Протоколы прикладного уровня

Курс читает Рогозин Н.О., каф. ИУ-7

Telnet

- Сетевой протокол для реализации текстового интерфейса по сети
- Был одним из первых протоколов удаленного сообщения
- Позволяет обслуживающей машине рассматривать все удаленные терминалы как стандартные "сетевые виртуальные терминалы" строчного типа, работающие в коде ASCII, а также обеспечивает возможность согласования более сложных функций (например, локальный или удаленный эхо-контроль, страничный режим, высота и ширина экрана и т.д.)

Telnet

- Протокол имеет симметричную структуру сообщения, позволяя двум терминалам обмениваться:

  1) прикладными данными

  2) командами протокола Telnet

- На прикладном уровне над TELNET находится либо программа поддержки реального терминала (на стороне пользователя), либо прикладной процесс в обсуживающей машине, к которому осуществляется доступ с терминала.

Пример настройки доступа в Packet Tracer

- Router(config)#line vty 0 4
- Router(config-line)#password cisco
- Router(config-line)#login
- Router(config)#service password-encryption

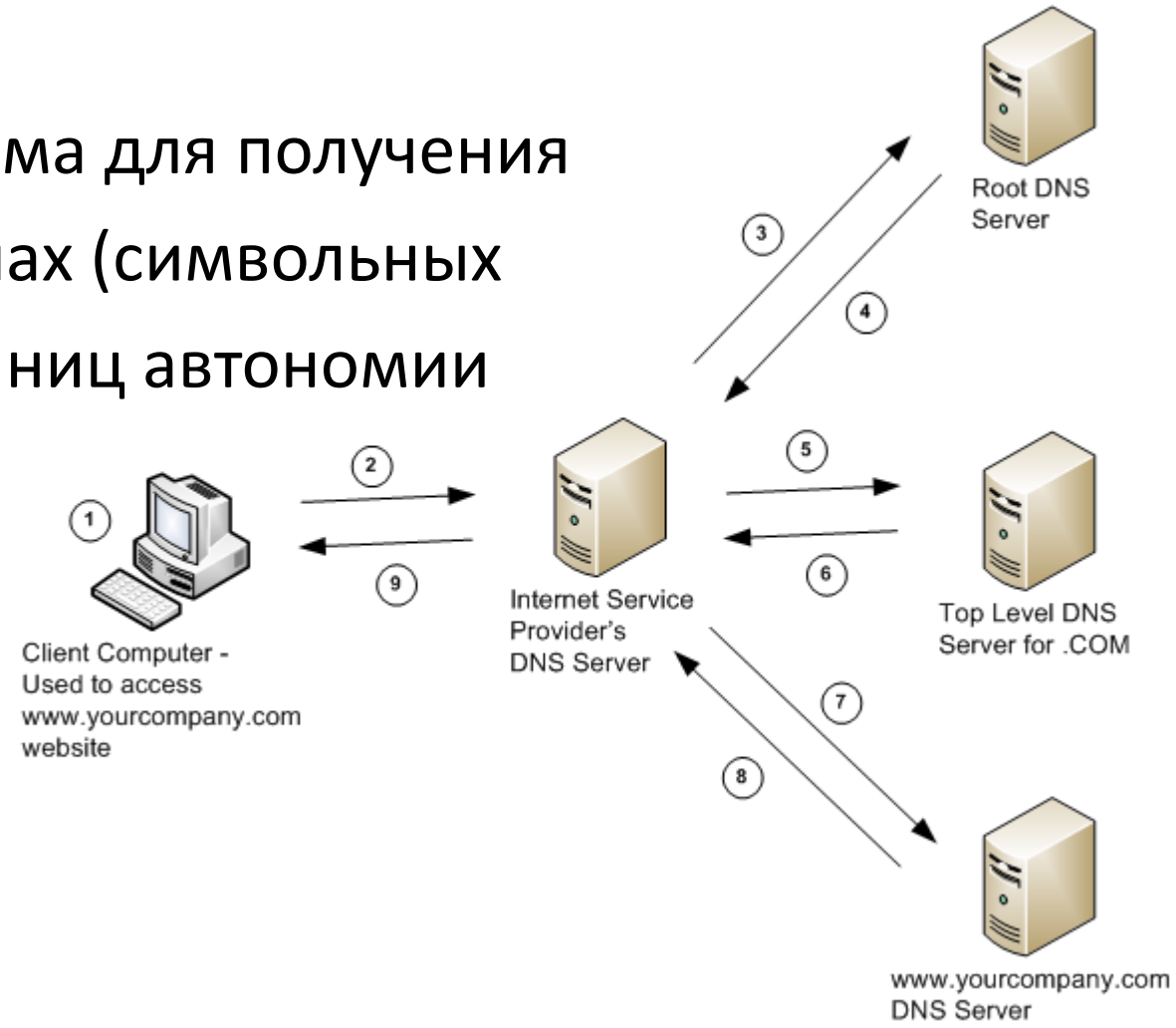
либо:

- Router(config)#aaa new-model
- Router(config)#username admin password 1234

(в рамках модели AAA (Authentication, Authorization, Accounting)). При этом появляется возможность использовать для аутентификации на устройстве RADIUS или TACACS сервер.

DNS

- Распределенная система для получения информации о доменах (символьных идентификаторах единиц автономии в сети Интернет).



DNS

- 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.

Зона DNS и файл отображений

- Часть пространства доменных имен, для которых некоторый сервер DNS имеет информацию об их отображениях на основе соответствующего текстового файла, называется **зоной DNS** данного сервера, а сам текстовый файл — **файлом зоны**.

Файл hosts (файл отображений)

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

# localhost name resolution is handled within DNS itself.
#   127.0.0.1       localhost
#   ::1             localhost
```

DNS

- 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.

Иерархические символьные имена

- 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.

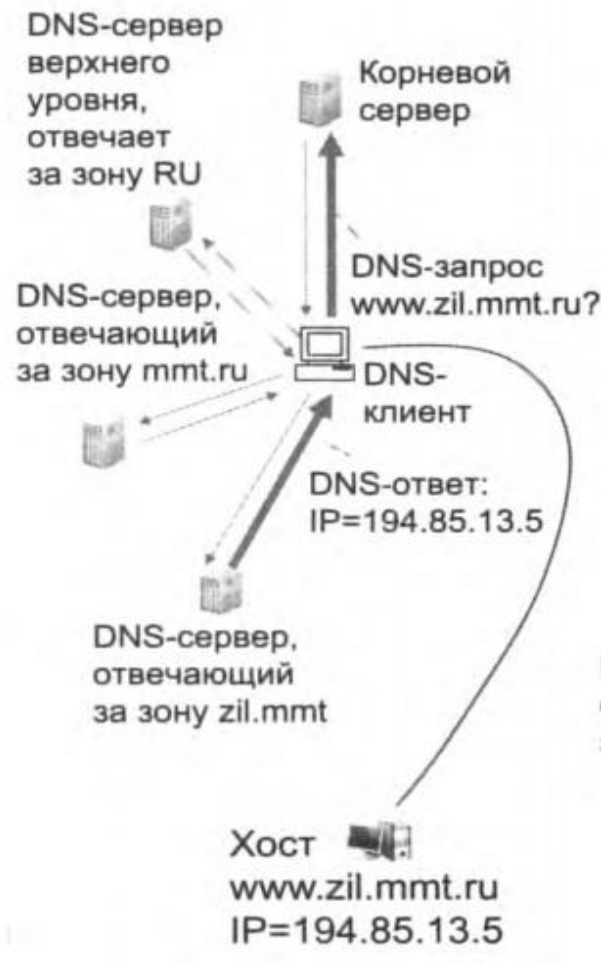
Иерархические символьные имена

- Корень базы данных DNS управляется центром INIC (Internet Network Information Center).
- Домены верхнего уровня назначаются для каждой страны, а также на организационной основе. Для обозначения стран используются трехбуквенные и двухбуквенные аббревиатуры, а для различных типов организаций используются следующие аббревиатуры:
  - com - коммерческие организации (например, microsoft.com);
  - edu - образовательные (например, mit.edu);
  - gov - правительственные организации (например, nsf.gov);

DNS

- Каждый домен DNS
  - Администрируется отдельной организацией, которая обычно разбивает свой домен на поддомены и передает функции администрирования этих поддоменов другим организациям.
  - 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 - FQDN (fully qualified domain name), которое включает имена всех доменов по направлению от хоста к корню.
  - Например: **students.bmstu.ru**

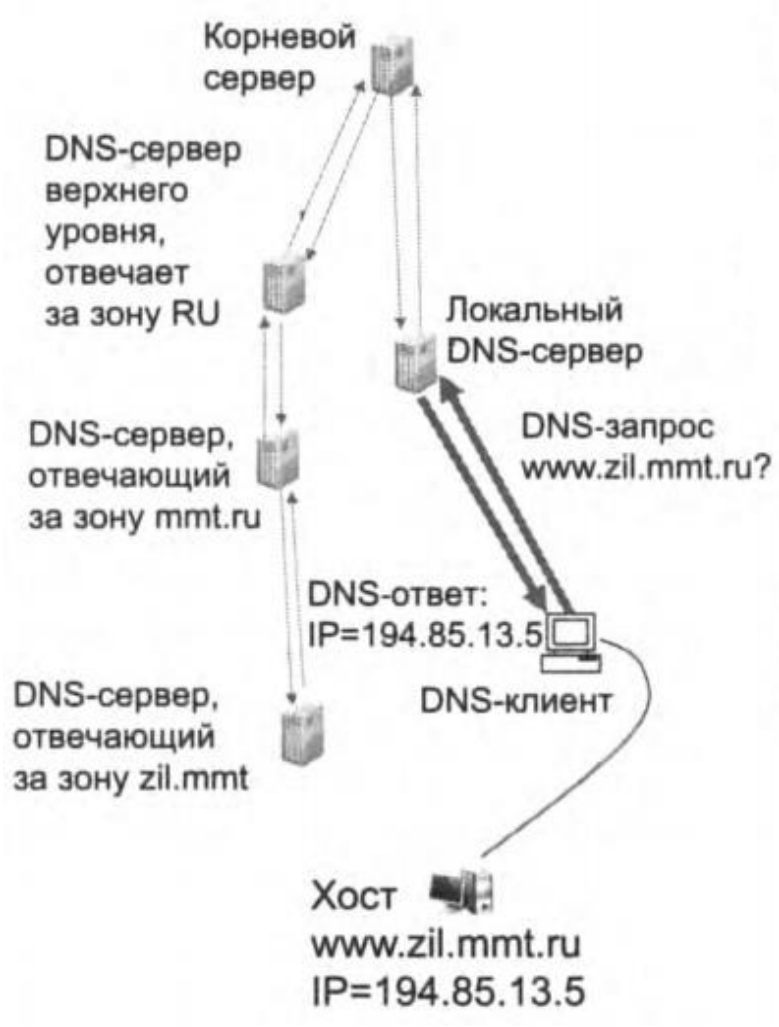
Итеративная процедура разрешения имен DNS



Итеративная процедура DNS

- DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени www.zil.mmt.ru хоста, для которого он хочет найти IP-адрес.
- Корневой DNS-сервер отвечает клиенту, указывая адреса DNS-серверов верхнего уровня, обслуживающих домен, заданный в старшей части запрошенного имени, в данном случае — домен ru.
- DNS-клиент делает следующий запрос к одному из предложенных ему DNS-серверов верхнего уровня, который отсылает его к DNS-серверу нужного поддомена (в примере это сервер, отвечающий за зону mmt.ru), и так далее, пока не будет найден DNS-сервер, в котором хранится отображение запрошенного имени на IP-адрес. Этот сервер дает окончательный ответ клиенту, который теперь может установить связь с хостом поIP-адресу 194.85.13.5.

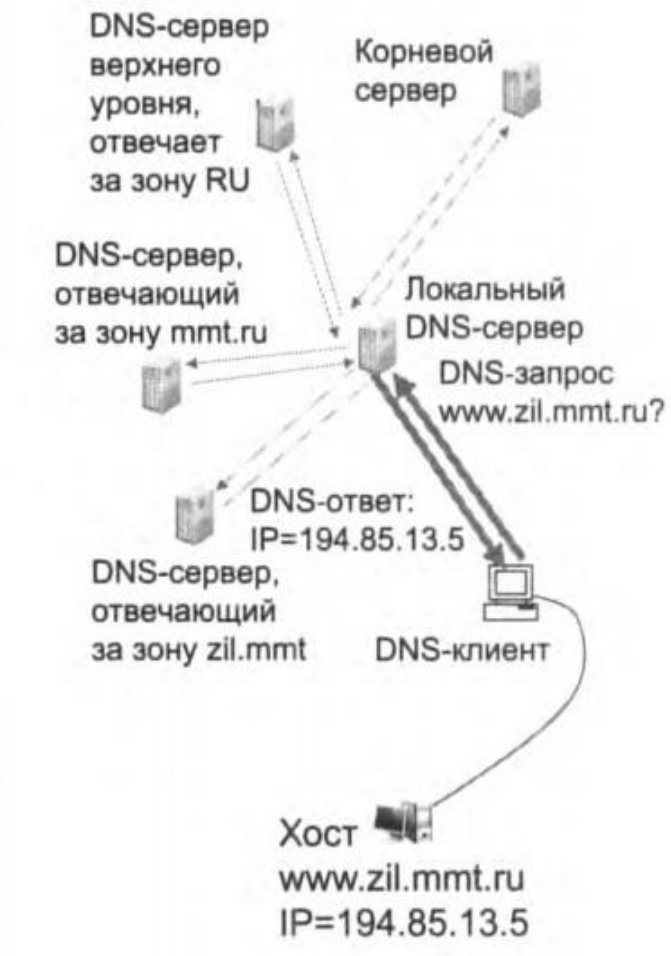
Рекурсивная процедура разрешения имен DNS



Рекурсивная процедура DNS

- DNS-клиент отправляет запрос к локальному DNS-серверу, то есть серверу, обслуживающему поддомен, которому принадлежит имя клиента.
- Если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту. Это может быть полномочный ответ (запрошенное имя входит в тот же поддомен, что и имя клиента) или неполномочный ответ (сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше).
- Если локальный DNS-сервер не знает ответа, то он обращается к корневому серверу, который переправляет запрос к DNS-серверу верхнего уровня (отвечающему за зону RU), который в свою очередь запрашивает нижележащий сервер (зона mmt), и так далее, пока запрос не дойдет до полномочного сервера, имеющего в своем файле зоны запись о запрошенном имени.

Смешанная процедура DNS



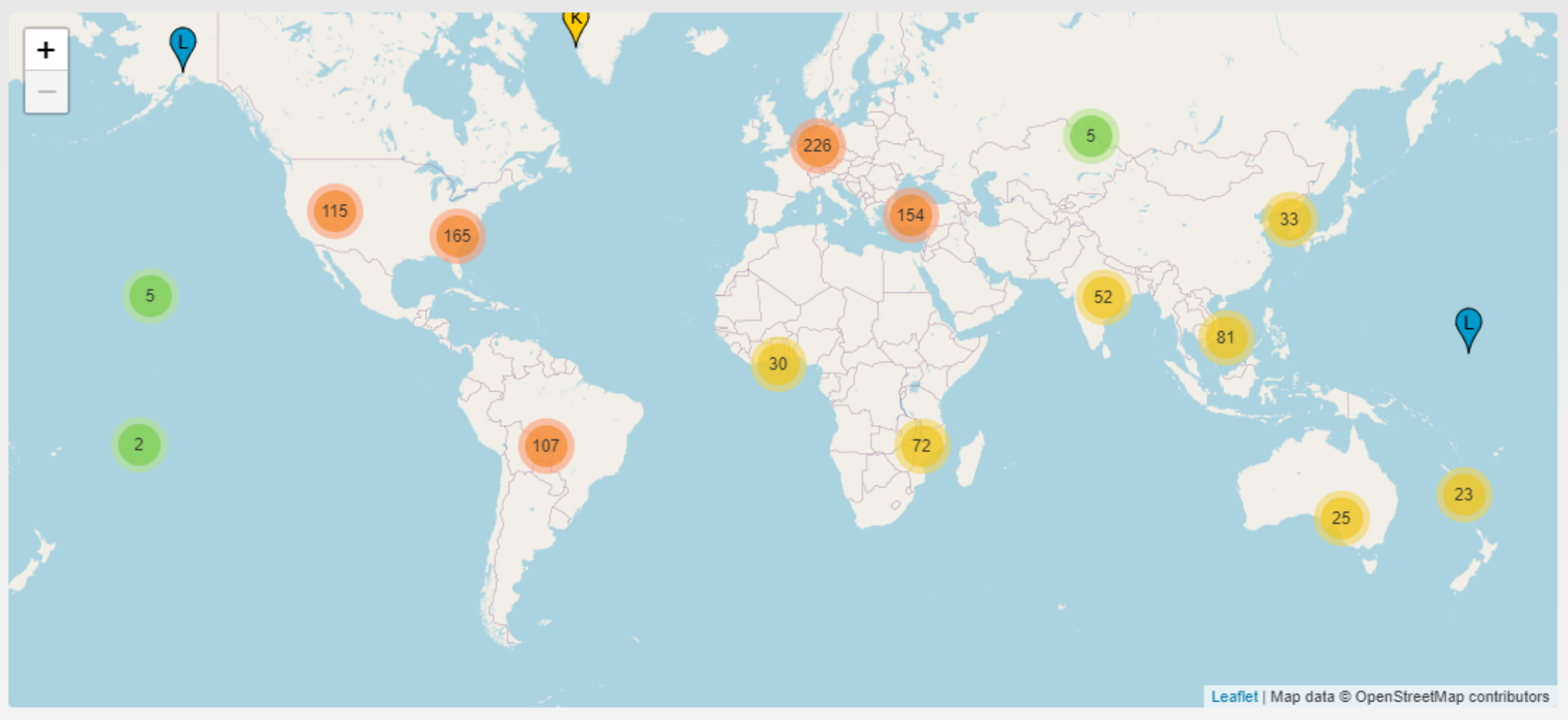
Смешанная процедура разрешения имен DNS

- Начальная часть процедуры, когда DNS-клиент передает запрос локальному DNS- серверу и поручает ему действовать от его имени, является рекурсивной.
- Затем, если локальный DNS-сервер не знает ответ, то он последовательно выполняет итеративные запросы к иерархии серверов точно так же, как это делал DNS-клиент в первом варианте. Получив ответ, локальный DNS-сервер передает его клиенту.

Корневые сервера

- Разрешение всех запросов, ответы на которые не находятся в кэше или файле зоны какого-либо DNS-сервера нижнего уровня, начинаются с обращения к одному из корневых серверов.
- Разработчики системы DNS понимали это, поэтому изначально было решено обеспечить высокую степень резервирования: было установлено 13 корневых серверов с именами a.root-servers.net, b.root-servers.net, c.root-servers.net,... m.root-servers.net и тринадцатью IP-адресами.
- С тех пор организация корневых DNS-серверов изменилась.
- Более 300 серверов
- Большая отказоустойчивость и производительность современной службы DNS.
- Все корневые серверы по-прежнему разделяют те же 13 имен (от a.root-servers.org до rn.root-servers.org) и 13 IP-адресов.
- Но теперь каждому имени и адресу соответствует кластер серверов. Например, имени f.root-servers.net соответствует 56 серверов, а имени Lroot-servers.net — 146.

Карта корневых серверов (октябрь 2020)



Обратная зона

- Система таблиц , которая хранит соответствие между IP-адресами и DNS-именами хостов в некоторой сети
- Находит нужное DNS-имя по IP адресу
- IP-адрес представляется в виде DNS -имени (в обратном порядке).
- Учитывая, что при записи IP-адрес а старшая часть является само й левой частью адреса, а при записи DNS-имени — самой правой, составляющие в преобразованном адресе указываются в обратном порядке
- Например, для адреса 192.31. 106.0 — 106.31.192 .

Актуальные проблемы DNS

- Denial of Service (DoS) атаки пропускной способности сети
- DoS атаки центрального процессора/потребление памяти
- Изменения в протоколах и системах безопасности
- Data Integrity
- Компрометация DNSKEY (ключ системы защиты DNSSEC)

Режимы DHCP

- Ручное назначение статических адресов;
- Автоматическое назначение статических адресов;
- Автоматическое распределение динамических адресов.

Ручной режим

- Администратор, помимо пула доступных адресов, снабжает DHCP-сервер информацией о жестком соответствии IP-адресов физическим адресам или другим идентификаторам клиентских узлов.
- DHCP-сервер, пользуясь этой информацией, всегда выдаст определенному DHCP-клиенту один и тот же назначенный ему администратором IP-адрес

Автоматическое назначение

- DHCP-сервер самостоятельно, без вмешательства администратора произвольным образом выбирает клиенту IP-адрес из пула наличных IP-адресов.
- Адрес дается клиенту из пула в постоянное пользование, то есть между идентифицирующей информацией клиента и его IP-адресом по-прежнему, как и при ручном назначении, существует постоянное соответствие.
- Оно устанавливается в момент первого назначения DHCP-сервером IP-адреса клиенту.
- При последующих запросах сервер возвращает клиенту тот же самый IP-адрес.

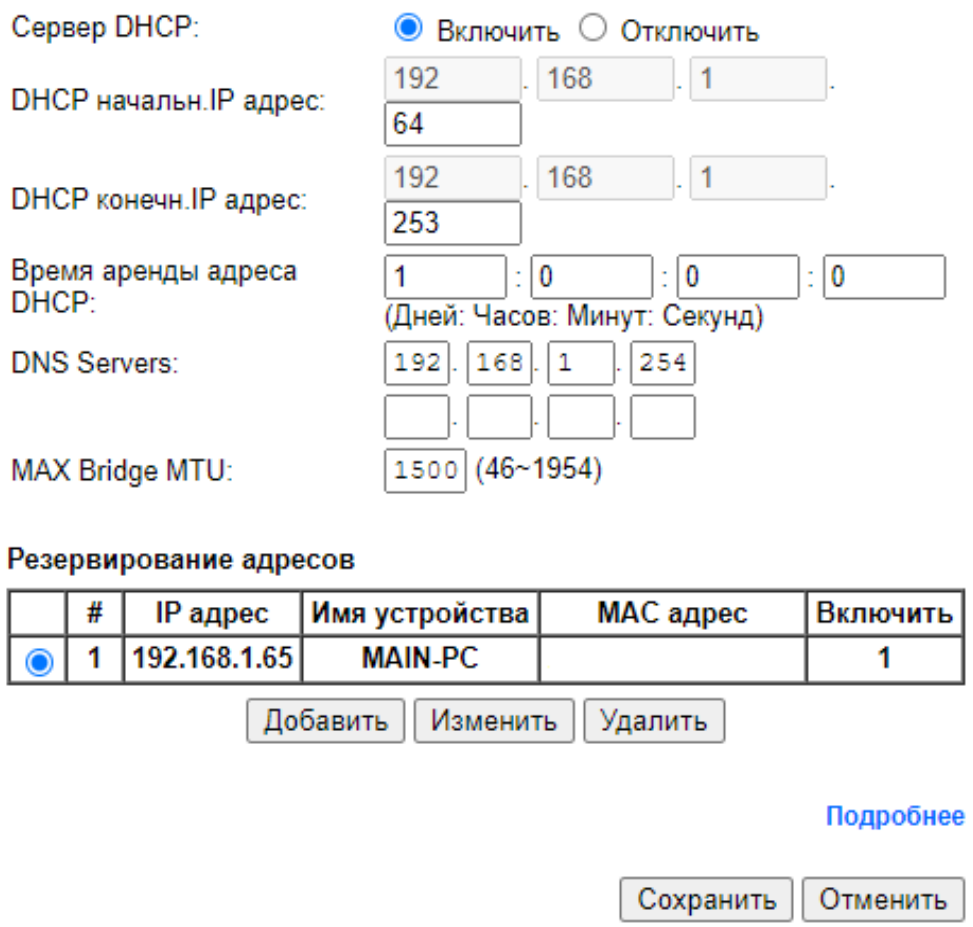
Динамическое распределение

- DHCP-сервер выдает адрес клиенту на ограниченное время, называемое сроком аренды.
- Когда компьютер, являющийся DHCP-клиентом, удаляется из подсети, назначенный ему IP-адрес автоматически освобождается.
- Когда компьютер подключается к другой подсети, то ему автоматически назначается новый адрес.
- Ни пользователь, ни сетевой администратор не вмешиваются в этот процесс. Это дает возможность впоследствии повторно использовать этот IP-адрес для назначения другому компьютеру.

Свойства набора адресов DHCP

- Диапазон адресов + набор исключений
- Маска подсети
- Длительность аренды
- DNS-сервер
- Шлюз по умолчанию

Пример использования



DHCP-агент

- Программное обеспечение, играющее роль посредника между DHCP-клиентами и DHCP-серверами (пример такого варианта — сеть 2).
- Связной агент переправляет запросы клиентов из сети 2 DHCP-серверу сети 3.
- Таким образом, один DHCP-сервер может обслуживать DHCP-клиентов нескольких разных сетей.

Порядок работы DHCP

1. Когда компьютер включают, установленный на нем DHCP-клиент посылает ограниченное широковещательное сообщение DHCP-поиска (IP-пакет с адресом назначения, состоящим из одних единиц, который должен быть доставлен всем узлам данной IP сети).
2. Находящиеся в сети DHCP-серверы получают это сообщение. Если в сети DHCP- серверы отсутствуют, то сообщение DHCP-поиска получает связной DHCP-агент. Он пересылает это сообщение в другую, возможно, значительно отстоящую от него сеть DHCP-серверу, IP-адрес которого ему заранее известен.

Порядок работы DHCP

3. Все DHCP-серверы, получившие сообщение DHCP-поиска, посылают DHCP-клиенту, обратившемуся с запросом, свои DHCP-предложения. Каждое предложение содержит IP-адрес и другую конфигурационную информацию. (DHCP-сервер, находящийся в другой сети, посылает ответ через агента.)
4. DHCP-клиент собирает конфигурационные DHCP-предложения от всех DHCP-серверов. Как правило, он выбирает первое из поступивших предложений и отправляет в сеть широковещательный DHCP-запрос. В этом запросе содержатся идентификационная информация о DHCP-сервере, предложение которого принято, а также значения принятых конфигурационных параметров.

Порядок работы DHCP

5. Все DHCP-серверы получают DHCP-запрос и только один выбранный DHCP-сервер посылает положительную DHCP-квитанцию (подтверждение IP-адреса и параметров аренды), а остальные серверы аннулируют свои предложения, в частности, возвращают в свои пулы предложенные адреса.
6. DHCP-клиент получает положительную DHCP-квитанцию и переходит в рабочее состояние.

Проблемы динамического назначения

- Возникают сложности при преобразовании символьного доменного имени в IP-адрес
- Трудно осуществлять удаленное управление и автоматический мониторинг интерфейса (например, сбор статистики), если в качестве его идентификатора выступает динамически изменяемый IP-адрес
- Усложняется фильтрация пакетов по IP-адресам

HTTP

- Служит для передачи гипертекстовой информации
- Главный протокол всемирной паутины (www)
- Существует несколько версий этого протокола: HTTP 1.0, HTTP 1.1, HTTP/2 и HTTP/3
- Обмен сообщениями идет по обычной схеме «запрос-ответ». Клиент и сервер обмениваются текстовыми сообщениями стандартного формата, то есть каждое сообщение представляет собой несколько строк обычного текста в кодировке ASCII.
- Для транспортировки HTTP-сообщений служит протокол TCP.

HTTP определяет порядок того как веб-клиенты запрашивают веб-страницы с веб-сервера и как сервер передает эти страницы клиентам.

Постоянные и непостоянные соединения

- При отправке каждой пары запрос-ответ через отдельное соединение говорят, что используются кратковременные, или **непостоянные соединения**;
- При отправке каждой пары через одно и то же TCP соединение — долговременные, или **постоянные соединения**.

Непостоянное соединение

- Допустим, есть адрес: http://www.bmstu.ru/home.index

1) HTTP-клиент инициирует TCP-соединение с сервером www.bmstu.ru по порту 80 (порт по умолчанию для HTTP). Этому TCP-соединению выделяются сокеты на клиентской и северной стороне.
2) HTTP-клиент отправляет запрос серверу через свой сокет. Запрос включает путь к базовому файлу /home.index

Непостоянное соединение

3) Процесс HTTP-сервера получает запрос через свой сокет, извлекает объект /home.index из своего места хранения (оперативной памяти или диска), помещает объект в ответное HTTP-сообщение и отправляет клиенту через свой сокет.
4) Процесс HTTP-сервера дает команду протоколу TCP закрыть соединение (на самом деле, TCP-соединение не разрывается до тех пор, пока сервер не получит информацию об успешном получении ответа клиентом).

Непостоянное соединение

5) HTTP-клиент получает ответ от сервера, и TCP-соединение разрывается. Сообщение указывает, что полученный объект — это HTMLфайл. Клиент извлекает файл из сообщения, обрабатывает его и находит ссылки на 10 объектов (файлов в формате JPEG).
6) Шаги с первого по четвертый повторяются для каждого из десяти JPEG-объектов.

Непостоянное соединение

- Когда браузер получает веб-страницу, он отображает ее на экране.
- Два различных браузера могут интерпретировать веб- страницу по-разному.
- Спецификации протокола HTTP (RFC 1945 и RFC 2616) определяют только протокол взаимодействия между программой клиента и программой сервера, но ничего не говорят о том, как вебстраница должна интерпретироваться клиентом.

Непостоянное соединение

- На самом деле большинство современных браузеров в режиме по умолчанию открывают от пяти до десяти параллельных TCP-соединений, и каждое из них обрабатывает одну транзакцию из запроса и ответа, а степень этого параллелизма может быть сконфигурирована пользователем.
- Если тот пожелает, число параллельных соединений можно установить равным единице, и в этом случае 10 соединений будут устанавливаться последовательно.

Постоянное соединение

- В случае с постоянным соединением сервер после отправки ответа клиенту оставляет TCP-соединение открытым.
- Через одно и то же соединение можно отправить последовательность запросов и ответов между одним и тем же клиентом и сервером.
- В частности, одно постоянное TCP-соединение позволяет передать всю веб-страницу (в примере выше это базовый HTML-файл и десять изображений).

Постоянное соединение

- Через одно постоянное соединение можно отправить одному и тому же клиенту много веб-страниц, размещенных на том же сервере.
- Эти запросы объектов могут быть сделаны один за другим, без ожидания ответов на обрабатываемый запрос (так называемая конвейеризация).
- Обычно HTTP-сервер закрывает соединение, когда оно не используется в течение определенного времени (настраиваемый интервал тайм-аута).
- Когда сервер получает последовательные запросы, он отправляет объекты также один за другим. По умолчанию HTTP использует постоянное соединение с конвейеризацией.

HTTP 1.0

- Поддерживается только режим кратковременных соединений, когда после передачи одного запроса и получения ответа TCP-соединение закрывается.
- Такой режим полностью соответствует концепции сервера без сохранения состояния, а это, как уже отмечалось, приводит к замедлению работы браузера и увеличению трафика из-за частого выполнения процедуры трехэтапного установления TCP-соединения.

HTTP1.1 (RFC 2616)

- По умолчанию применяются постоянные соединения и конвейерный режим.
- Соединение разрывается по инициативе либо браузера, либо сервера за счет отправки специального токена разрыва соединения в HTTP-пакете.
- Веб-сервер обычно использует таймер неактивности пользователя для того, чтобы разорвать соединение по тайм-ауту и не тратить ресурсы памяти на неактивные соединения.

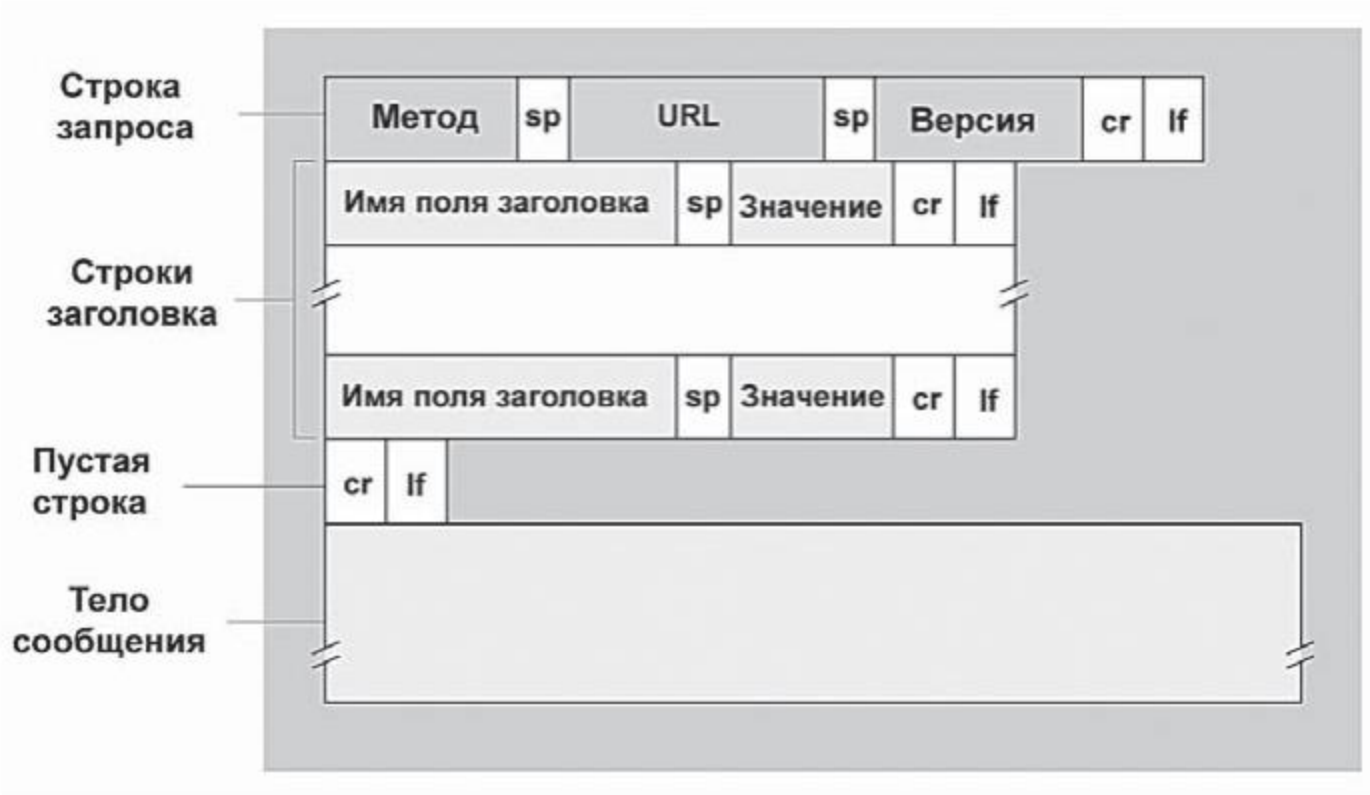
HTTP/2 (RFC 7540)

- Вместо использования отдельных TCP-соединений для передачи каждого запроса и ответа, приводившего к простоям из-за того, что новый запрос не может быть послан без получения ответа, теперь используется одно TCP-соединение для мультиплексирования нескольких запросов, которые могут быть посланы практически одновременно;
- Приоритезация запросов к веб-серверу, благодаря которой сервер знает, какой запрос более важен веб-браузеру для построения страницы
- Введение режима Server Push, при котором веб-сервер может передать веб-браузеру не только запрашиваемые ресурсы, но и те, которые, по мнению веб-сервера, скоро пона добятся веб-браузеру
- Компрессия заголовков сообщений HTTP, значительно сокращающая длину сообщения за счет компрессии таких потенциально длинных полей, как куки

HTTP/3

- На текущий момент не стандартизирован
- Заменяет протокол TCP на новый протокол QUIC, являющийся транспортным протоколом, работающим поверх UDP, и более быстро, чем TCP, устанавливающая соединения и обрабатывающая потерю и искажения данных.
- Протокол QUIC первоначально был разработан компанией Google и уже применяется в браузере Chrome этой компании.

Формат сообщения-запроса



Формат сообщения-ответа

Методы

- Метод HEAD аналогичен методу GET, но запрашиваются только метаданные заголовка HTML-страницы.
- Метод POST используется клиентом для отправки данных на сервер: сообщений электронной почты, ключевых слов в запросе поиска, веб-формы.
- Метод PUT используется клиентом для размещения некоторого объекта на сервере, на который указывает URL-адрес.

Методы

- Метод DELETE указывает серверу на то, что некоторый объект на сервере, определяемый URL-адресом, необходимо удалить.
- Методы GET и HEAD считаются безопасными1 для сервера, так как они только передают информацию клиенту, а методы POST, PUT и DELETE — опасными, поскольку передают информацию на сервер.
- Наибольшую угрозу представляют два последних метода, так как они непосредственно указывают на объект на сервере. Используя эти методы, злоумышленник может атаковать сервер, заменяя или удаляя некоторые его объекты.

Форматы стартовых строк и заголовков

Обобщенная структура сообщения	**HTTP-запрос**	**HTTP-ответ**
Стартовая строка (всегда должна быть первой строкой сообщения; обязательный элемент)	Формат запроса Метод/ URL HTTP/1.x. Пример: GET /books/books.htm HTTP/1.1	Формат ответа: HTTP/1.x КодСостояния Фраза. Пример: HTTP/1.1 200 OK
Заголовки (следуют в произвольном порядке; могут отсутствовать)	Заголовок о DNS-имени компьютера, на котором расположен веб-сервер. Пример: Host: www.olifer.co.uk	Заголовок о времени отправления данного ответа. Пример: Date: 1 Jan 2009 14:00:30
	Заголовок об используемом браузере. Пример: User-agent: Mozilla/5.0	Заголовок об используемом веб-сервере. Пример: Server: Apache/1.3.0 (Unix)
	Заголовок о предпочтительном языке. Пример: Accept-language: ru	Заголовок о количестве байтов в теле сообщения. Пример: Content-Length: 1234
	Заголовок о режиме соединения. Пример: Connection: close	Заголовок о режиме соединения. Пример: Connection: close
Пустая строка		
Тело сообщения (может отсутствовать)	Здесь могут быть расположены ключевые слова для поисковой машины или страницы для передачи на сервер	Здесь может быть расположен текст запрашиваемой страницы

Классы кодов состояний

- 1xx — информация о процессе передачи;
- 2xx — информация об успешном принятии и обработке запроса клиента (в таблице в примере стартовой строки ответа приведен код и соответствующая фраза 200 OK, сообщающий клиенту, что его запрос успешно обработан);
- 3xx — информация о том, что для успешного выполнения операции нужно произвести следующий запрос по другому URL-адресу, указанному в дополнительном заголовке Location;

Классы кодов состояний

- 4xx — информация об ошибках со стороны клиента (при указании адреса несуществующей страницы браузер выводит на экран сообщение 404 Not Found);
- 5xx — информация о неуспешном выполнении операции по вине сервера (например, сообщение 505 http Version Not Supported говорит о том, что сервер не поддерживает версию HTTP, предложенную клиентом).

Cookie

- Определенный в документе RFC 6265562 механизм сохранения данных для идентификации пользователя
- Позволяет веб-сайтам отслеживать состояние пользовательского соединения.
- Подавляющее большинство коммерческих веб-сайтов используют данный механизм.

Принцип работы

- Пользователь А посещает сайт, используя браузер
- Когда запрос приходит на веб-сервер Amazon, он создает уникальный идентификационный номер, а также запись в своей базе данных, которая индексируется этим идентификационным номером.
- Затем вебсервер Amazon посылает ответ браузеру Сьюзен, включающий в HTTPсообщение заголовок Set-cookie:, и в нем содержится соответствующий идентификационный номер.

Принцип работы

- Когда браузер получает ответное HTTP-сообщение, он видит заголовок Set-cookie: и добавляет строку в специальный cookie-файл (имя сервера и идентификационный номер из заголовка Set-cookie)
- Каждый раз, когда пользователь запрашивает веб-страницу, браузер обращается к своему cookie-файлу, извлекает идентификационный номер этого сайта и помещает строку cookie-заголовка, включающую идентификационный номер, в HTTP запрос.
- Сайту точно известно, какие страницы посетил пользователь, в каком порядке и сколько раз

Прокси-сервер

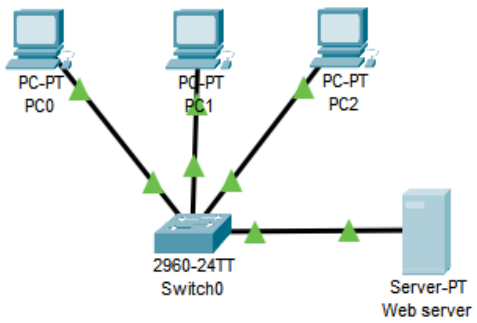
Прокси-сервер

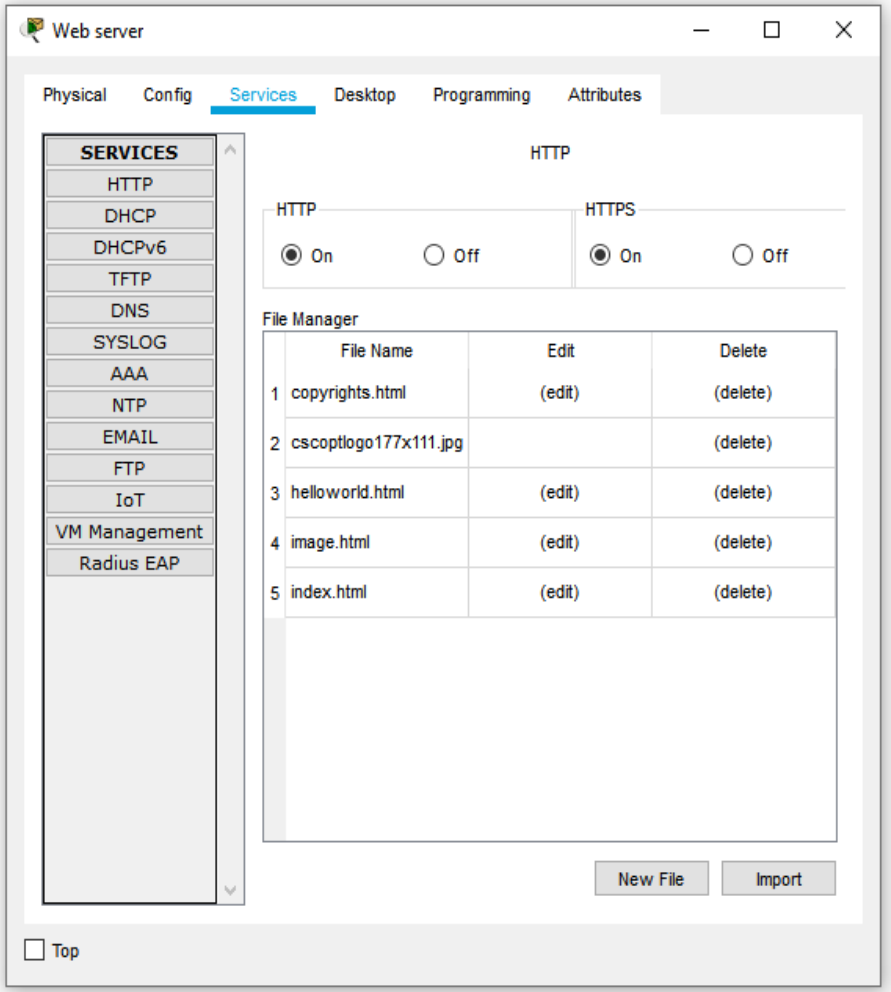
- Веб-кэш, также называемый прокси-сервером — это элемент сети,который обрабатывает HTTP-запрос в дополнение к «настоящему» вебсерверу.
- Для этого на прокси-сервере имеется собственное дисковое хранилище, куда помещаются копии недавно запрошенных объектов.

Прокси-сервер

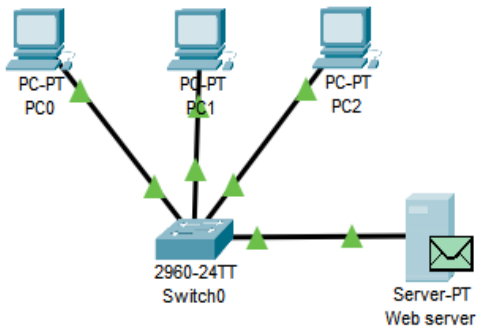
- позволяет уменьшить время ответа на запрос клиента, особенно если полоса пропускания между клиентом и вебсервером намного меньше, чем между клиентом и прокси-сервером. (высокоскоростное соединение, поэтому прокси-сервер способен доставить объект клиенту очень быстро)
- может уменьшить трафик в сети доступа организации, позволяет снизить расходы и положительно сказывается на производительности приложений, использующих сеть

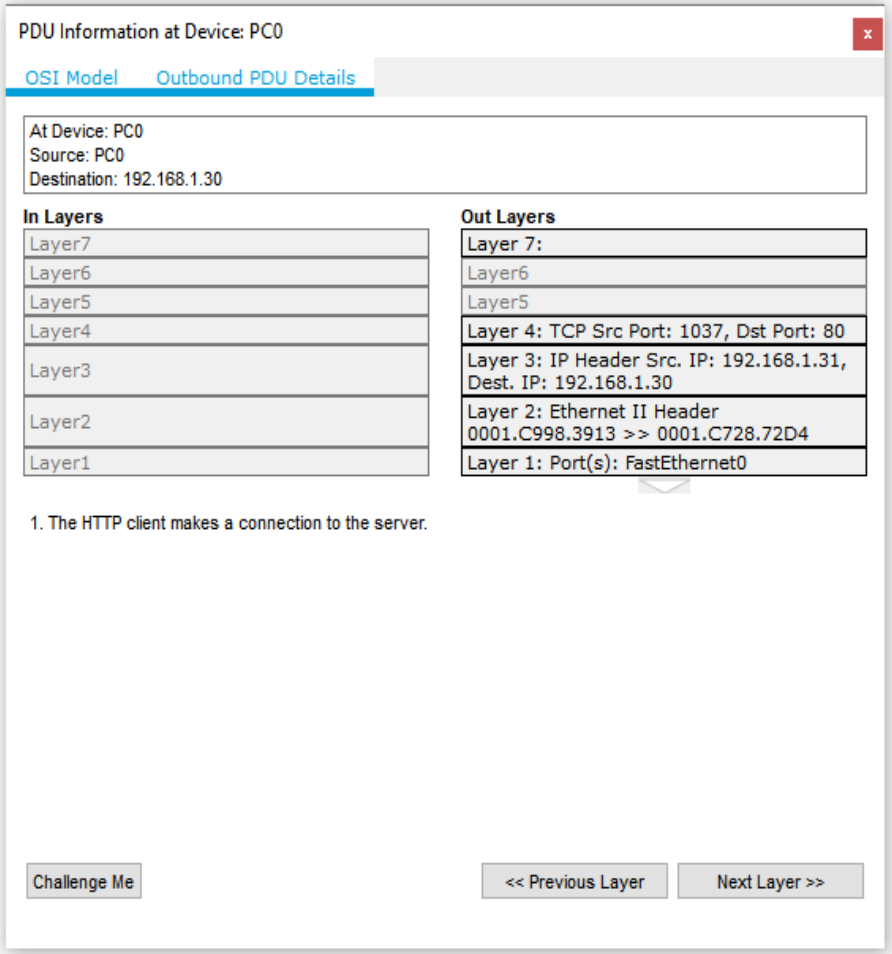
Пример настройки в Packet Tracer



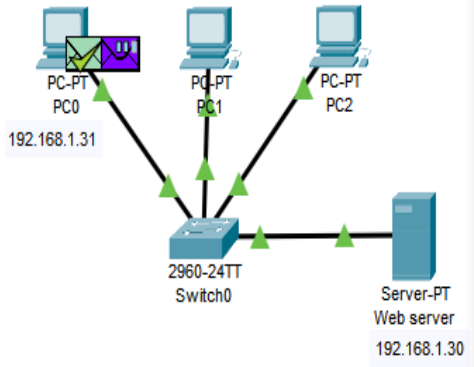


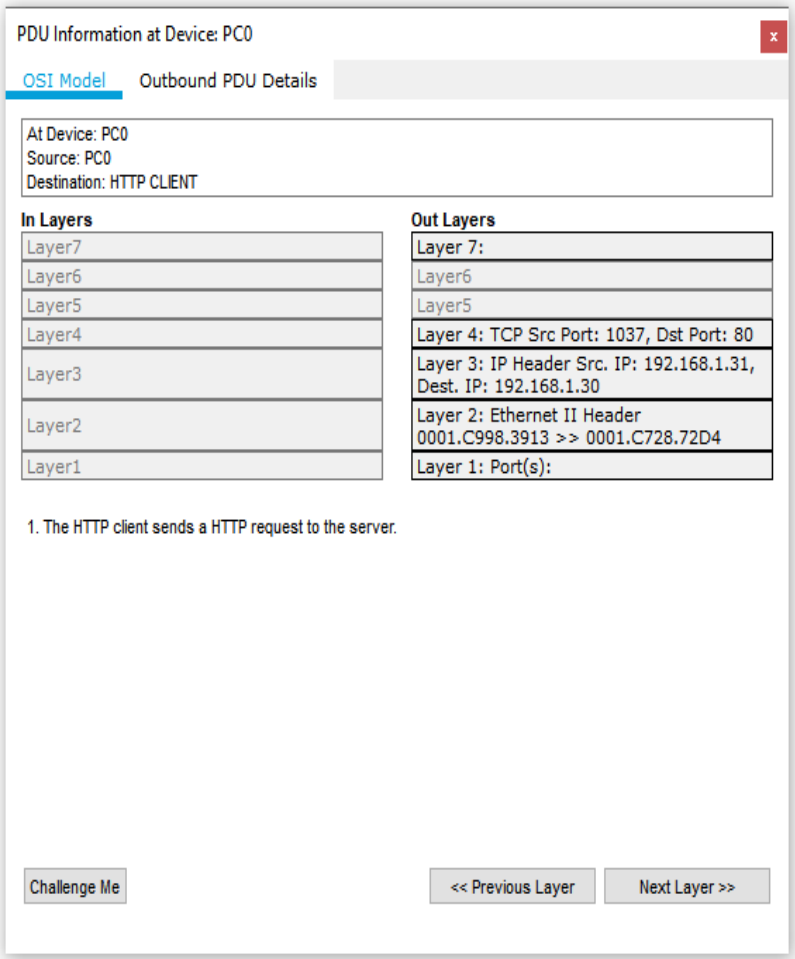
Пример настройки в Packet Tracer





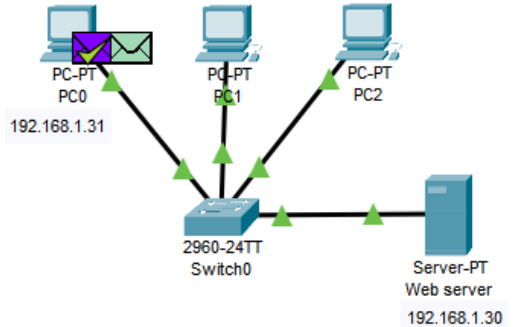
Пример настройки в Packet Tracer

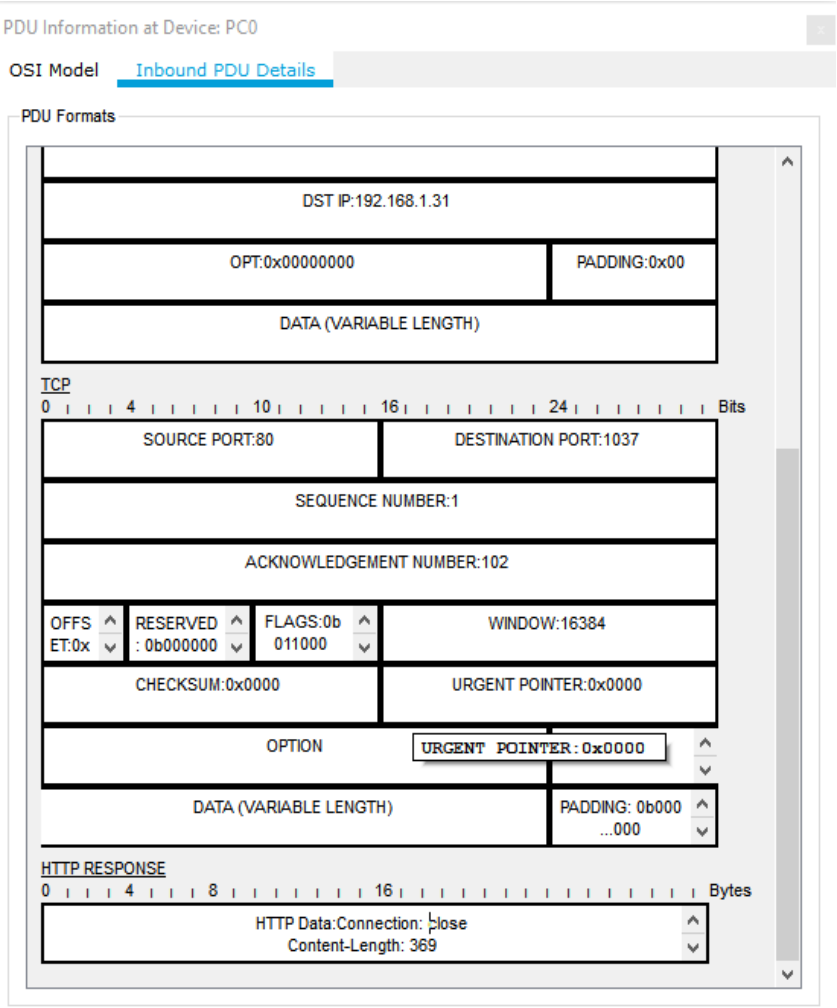




Vis.	Time(sec)	Last Device	At Device	Type
	0.000	--	PC0	TCP
	0.001	PC0	Switch0	TCP
	0.002	Switch0	Web server	TCP
	0.003	Web server	Switch0	TCP
	0.004	Switch0	PC0	TCP
	0.004	--	PC0	HTTP

Пример настройки в Packet Tracer





Пример настройки в Packet Tracer