

# Нужно ли переходить на http/2

**Медведик Давид**

Старший инженер-программист в “Positive technologies”

# Немного обо мне



Медведик Давид

LinkedIn: [www.linkedin.com/in/medvedik-david](https://www.linkedin.com/in/medvedik-david)

Telegram: @dmedvedik

Последние 7 лет пишу на C++

## Positive Technologies

- Telecom Attack Discovery

## EPAM

- **Acronis**
  - Acronis True Image

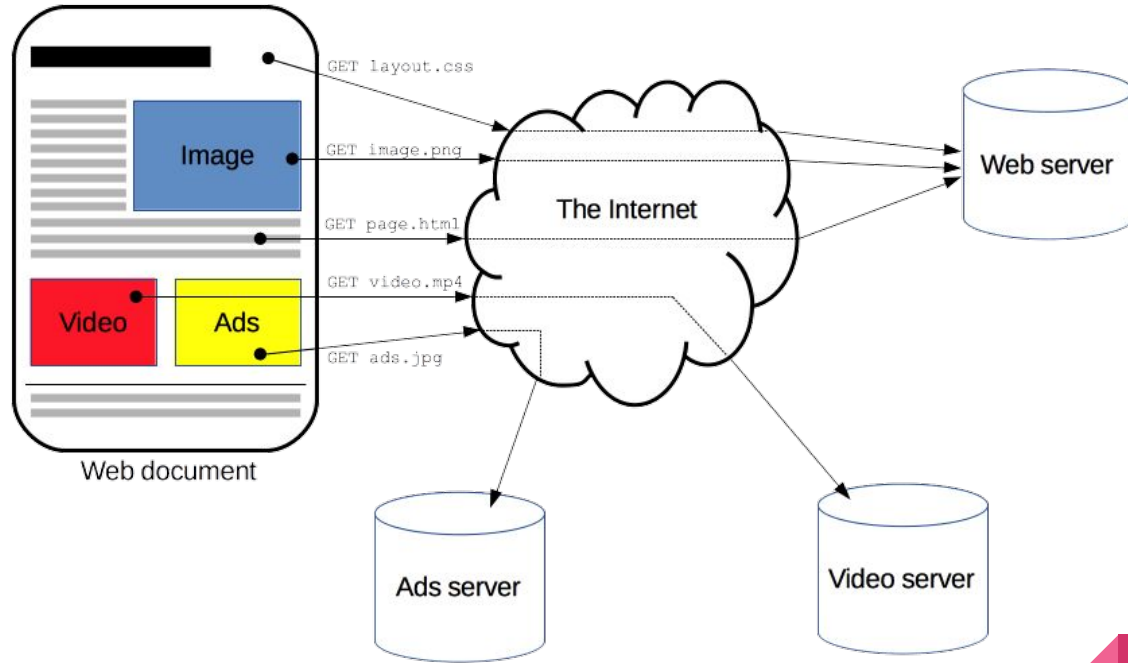
## MFI Soft

- Internet traffic filtering systems

## Orion Innovation

- **Ericsson**
  - SS7 protocols
- **Honeywell**
  - Smart meter

# HTTP



# История

## HTTP 0.9 (1991)

- Простой текстовый ASCII протокол
- Автор Тим Бернерс-Ли

## HTTP/1.0 (1996)

- Выпущен информационный документ RFC 1945
- Поддержка версии в запросе
- Поддержка заголовков (Headers)

## HTTP/1.1 (1997)

- Черновик RFC 2068
- Internet Standard RFC 2616 (с 1999 года)
- Encoding, Chunks, Keep Alive, VHost ...



# История

## SPDY (2012)

- Разработан Google для Chrome
- Binary (gzip) поверх TLS

## HTTP/2 (2015)

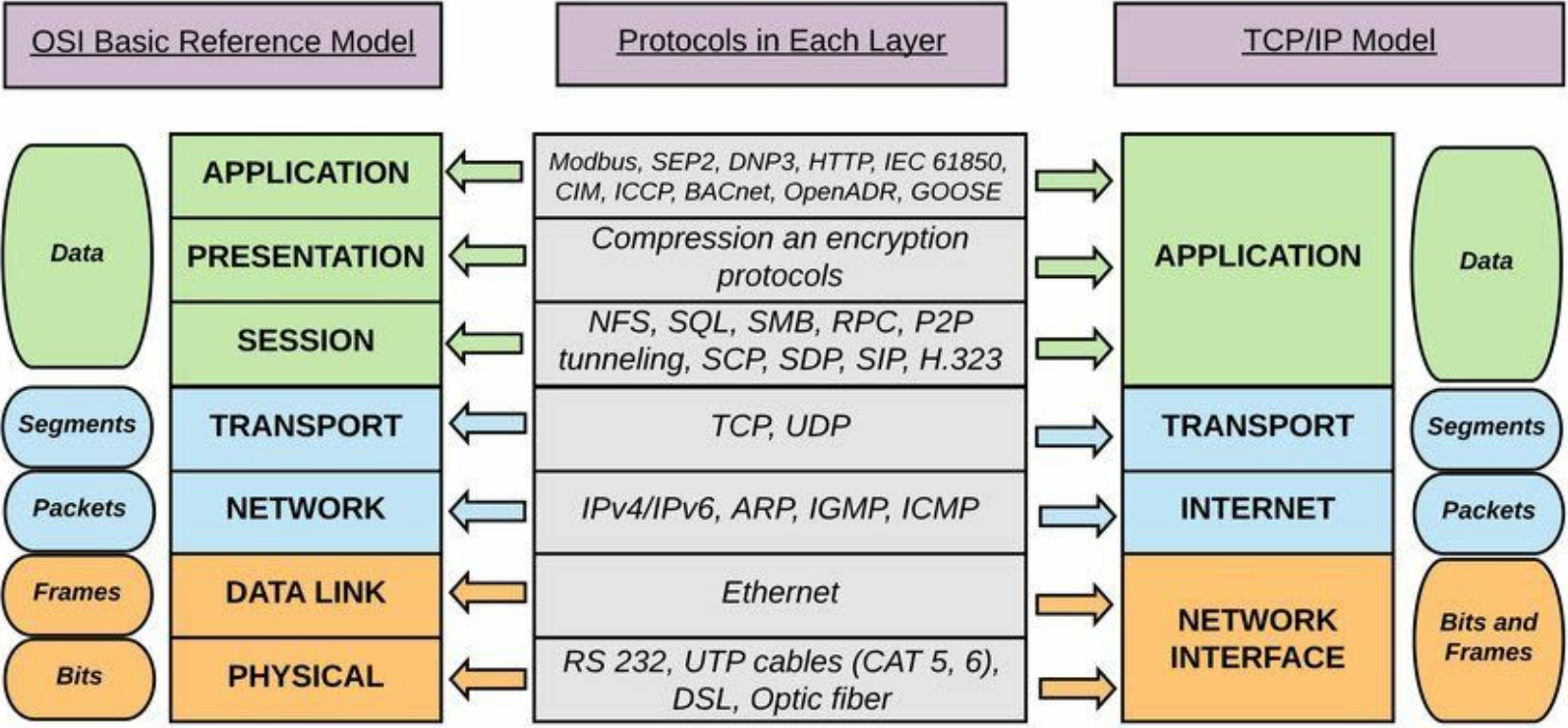
- Internet Standard RFC 7540
- Стандартная версия SPDY

## HTTP/3 (2018)

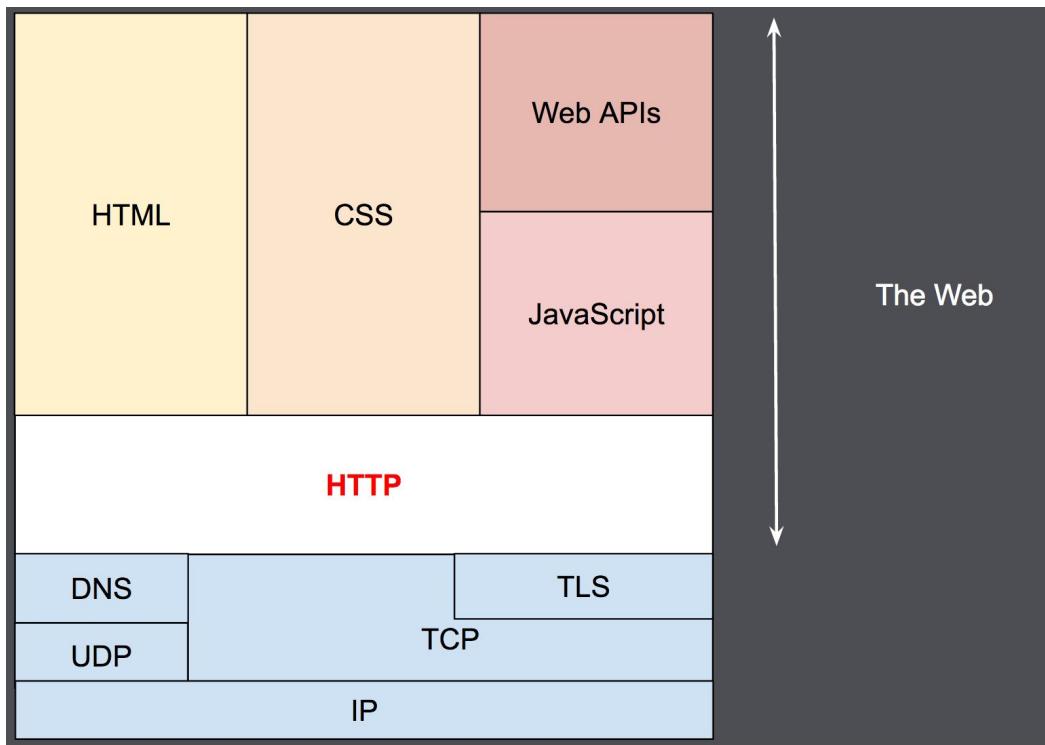
- HTTP поверх QUIC, UDP протокол
- В работе черновик IETF
- Поддержка браузеров с cURL (с 2019)
- Nginx Tech Preview (10 июня - 2020)



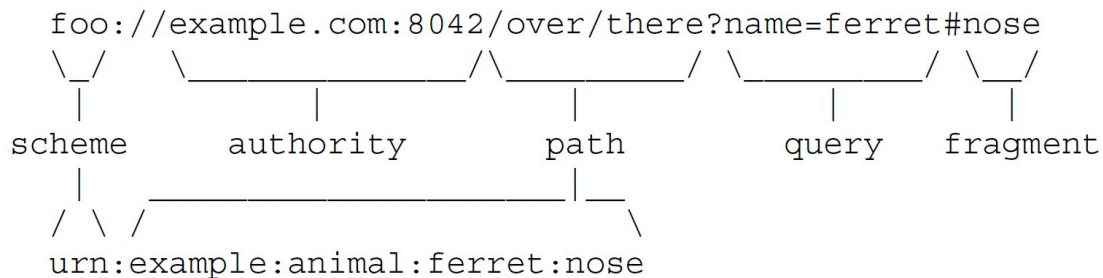
# OSI Model



# Http стек



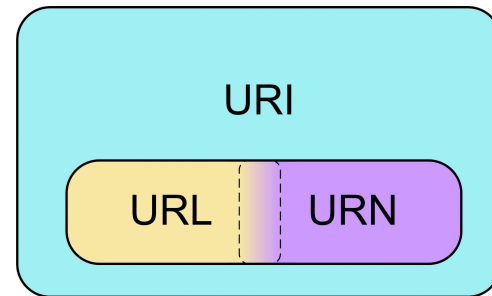
# Адресация



**URI** - Uniform Resource Identifier

**URL** - Uniform Resource Locator

**URN** - Uniform Resource Name



Описание стандарта **RFC 3986**



# HTTP Сообщение

Первая строка запроса (ответа)

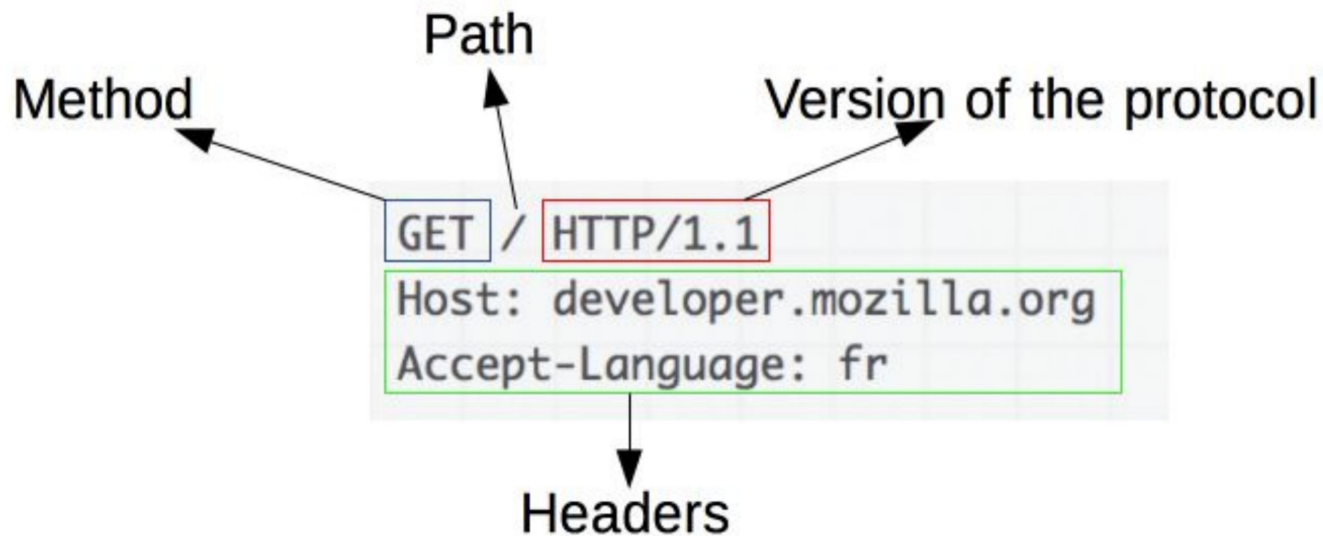
Заголовки запроса (ответа)

HeaderName: HeaderValue  
Header-Name: Header Value

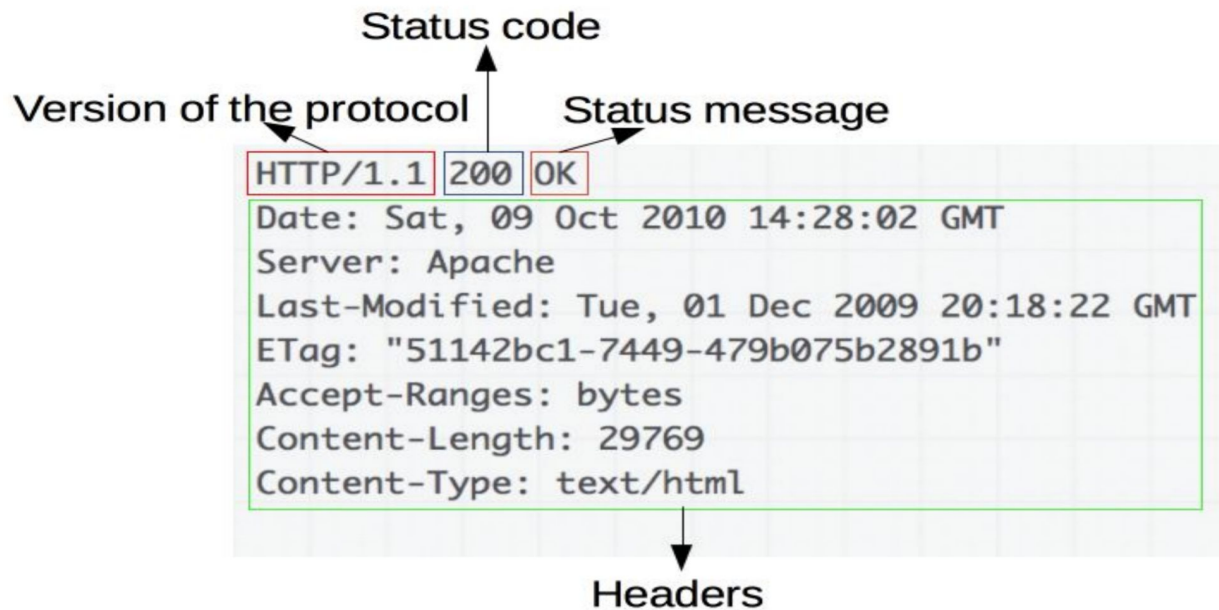
Пустая строка

Данные (HTML , JSON и т.д)


# HTTP Запрос



# HTTP Ответ



# Особенности HTTP/1.1

- Заголовок **Host** (виртуальные хосты)
  - Заголовок **Connection: keep-alive** (Постоянное соединение)
  - Поддержка **pipelining** (Несколько запросов подряд)
  - Заголовки **Accept / Content** (Language, Encoding Type)  
для определения запроса и передачи документов
  - Заголовок **Transfer-Encoding** (gzip, chunked)
  - Расширенная поддержка кеширования (**Etag, If-Unmodified-Since** и т.д.)
- 

# Заголовки HTTP/1.1

## General Headers

- Могут включаться в запрос и ответ

## Request Headers

- Только в запросах сервера

## Response Headers

- Только в ответах сервера

## Entity Headers

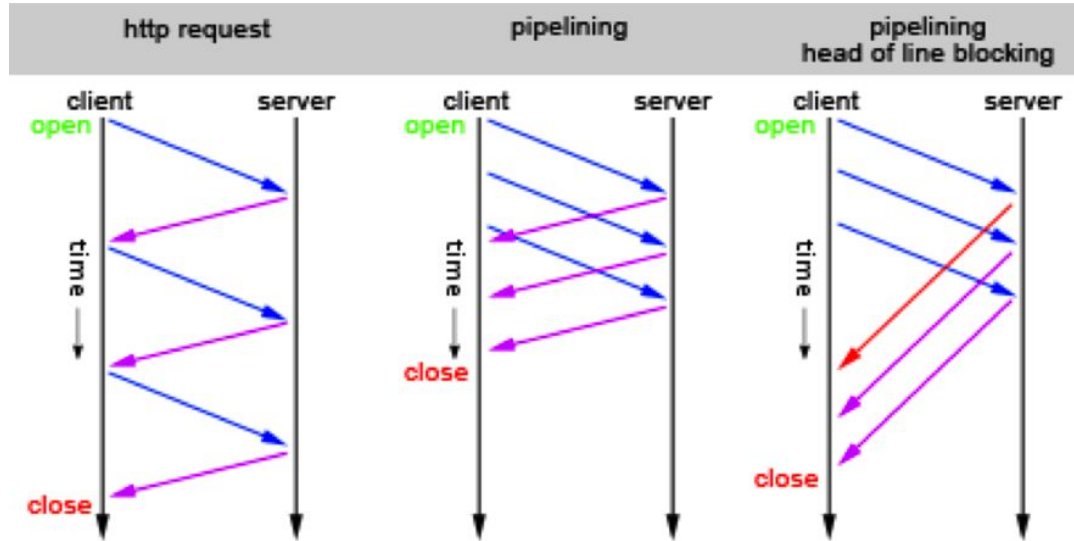
- Заголовки сущностей (Content)



# Методы HTTP/1.1

Метод	Безопасный	Идемпотентный	Кэшируемый
GET	Да	Да	Да
HEAD	Да	Да	Да
PUT	Нет	Да	Нет
DELETE	Нет	Да	Нет
POST	Нет	Нет	Нет *
OPTIONS	Да	Да	Нет
CONNECT	Нет	Нет	Нет
TRACE	Нет	Да	Нет
PATCH	Нет	Нет	Нет

# HTTP requests + HOL



# Browsers TCP Limitations



140



BrowserVersion	ConnectionsPerHostname	MaxConnections
Chrome34/32	6	10
IE9	6	35
IE10	8	17
IE11	13	17
Firefox27/26	6	17
Safari7.0.1	6	17
Android4	6	17
ChromeMobile18	6	16
IE Mobile9	6	60

The first value is **ConnectionsPerHostname** and the second value is **MaxConnections**.

<https://stackoverflow.com/questions/985431/max-parallel-http-connections-in-a-browser>



# Оптимизация

## Тюнинг TCP

```
sysctl -a | grep tcp
```

- [https://en.wikipedia.org/wiki/TCP\\_tuning](https://en.wikipedia.org/wiki/TCP_tuning)
- [Статья от Dropbox](#)
- [Статья от Google Cloud](#)

## Тюнинг веб сервера (nginx)

- <https://www.nginx.com/blog/tuning-nginx/>



# Недостатки HTTP

- Открытый текстовый формат
- HOL Blocking (проблема первого пакета)
- Pipelines != Parallel
- Избыточность данных
- Ограничения TCP (количество одновременных соединений)

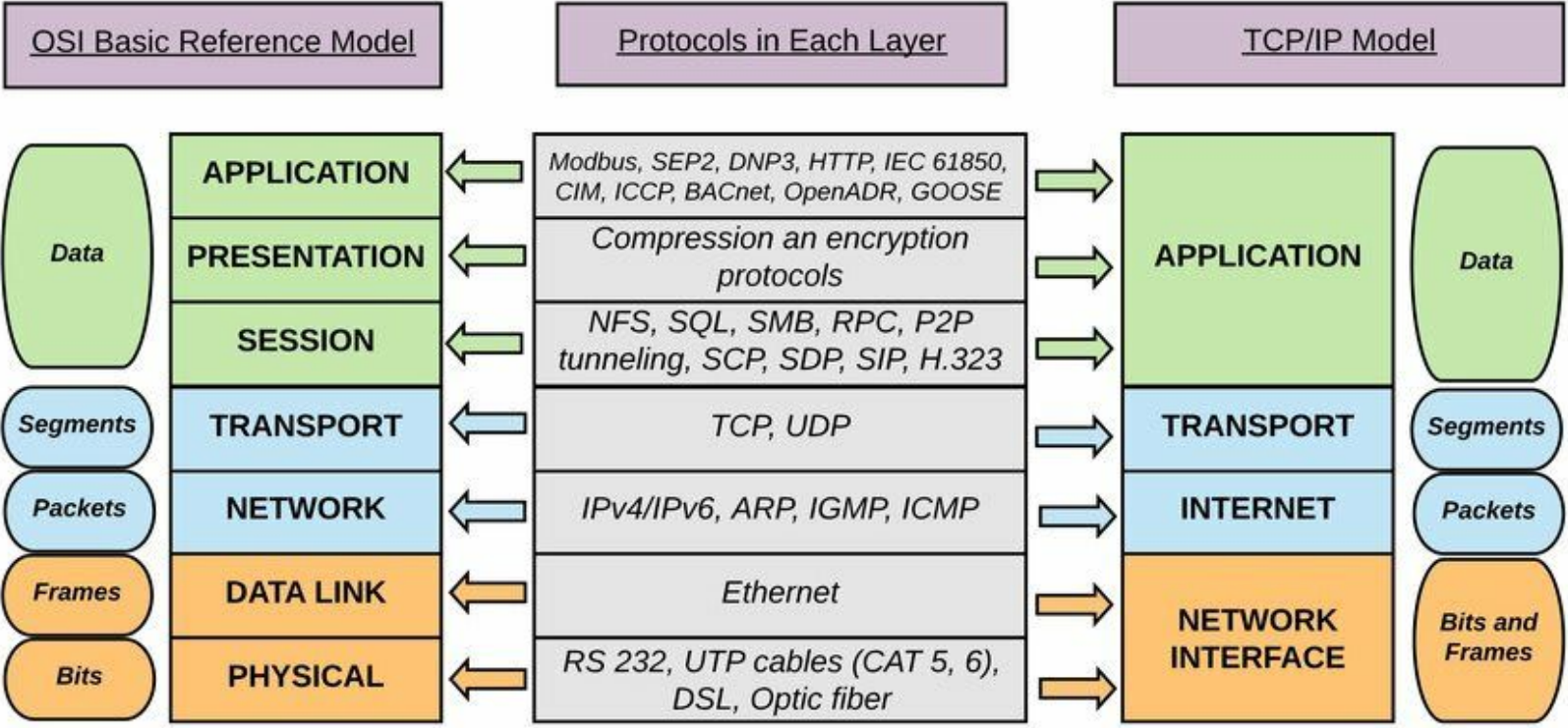
Всё исправлено в HTTP/3





HTTPS

# OSI Model

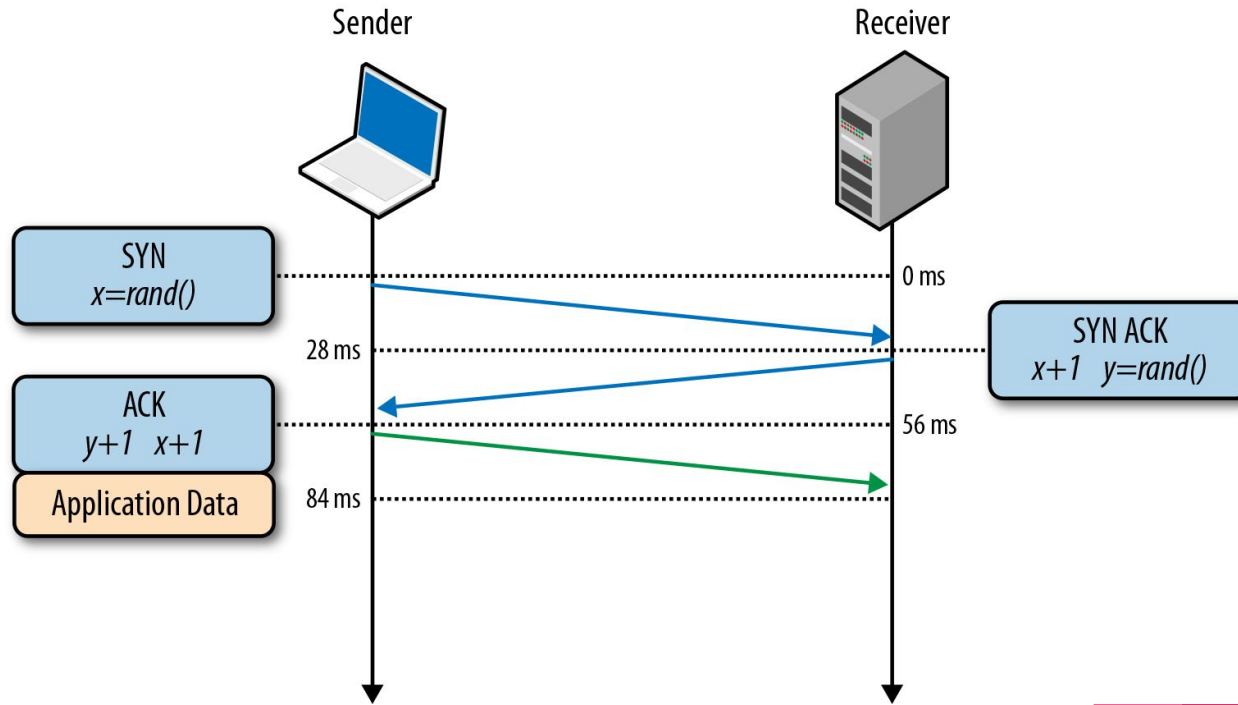


# HTTPS

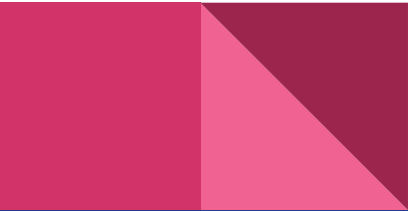
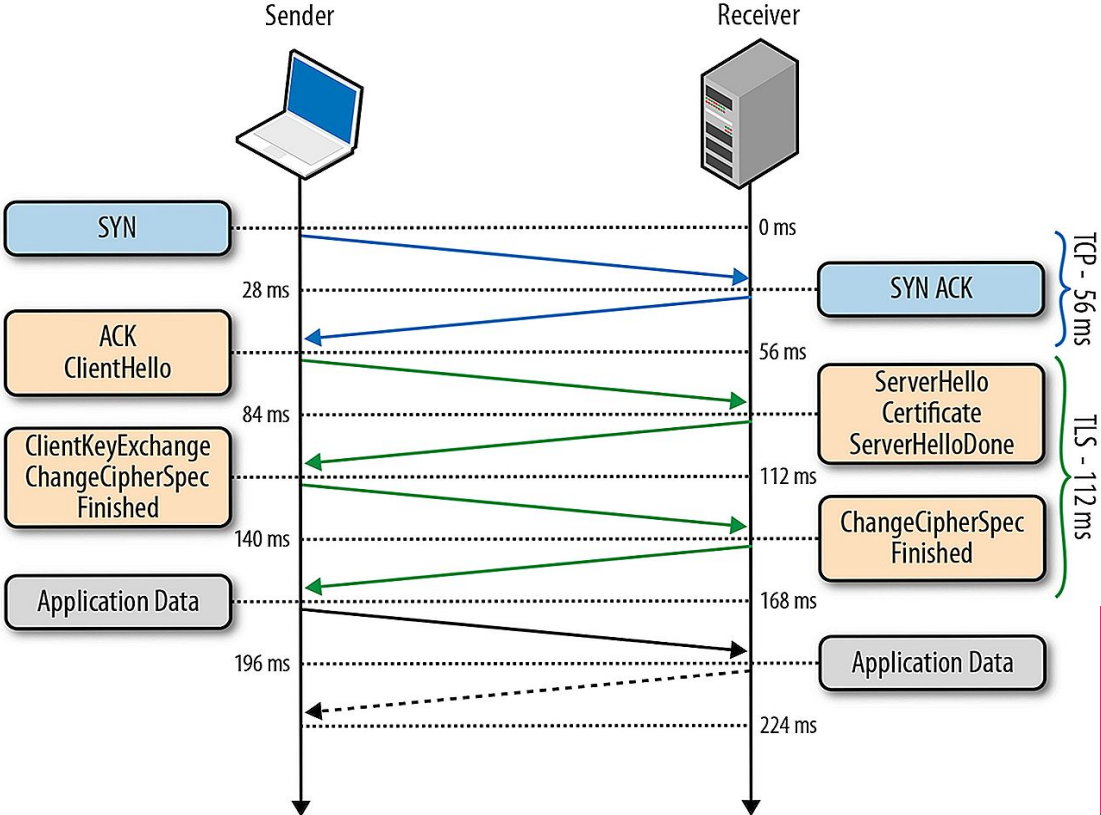
- Использует асимметричное шифрование для установления сессии
- При повторном соединении, если сессия совпадает, может использовать ускоренную процедуру
- Узкое место - скорость
- Внутри всё тот же HTTP
- Point-To-Point соединение



# TCP Handshake



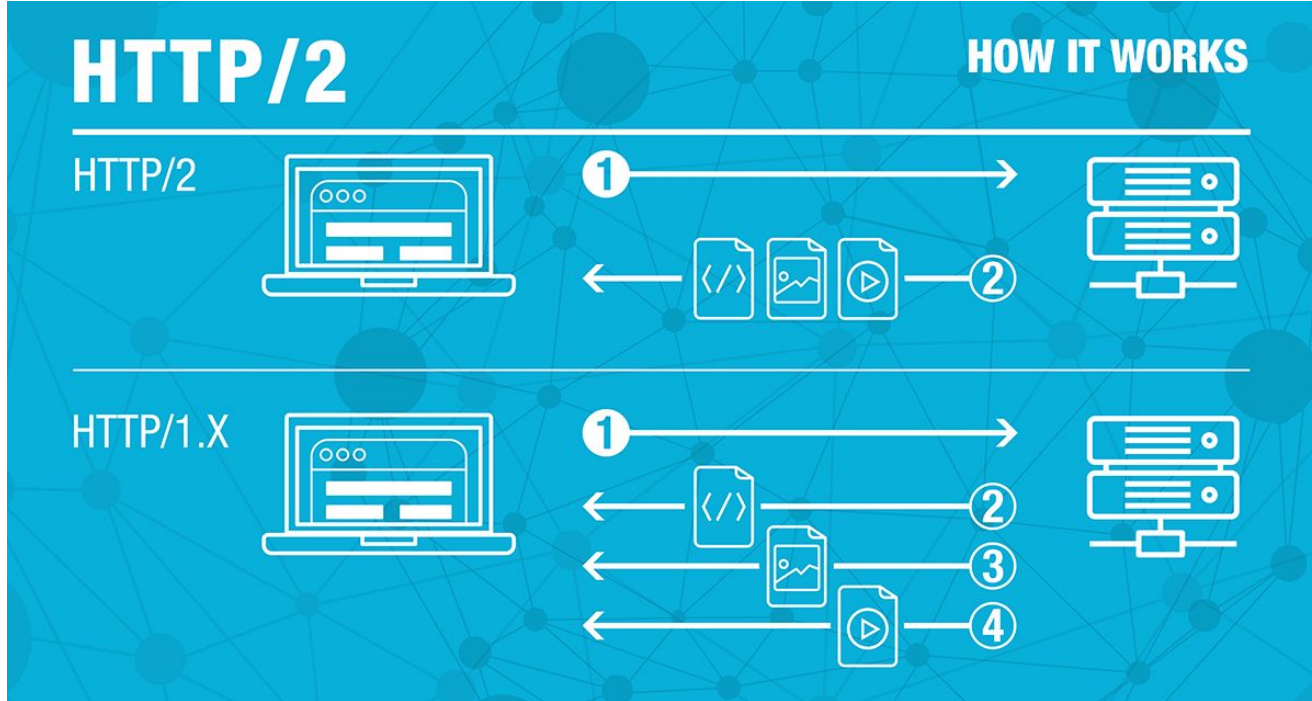
# HTTPS Handshake



HTTP/2



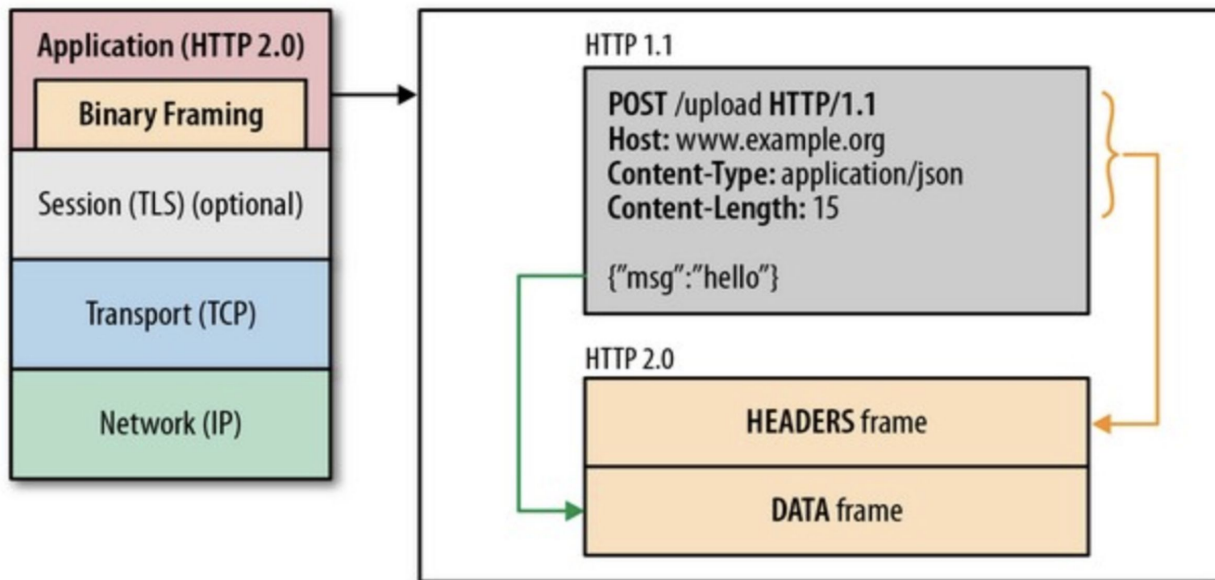
# HTTP/2



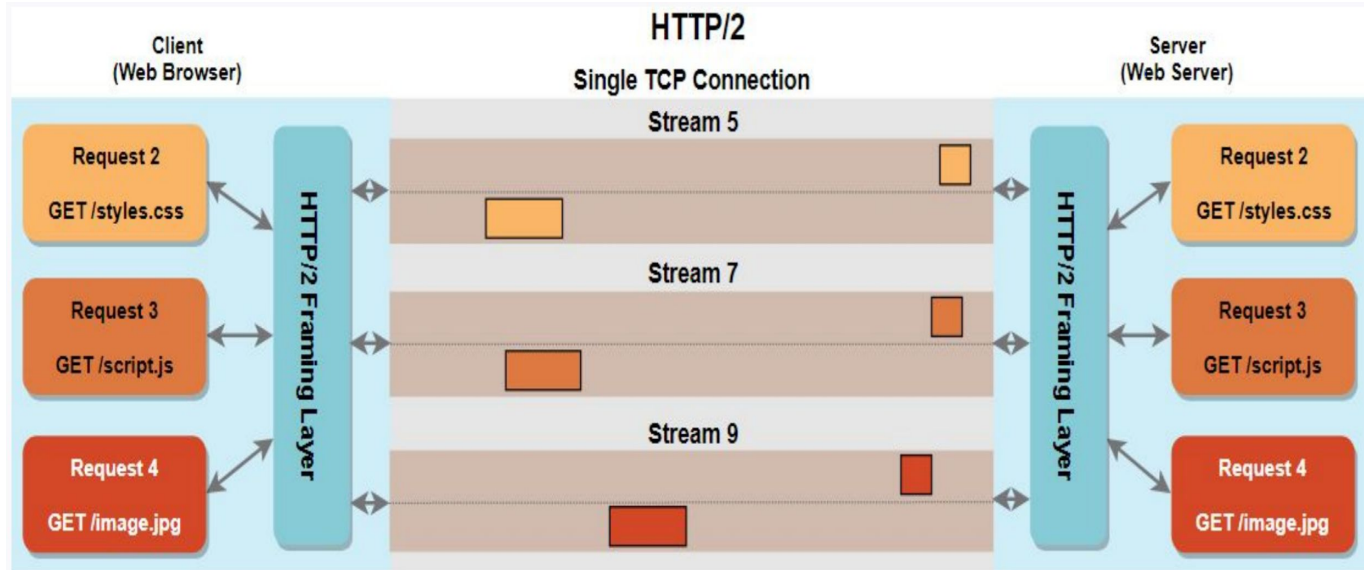
# Особенности http/2

- **Бинарный протокол (TCP в качестве транспорта)**
- Использует каналы (**streams**) для передачи кадров (**frames**)
- **Включает в себя механизмы**
  - Мультиплексирование (**multiplexing**)
  - Сжатие заголовков (**HPACK**)
  - Отправка сервером не запрошенного содержимого (**server push**)
  - Управление размером буфера соединения (**flow control**)
  - Приоритизация запросов (**stream priority**)
  - Зависимости между каналами (**stream dependencies**)
- **Де-факто работает поверх TLS \***

# Принцип работы



# HTTP/2 - Streaming multiplexing



# HTTP/2 Framing

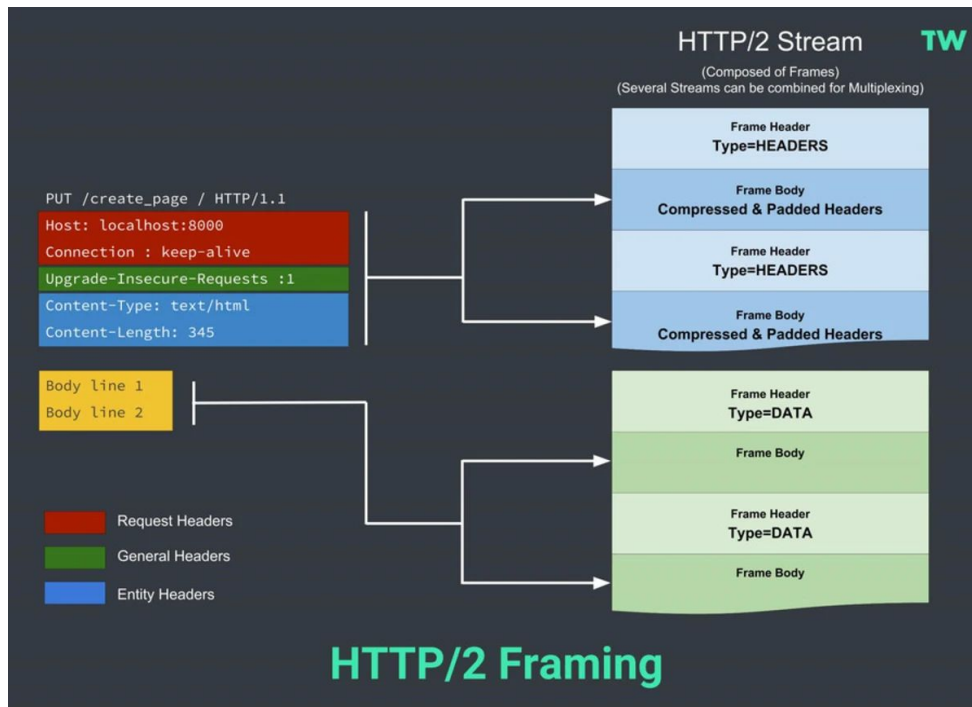
## Бинарный формат данных

## Различные типы фреймов

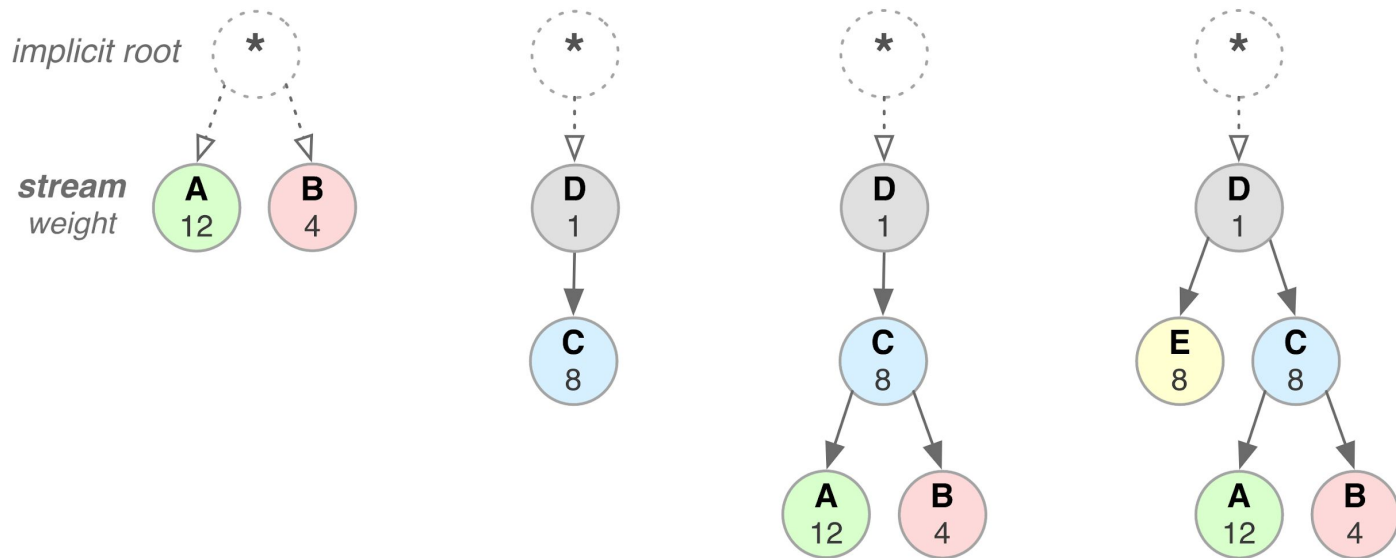
- HEADERS
- DATA
- PRIORITY
- PUSH\_PROMISE
- RST\_STREAM
- PING
- WINDOW\_UPDATE
- SETTINGS
- CONTINUATION
- GOAWAY

## Security Paddings

- Для защиты mitm атак

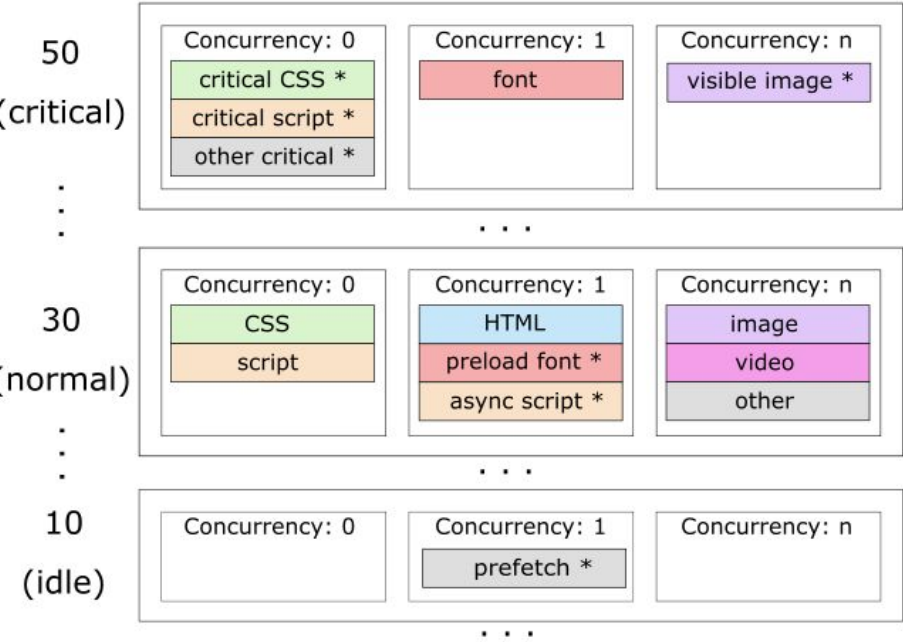


# Управление зависимостями и приоритетом

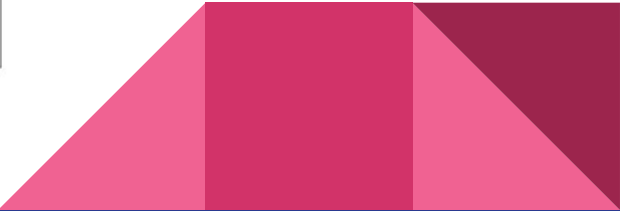


# Управление зависимостями и приоритетом

Priority (0-63)



\* If Detectable



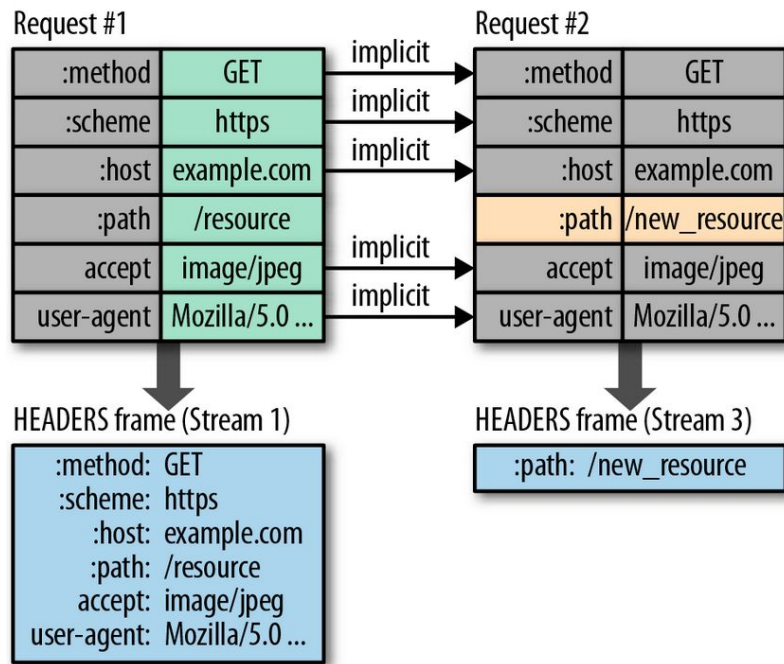
# Сжатие заголовков HPACK

## Использует несколько алгоритмов сжатия

- Статический словарь
- Динамический словарь
- В основе кодирования лежит алгоритм Хаффмана

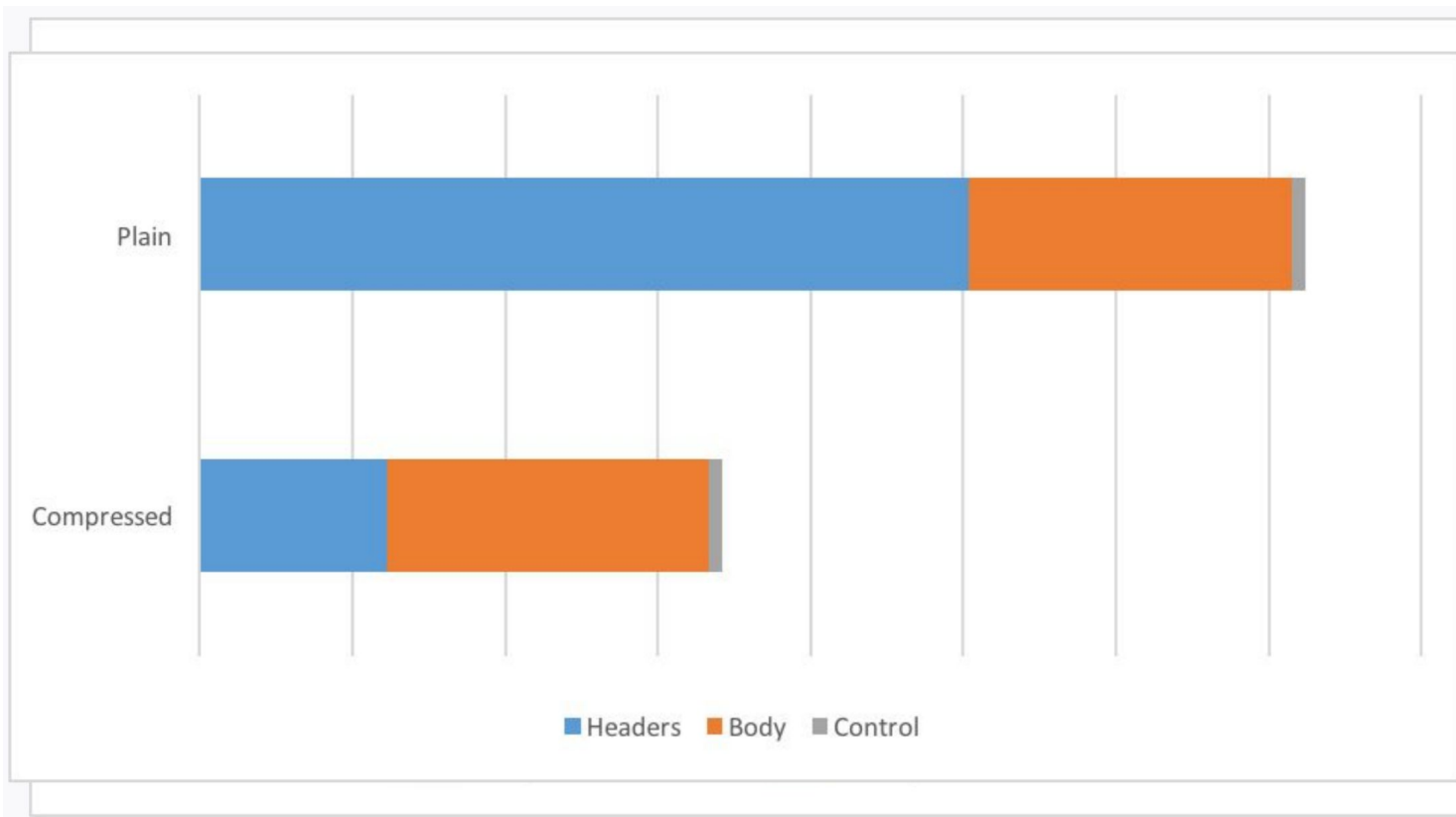
## Основной принцип

- Передаются только измененные заголовки
- Для неизменяемых указывается ссылка на словарь

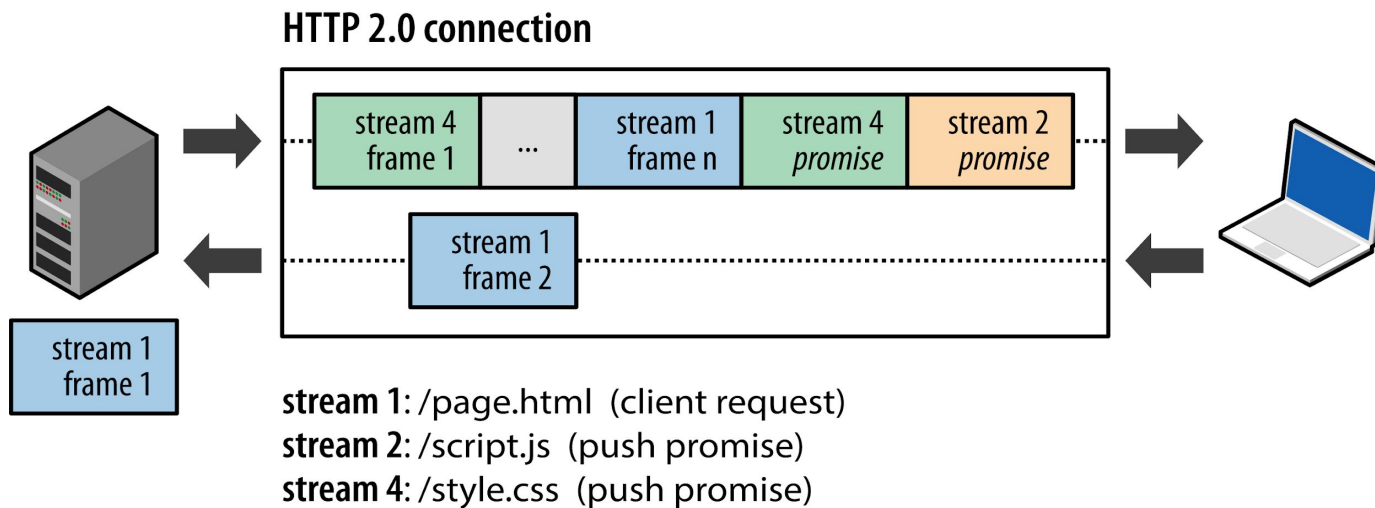




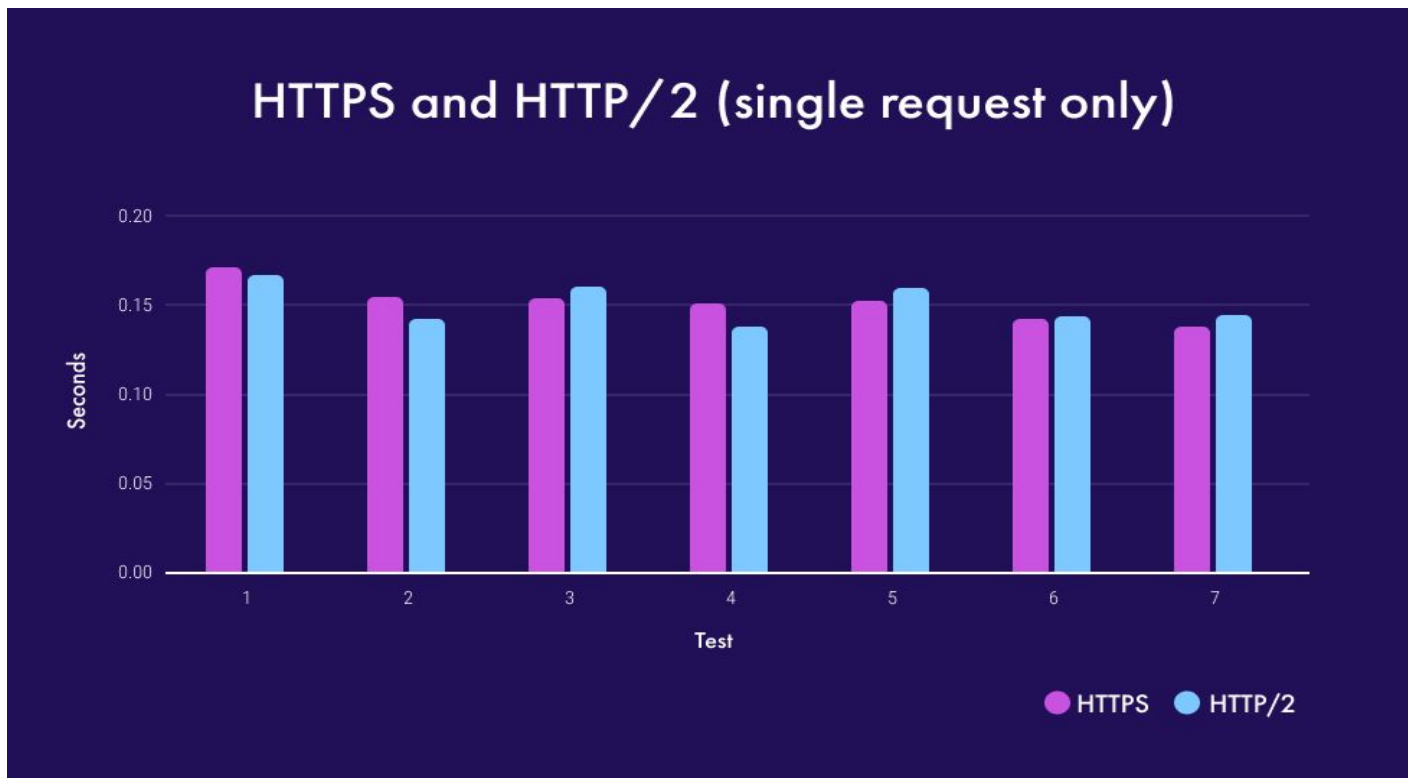
# Сжатие заголовков HPACK



# Server Push



# Performance Benchmark



HTTP/3

# Транспорт через UDP

## gQUIC (2012)

- Автор **Jim Roskind** (ex Google, ex Netscape)
- Транспортный протокол общего назначения
- <https://www.youtube.com/watch?v=FhDDwmOyRmk&t=2s>



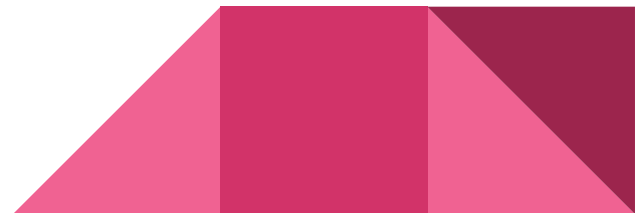
## QUIC - черновик IETF (in progress since 2015)

<https://datatracker.ietf.org/doc/draft-ietf-quic-transport/>

Переводы статей QUIC & HTTP/3 от Cloudflare на Habr

<https://habr.com/en/company/voximplant/blog/430436/>

<https://habr.com/en/post/438810/>



# HTTP/3

## HTTP/3 (2018)

- HTTP over QUIC, UDP протокол
- В работе черновик IETF
- Поддержка браузеров и cURL (с 2019)
- Nginx Tech Preview (10 июня 2020)

## quiche (2019)

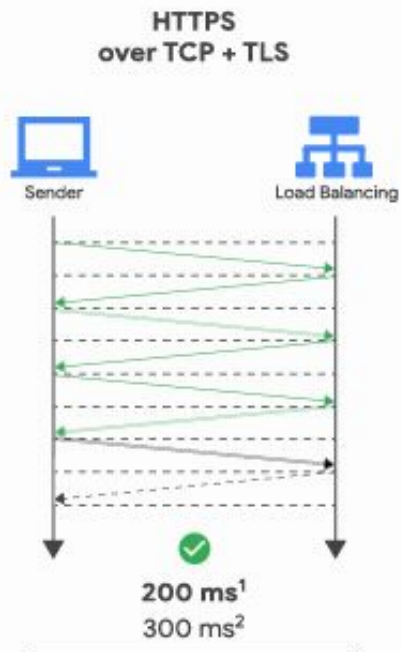
- Имплементация HTTP/3 от Cloudflare (Rust)
- Патч для nginx в 2019 году
- в работе ветка nginx-quic

## curl

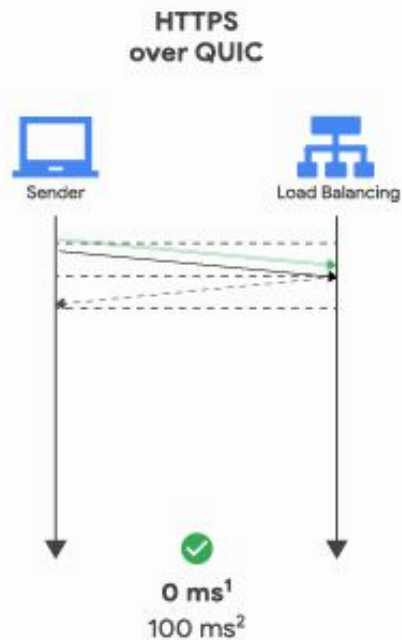
- Experimental (since July 2019)
- <https://github.com/curl/curl/blob/master/docs/HTTP3.md>



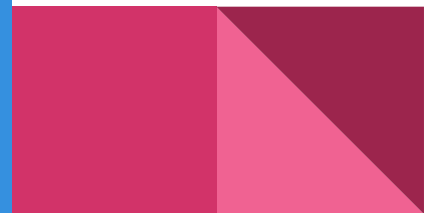
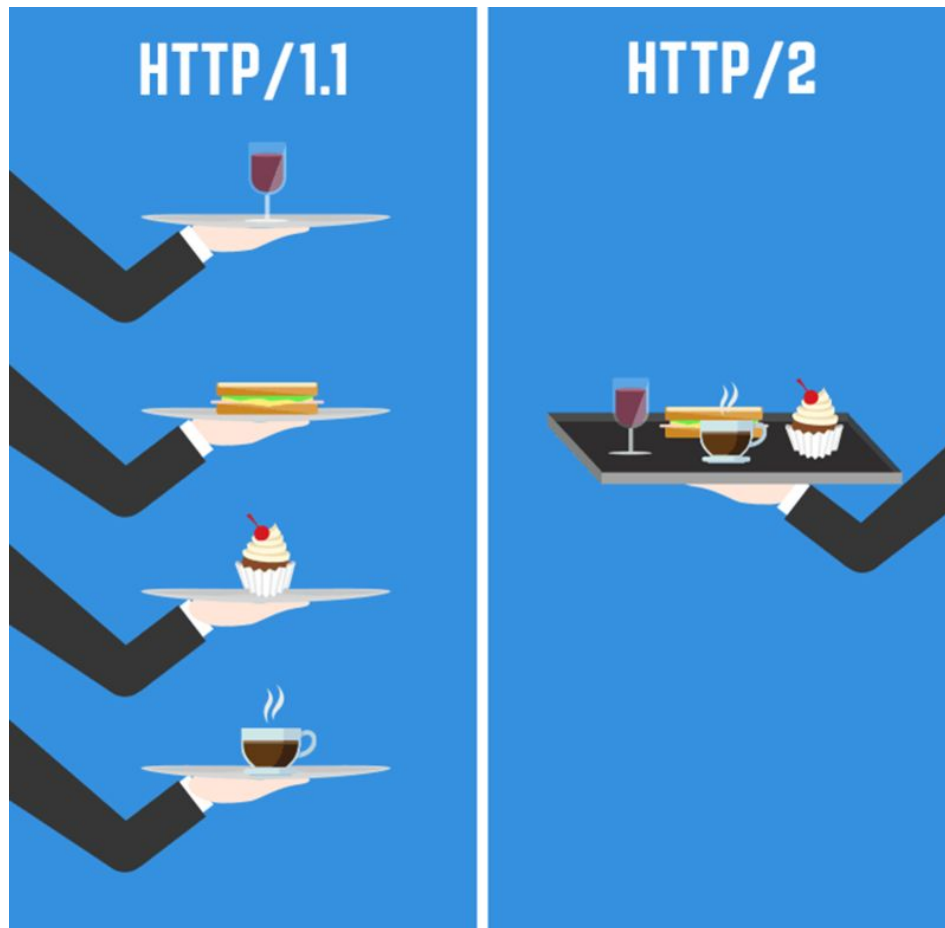
# Принцип работы



1. Repeat connection
2. Never talked to server before



# Рефлексия





# Спасибо за внимание!

- История http
- Адресация
- Особенности HTTP/1.1
- Методы HTTP/1.1
- Проблема H0L
- Ограничение TCP
- HTTPS
- HTTP/2 Streaming multiplexing
- HTTP/2 Framing
- Управление зависимостями и приоритетом
- Сжатие заголовков HPACK
- Server Push
- HTTP/3

# Нужно ли переходить на http/2

LinkedIn: [www.linkedin.com/in/medvedik-david](https://www.linkedin.com/in/medvedik-david)  
Telegram: @dmedvedik

---