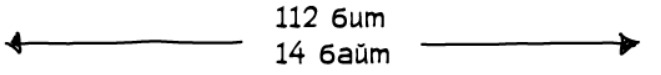
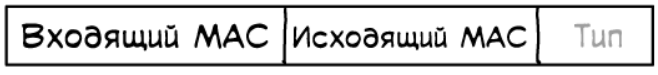
Сетевые уровни

Считаю, что это не всегда полезно, но все-равно стоит знать, что такое "уровень 4".

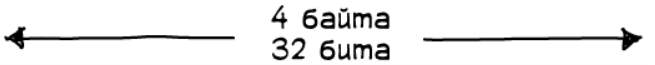
Сетевые уровни в основном относятся к разным секциям в пакете.





Уровень 1: Провода и радиоволны.

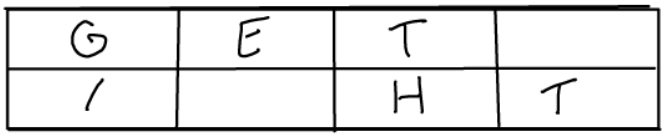
Уровень 2: Локальная сеть/wifi. Его понимает твоя сетевая карта.



Уровень 3: IP-адреса. Маршрутизаторы смотрят сюда, чтобы решить, куда отправлять пакет.

Уровень 4: TCP или UDP. Здесь определены порты.

Уровни 5+6: в целом их тут нет (хотя люди называют SSL "уровень 5").



Уровень 7: HTTP и всякое такое. Маршрутизаторы в основном игнорируют этот уровень. DNS-запросы, почта и другое находятся здесь.

Твой домашний маршрутизатор работает с уровнями 2+3+4

Сетевая утилита 3 уровня

↑

игнорирует уровень 4 и дальше

Я знаю только про IP-адреса! Не имею понятия, что такое порт и уж тем более ничего не знаю о содержимом пакета.

Приложения в основном переживают за 7 уровень, но они говорят ОС, какой IP и порт использовать.

Прикол в том, что уровни в основном не зависят друг от друга. Можно изменить IP-адрес (3 уровень) и не переживать за уровни 4+7.

Сетевая карта в твоем компьютере обращает внимание только на 1+2 уровни.

Что такое порт?

Порты – это часть протоколов TCP и UDP
(порт TCP 999 и порт UDP 999 не одно и то же).

Отправляя TCP-сообщение, ты хочешь передать его определенному типу программы.
Вот так будет не очень:

Мы хотим запускать разные типы программ на одном сервере:

Поэтому у каждого TCP-пакета есть номер порта в промежутке от 1 до 65535, которому он предназначается:

Netstat и lsof расскажут тебе, какие порты на компьютере используются.

Некоторые стандартные порты:

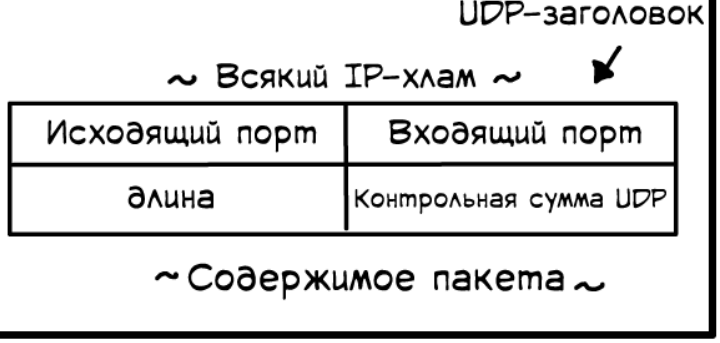
DNS: UDP порт 53
HTTP: TCP порт 80
HTTPS: TCP порт 443
SMTP (почта): TCP порт 25
Minecraft: TCP + UDP 25565

UDP

user datagram protocol

(протокол пользовательских датаграмм)

DNS-запросы отправляются через UDP. UDP очень простой протокол. Пакеты выглядят вот так:



"Протокол потерянных данных"

Отправляя UDP-пакеты они могут придти:

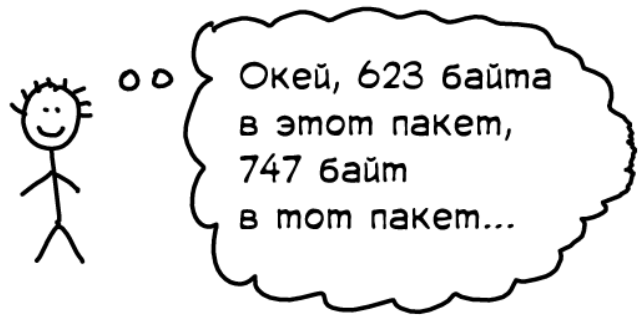
- Беспорядочно.
- Никогда.

На самом деле потеряться может любой пакет, но UDP не сделает ничего, чтобы тебе помочь.

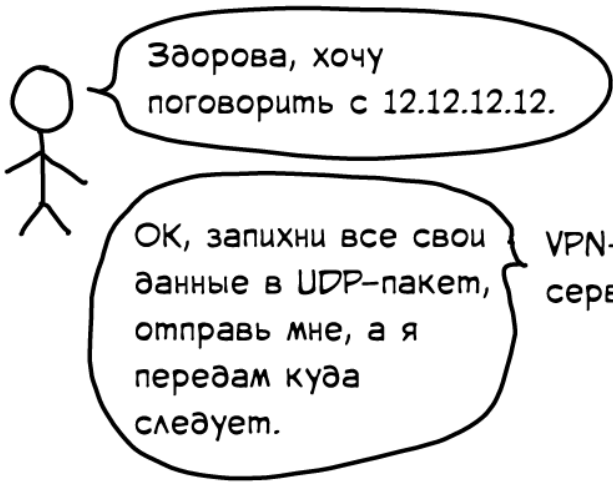
Размеры пакетов ограничены

*Размеры пакетов вообще суперски интересная тема. Гугли "MTU".

Придется решать, как организовывать данные в пакетах вручную:



VPNки пользуют UDP



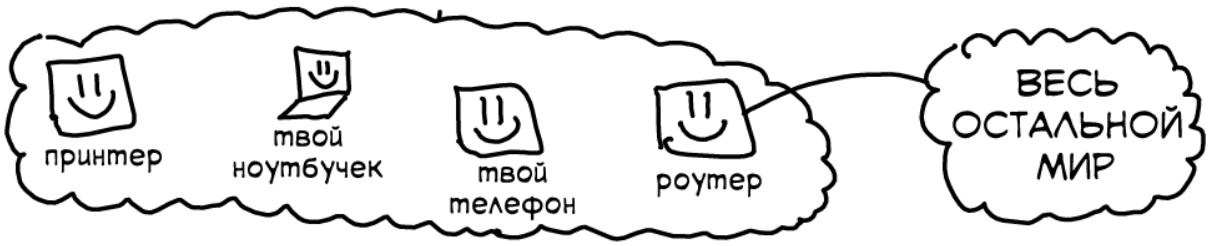
Потоковое видео часто передают по UDP.

Читай http://hpbn.co/webrtc (или просто по WebRTC), там ОФИГЕННОЕ обсуждение использования UDP как протокола в реальном времени.

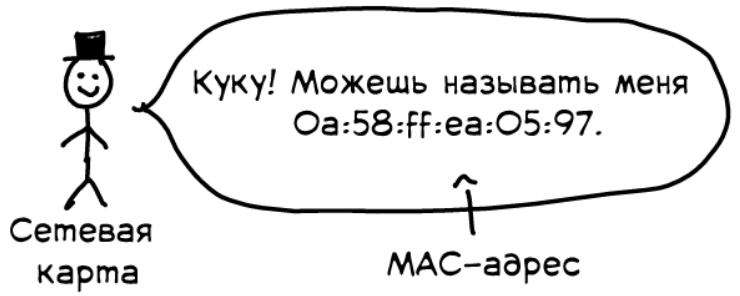
Локальные сети

Как поговорить с компьютером в этой же комнате

Каждый компьютер находится в подсети. Твоя подсеть–это все компьютеры, с которыми можно общаться напрямую.

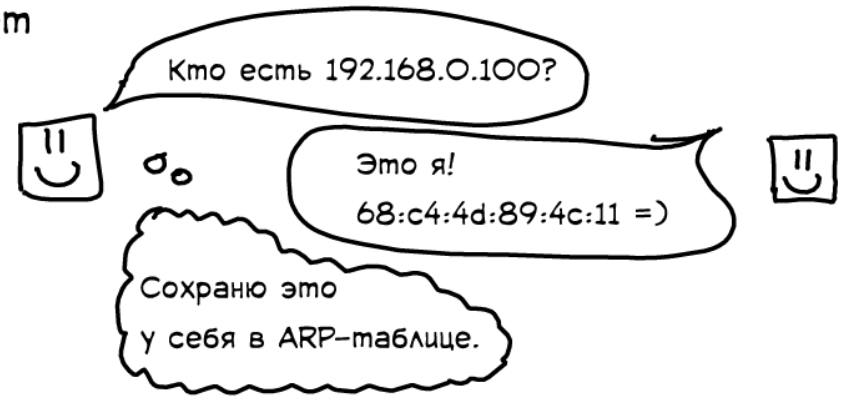


Что значит "разговаривать с компьютером напрямую"? Ну, у каждого компьютера в интернете есть сетевая карта с MAC–адресом.



У твоего ноутбучека поменяется IP–адрес, если подключиться к другой сети, но MAC останется прежним.

Отправляя пакет компьютеру в своей подсети, ты указываешь в нем его MAC–адрес. Чтобы определить правильный MAC, твой компьютер использует ARP–протокол (Address Resolution Protocol, протокол определения адресов).



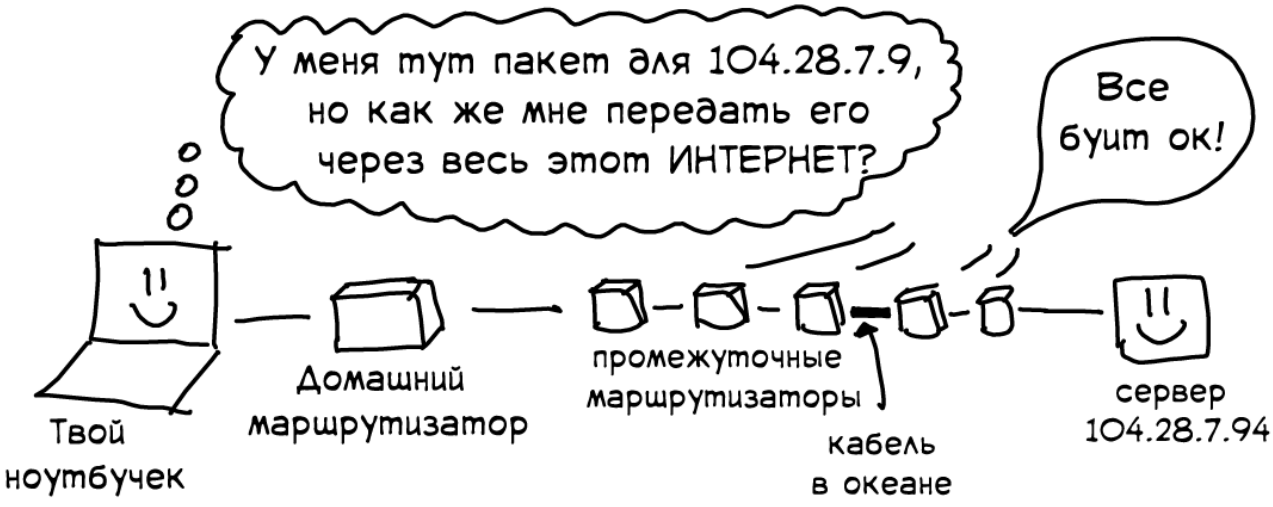
Выполни `arp -na` чтобы узнать содержимое ARP–таблицы на своем компьютере. Должно выглядеть как–то так:

```
$ arp -na
? (192.168.1.120) at 94:53:30:91:98:c8 [ether] on wlp3s0
```

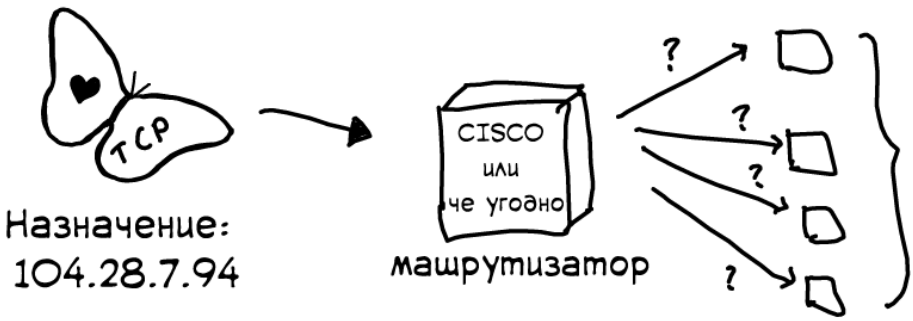
MAC для 192.168.1.120 (мой принтер)

моя wifi карта

Как пакеты отправляются за океан

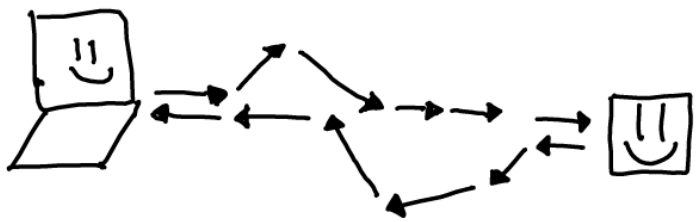


Когда пакет доезжает до маршрутизатора



Маршрутизаторы пользуются протоколом BGP, чтобы решить, на какой маршрутизатор передать пакет дальше.

Пакет может ездить по КУЧЕ разных путей, добираясь до одного и того же места назначения.



Путь пройденный им от A -> B, может отличаться от пути B -> A

УПРАЖНЕНИЕ: Запусти `traceroute firstvds.ru` и посмотри шаги, которые пройдет пакет до firstvds.ru.

Время нотаций!

10.0.0.0/8 132.5.23.0/24

Для обозначения групп IP-адресов человеки пользуются CIDR-нотацией.

Примеры CIDRов

CIDR	Диапазон IP
10.0.0.0/8	10.*.*.*
10.9.0.0/16	10.9.*.*
10.9.8.0/24	10.9.8.*

Важные примеры

10.0.0.0/8 и 192.168.0.0/16 и 172.16.0.0/12 зарезервированы для локальных сетей.

В CIDR-нотации "/n" обозначает 2^{32-n} IP-адресов.
Так что /24 будет означать $2^8 = 256$ IP.

Очень важно представлять группы IP-адресов как можно более эффективно, ведь у маршрутизаторов ОЧЕНЬ МНОГО РАБОТЫ.

Маршрутизатор

Принадлежит ли 192.168.3.2 подсети 192.168.0.0/16? Я могу провести очень быстрые битовые вычисления, чтобы это узнать!

10.9.0.0 в двоичном коде выглядит так:

00001010 00001001 00000000 00000000

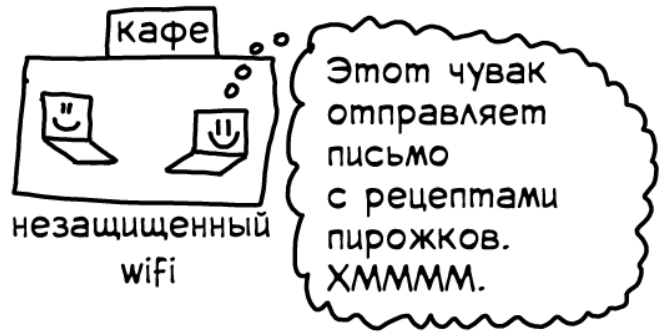
(первые 24 бита)

10.9.0.0/24 – это все IP-адреса с одинаковыми первыми 24 битами, что и 10.9.0.0!

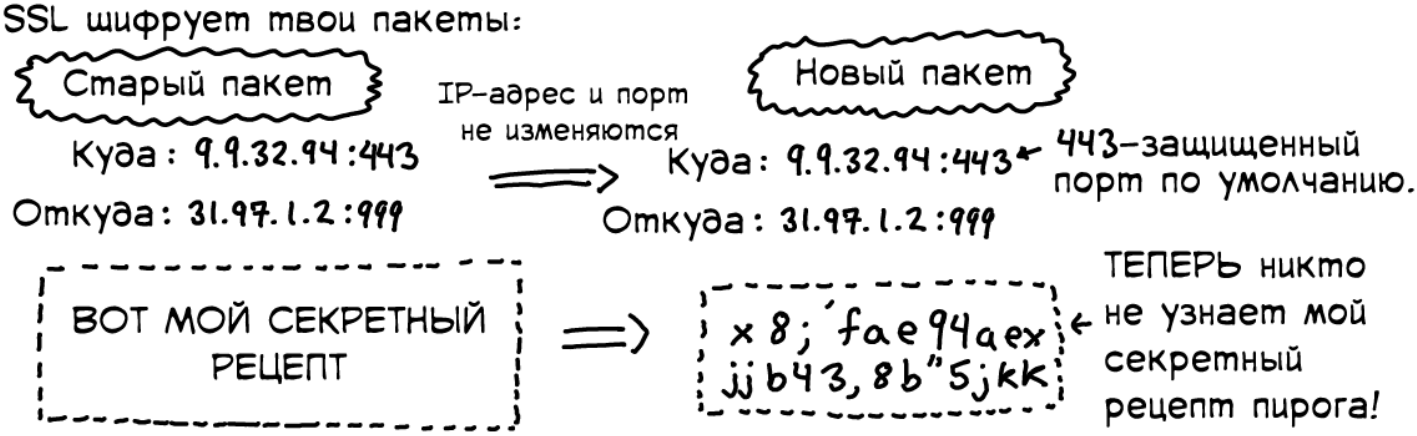
SSL / TLS

(TLS – новая версия SSL)

Когда ты отправляешь пакет через интернет, потенциально его могут прочитать ОЧЕНЬ МНОГИЕ.



SSL шифрует твои пакеты:



Вот что случится, если зайти на https://jvns.ca:

(очень упрощенно)

Когда клиент и сервер договорились о сессионном ключе, они могут шифровать любые данные, которые им только вздумается.

Чтобы посмотреть сертификат у jvns.ca выполните:

```
$openssl s_client -connect jvns.ca:443 -servername jvns.ca
```

TSL вообще очень непрост. Можно воспользоваться утилитой типа SSL Labs, чтобы проверить безопасность на твоем сайте.

wireshark

Wireshark – это ОФИГЕННАЯ тулза для анализа пакетов. Вот упражнения для ее изучения! Запусти:

```
sudo tcpdump port 80 -w http.pcap
```

Пока процесс выполняется, открой firstvds.ru в браузере. Нажми ctrl+c, чтобы остановить tcpdump. Теперь у нас есть pcap-файл!

Открой http.pcap в Wireshark.

Попытайся ответить на несколько вопросов:

① Какие HTTP-заголовки твой браузер отправил на firstvds.ru?

(подсказка: поищи `frame contains "GET"`)

② Сколькими пакетами ты обменялся с сервером firstvds.ru?

(подсказка: поищи `ip.dst == 212.109.222.30` тут напиши IP из "ping firstvds.ru")

С Wireshark можно легко разобраться:

- В IP-адресах и портах.
- SYNах и ACKах для TCP-трафика.
- Что вообще происходит с DNS-запросами.
- Да и вообще с кучей всего. Отличный способ разнюхать все и научиться чему-то.

♡ СПАСИБО ♡
ЧТО ПРОЧИТАЛИ

Если хотите узнать больше о работе сети:

→ Делайте сетевые запросы! Играйтесь с

dig traceroute tcpdump ifconfig
netcat wireshark netstat

→ Мануал от beej о сетевом программировании – это очень полезный и смешной гайд про API сокетов на UNIX-системах.

→ beej.us/guide/bgnet ←

→ High Performance Browser Networking

Просто ★ опупенный ★ и практичный гайд ко всему, что тебе потребуется знать о сетях, чтобы делать быстрые веб-сайты. Можно читать бесплатно здесь:

→ hpbn.co ←

Спасибо Камалу Маруби, Крису Каничу и Аде Мунро за проверку всего этого.

Обложку нарисовала офигительная Лиз Бэйли.