

HTTP/1

VS

HTTP/2

HTTP (*hyper text transport protocol*)

-это набор правил, согласно которым передаются данные между клиентом и сервером.

Изначально этими данными были гипертекстовые документы в формате HTML. Теперь это произвольные данные. Например, картинки, css, js.

Версии HTTP

- HTTP 0.9 – экспериментальная версия, 1991
- HTTP 1 – первая официальная версия протокола, 1996
- HTTP 1.1 – расширение первой версии HTTP, 1997
- HTTP 2 – современная версия, 2015

Структура HTTP-сообщения

Стартовая строка – определяет тип сообщения.

GET /wiki/article HTTP/1.1

Host: ru.wikipedia.org

Заголовки – характеризуют тело сообщения, параметры передачи и прочие сведения.

Date: Thur, 01 Oct 2020 01:24:22 GMT+3

Server: Apache

Content-Type: text/html

Тело сообщения – непосредственно данные.

Примеры заголовков

POST / HTTP/1.1

```
Host: localhost:8000
User-Agent: Mozilla/5.0 (Macintosh;... )... Firefox/51.0
Accept: text/html,application/xhtml+xml,..., */*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: multipart/form-data; boundary=-12656974
Content-Length: 345
```

Request headers

General headers

Entity headers

-12656974
(more data)

HTTP/1.1 200 OK

```
Access-Control-Allow-Origin: *
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: text/html; charset=utf-8
Date: Wed, 10 Aug 2016 13:17:18 GMT
Etag: "d9b3b803e9a0dc6f22e2f20a3e90f69c41f6b71b"
Keep-Alive: timeout=5, max=999
Last-Modified: Wed, 10 Aug 2016 05:38:31 GMT
Server: Apache
Set-Cookie: csrftoken=.....
Transfer-Encoding: chunked
Vary: Cookie, Accept-Encoding
X-Frame-Options: DENY
```

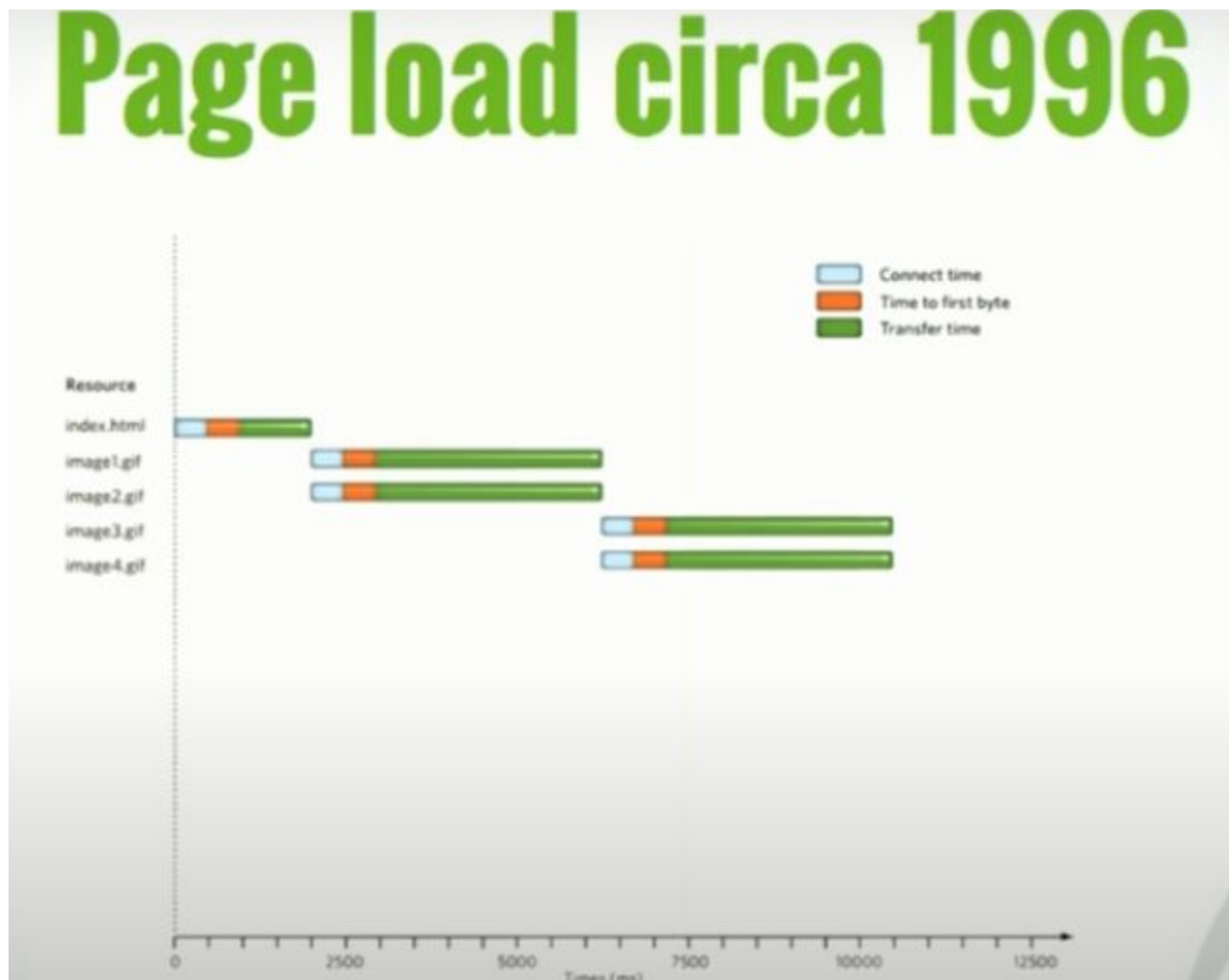
Response headers

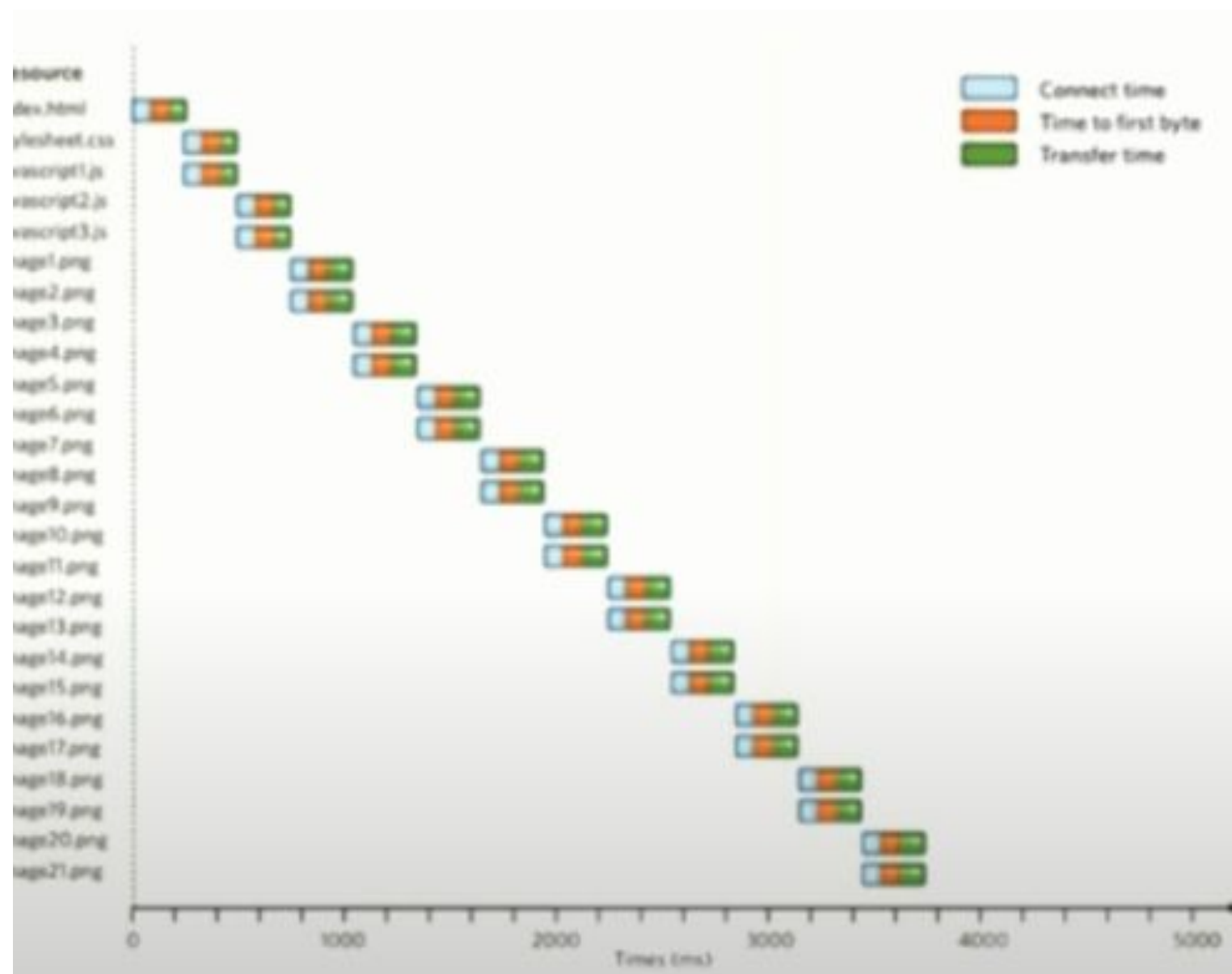
Entity headers

General headers

(body)

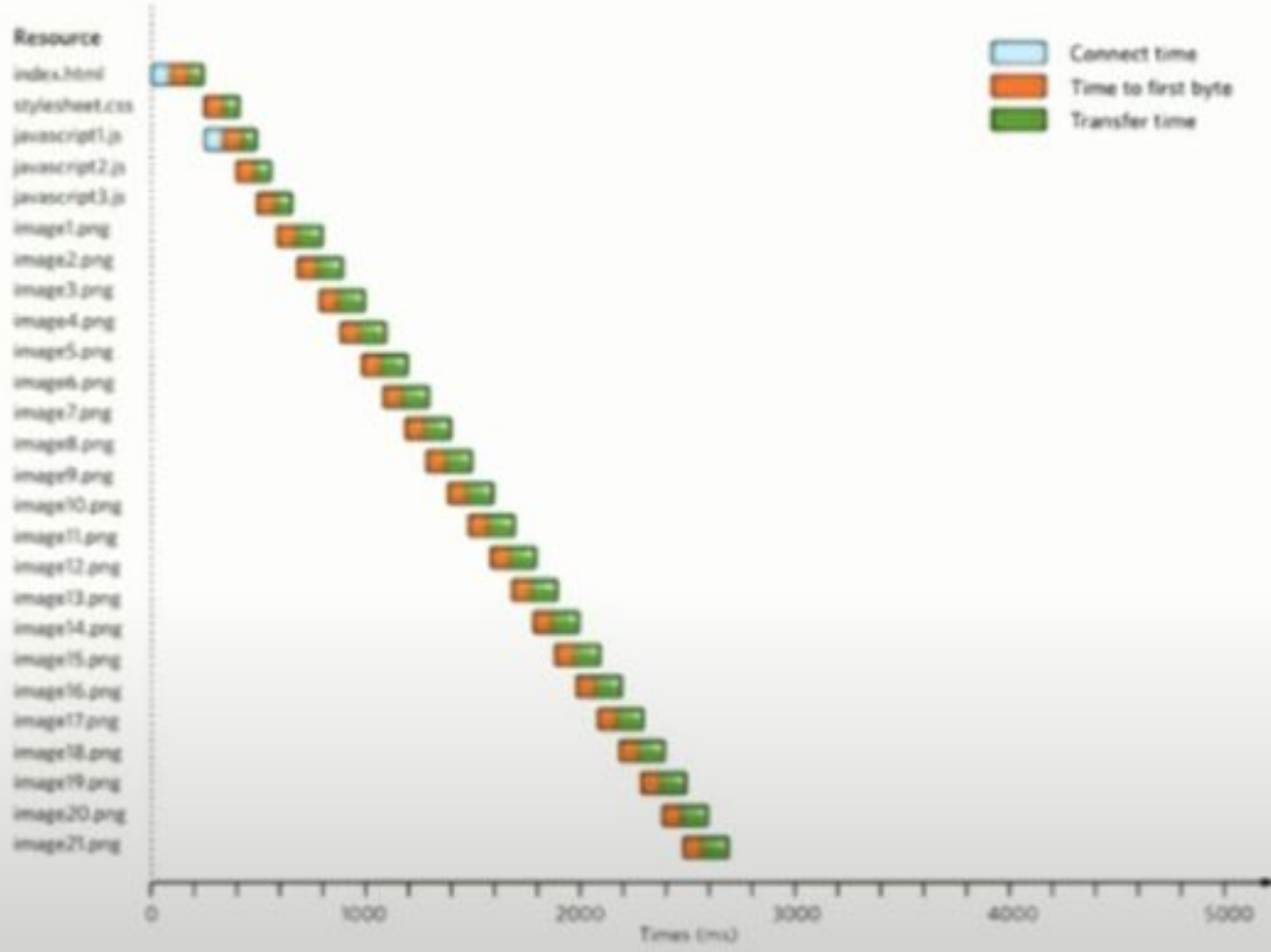
- Создание соединения
- Запрос
- Загрузка данных



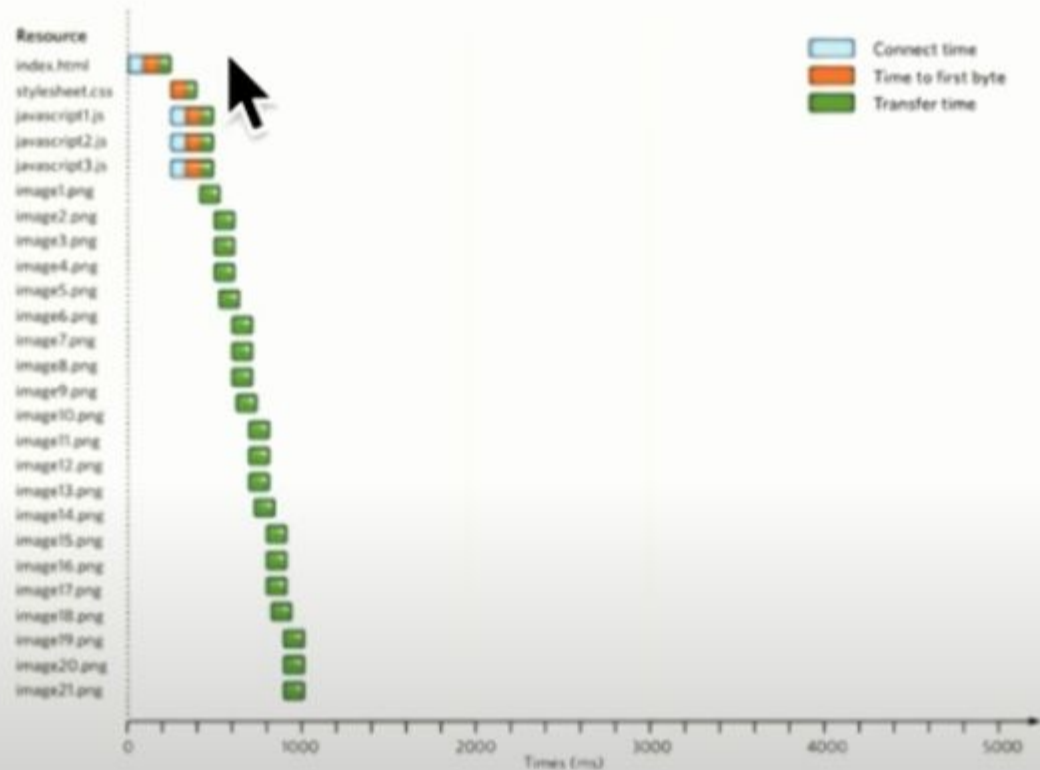


HTTP 1.1

Connection re-use



Pipelining - if it worked



Performance optimizations

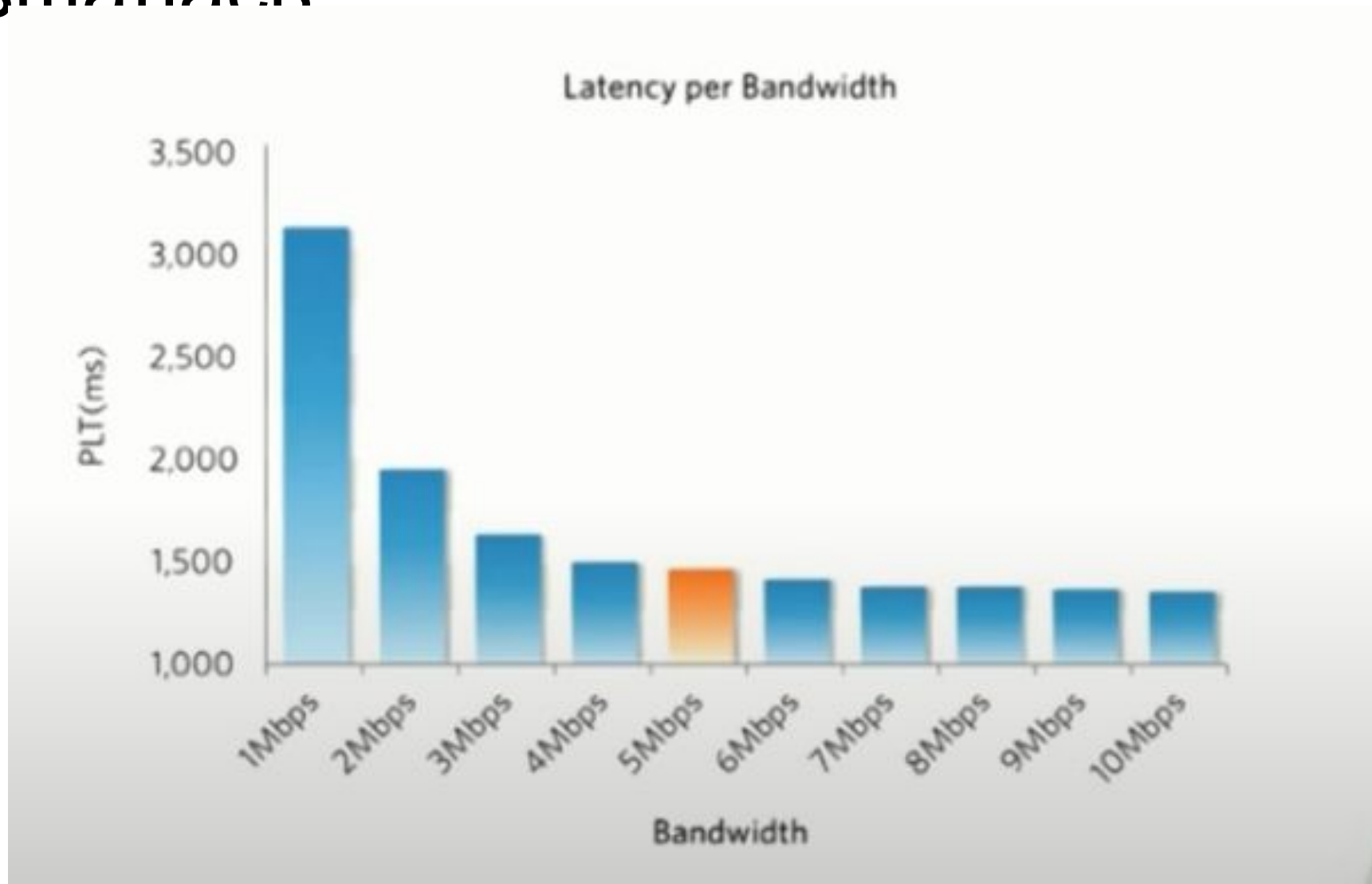
a.k.a. "ugly hacks"

- **Spriting**
- **Inlining**
- **Concatenating**
- **Sharding**



Spriting

С увеличением пропускной способности задержка при передаче запроса почти не уменьшается



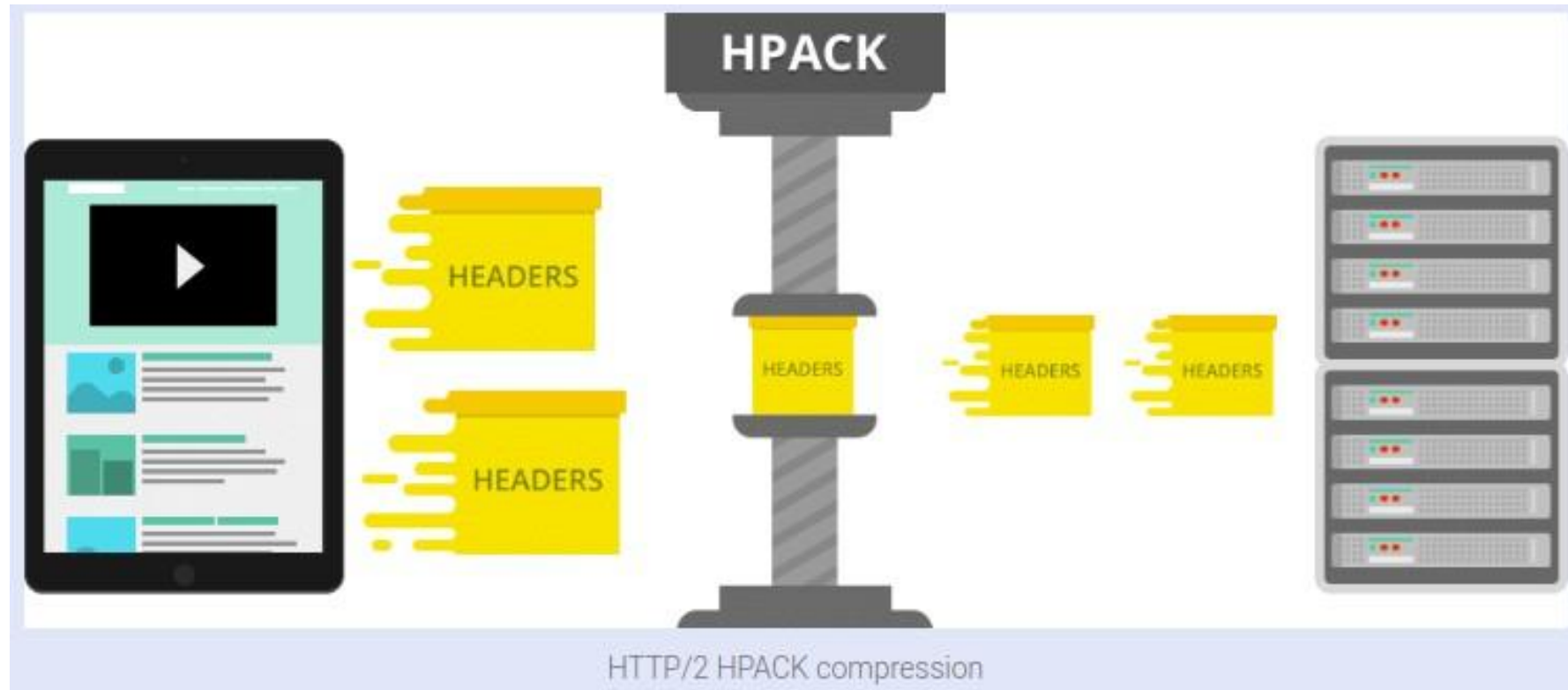
Бинарность

HTTP/1 – запросы передаются в виде текстовых сообщений,
HTTP/2 — в двоичном формате.

Поэтому протокол более эффективен при парсинге, более компактный при передаче, подвержен меньшему количеству ошибок.

Компрессия заголовков HPACK

В HTTP/1 заголовок отправляется в виде простого текста. А учитывая, что может отправляться очень много запросов, все заголовки могут занимать большой объём. Поэтому в HTTP/2 есть сжатие заголовков HPACK, которое позволяет существенно сократить объём передаваемой информации.



Мультиплексирование

В HTTP 1.1 браузеры используют множественные подключения к серверу для загрузки веб-страницы, причем, количество таких соединений ограничено. Но это не решает проблему с блокированием канала медленными пакетами. Тогда как в HTTP/2 используется мультиплексирование, которое позволяет браузеру использовать одно соединение TCP для всех запросов.

В HTTP/2 все файлы подгружаются параллельно. Запросы и ответы разделяются по фреймам с мета-данными, которые ассоциируют запросы и ответы. Так что они не перекрывают друг друга и не вызывают путаницы. При этом ответы получаются по мере их готовности, следовательно, тяжелые запросы не будут блокировать обработку и выдачу более простых объектов.

Приоритизация

Вместе с мультиплексированием появилась приоритизация трафика. Запросам можно назначить приоритет на основе важности и зависимости.

Так что при загрузке веб-страницы браузер будет в первую очередь получать важные данные, CSS-код, к примеру, а все второстепенное обрабатывается в последнюю очередь.

Server Push

При использовании протокола HTTP/1 браузер запрашивает страницу, сервер отправляет в ответ HTML и ждет, пока браузер его обработает и запросит все необходимые файлы: JavaScript, CSS и фото. Поэтому в новый протокол внедрили функцию под названием Server Push.

Она позволяет серверу сразу же, не дожидаясь ответа веб-браузера, добавить нужные по его мнению файлы в кэш для быстрой выдачи.