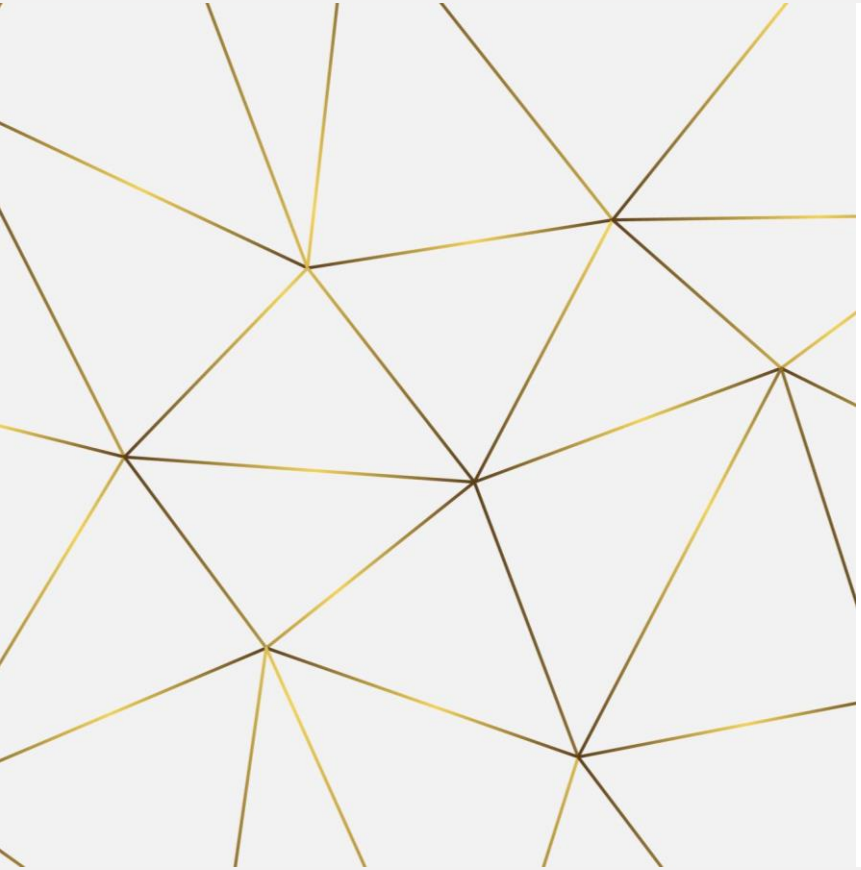
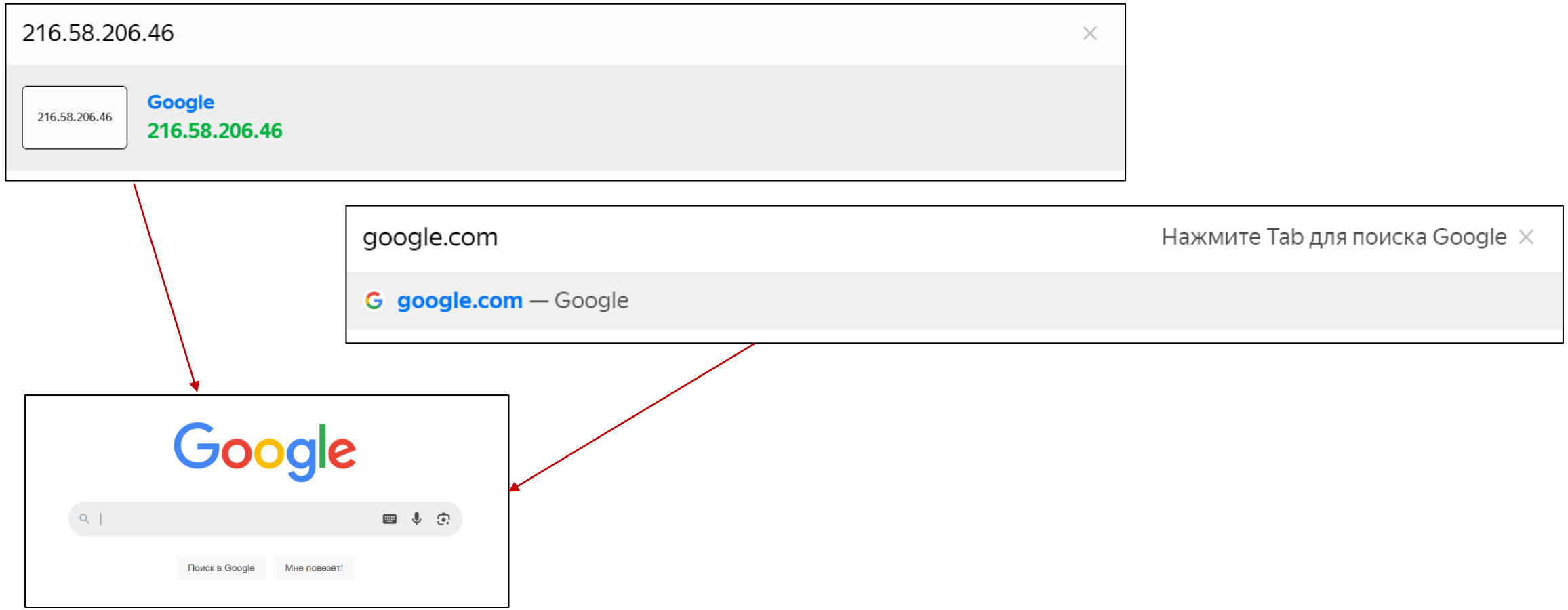


Система доменных имен (DNS)

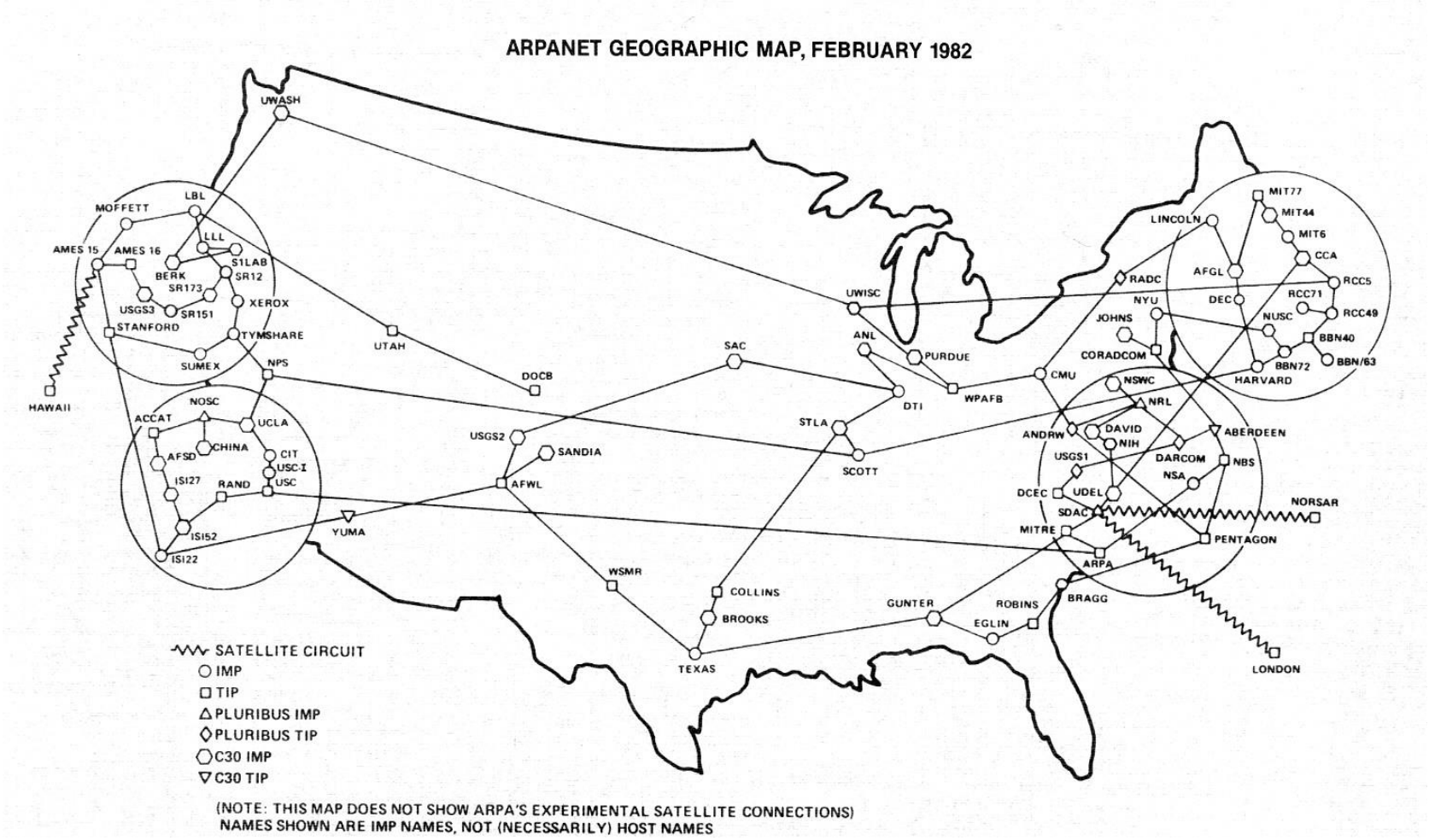
- ОПРЕДЕЛЕНИЕ, ОСНОВНЫЕ ФУНКЦИИ DNS, ИЕРАРХИЯ ДОМЕНОВ
- ПРОЦЕДУРА РАЗРЕШЕНИЯ ИМЕН
- СЛУЖБА И ПРОТОКОЛ DNS
- УТИЛИТЫ ТЕСТИРОВАНИЯ РАБОТЫ СЛУЖБЫ

Выполнила: Синебухова Елизавета, 09-231



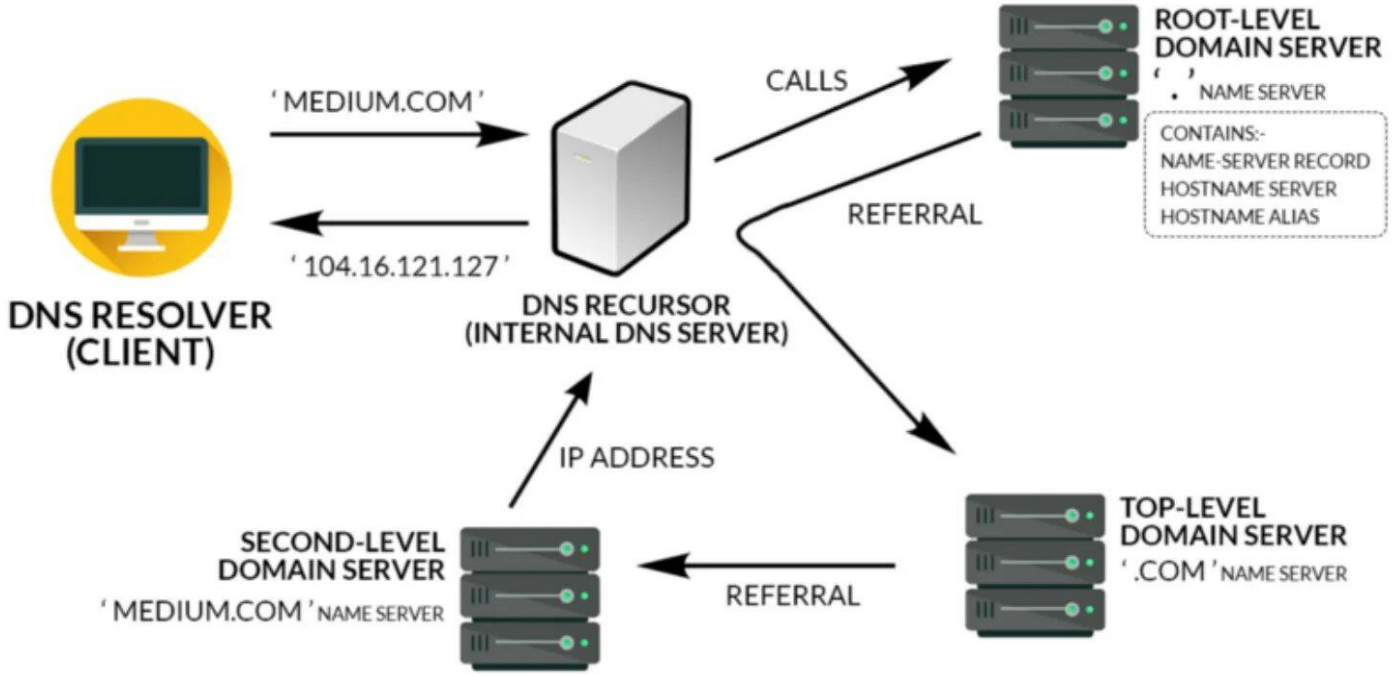
Хотя программы теоретически могут обращаться к веб-страницам, почтовым ящикам и другим ресурсам по сетевым адресам компьютеров (например, IP), на которых хранится данная информация, пользователям тяжело запоминать такие адреса.

История



ARPANET - предшественник сети INTERNET, данные хранились в файле hosts и обновлялись вручную

Появление DNS



Однако файл hosts становился большим, страдала скорость синхронизации информации, и **в 1983 году американским инженером Полом Мокапетрисом** была разработана служба имен доменов (**DNS, Domain Name System**).

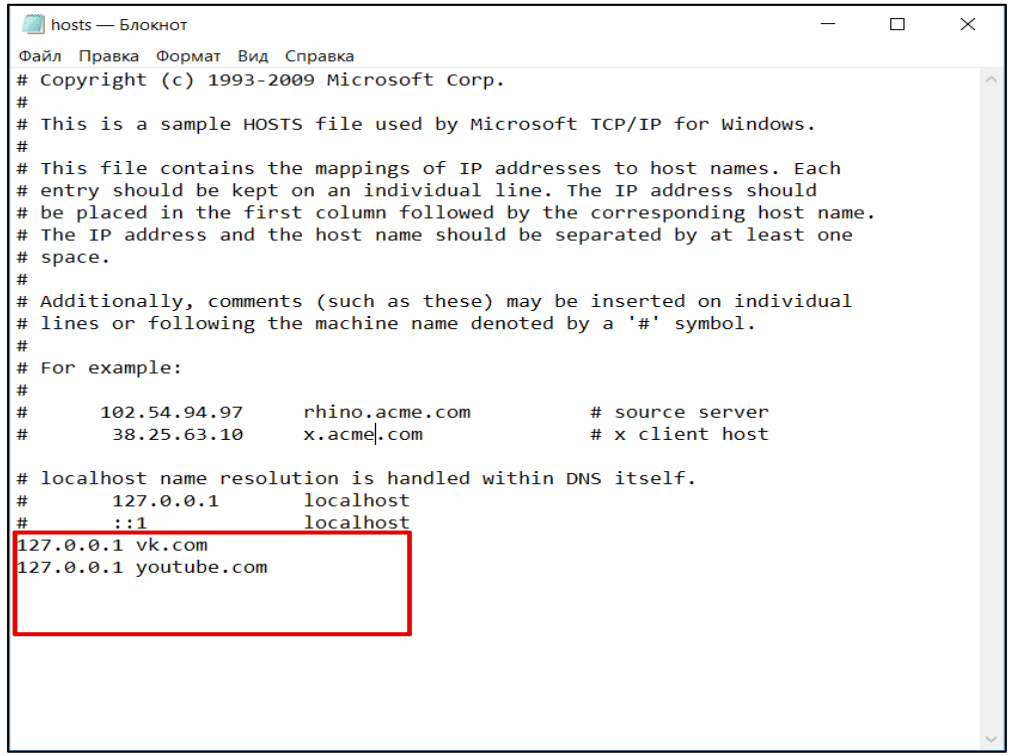
Определение дано в **RFC(Request For Comments)** 1034, 1035, 2181.

Hosts


```
; The format for entries is:
;
; NET : NET-ADDR : NETNAME :
; GATEWAY : ADDR, ADDR : NAME : CPUTYPE : OPSYS : PROTOCOLS :
; HOST : ADDR, ALTERNATE-ADDR (if any): HOSTNAME,NICKNAME : CPUTYPE :
;   OPSYS : PROTOCOLS :
;
; Where:
;;  ADDR = internet address in decimal, e.g., 10.0.0.73
;;  CPUTYPE = machine type (PDP-11/70, VAX-11/780, FOONLY-F3, C/30, etc.)
;;  OPSYS = operating system (UNIX, TOPS-20, TENEX, ITS, etc.)
;;  PROTOCOLS = transport/service (NCP/FTP,TCP/TELNET,TCP/FTP, etc.)
;;  : (colon) = field delimiter
;;  :: (2 colons) = null field

NET : 1.0.0.0 : BBN-PR-TEMP :
NET : 2.0.0.0 : SF-PR-1-TEMP :
NET : 3.0.0.0 : BBN-RCC :
NET : 4.0.0.0 : SATNET :
NET : 5.0.0.0 : DEMO-PR-1-TEMP :
NET : 6.0.0.0 : SF-PR-2-TEMP :
NET : 7.0.0.0 : BBN-NET :
```

Так выглядел файл hosts.txt, который использовался во времена ARPANET



Так выглядит hosts.txt сейчас

Особенности DNS

Распределенная система

- Нет единого сервера, на котором описываются имена хостов

Делегирование ответственности

- Пространство имен разделено на отдельные части – домены
- За каждый домен отвечает отдельная организация

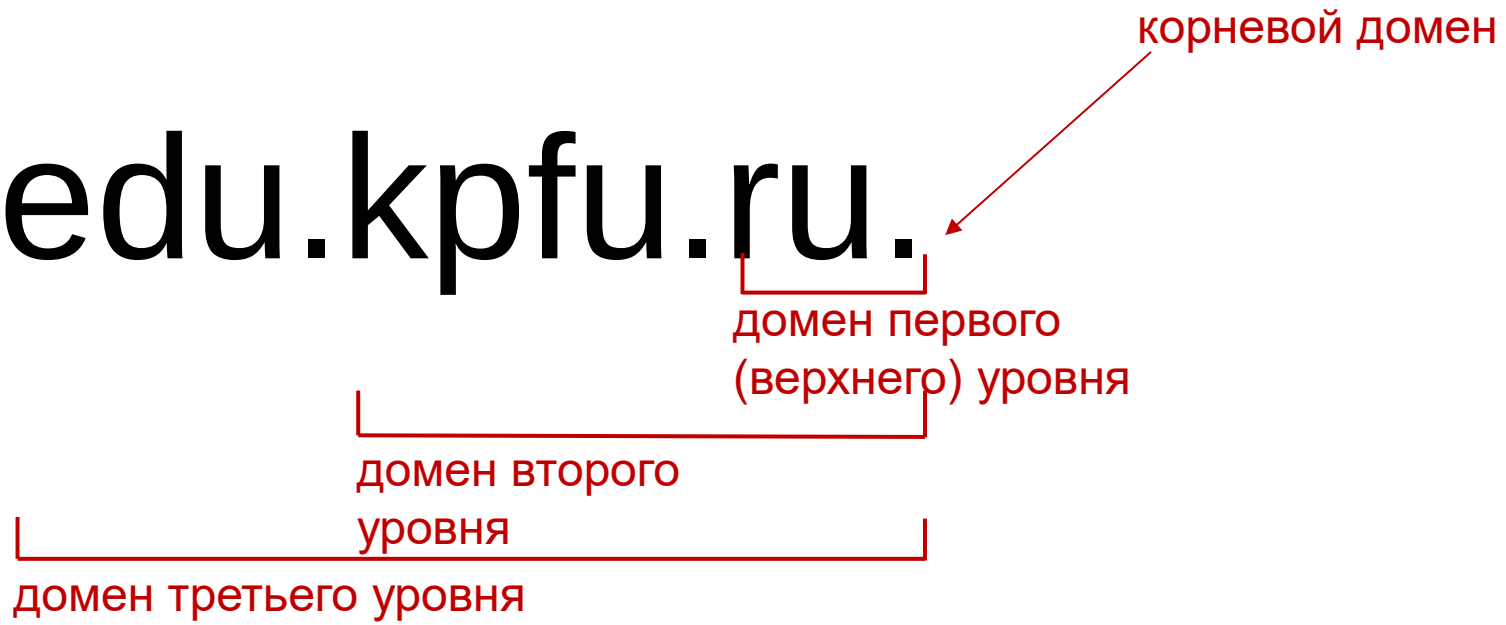
Надежность

- Дублирование серверов DNS

Основные функции DNS

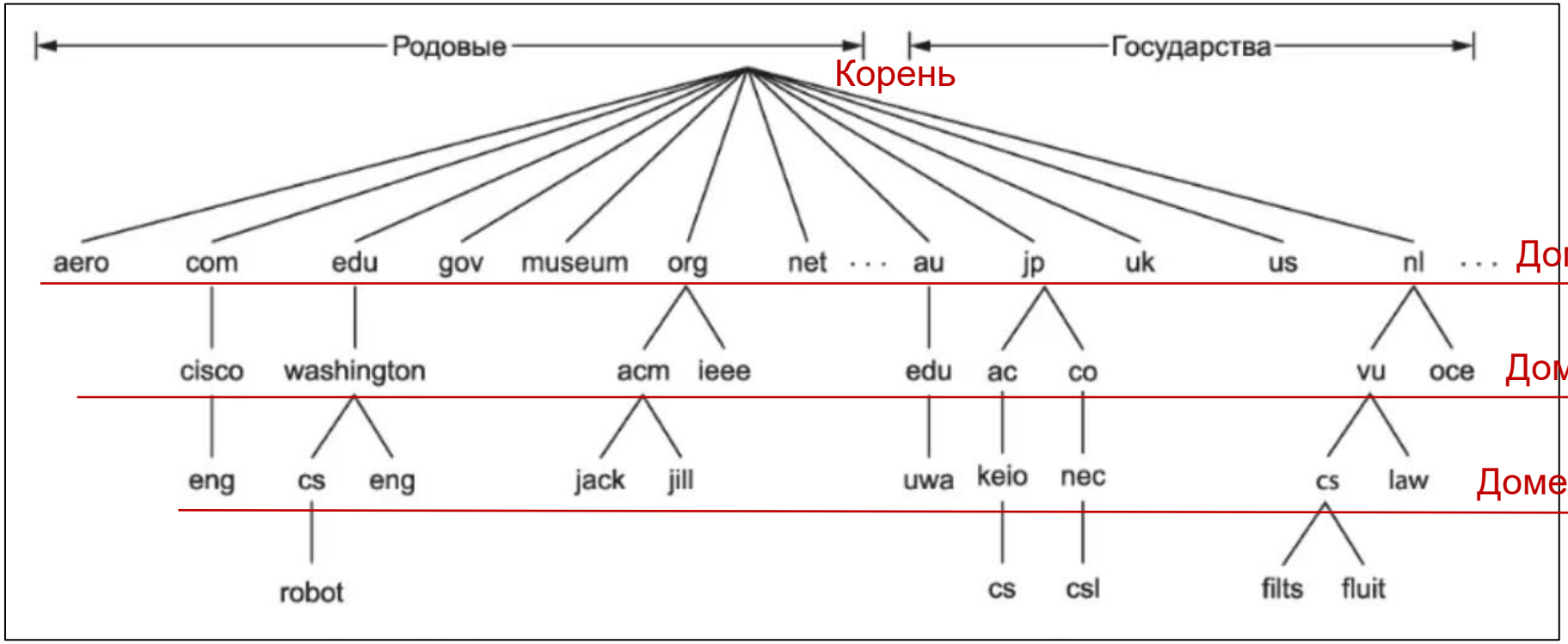
1. Преобразование доменных имен в IP-адреса. **Главная функция DNS — сопоставление понятных для человека доменных имен** с IP-адресами, которые используются компьютерами для взаимодействия в сети.
2. Обратное преобразование (Reverse DNS Lookup). Обеспечивает обратное преобразование, то есть сопоставление IP-адреса с доменным именем. Это используется, например, для проверки подлинности сервера.
3. Распределение нагрузки. DNS позволяет использовать балансировку нагрузки, предоставляя разные IP-адреса для одного и того же доменного имени в зависимости от географического положения пользователя или текущей загруженности серверов.
4. Распределённое управление именами. DNS организована как распределённая система с иерархической структурой. Это позволяет разным организациям управлять своими доменными зонами независимо друг от друга.
5. Маршрутизация почты. DNS используется для маршрутизации электронной почты с помощью записей типа MX (Mail Exchange), указывающих, куда доставлять письма для определённого домена.
6. Обеспечение стабильности и масштабируемости Интернета. Благодаря иерархической и распределённой архитектуре, DNS поддерживает работу огромного количества доменных имен, обеспечивая отказоустойчивость и высокую производительность.
7. Поддержка других сетевых сервисов. DNS помогает находить и подключаться к различным сервисам, таким как SIP для VoIP, XMPP для обмена сообщениями, и другие, используя записи типа SRV (Service).
8. Поддержка алиасов и редиректов. С помощью записей CNAME DNS позволяет создавать псевдонимы для доменных имен, чтобы несколько имен могли указывать на один и тот же ресурс.
9. Кеширование запросов. DNS-серверы хранят результаты запросов в кеше для ускорения повторных обращений и снижения нагрузки на глобальную DNS-систем.

Структура доменного имени



Имена доменов могут быть **абсолютными** и **относительными**. Абсолютное имя домена всегда оканчивается точкой, тогда как относительное имя – нет. Полные доменные имена называются FQDN, Fully Qualified Domain Name, "полностью определенное имя домена" или просто "полное имя".

Иерархия доменов



Интернет концептуально разделен на более чем 250 доменов верхнего уровня (top-level domains). Доменами называют в Интернете множество хостов, объединенных в логическую группу. Каждый домен верхнего уровня подразделяется на поддомены (subdomains), которые, в свою очередь, также могут состоять из других доменов и т. д. Все эти домены можно рассматривать в виде дерева, показанного на рисунке. Листьями дерева являются домены, не разделяющиеся на поддомены (но состоящие из хостов, конечно). Такой конечный домен может состоять из одного хоста или может представлять компанию и содержать в себе тысячи хостов.

Таблица 7.1. Родовые домены верхнего уровня

Домен	Использование	Дата основания	Ограниченный?
com	Коммерческие цели	1985	Нет
edu	Образовательные учреждения	1985	Да
gov	Правительство	1985	Да
int	Международные организации	1985	Да
mil	Военные	1985	Да
net	Сетевые провайдеры	1985	Нет
org	Некоммерческие организации	1985	Нет
aero	Авиатранспорт	2001	Да
biz	Бизнес	2001	Нет
coop	Кооперативы	2001	Да
info	Информация	2002	Нет
museum	Музеи	2002	Да
name	Люди	2002	Нет
pro	Профессионалы	2002	Да
cat	Каталония	2005	Да
jobs	Занятость	2005	Да
mobi	Мобильные устройства	2005	Да
tel	Контактная информация	2005	Да
travel	Индустрия путешествий	2005	Да

Как зарегистрировать домен?

Зарезервировать домен второго уровня, такой как имя _компании.com, просто. Домены высшего уровня управляются регистраторами (registrars), назначенными ICANN.

Для того чтобы получить имя, нужно просто обратиться к соответствующему регистратору (в данном случае com) и проверить, доступно ли желаемое имя и не является ли оно чьей-либо торговой маркой. Если все в порядке, заказчик регистрируется и за небольшую ежегодную абонентскую плату получает домен второго уровня.

Не существует каких-либо правил, запрещающих регистрацию под несколькими доменами верхнего уровня. Большие компании зачастую именно так и поступают (например, sony.com, sony.net u sony.nl).

Записи ресурсов доменов

Истинное назначение системы DNS заключается в преобразовании доменных имен в записи ресурсов. Запись ресурса состоит из пяти частей.

```
Domain_name    Time_to_live    Class    Type    Value
```

Поле **Domain _name (имя домена)** обозначает домен, к которому относится текущая запись. Является первичным ключом поиска, используемым для выполнения запросов.

Поле **Time_to_live (время жизни)** указывает, насколько стабильно состояние записи.

Третьим полем каждой записи является поле **Class (класс)**.

Поле **Type (тип)** означает тип DNS-записи. Их существует довольно много. Важные типы записей будут перечислены в следующем слайде.

Последнее поле записи ресурса — это поле **Value (значение)** - может быть числом, именем домена или текстовой ASCII-строкой. Смысл поля зависит от типа записи. Краткое описание поля Value для каждого из основных типов записей дано в таблице на следующем слайде.

Основные типы записей ресурсов доменов

Тип	Смысл	Значение
SOA	Начальная запись зоны	Параметры для этой зоны
A	IPv4-адрес хоста	Целое число, 32 двоичных разряда
AAAA	IPv6-адрес хоста	Целое число, 128 двоичных разрядов
MX	Обмен почтой	Приоритет, с которым домен желает принимать электронную почту
NS	Сервер имен	Имя сервера для этого домена
CNAME	Каноническое имя	Имя домена
PTR	Указатель	Псевдоним IP-адреса
SPF	Правила отправки почты	Правила отправки почты, закодированные в текстовом виде
SRV	Сервис	Хост, предоставляющий данный сервис
TXT	Текст	Не интерпретируемый ASCII-текст

Часть возможной базы данных домена cs.vu.nl


```
; Официальная информация для cs.vu.nl
cs.vu.nl.      86400  IN  SOA  star boss (9527,7200,7200,241920,86400)
cs.vu.nl.      86400  IN  MX   1 zephyr
cs.vu.nl.      86400  IN  MX   2 top
cs.vu.nl.      86400  IN  NS   star

star           86400  IN  A    130.37.56.205
zephyr         86400  IN  A    130.37.20.10
top            86400  IN  A    130.37.20.11
www            86400  IN  CNAME  star.cs.vu.nl
ftp            86400  IN  CNAME  zephyr.cs.vu.nl

flits          86400  IN  A    130.37.16.112
flits          86400  IN  A    192.31.231.165
flits          86400  IN  MX   1 flits
flits          86400  IN  MX   2 zephyr
flits          86400  IN  MX   3 top

rowboat            IN  A    130.37.56.201
                   IN  MX   1 rowboat
                   IN  MX   2 zephyr

little-sister      IN  A    130.37.62.23

laserjet           IN  A    192.31.231.216
```

Что такое DNS-сервер?

С точки зрения программно-аппаратных средств, DNS-сервер это:

- Выделенный сервер или компьютер, спроектированный для обработки DNS-запросов. Этот сервер может быть расположен в дата-центре, внутри организации или быть частью сетевой инфраструктуры интернет-провайдера.
- Служба DNS, которая «прослушивает» определенные сетевые порты (53 UDP и TCP) и принимает входящие DNS-запросы.
- База данных DNS-записей, содержащая информацию о доменных именах и соответствующих им IP-адресах. Записи в базе могут создаваться вручную или автоматически обновляться на основе конфигурации сервера и зон DNS.
- Кэширование полученных записей DNS. DNS-сервер временно сохраняет информацию о запросах, чтобы ускорить ответы на будущие запросы. Кэширование позволяет избегать постоянного обращения к другим DNS-серверам для одних и тех же запросов.
- Обработка запросов. DNS-сервер анализирует запрос, проверяет свою базу данных и кэш для поиска соответствующей записи DNS. Затем отправляет ответ клиенту, содержащий IP-адрес, связанный с запрошенным доменным именем.

Что такое DNS-сервер?

Выделяют 4 вида ДНС-серверов:

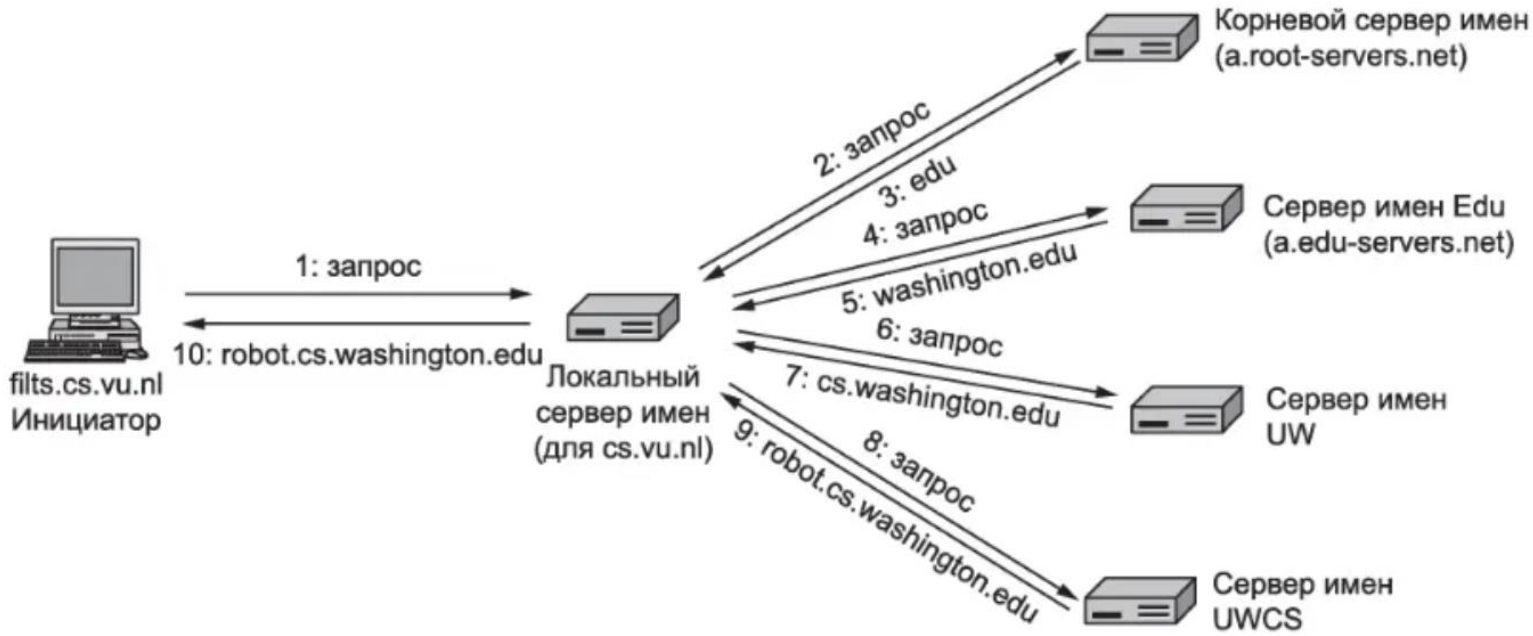
1. Рекурсивные. Обычно используются интернет-провайдерами или в рамках корпоративной сети. Они принимают DNS-запросы от клиентов и выполняют все необходимые запросы для поиска и возврата запрошенных DNS-записей.

2. Авторитетные. Содержат «авторитетную информацию» о конкретных доменных зонах и записях DNS. Они отвечают на запросы, связанные с этими зонами.

3. Кэширующие. Специализируются на кэшировании записей DNS и ускоряют процесс обработки запросов, так как могут предоставлять кэшированные ответы, если они имеют соответствующие записи.

4. Публичные. Предоставляются общедоступными службами (например, Google Public DNS или OpenDNS). Они могут использоваться для улучшения скорости и безопасности подключения к интернету.

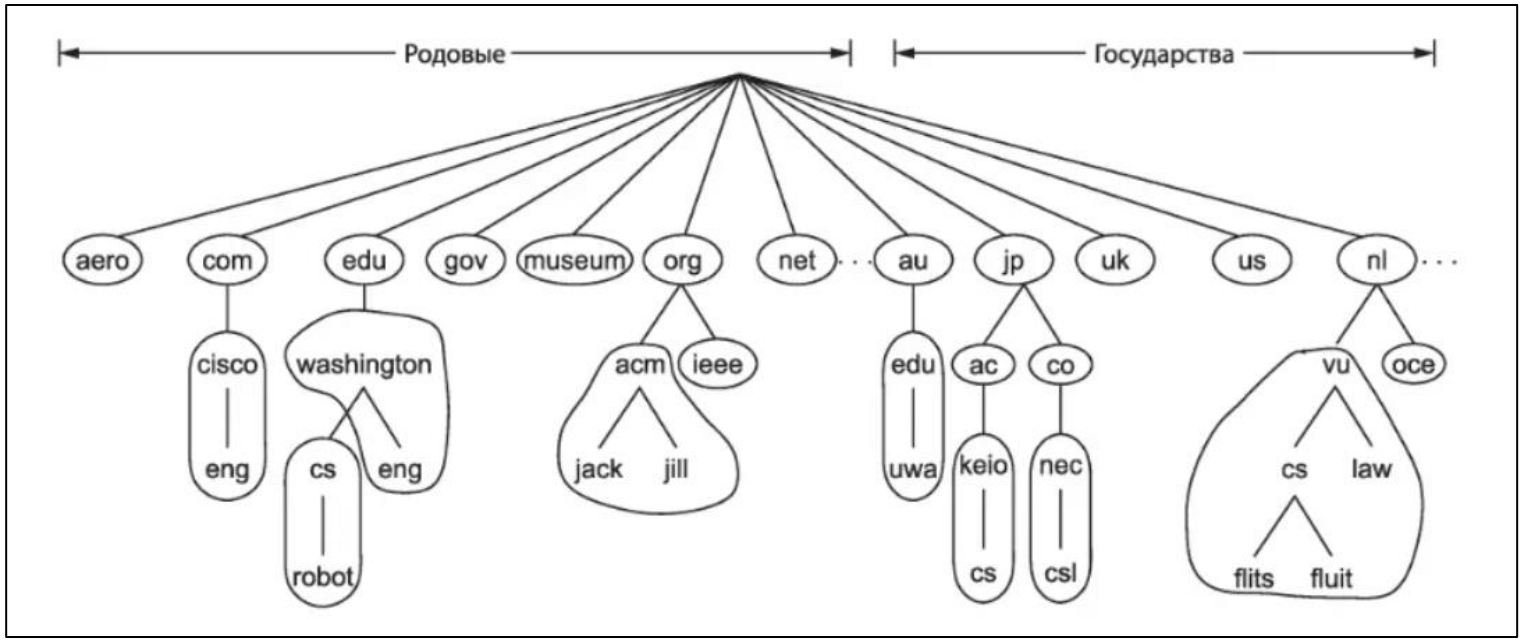
Типы DNS-запросов



На рисунке используется два разных механизма запроса. Когда хост fits.cs.vu.nl отсылает запрос на локальный сервер имен, этот сервер выполняет запрос от имени flits, пока не получит ответ, который можно будет вернуть. Он не возвращает частичных ответов. Они могут быть полезными, но в запросе о них нет ни слова. Этот механизм называется **рекурсивным запросом** (recursive query).

С другой стороны, корневой сервер имен (и каждый последующий) не продолжает рекурсивно запрос локального сервера имен. Он возвращает лишь частичный ответ и переходит к следующему запросу. Локальный сервер имен отвечает за продолжение поиска ответа, направляя следующие запросы. Этот механизм называется **итеративным запросом** (iterative query).

Серверы имен



Теоретически, один сервер мог бы содержать всю базу данных DNS и отвечать на все запросы к ней. На практике этот сервер оказался бы настолько перегруженным, что был бы просто бесполезным. Более того, если бы с ним когда-нибудь что-нибудь случилось, то весь Интернет не работал бы.

Чтобы избежать проблем, связанных с хранением всей информации в одном месте, пространство имен DNS разделено на **непересекающиеся зоны (zones).** Один возможный способ разделения пространства имен на зоны изображен на рисунке. Каждая очерченная зона содержит часть общего дерева доменов.

Каждая зона также ассоциируется с одним или более сервером имен. Это хосты, на которых находится база данных для зоны. Обычно у зоны есть один основной сервер имен, который получает информацию из файла на своем диске, и один или более второстепенных серверов имен, получающих информацию с основного сервера имен. Для повышения надежности некоторые серверы имен могут быть расположены вне зоны.

Реверсная зона

Иногда возникает потребность произвести обратное преобразование: по IP-адресу узнать доменное имя хоста. Раньше это считалось как-то прям даже обязательным, но сейчас народ несколько подрасслабился и перестал за этим следить.

Это делается с помощью так называемых **реверсных зон**: записей DNS, в которых ключом поиска является IP-адрес, а ответом является имя.

Но поскольку в философии DNS все есть домен, то IP-адреса приходится записывать в виде псевдодоменов:

- 44.33.22.11.in-addr.arpa - для IPv4, адрес был 11.22.33.44
- 4.4.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.3.0.0.2.2.0.0.1.1.0.0.ip6.arpa - для IPv6, адрес был 11:22:33::44

Обратите внимание на запись адреса задом наперед и на использование специальных служебных псевдодоменов in-addr.arpa и ip6.arpa.

Реверсной зоной распоряжается та же организация, которая распоряжается соответствующим диапазоном IP адресов.

Процедуры разрешения имен. Итеративная

Процесс поиска адреса по имени называется **разрешением имен (name resolution).**

Существуют две основные схемы разрешения DNS-имен. В первом варианте работу по поиску IP-адреса координирует DNS-клиент:

- DNS-клиент обращается к корневому DNS-серверу с указанием полного доменного имени
- DNS-сервер отвечает, указывая адрес следующего DNS-сервера, обслуживающего домен верхнего уровня, заданный в старшей части запрошенного имени;
- DNS-клиент делает запрос следующего DNS-сервера, который отсылает его к DNS-серверу нужного поддомена, и т. д., пока не будет найден DNS-сервер, в котором хранится соответствие запрошенного имени IP-адресу. Этот сервер дает окончательный ответ клиенту.

Такая схема взаимодействия называется **нерекурсивной или итеративной**, когда клиент сам итеративно выполняет последовательность запросов к разным серверам имен. Так как эта схема загружает клиента достаточно сложной работой, она применяется редко.

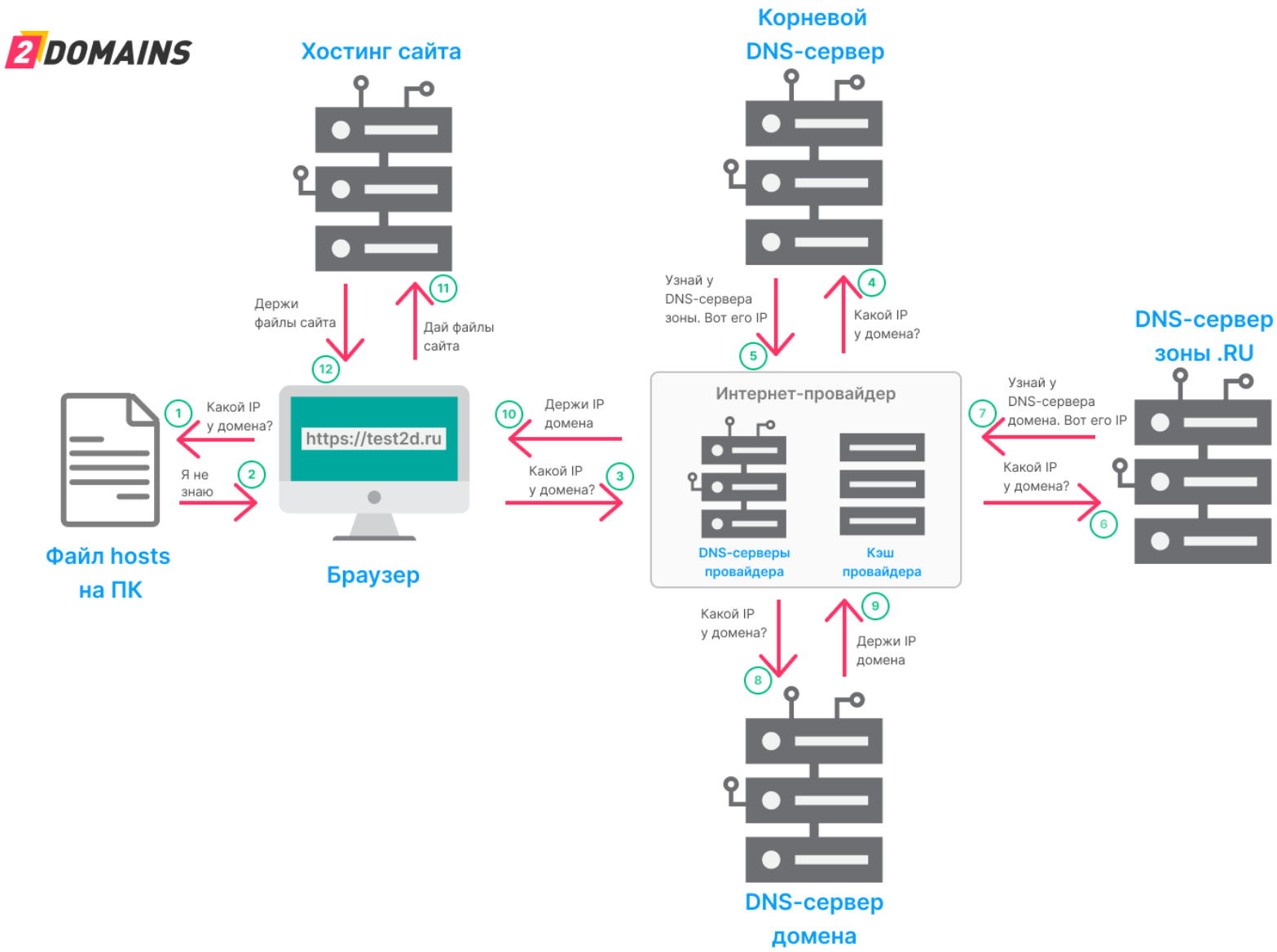
Процедуры разрешения имен. Рекурсивная

Во втором варианте реализуется **рекурсивная** процедура:

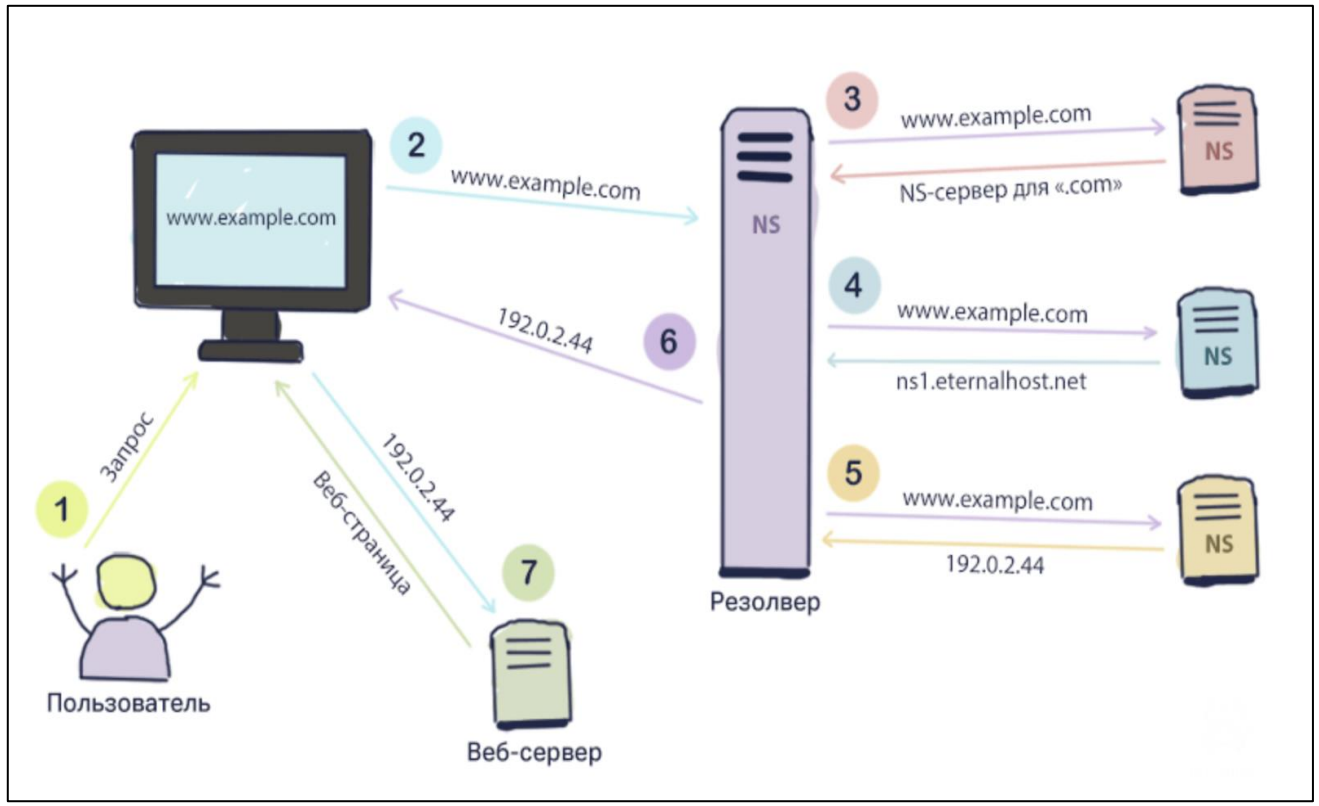
- DNS-клиент запрашивает локальный DNS-сервер, то есть тот сервер, который обслуживает поддомен, к которому принадлежит имя клиента;
- если локальный DNS-сервер знает ответ, то он сразу же возвращает его клиенту; это может соответствовать случаю, когда запрошенное имя входит в тот же поддомен, что и имя клиента, а также может соответствовать случаю, когда сервер уже узнавал данное соответствие для другого клиента и сохранил его в своем кэше;
- если же локальный сервер не знает ответ, то он выполняет итеративные запросы к корневому серверу и т. д. точно так же, как это делал клиент в первом варианте; получив ответ, он передает его клиенту, который все это время просто ждал его от своего локального DNS-сервера.

В этой схеме клиент перепоручает работу своему серверу, поэтому схема называется **косвенной или рекурсивной**. Практически все DNS-клиенты используют рекурсивную процедуру.

Процедуры разрешения имен. Рекурсивная



Рекурсивные резолверы и кеширование



Рекурсивные резолверы играют ключевую роль в работе DNS-системы. Когда клиент, например браузер, отправляет запрос на перевод доменного имени в IP-адрес, рекурсивный резолвер начинает процесс поиска нужной информации. Он действует как посредник, который обращается к различным уровням DNS-иерархии: сначала к корневым серверам, затем к серверам доменов верхнего уровня (TLD) и, наконец, к авторитетным серверам, содержащим окончательные записи. Рекурсивный резолвер выполняет всю тяжелую работу по поиску IP-адреса, возвращая клиенту уже готовый ответ.

Чтобы оптимизировать этот процесс и уменьшить задержки, рекурсивные резолверы используют кеширование. Когда резолвер получает ответ от авторитетного сервера, он сохраняет эту информацию в своей памяти на время, определенное параметром TTL (Time to Live). Кеширование позволяет значительно сократить количество запросов к внешним серверам, так как повторные запросы на тот же домен будут обслуживаться из кеша, что обеспечивает более быстрый доступ к часто запрашиваемым ресурсам и уменьшает нагрузку на инфраструктуру DNS.

DNS как протокол

DNS принадлежит к семейству интернет-протоколов и модели OSI (а именно **прикладному уровню**). Он по умолчанию содержится во всех операционных системах, чтобы компьютер мог общаться с другими машинами по сети.
DNS-протокол работает на основе клиент-серверной архитектуры. Использует **TCP и UDP** для обмена данными между клиентами и DNS-серверами, использует **порт 53**.

DNS использует оба этих протокола в зависимости от типа запроса. Большинство DNS-запросов используют UDP, так как он обеспечивает быструю и эффективную передачу данных без установления соединения. Например, когда вы вводите URL в браузере, ваш компьютер отправляет DNS-запрос через UDP, и DNS-сервер отвечает с минимальной задержкой. Однако для более крупных запросов или зоновых передач (zone transfers) используется TCP, так как он обеспечивает надежность и гарантирует доставку всех пакетов данных.

UDP является более легковесным протоколом по сравнению с TCP. Он не требует установления соединения перед передачей данных, что делает его более быстрым и эффективным для небольших запросов, таких как DNS-запросы. Однако UDP не обеспечивает гарантии доставки данных, что может быть проблемой для более крупных или критически важных запросов.

TCP, с другой стороны, обеспечивает надежную передачу данных, устанавливая соединение между отправителем и получателем перед началом передачи. Он использует механизмы подтверждения и повторной передачи для обеспечения целостности данных. В контексте DNS TCP используется для зоновых передач, когда необходимо передать большие объемы данных между DNS-серверами.

Кто поддерживает DNS-сервера?

Например, сервер a.root-servers.net имеет IP-адрес 198.41.0.4 и находится в ведении компании. Когда вы вводите адрес интернет-ресурса в строку браузера, он отправляет запрос на DNS-сервер отвечающий за корневую зону. Таких серверов 13 и они управляются различными операторами и организациями.

Каждый из этих операторов предоставляет данную услугу бесплатно, а также обеспечивает бесперебойную работу, поскольку при отказе любого из этих серверов станут недоступны целые зоны интернета. Ранее корневые DNS-серверы, являющиеся основой для обработки всех запросов о доменных именах в интернете, располагались в Северной Америке. Однако с внедрением технологии альтернативной адресации они «распространились» по всему миру, и фактически их число увеличилось с 13 до 123, что позволило повысить надёжность фундамента DNS.

Например, в Северной Америке находятся 40 серверов (32,5%), в Европе – 35 (28,5%), еще 6 серверов располагаются в Южной Америке (4,9%) и 3 – в Африке (2,4%). Если взглянуть на карту, то DNS-серверы расположены согласно интенсивности использования интернет-инфраструктуры.

Утилиты тестирования работы службы DNS

1. nslookup (Windows, Linux, macOS)

Используется для выполнения запросов к DNS-серверам. Выводит IP-адрес домена.

Пример:

```
nslookup example.com
```

2. dig (Linux, macOS)

Мощная утилита для проверки работы DNS. Показывает детальную информацию о запросах и ответах. Выводит данные об IP, алиасах, сервере, времени запроса и т.д.

Пример:

```
dig example.com
```

3. host (Linux, macOS)

Простой инструмент для разрешения доменных имен и IP-адресов.

Пример:

```
host example.com
```

4. ping (Windows, Linux, macOS)

Проверяет доступность домена или IP-адреса.

Пример:

```
ping example.com
```

Утилиты тестирования работы службы DNS

5. traceroute (Linux, macOS) / tracert (Windows)

Отслеживает путь к домену через маршрутизаторы, что позволяет выявить проблемы с маршрутизацией.

Пример:

```
traceroute example.com
```

6. tcpdump или wireshark

Используются для анализа сетевого трафика, включая запросы и ответы DNS.

7. systemd-resolve (Linux, systemd)

Инструмент для тестирования и управления настройками DNS.

Пример:

```
systemd-resolve example.com
```

8. ipconfig(Windows)

(Internet Protocol Configuration) — это утилита, которая отображает текущую сетевую конфигурацию компьютера в Windows. Просмотр IP-адреса, маски подсети, шлюза по умолчанию и DNS-серверов.

Как DNS работает на практике?

Давайте посмотрим, как DNS работает на практике, используя команду `nslookup`.

Допустим, мы хотим найти IP-адрес для `google.com`. Мы можем использовать команду `nslookup` для поиска IP-адреса:

```
nslookup google.com
```

Вывод будет выглядеть примерно так:

```
Server:  UnKnown
Address:  192.168.1.1

Non-authoritative answer:
Name:    google.com
Addresses:  2a00:1450:4007:807::200e
          142.250.75.238
```

Как DNS работает на практике?

Для отслеживания маршрута до сервера можно использовать `tracert` :

```
tracert google.com
```

Вывод будет выглядеть примерно так:

```
Tracing route to google.com [216.58.214.78]
over a maximum of 30 hops:

  1     1 ms     1 ms     2 ms  192.168.1.1
  2     3 ms     2 ms     2 ms  10.218.0.114
  3     2 ms     2 ms     2 ms  cable-pppoe-89-251-153-134.kzn.hitv.ru
[89.251.153.134]
       … …
 25    62 ms    63 ms    62 ms  142.250.224.93
 26    62 ms    62 ms    62 ms  par10s39-in-f14.1e100.net [216.58.214.78]

Trace complete.
```

Как DNS работает на практике?

Можно использовать команду `ping` для проверки доступности домена или IP-адреса:

```
ping google.com
```

Вывод будет выглядеть примерно так:

```
Pinging google.com [142.250.75.238] with 32 bytes of data:
Reply from 142.250.75.238: bytes=32 time=62ms TTL=112
Reply from 142.250.75.238: bytes=32 time=61ms TTL=112
Reply from 142.250.75.238: bytes=32 time=62ms TTL=112
Reply from 142.250.75.238: bytes=32 time=61ms TTL=112

Ping statistics for 142.250.75.238:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 61ms, Maximum = 62ms, Average = 61ms
```

Как работает DNS на практике? 31

Извлечение записей DNS для домена

Вы можете использовать `dig` для получения различных типов записей DNS для доменного имени. Например, чтобы получить все A записи для домена, вы можете использовать следующую команду:

```
dig example.com A
```

Это вернёт список всех `A` записей, связанных с доменом example.com.

Аналогично, чтобы получить все `MX` записи для домена, вы можете использовать следующую команду:

```
dig example.com MX
```

Как очистить DNS кэш?

Бывают случаи, когда необходимо очистить кэш DNS на локальной машине. Вы можете использовать команду `ipconfig` для этого в Windows и команду `dscacheutil` для этого в MacOS.

Вот команда для очистки кэша DNS в Windows:

```
ipconfig /flushdns
```

Если кэш успешно очищен, вы увидите сообщение:

```
Successfully flushed the DNS Resolver Cache.
```

В MacOS можно использовать следующую команду:

```
dscacheutil -flushcache
```

Тест

Вопрос 1

Какую **основную** функцию выполняет система DNS?

a. Хранение файлов в интернете
b. Преобразование доменных имен в IP-адреса
c. Преобразование IP-адресов в доменные имена
d. Передача данных между серверами

Вопрос 2

Какой тип DNS-записи используется для указания IPv4-адреса?

a. AAAA
b. A
c. MX
d. TXT

Вопрос 3

Какая запись используется для указания почтового сервера?

a. TXT
b. CNAME
c. A
d. MX

Вопрос 4

Что такое реверсная зона?

a. Зона, которая используется для хранения записей, связывающих доменные имена с IP-адресами.
b. Зона, которая используется для хранения PTR-записей, связывающих IP-адреса с доменными именами.
c. Зона, которая используется для хранения MX-записей, указывающих на почтовые серверы.
d. Зона, которая используется для хранения SRV-записей, указывающих на сервисы.

Вопрос 5

Что происходит, если DNS-запись отсутствует в кэше?

a. Пользователь получает сообщение об ошибке
b. Запрос перенаправляется на корневой сервер
c. Сервер создает новую запись автоматически
d. Запрос остается необработанным

Вопрос 6

Какой порт используется для работы DNS по умолчанию?

a. 25
b. 53
c. 443
d. 80

Вопрос 7

Что такое TTL (Time To Live) в DNS?

a. Время, за которое запрос должен быть обработан
b. Время жизни DNS-записи в кэше
c. Максимальное время работы сервера DNS
d. Интервал между обновлениями зон в DNS

Вопрос 8

Какой из этих доменов является доменом верхнего уровня?

a. google.com
b. .org
c. mail.google.com
d. www.google.com

Вопрос 9

Что происходит при отсутствии ответа от DNS-сервера?

a. Запрос направляется к корневому серверу
b. Происходит кэширование пустого ответа
c. Пользователь получает сообщение об ошибке
d. Запрос повторяется бесконечно

Вопрос 10

Какой из перечисленных серверов отвечает за хранение оригинальных записей домена?

a. Рекурсивный сервер
b. Кеширующий сервер
c. Авторитетный сервер
d. Прокси-сервер

Вопрос 11

Какой формат используется для передачи DNS-запросов и ответов?

a. JSON
b. XML
c. UDP или TCP
d. HTTPS

Вопрос 12

Для чего используется команда `nslookup`?

a. Для проверки скорости интернет-соединения
b. Для проверки DNS-записей и диагностики DNS
c. Для настройки сетевого интерфейса
d. Для сканирования открытых портов на удалённом сервере

Вопрос 13

Чем отличается рекурсивная процедура разрешения DNS-имен от итеративной?

a. В рекурсивной процедуре DNS-клиент сам выполняет последовательные запросы к разным DNS-серверам, а в итеративной — перекладывает эту задачу на локальный DNS-сервер.
b. В рекурсивной процедуре локальный DNS-сервер выполняет все запросы к другим DNS-серверам, а клиент только ожидает ответа, тогда как в итеративной процедуре клиент сам выполняет последовательные запросы.
c. Рекурсивная процедура требует больше времени для выполнения, чем итеративная, так как DNS-серверы обрабатывают запросы медленнее.
d. Итеративная процедура используется только для запросов к корневым DNS-серверам, а рекурсивная — для всех остальных случаев.

Спасибо за внимание!