NETWORKING!

Журнал о компьютерных сетях

ACK!

Юля Эванс

Действующие лица

У тебя дома

Твой ноутбучек (все картинки котиков там)

Твоя программа

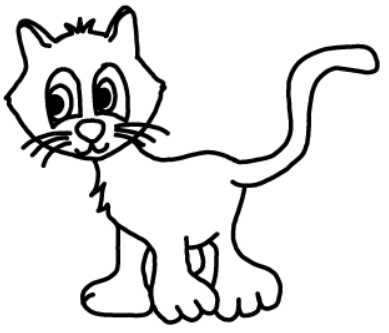
Операционная система (знает, как работать с сетью)

Твой домашний маршрутизатор

С этими компьютерами мы будем общаться

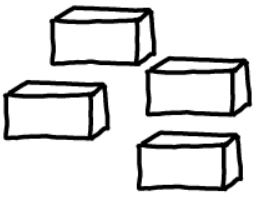
Сервер (тут лежит картинка котика)

DNS-сервер (знает на каком сервере хостится jvns.ca)

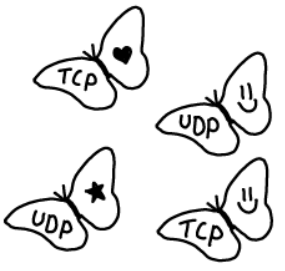


Та самая картика котика, которую мы хотим скачать

Между всем этим



Промежуточные маршрутизаторы в интернете



Пакеты!

Што здесь?!

Привет! Я – Джулия

Твиттер: @bOrk

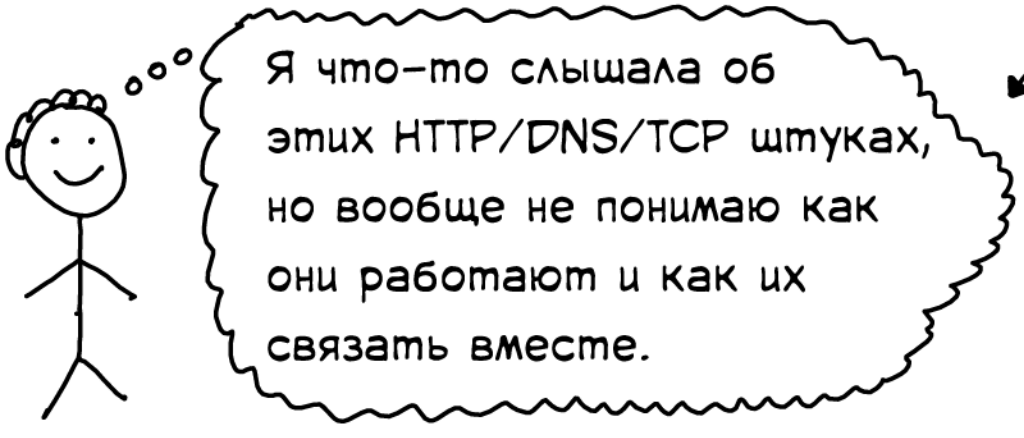
Бложик: http://jvns.ca

Я выложила картинку котика в интернете по этому адресу:

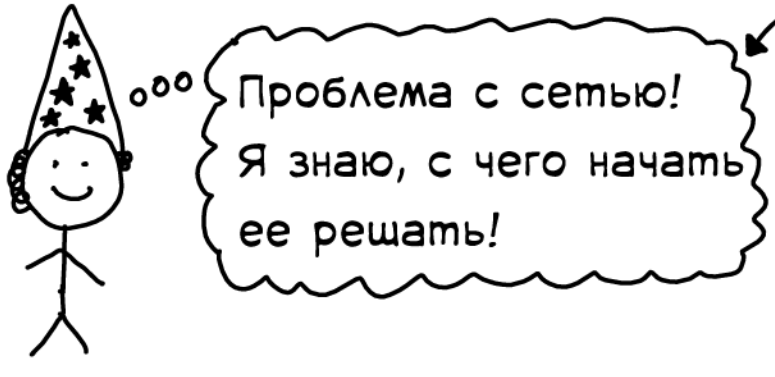
★ jvns.ca/cat.png ★ (можно смотреть!)

Из этого журнала ты узнаешь ВСЁ (но это не точно), что должно произойти, чтобы эта картинка попала с моего сервера к тебе на компьютер.

Моя задача помочь тебе перестать думать так...



И начать думать вот так...



Наш главный герой: ПАКЕТ

Все данные в интернете передаются в ПАКЕТАХ. Пакет – это последовательность бит (0100101110 11....), разделенная на секции (или "заголовки").

Вот как выглядит UDP-пакет, в котором написано "mangotea". В нем всего 50 байт! (400 бит)

84 бита

Входящий MAC	Исходящий MAC	Тип

Заголовок кадра локальной сети (14 байт)

4 байта (32 бита)

ver	hlen	TOS	Длина пакета
идентификация		flg	Смещение фрагмента
TTL	протокол	Контрольная сумма заголовка	
Исходящий IP-адрес			
Входящий IP-адрес			

160 бит
Заголовок IP 20 байт

Тут вся информация о том, на какой IP-адрес маршрутизатору нужно отправить пакет.

64 бита

Исходящий порт	Входящий порт
длина	Контрольная сумма UDP

Заголовок UDP 8 байт

(у TCP-пакета тут был бы TCP-заголовок)

m	a	n	g
o	t	e	a

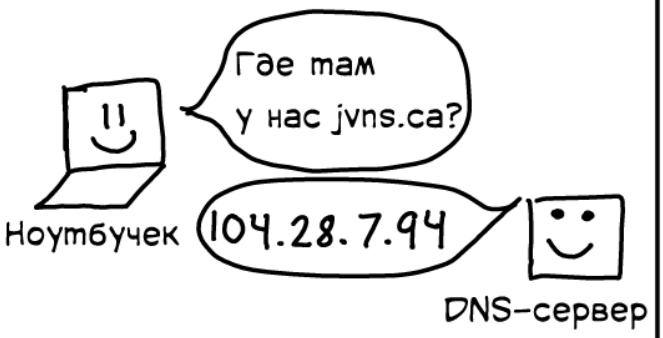
Сюда заезжает все содержимое пакета. Каждый символ ASCII занимает 1 байт, поэтому "mangotea" = 8 байт
64 бита

Необходимые действия, чтобы скачать картинку котика

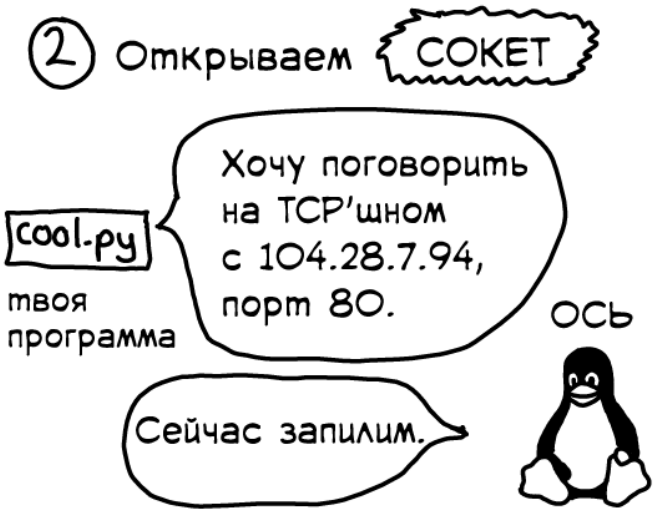
Конечно же с jvns.ca/cat.png

Когда ты загружаешь картинку, в сети выполняется ОЧЕНЬ много операций. Вот основные действия, в которых мы разберемся на следующих страницах.

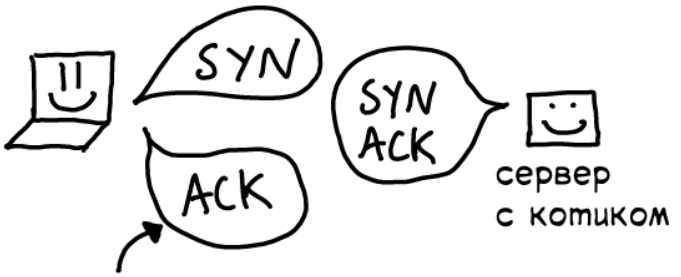
1 Определяем IP-адрес jvns.ca



2 Открываем СОКЕТ

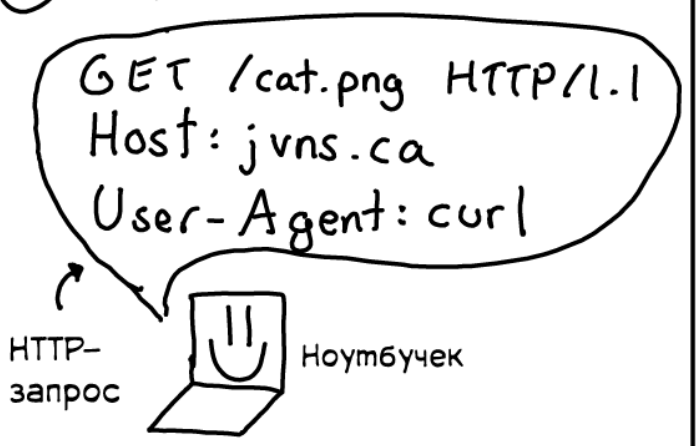


3 Открываем TCP-соединение к 104.28.7.94, порт 80

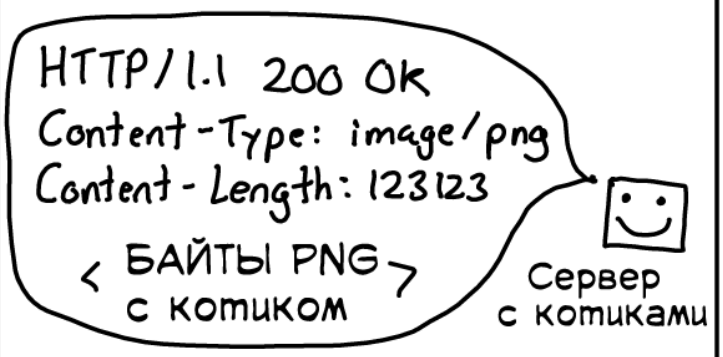


Это называется "рукопожатие"
Разберемся с этим на странице про TCP.

4 Запрашиваем котика



5 Получаем котика

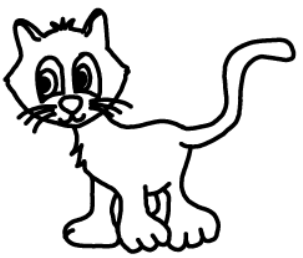


6 Прибираемся

-> Закрываем соединение, но это не точно,

-> складываем байты от PNG в файл, но это не точно,

-> смотрим на котика, инфа 100%.

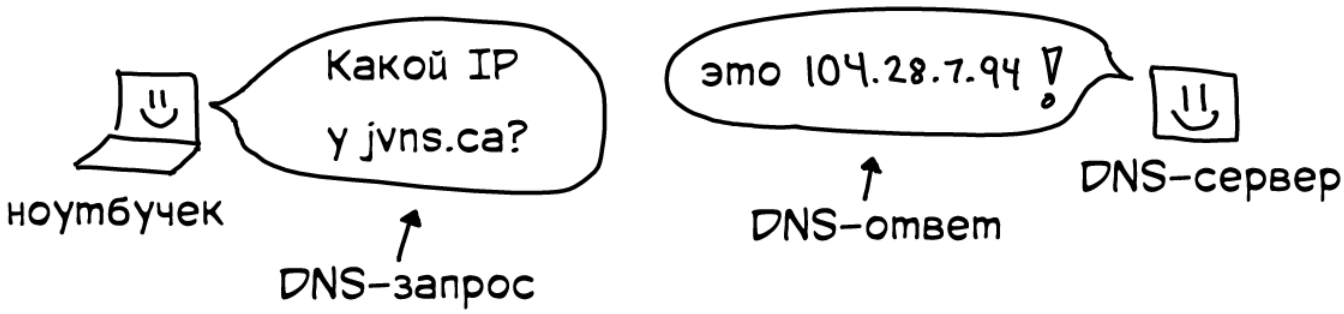


DNS

Шаг ① : определяем IP-адрес jvns.ca.

Вся сетевая активность происходит через передачу пакетов. Чтобы отправить пакет на сервер в интернете, тебе нужно знать его {IP-АДРЕС}, например 104.27.7.94.

Jvns.ca и firstvds.ru – это доменные имена. DNS ("Domain Name System", "Система Доменных Имен") – протокол, которым мы воспользуемся для определения IP-адреса доменного имени.



Обычно и DNS-запрос, и DNS-ответ это UDP-пакеты.

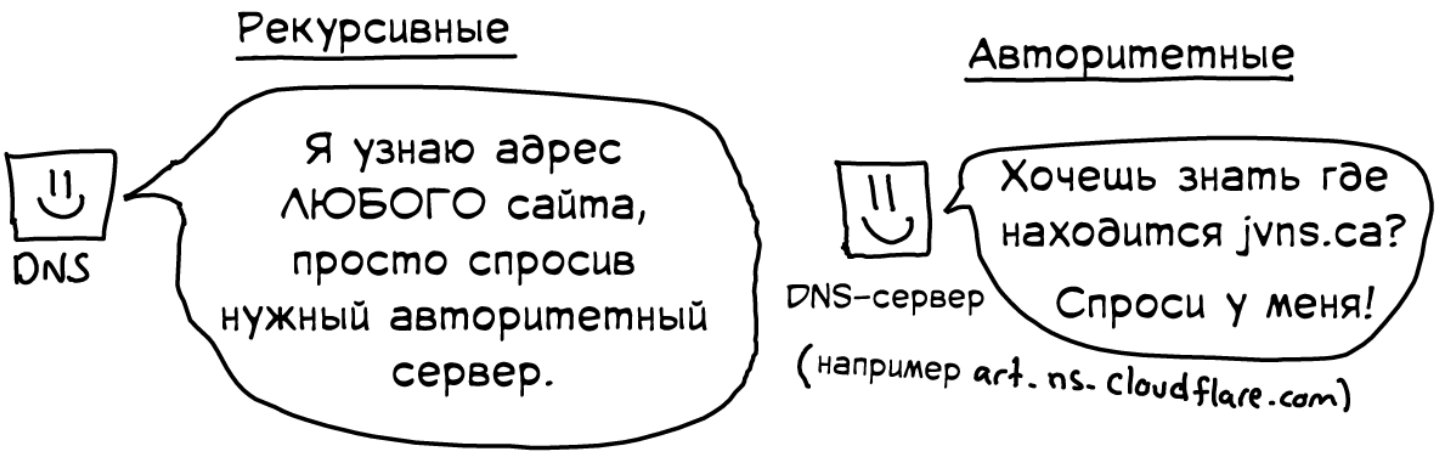
Когда ты запускаешь `$ curl jvns.ca/cat.png` :

Curl вызывает функцию "getaddrinfo" со значением "jvns.ca".	Getaddrinfo определяет системный DNS-сервер (например 8.8.8.8).	Getaddrinfo отправляет этому серверу запрос.	IP-адрес получен! 104.28.7.94

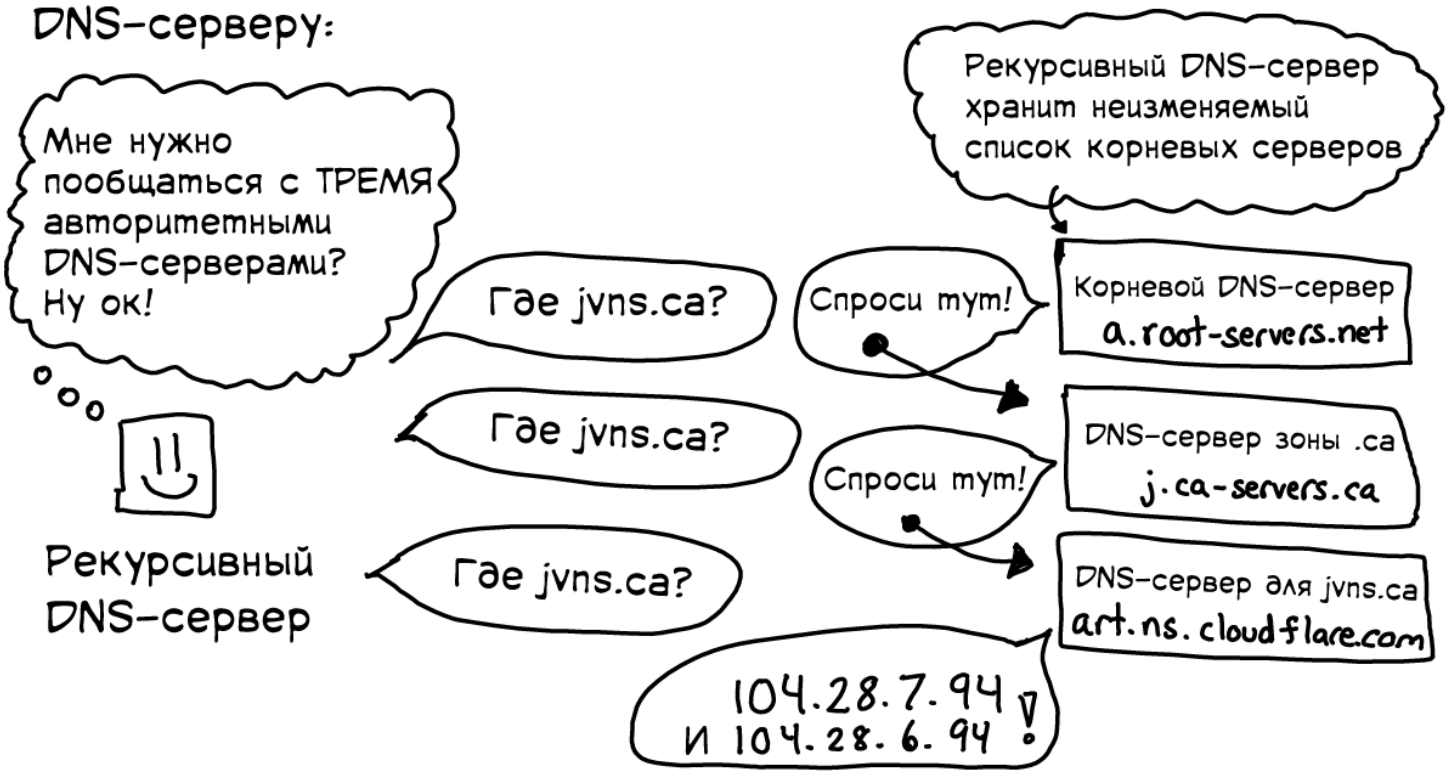
DNS-сервер по умолчанию обычно прописан в /etc/resolv.conf.

8.8.8.8 – это DNS-сервер Гугла и им пользуется куча народа. Отличный выбор для большинства задач!

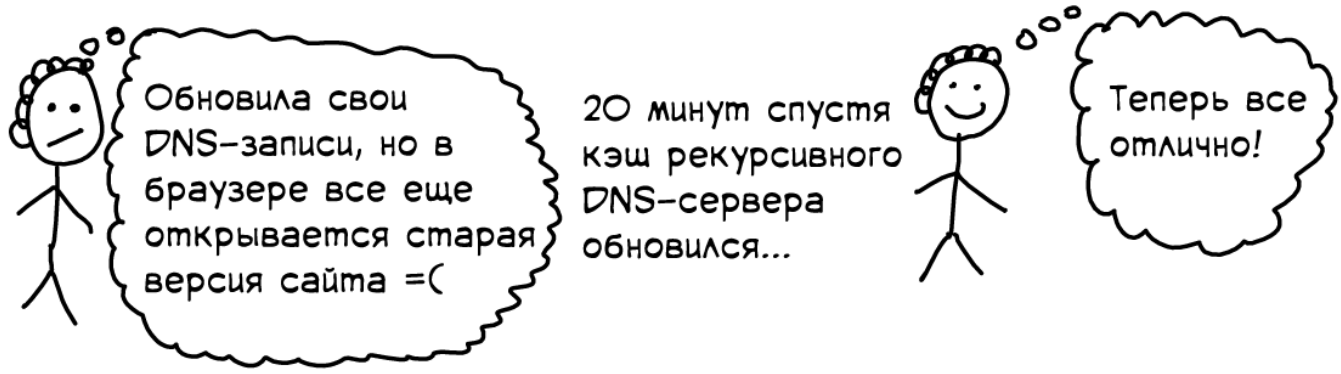
DNS-серверы бывают 2-х видов:



Вот что произойдет, если отправить запрос к рекурсивному DNS-серверу:

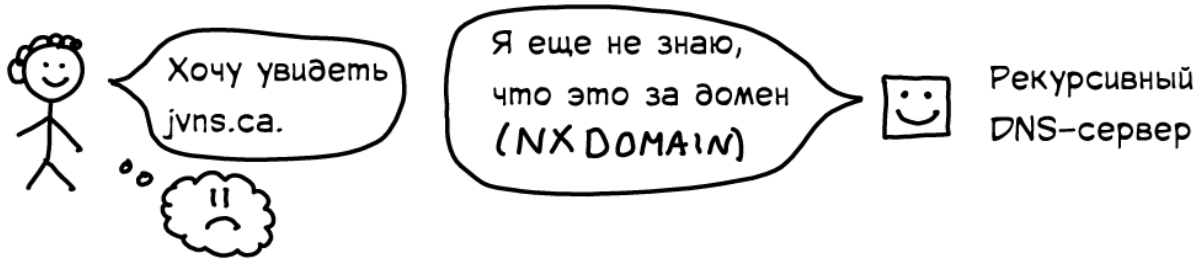


Рекурсивные DNS-серверы обычно кэшируют записи. У каждой DNS-записи есть TTL ("Time to Live", "время жизни"), который определяет время ее хранения в кэше. Обычно их нельзя заставить обновить свой кэш, придется просто ждать:



Давай делать ♡ DNS-запросы ♡

Когда настраиваешь DNS для нового домена, частенько случается следующее:



Чтобы понять, что происходит, можно отправлять DNS-запросы из командной строки:

```
$ dig jvns.ca

;; ANSWER SECTION
jvns.ca   268 IN A 104.28.6.94
jvns.ca   268 IN A 104.28.7.94

;; SERVER 127.0.1.1#53
```

У одного домена может быть МНОГО IP-адресов.

Эта запись истекает через 268 секунд.

"A" запись это IP-адрес.

DNS-сервер, который я использую.

```
$ dig @8.8.8.8 jvns.ca
```

8.8.8.8 это рекурсивный DNS-сервер Google. @8.8.8.8 отправляет запрос этому серверу, вместо дефолтного.

```
$ dig +trace jvns.ca

.          502441 IN NS   h.root-servers.net
ca.        172800 IN NS   c.ca-servers.net
jvns.ca.   86400  IN NS   art.ns.cloudflare.com
jvns.ca.   300    IN A    104.28.6.94
```

корневой DNS-сервер!

Dig +trace делает практически то же самое, что и рекурсивный DNS-сервер, когда нужно определить IP твоего домена.

Это те 3 авторитетные сервера с которыми должен связаться рекурсивный сервер, чтобы определить IP для jvns.ca.

СОКЕТЫ

Шаг (2): теперь когда у нас есть IP-адрес, нам нужно открыть сокет! Давай узнаем, что это такое.

Твоя программа ничего не знает о TCP.

ХБЗ что такое TCP, мне просто нужно открыть страницу.

code.py
программа

Не боись, я могу помочь!

ОС

Чтобы использовать сокет мы:

Шаг 1: просим сокет у ОС.

Шаг 2: подключаем сокет к IP-адресу и порту.

Шаг 3: пишем в сокет, чтобы отправить данные.

4 стандартных вида сокетов

TCP
для TCP

UDP
для UDP

raw
для ЗАПРЕДЕЛЬНОЙ ВЛАСТИ. Ping пользует такой сокет, чтобы слать ICMP пакеты.

unix
чтобы общаться с программами на одном компьютере.

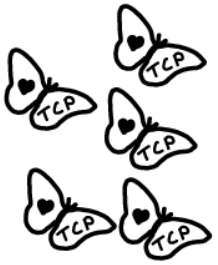
Когда ты соединяешься с TCP-сокетом.

(разберемся в том, что такое SYN ACK чуть позже)

Когда ты записываешь данные в сокет

code.py
программа
-> записывает гору данных ♥♥♥♥♥

делит данные на пакеты, чтобы отправить.



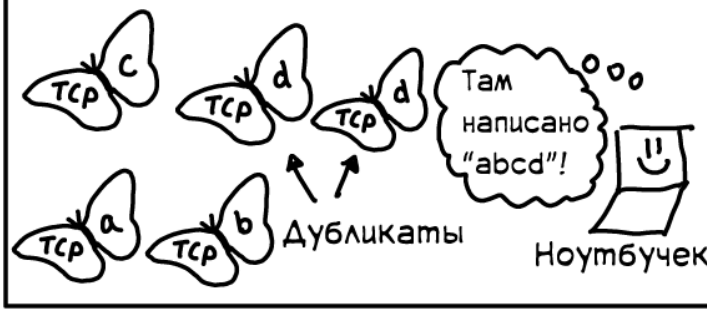
Интерфейс сокетов просто офигенен! Операционная система делает для меня очень много.

TCP: как гарантированно получить котика

Шаг ③ в нашем плане это "открываем TCP-соединение!". Выясним, что это же за штука такая "TCP".

Иногда пакеты теряются в интернете.

TCP поможет надежно передать поток данных, даже если пакеты потеряются или придут в случайном порядке.



Так ты хочешь знать, как работает TCP? Ну вот так!

Как узнать, в каком порядке должны приходить пакеты:

В каждом пакете есть информация о диапазоне находящихся в нем байт.

Вот такая:

once upon a ti ← байты 0-13
agical oyster ← байты 30-42
me there was a m ← байты 14-29

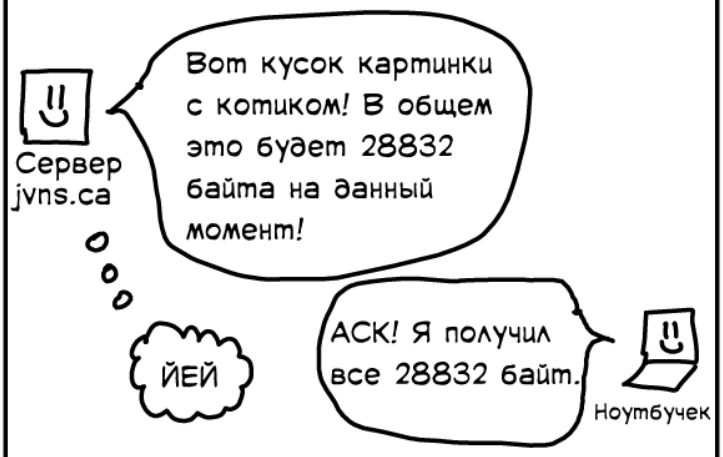
Получатель сможет собрать это в единое целое:

"once upon a time there was a magical oyster"

Положение первого байта (в нашем примере это 0,14,30) называется «номером последовательности».

Как справиться с потерянными пакетами:

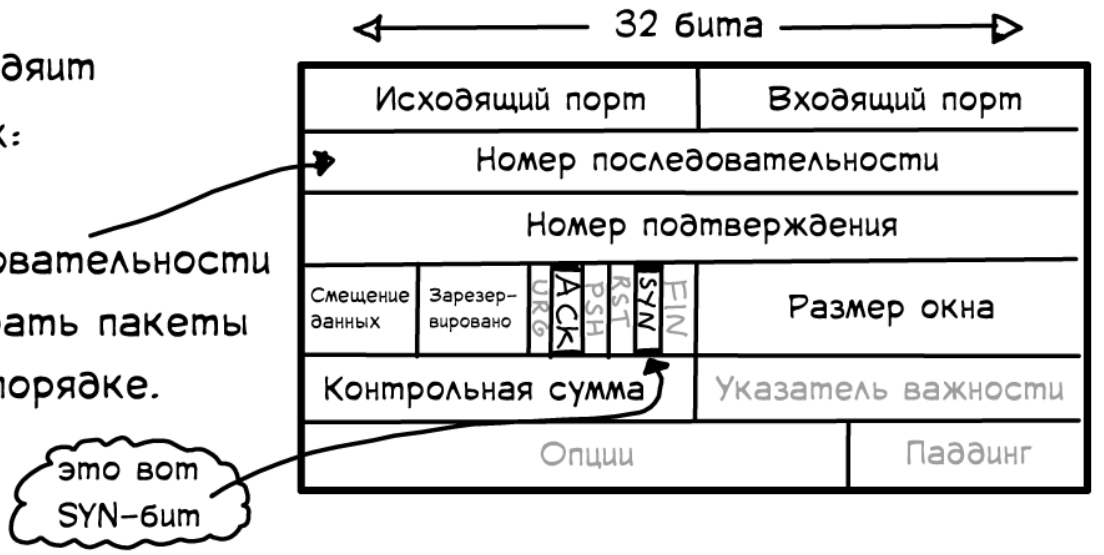
Принимая данные TCP, вам необходимо ACKтвердить (подтвердить) их: (ACK)



Если сервер не получит ACKтверждения, он отправит данные заново.

Рукопожатие TCP

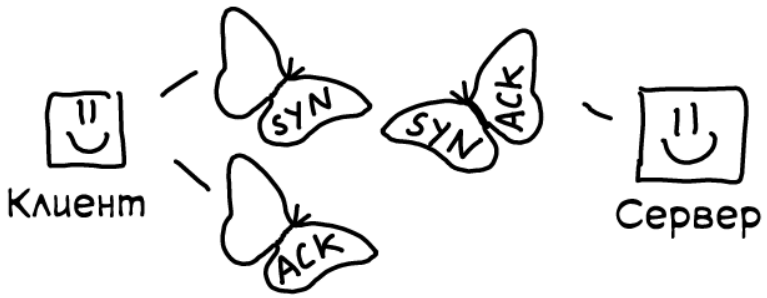
Вот так выглядит TCP-заголовок:



Номер последовательности позволит собрать пакеты в правильном порядке.

это вот SYN-бит

Любое TCP-соединение начинается с "рукопожатия". Это нужно, чтобы обе стороны соединения могли общаться друг с другом.



Но что такое "SYN" и "ACK"? В TCP-заголовках можно установить 6 битовых флагов (SYN, ACK, RST, FIN, PSH, URG), которые видно на диаграмме выше. SYN-пакет – это пакет, в котором SYN флаг равен 1.

Если у тебя возникает ошибка типа "в соединении отказано" и "таймаут соединения", значит рукопожатие не завершилось!

Я выполнила `sudo tcpdump host jvns.ca` в одном терминале

и `curl jvns.ca` в другом.

Вот кусочек вывода:

```
localhost:51104 > 104.28.6.94:80   Flags [S]
104.28.6.94:80  > localhost:51104  Flags [S.]
localhost:51104 > 104.28.6.94:80   Flags [.]
```

TCP-рукопожатие

IP-адрес jvns.ca

S это SYN

. это ACK

HTTP

Шаг ④ : Наконец-то мы можем отправить запрос на cat.png!

Каждый раз, когда ты открываешь веб-страницу или картинку онлайн, используется HTTP.

HTTP очень простой текстовый протокол. Даже больше, он настолько простой, что ты можешь руками состряпать HTTP-запрос прямо сейчас. Давай попробуем!!!

Сначала давай создадим файл request.txt.

```
GET / HTTP/1.1
Host: ask.metafilter.com
User-Agent: zine
(в конце добавьте два пустых переноса строки)
```

Разберемся в Host части совсем скоро

После выполним:

```
cat request.txt | nc metafilter.com 80
```

Команда `nc` ("netcat") устанавливает TCP-соединение с metafilter.co и отправляет HTTP-запрос, который ты только что создал! В ответ придет вот такое:

```
200 OK
Content-Length: 120321
...заголовки...

Кучка всякого HTML
```

HTTP/2 – это следующая версия HTTP. Он очень сильно отличается, но у нас закончилось место.

Важность HTTP-заголовков

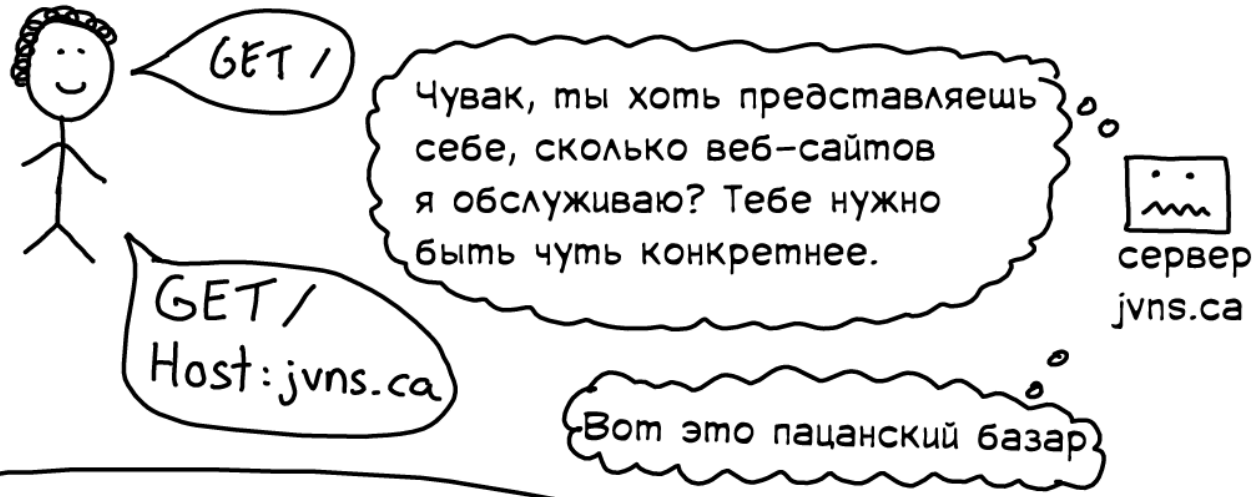
Вот так выглядит HTTP-запрос:

```
GET /cat.png HTTP/1.1
Host: jvns.ca
User-Agent: zine
```

Строки User-Agent и Host: называются "заголовками".

В них есть дополнительная информация для веб-сервера о том, какая страница тебе нужна.

Заголовок HOST ← мой любимый!



Большинство серверов обслуживает множество разных сайтов. Заголовок HOST позволяет выбрать нужный!

Серверы также отправляют заголовки ответа с дополнительной информацией о нем.

Еще полезные заголовки:

User-Agent

Куча серверов проверяют его, чтобы узнать, старый ли у тебя браузер или ты вообще бот.

Accept-Encoding

Хочешь сэкономить трафик? Установи тут "gzip" и сервер, возможно, сожмет свой ответ.

Cookie

Когда ты авторизовался на каком-то сайте, то твой браузер отправляет данные в этом заголовке! Так сервер узнает, что ты залогинен.

...а теперь у меня есть ЕЩЕ

Мы познакомились с основами того, как скачать картинку с котиком! Но вообще-то нужно знать много больше! Давай поговорим на еще несколько тем.

Познакомимся немного больше с сетевыми протоколами:

- Что же такое **порт**.
- Как же **пакет** собирается по частям.
- Безопасность: как работает **SSL**.
- Разные **сетевые слои**.
- **UDP** и почему он офигенен.

А также с тем, как пакеты перемещаются с места на место:

- Как отправляются пакеты в **локальной сети**.
- Как пакеты добираются **от твоего дома до jvns.ca**.
- Сетевая **нотация**.