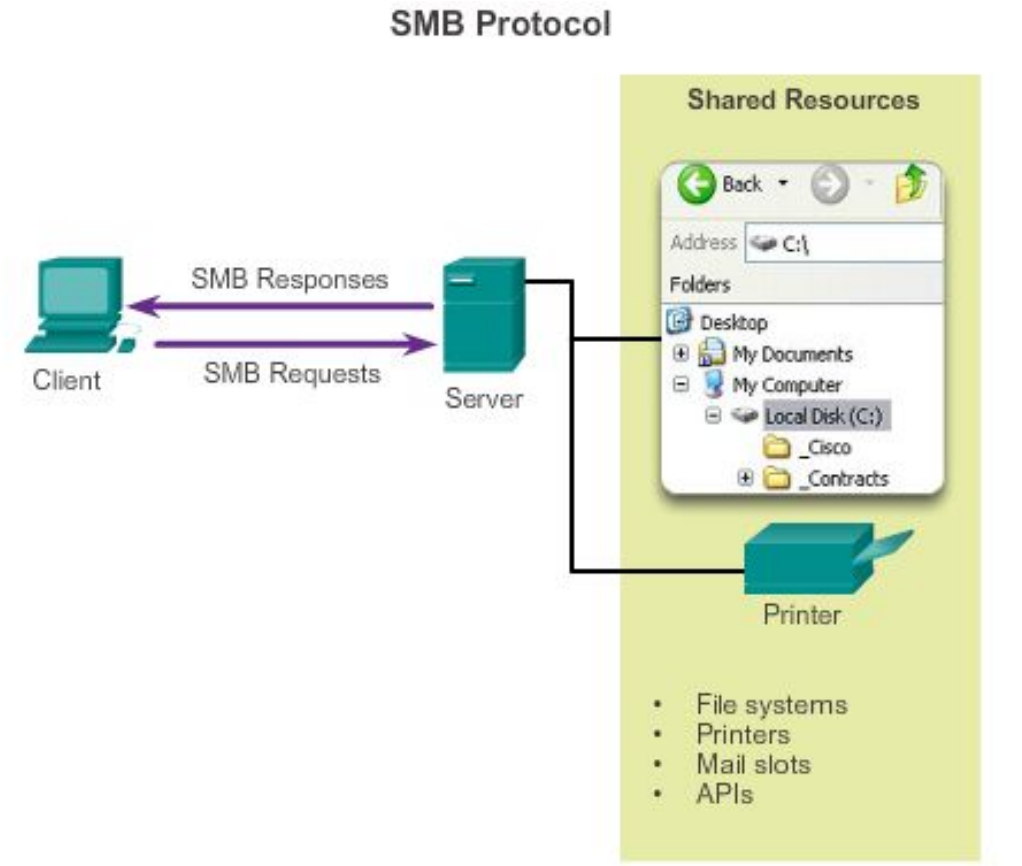
Клиент может загрузить (принять) данные с сервера или отправить данные на сервер.

Блок сообщений сервера

Протокол SMB описывает доступ к файловой системе и способ запроса файлов клиентами.

Он также описывает связь между процессами SMB.

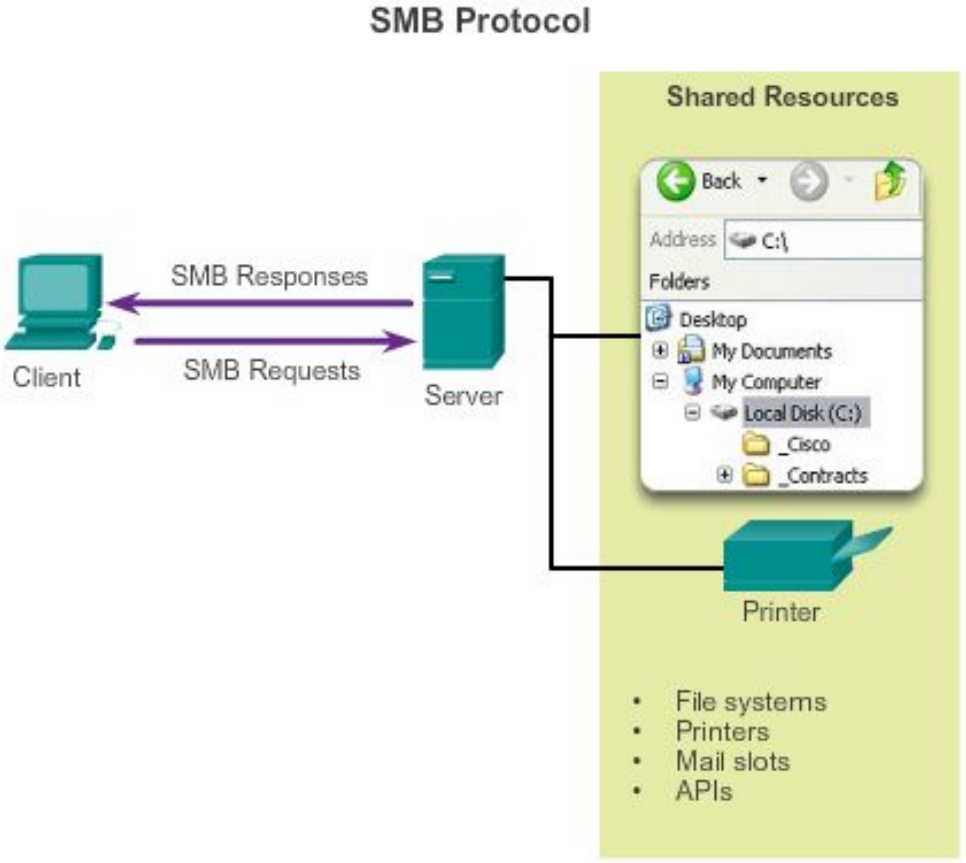
Все сообщения SMB имеют общий формат.



SMB is a client-server, request-response protocol. Servers can make their resources available to clients on the network.

Блок сообщений сервера (SMB)

- Клиенты устанавливают долгосрочное соединение с серверами
- После установления соединения пользователь может осуществлять доступ к ресурсам на сервере так, как если бы эти ресурсы были локальными на клиентском узле

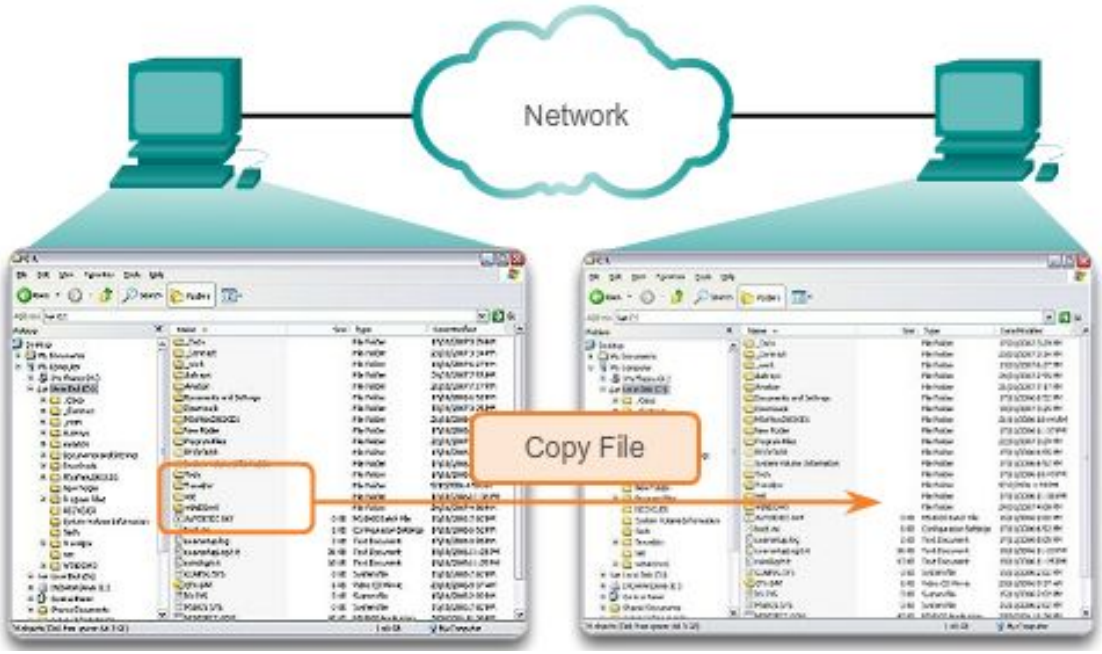


SMB is a client-server, request-response protocol. Servers can make their resources available to clients on the network.

Блок сообщений сервера

С помощью сообщений SMB можно выполнять следующие действия:

- осуществлять запуск, аутентификацию и завершение сеансов;
- управлять доступом к файлам и принтерам;
- разрешать приложению отправлять сообщения на другое устройс



Интернет вещей

Тенденции, такие как «принеси на работу своё собственное устройство» (BYOD), доступ из любой точки мира, виртуализация и межмашинные подключения (m2m) открывают путь для новых классов приложений.

Ожидается, что к 2020 г. будут связаны между собой около 50 миллиардов устройств.

За один только 2010 год было разработано более 350 000 приложений, которые были загружены более трёх миллионов раз.

Всё это приводит к созданию интуитивных связей между пользователями, процессами, данными и вещами в сетях.

Интернет вещей

Внедрение цифровых технологий, таких как смарт-теги и расширенные сетевые возможности, реализованные в простых изделиях, от велосипедов и бутылок до холодильников и автомобилей, и подключение их к Интернету позволит людям и компаниям взаимодействовать друг с другом новыми и почти невообразимыми способами.

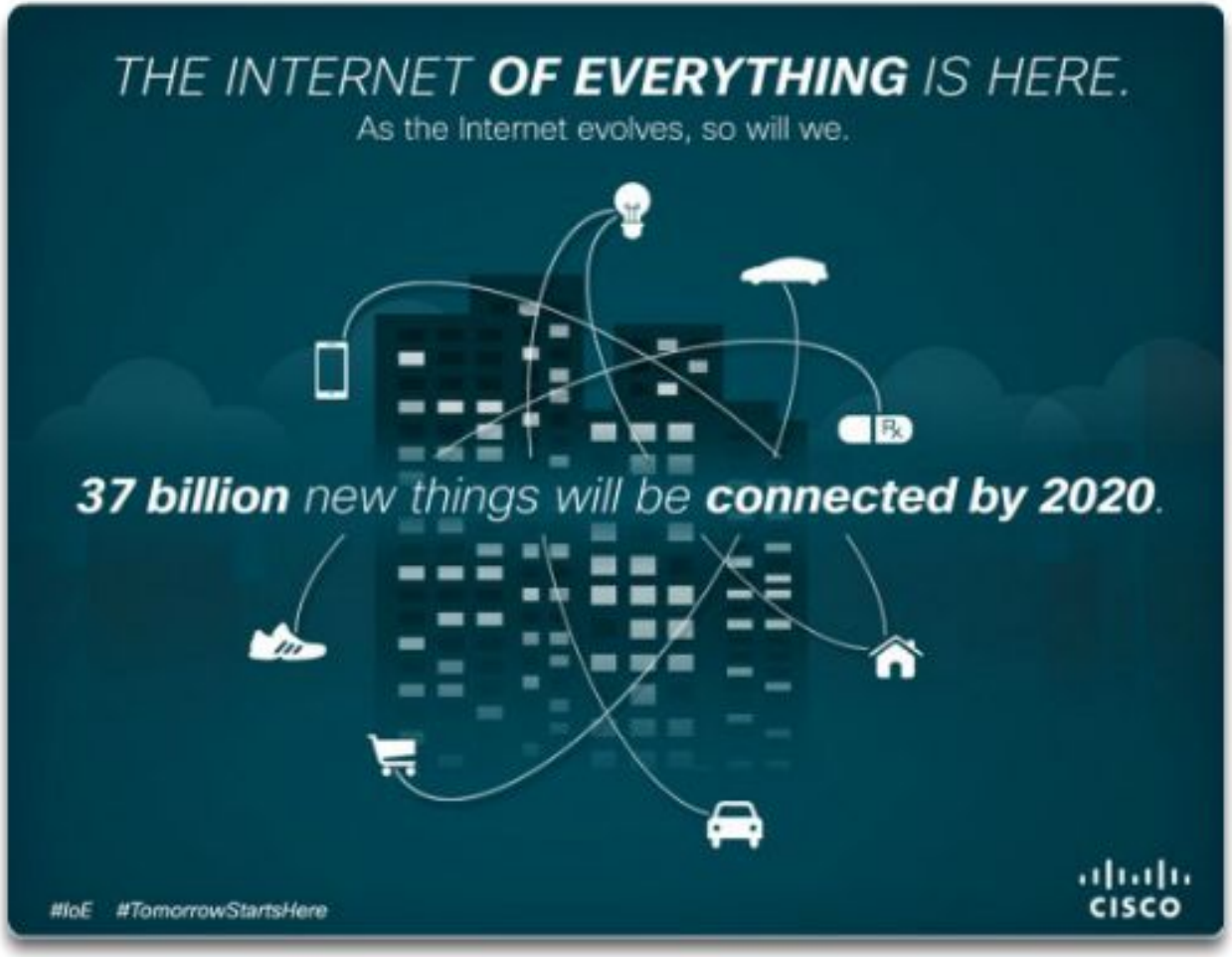
Объекты смогут собирать, получать и отправлять информацию пользователям и другим подключённым объектам.

Более 100 миллионов торговых автоматов, транспортных средств, систем пожарной сигнализации и других устройств уже сегодня автоматически обмениваются своими данными.

А по прогнозам рыночных аналитиков компании Berg Insight эта цифра приблизится в 2016 г. к 360 миллионам.

Интернет вещей

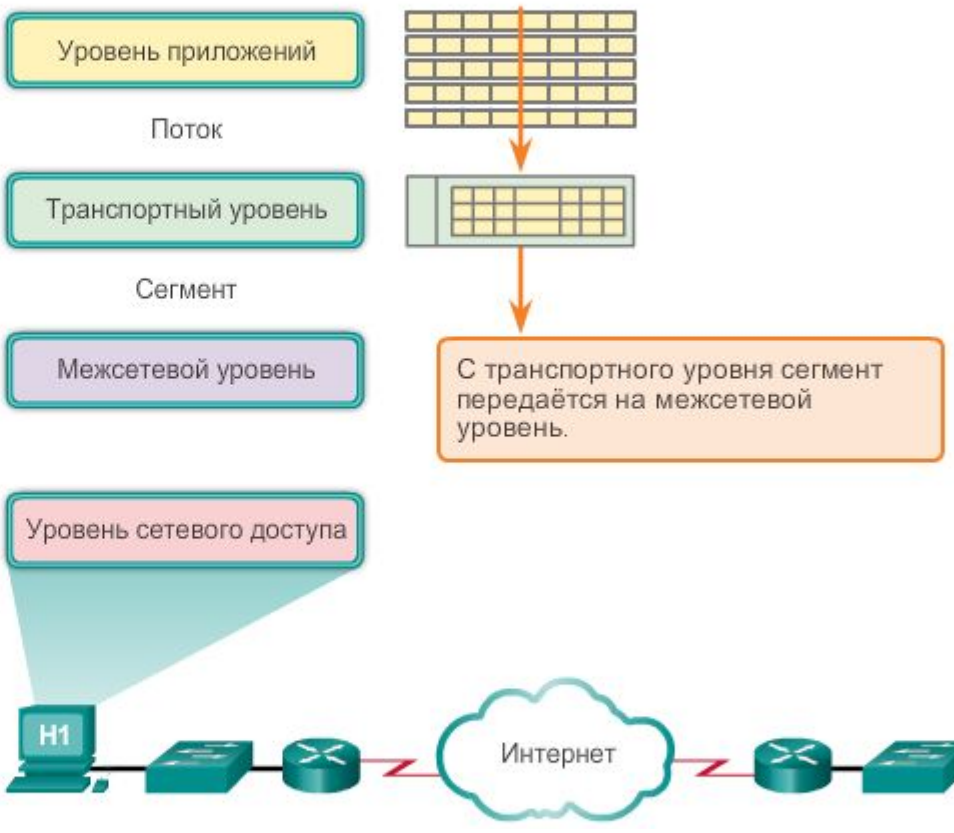
Как показано на рисунке, эта новая волна в интернет-разработке известна под названием «Всеобъемлющий Интернет».



Передача сообщения по сети

При использовании модели TCP/IP полный процесс обмена данными состоит из шести шагов.

Первый шаг — это создание данных на уровне приложений исходного оконечного устройства.



Передача сообщения по сети

В этом случае после создания запроса веб-клиента (HTTP GET) эти данные будут закодированы, сжаты и, если необходимо, зашифрованы.

Этот процесс выполняется на уровне приложений модели TCP/IP, но к нему также относятся функции, описанные уровнем приложений, уровнем представления и сеансовым уровнем модели OSI.

Передача сообщения по сети

Шаг 2. Сегментация и первоначальная инкапсуляция данных по мере их прохождения по стеку протоколов.

На транспортном уровне сообщение HTTP GET будет разбито на более мелкие и более управляемые части, в каждую из которых будет добавлен заголовок транспортного уровня.

В заголовках транспортного уровня находятся индикаторы, по которым можно будет воссоздать сообщение.

В заголовок также добавляется идентификатор — номер порта 80.

Он сообщает конечному серверу, что сообщение предназначено для приложения веб-сервера.

Также добавляется сгенерированный случайным образом исходный порт, чтобы клиент смог получить ответное сообщение и переслать его соответствующему клиентскому приложению.

Передача сообщения по сети

Шаг 3. Адресация - далее в сегменты добавляются идентификаторы адреса.

Так же, как существует несколько уровней протоколов, которые подготавливают данные для передачи в место назначения, существует несколько уровней адресации для обеспечения доставки данных.

Задача сетевого уровня — добавить адресацию, чтобы обеспечить передачу данных с исходного узла на узел, который их использует.

На сетевом уровне это выполняется путём инкапсуляции каждого сегмента в заголовок IP-пакета.

Заголовок IP-пакета содержит IP-адреса исходного и оконечного устройств.

Передача сообщения по сети

Шаг 4. Подготовка к передаче – далее пакет передаётся на уровень сетевого доступа для генерации данных в физической среде.

Для этого на уровне сетевого доступа пакет сначала должен быть подготовлен к передаче путём помещения его в кадр с заголовком и трейлер.

Этот кадр содержит физический адрес (MAC) исходного узла, а также физический адрес (MAC) следующего узла на пути к месту назначения.

Кадр после подготовки на уровне сетевого доступа с добавлением IP-адресов источника и назначения кодируется в последовательность битов, а затем — в электрические импульсы или вспышки света, которые передаются по кабелям сети.

Передача сообщения по сети

Шаг 5. Передача данных - данные передаются в сетевой инфраструктуре, которая состоит из носителя и промежуточных устройств.

Инкапсулированное сообщение при движении по сети может передаваться по различным носителям и типам сети.

Уровень сетевого доступа определяет способы передачи кадра в носитель и вывода из него, что иначе называется управлением доступом к среде передачи данных.