

From HTTP 1.1 to HTTP 2.0

# Bandwidth and Latency

## Bandwidth

*In computing, bandwidth is the bit-rate of available or consumed information capacity expressed typically in metric multiples of bits per second. Various, bandwidth may be characterized as network bandwidth, data bandwidth, or digital bandwidth.*

## Latency

*Latency is a time interval between the stimulation and response, or, from a more general point of view, as a time delay between the cause and the effect of some physical change in the system being observed.*

Delay	User reaction
0 - 100ms	Instant
100 - 300ms	Slight perceptible delay
300 - 1000ms	Task focus, perceptible delay
1 s+	Mental context switch
10 s+	I'll come back later...



**"1000 ms  
time to  
glass  
challenge"**

- *Simple user-input must be acknowledged within ~100 milliseconds.*
- *To keep the user engaged, the task must complete within 1000 milliseconds.*

***Ergo, our pages should render within 1000 milliseconds.***

# The World Wide Web in 1996



The screenshot shows the Yahoo! homepage from 1996. At the top, the "YAHOO!" logo is prominently displayed in red. To its left are three small icons: a green one with a star, a red one with a heart, and a purple one with a flower. Below these icons are the words "NEW", "COOL", and "RANDOM". To the right of the logo are three more icons: a yellow one with a star, a blue one with a heart, and a red one with a flower. Below these icons are the words "HEAD LINES", "YAHOO! INFO", and "ADD URL".

Below the logo, there are several links: "Yahoo! Deutschland", "CLICK HERE TO VISIT THE STARS", "LOS ANGELES", and "Weekly Picks".

In the center, there is a search bar with a "Search" button and a link to "Options".

Below the search bar, there are several links: "Yellow Pages", "People Search", "City Maps", "News Headlines", "Stock Quotes", and "Sports Scores".

At the bottom, there is a list of categories, each followed by a list of sub-categories and a link to "More":

- [Arts](#) - - [Humanities](#), [Photography](#), [Architecture](#), ...
- [Business and Economy \[Xtra!\]](#) - - [Directory](#), [Investments](#), [Classifieds](#), ...
- [Computers and Internet \[Xtra!\]](#) - - [Internet](#), [WWW](#), [Software](#), [Multimedia](#), ...
- [Education](#) - - [Universities](#), [K-12](#), [Courses](#), ...
- [Entertainment \[Xtra!\]](#) - - [TV](#), [Movies](#), [Music](#), [Magazines](#), ...
- [Government](#) - - [Politics \[Xtra!\]](#), [Agencies](#), [Law](#), [Military](#), ...
- [Health \[Xtra!\]](#) - - [Medicine](#), [Drugs](#), [Diseases](#), [Fitness](#), ...
- [News \[Xtra!\]](#) - - [World \[Xtra!\]](#), [Daily](#), [Current Events](#), ...
- [Recreation and Sports \[Xtra!\]](#) - - [Sports](#), [Games](#), [Travel](#), [Autos](#), [Outdoors](#), ...
- [Reference](#) - - [Libraries](#), [Dictionaries](#), [Phone Numbers](#), ...
- [Regional](#) - - [Countries](#), [Regions](#), [U.S. States](#), ...
- [Science](#) - - [CS](#), [Biology](#), [Astronomy](#), [Engineering](#), ...
- [Social Science](#) - - [Anthropology](#), [Sociology](#), [Economics](#), ...
- [Society and Culture](#) - - [People](#), [Environment](#), [Religion](#), ...

# The World Wide Web Today

**YAHOO!**

Buscar en la Web

Iniciar sesión

Correo

Correo

Noticias

Deportes

Finanzas

Celebrity

Vida y Estilo

Cine

Horóscopo

Videos


Más >

eBay

Amazon

Meetic

Al utilizar Yahoo, aceptas que nosotros y nuestros socios podamos definir cookies para distintos fines, tales como personalizar el contenido y la publicidad.



### 10 trucos para acelerar tu metabolismo

No tienes que pasarte el día en el gimnasio, basta con entrenamientos en intervalos de alta intensidad para quemar calorías, y sin dieta [Maneras rápidas de perder peso](#) »

1-5 de 45

II


Titulares

Noticias

Deportes

Finanzas

Celebrity



### Liga - De Gea-United 2019: Algunas sorprendentes preguntas sin respuesta


El portero David de Gea ha renovado su contrato con el Manchester United y pone punto y final a uno de los grandes culebrones de los últimos tiempos.

Eurosport

### Los 10 lugares donde mejor se come de España

Desde Sevilla a San Sebastián haciendo parada en Cádiz, Madrid o Segovia, nos vamos a comer el país, bocado a bocado

Skyscanner Patrocinado



Lo más buscado

1 [Liga BBVA](#)

2 [US Open](#)

3 [Casas rurales](#)

4 [Eurobasket 2015](#)

5 [Horóscopo](#)

6 [Oferta hoteles](#)

7 [Lionel Messi](#)

8 [Vestidos mujer](#)


9 [Floyd Mayweather](#)

10 [Previsión tiempo](#)

NUEVO FORD ECOSPORT

>Apertura Sin Llave

Desde 12.990€



Descúbrelo

Go Further

29660, Marbella (Ubicación actual)

27°F | °C

Buen tiempo

Hoy

Lu.

Ma.

29° 19°

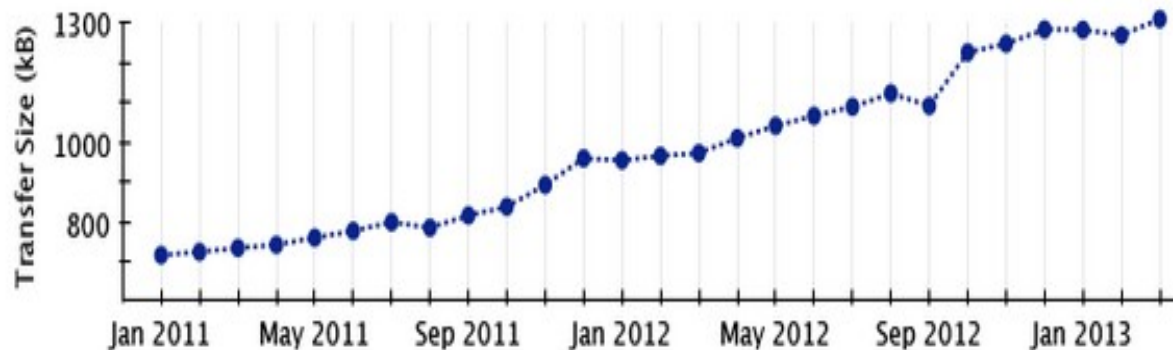
29° 19°

28° 18°

Ver más »

Establecer YAHOO! como página de inicio

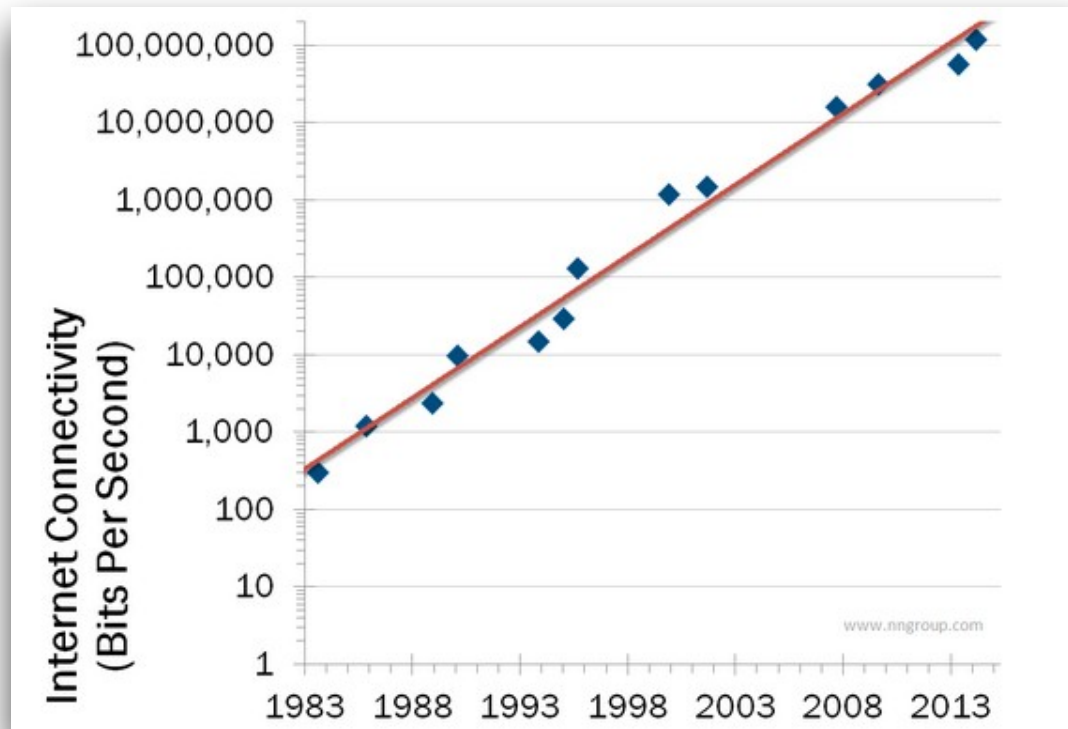
# Our applications are **complex,** **and growing...**



Content Type	Desktop		Mobile	
	Avg # of requests	Avg size	Avg # of requests	Avg size
HTML	10	56 KB	6	40 KB
Images	56	856 KB	38	498 KB
Javascript	15	221 KB	10	146 KB
CSS	5	36 KB	3	27 KB
Total	86+	1169+ KB	57+	711+ KB

← **Ouch!**

# Nielsen's Law of Bandwidth



1984 - 300 bps

50% Growth per Year

# Bandwidth by Country

Position	Country	Speed (Mbps)
1	South Korea	25.3
2	Hong Kong	16.3
3	Japan	15
4	Switzerland	14.5
...	...	...
12	United States	11.5
13	Belgium	11.4
...	...	...
24	Germany	8.7
...	...	...
28	Spain	7.8
...	...	...
30	Australia	6.9
31	France	6.9
...	...	...
55	Bolivia	1.1

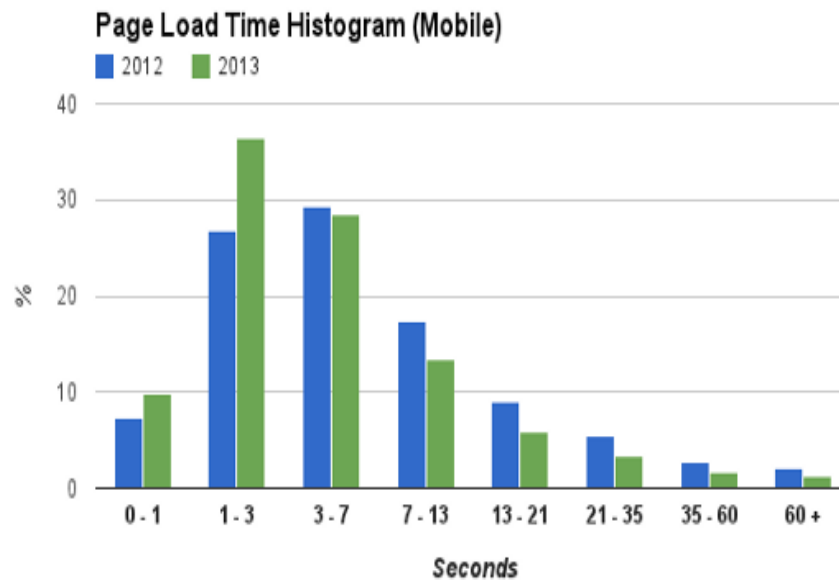
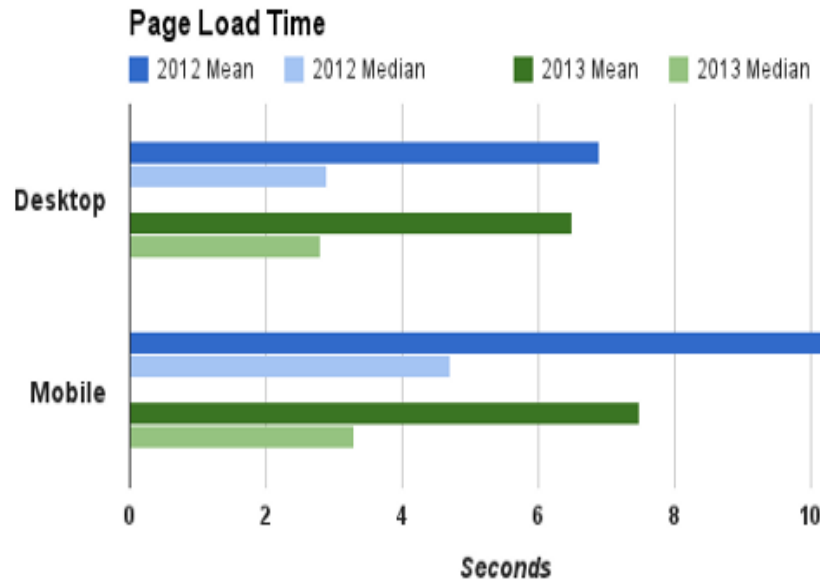


# Mobile Networks

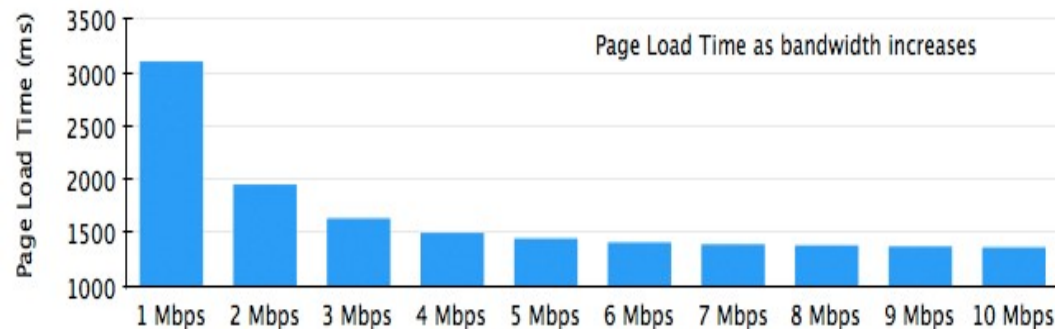
Region	Average Speed (Mbps)
Europe	20.4
North America	9.6
Asia Pacific	8.8
South America	7.0
Africa	4.8

Desktop: ~3.1 s

Mobile ~3.5 s



# Latency vs. Bandwidth impact on Page Load Time



*Single digit  
perf  
improvement  
after 5 Mbps*



Average household is running on a **5 Mbps+** connection. Ergo, **average consumer would not see an improvement in page loading time by upgrading their connection.**

(doh!)

[Bandwidth doesn't matter  
\(much\)](#) - Google

- Improving bandwidth is "easy"...
  - 60% of new capacity through upgrades in past decade + unlit fiber
  - *"Just lay more fiber..."*
- **Improving latency is expensive... impossible?**
  - Bounded by the speed of light - oops!
  - We're already within a small constant factor of the maximum
  - *"Shorter cables?"*






















**\$80M /  
ms**

Latency is the new Performance  
Bottleneck

@igrigo  
rik

And latency is per connection

# Typical Web Page

Name	Method	Status	Type	Initiator	Size	Time	▲
 comunes.css	GET	200	stylesheet	(index):37	(from cache)	42 ms	
 pxlctl2.gif?r=2&m=1&s=25c10673e...	GET	302	gif	http://pxlctl.elpais.com/pxlc...	508 B	58 ms	
 advert.gif?648618153	GET	200	gif	(index):164	392 B	59 ms	
 caja_tlife.css	GET	200	stylesheet	(index):1619	793 B	70 ms	
 pxlctl2.gif?m=1&r=2&w=2123515	GET	302	gif	http://elpais.com/pxlctl.gif?...	347 B	80 ms	
 ?script=0&random=4019638651&ip...	GET	200	gif	http://www.google.com/ads...	343 B	94 ms	
 gpt.js	GET	304	script	pbs.slots.js:1	363 B	95 ms	
 rta.js?netId=4045&cookieName=crt...	GET	200	script	pbs.slots.js:1	457 B	118 ms	
 pxlctl.gif?m=1&r=2&w=2123515	GET	302	text/html	(index):164	241 B	121 ms	
 elpais.com	GET	200	document	http://www.elpais.es/	61.9 KB	123 ms	
 ?script=0&random=4019638651	GET	302	text/html	http://googleads.g.doublecli...	380 B	138 ms	
 rep.gif?ver=1&typ=pgv&rnd=iemvn...	GET	200	gif	(index):165	404 B	147 ms	
 bid?src=3226&u=http%3A%2F%2Felp...	GET	200	script	amzn_ads.js:1	217 B	191 ms	
 s64909375414717?AQB=1&pccr=tr...	GET	302	text/plain	http://prisacom.112.2o7.net...	811 B	213 ms	
 ?value=0&guid=ON&script=0	GET	302	gif	(index):164	651 B	226 ms	
 ?idsite=255&url=http%3A%2F%2Felp...	GET	200	xhr	vrs.20150907.js:6	337 B	355 ms	
 ?transport=jsonp&idsite=255&url=h...	GET	200	xhr	vrs.20150907.js:6	1.1 KB	369 ms	
 s64909375414717?AQB=1&ndh=1&...	GET	302	text/plain	(index):165	1.8 KB	432 ms	
 ads?gdfp_req=1&correlator=231131...	GET	200	script	pubads_impl_72.js:189	29.4 KB	528 ms	

# Optimization Possibilities with HTTP 1.1

# Optimizations...

## **Reduce DNS lookups**

Every hostname resolution requires a network roundtrip, imposing latency on the request and blocking the request while the lookup is in progress.

## **Make fewer HTTP requests**

No request is faster than a request not made:  
eliminate unnecessary resources on your pages.

## **Use a Content Delivery Network**

Locating the data geographically closer to the client can significantly reduce the network latency of every TCP connection and improve throughput.



# Optimizations...

## Add an Expires header and configure Etags

An Expires header can be used to specify the cache lifetime of the object, allowing it to be retrieved directly from the user's cache and eliminating the HTTP request entirely.

ETags and Last-Modified headers provide an efficient cache revalidation mechanism—effectively a fingerprint or a timestamp of the last update.

### Response Headers

- . Access-Control-Allow-Origin:\*
- . Cache-Control:max-age=604800
- . Connection:keep-alive
- . Date:Wed, 2 Mar 2016 19:56:34 GMT
- . ETag:"~~13630e1-b438-524daace96280~~"
- . Expires:Mon, 30 Nov 2015 19:56:34 GMT
- . Last-Modified:Thu, 16 Apr 2015 10:26:33 GMT
- . Link:https://domain.com/foobar.css
- . X-Edge-Location:usch

# Optimizations...

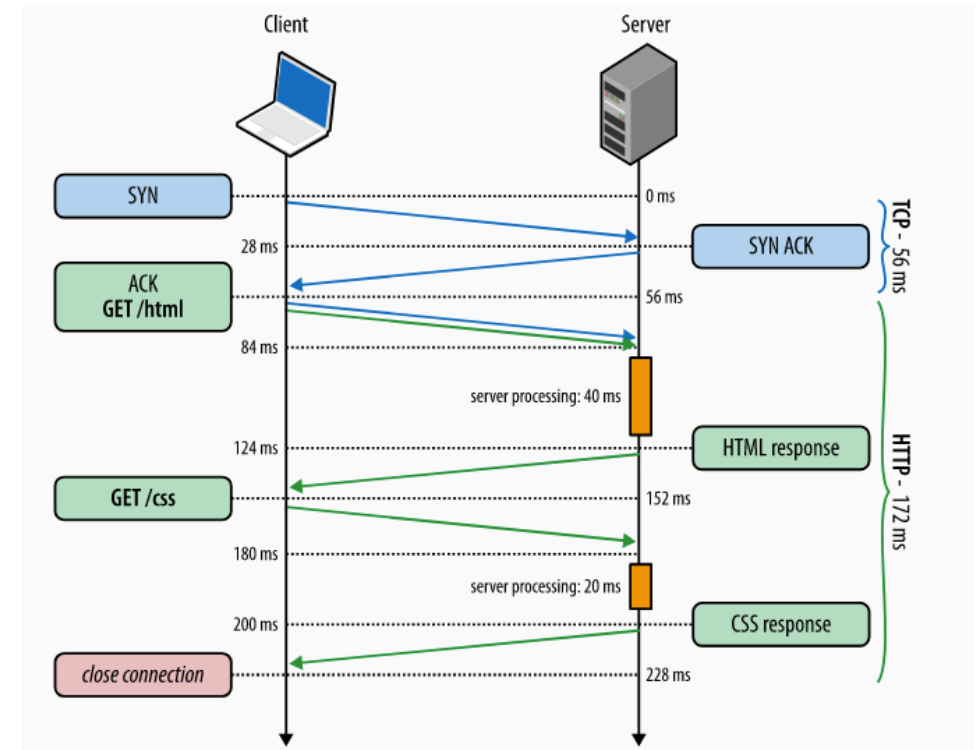
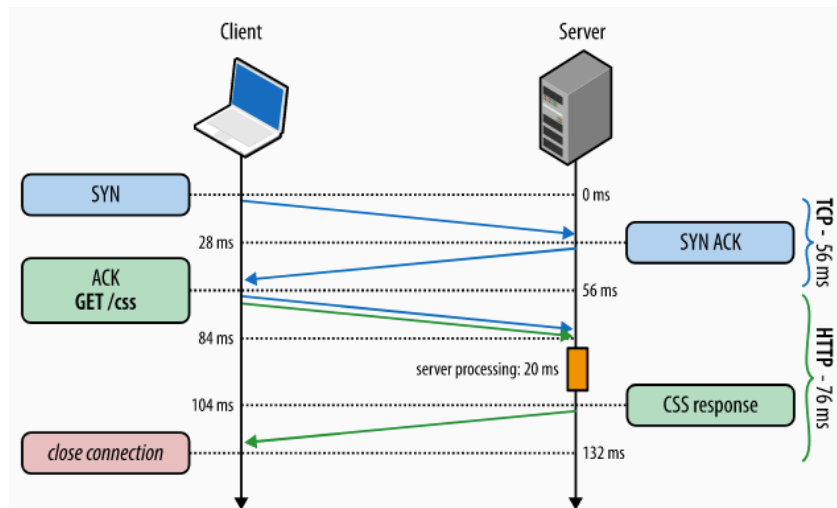
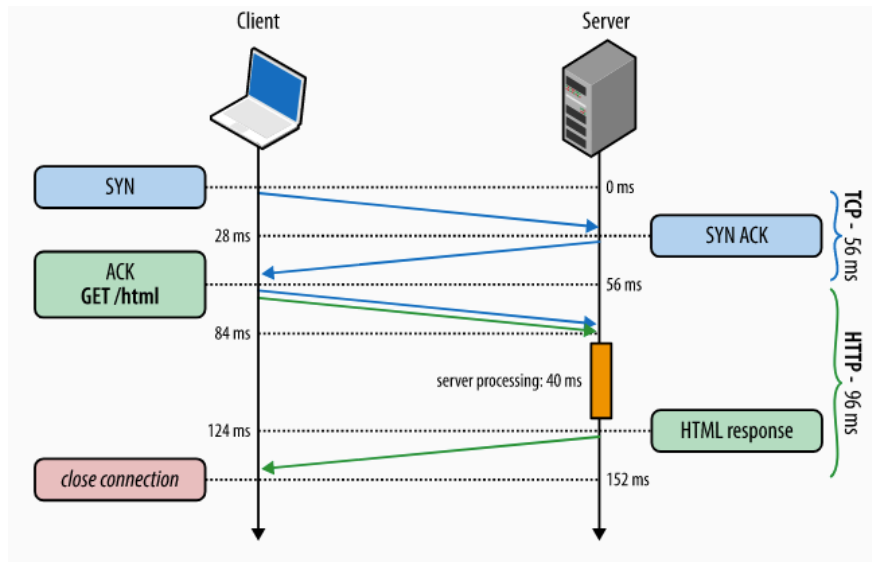
## **Gzip assets**

All text-based assets should be compressed with Gzip when transferred between the client and the server.

On average, Gzip will reduce the file size by 60–80%, which makes it one of the simpler (configuration flag on the server) and high-benefit optimizations you can do.

## **Avoid HTTP redirects**

HTTP redirects can be extremely costly, especially when they redirect the client to a different hostname, which results in additional DNS lookup, TCP connection latency, etc.



# Using Keep-Alive

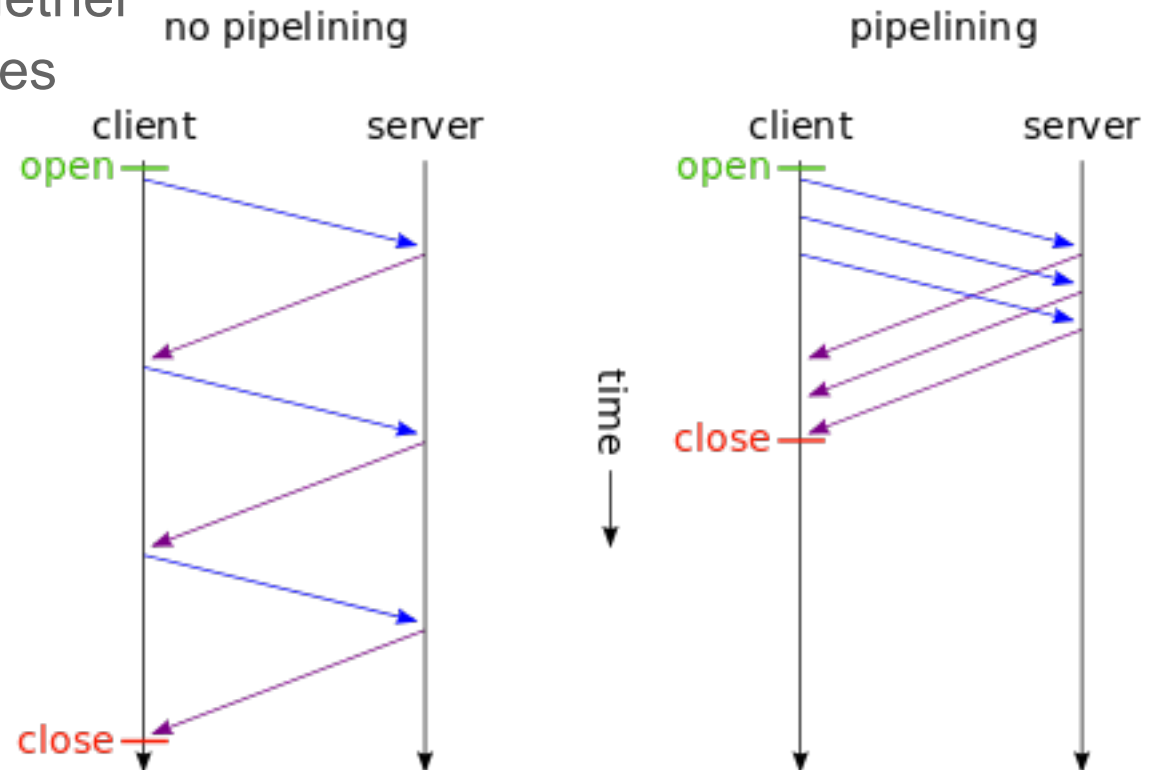
# Keep Alive-Pipelining

Using a single connection to send multiple successive requests

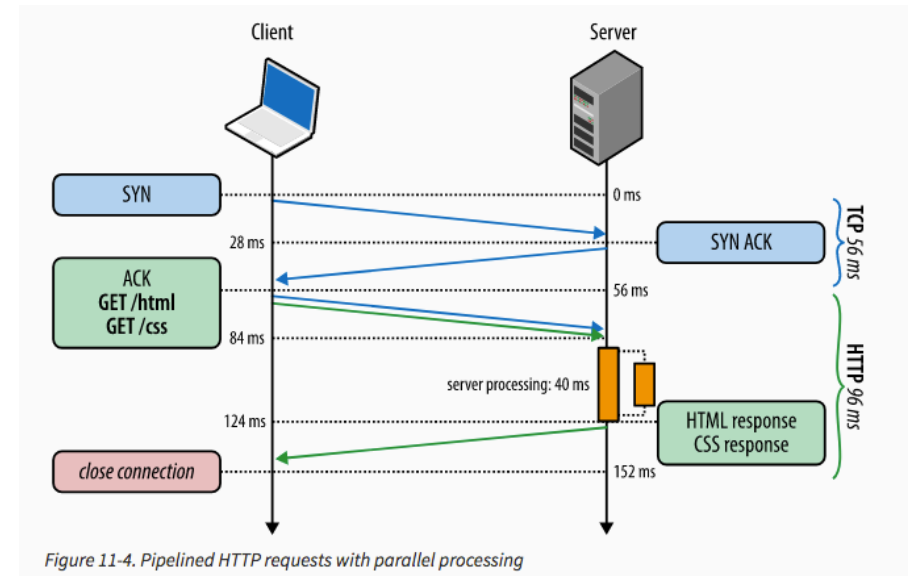
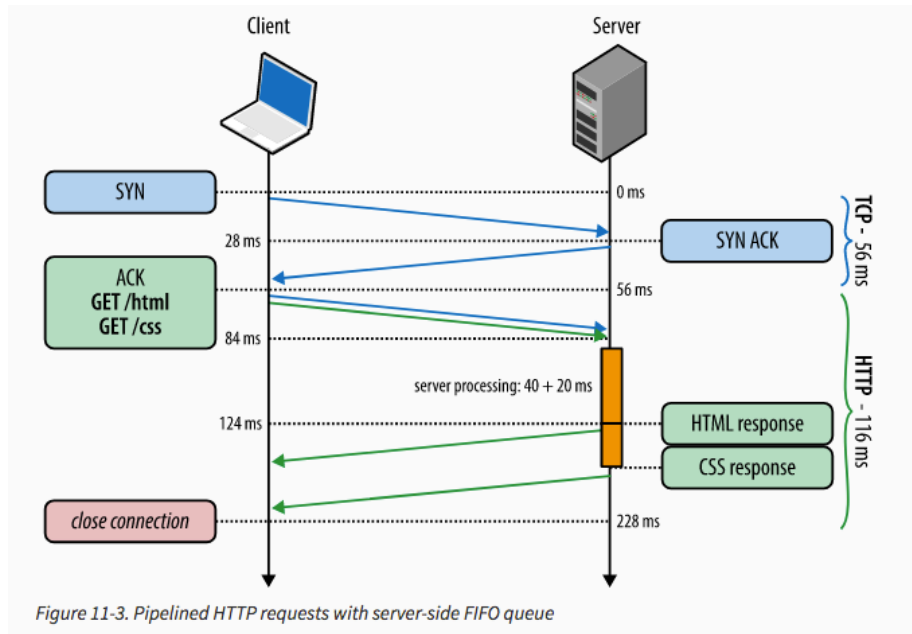
Pipelining Requests

Send several requests together

Head of line blocking issues



# Keep-alive + pipelining



# Keep Alive-Pipelining

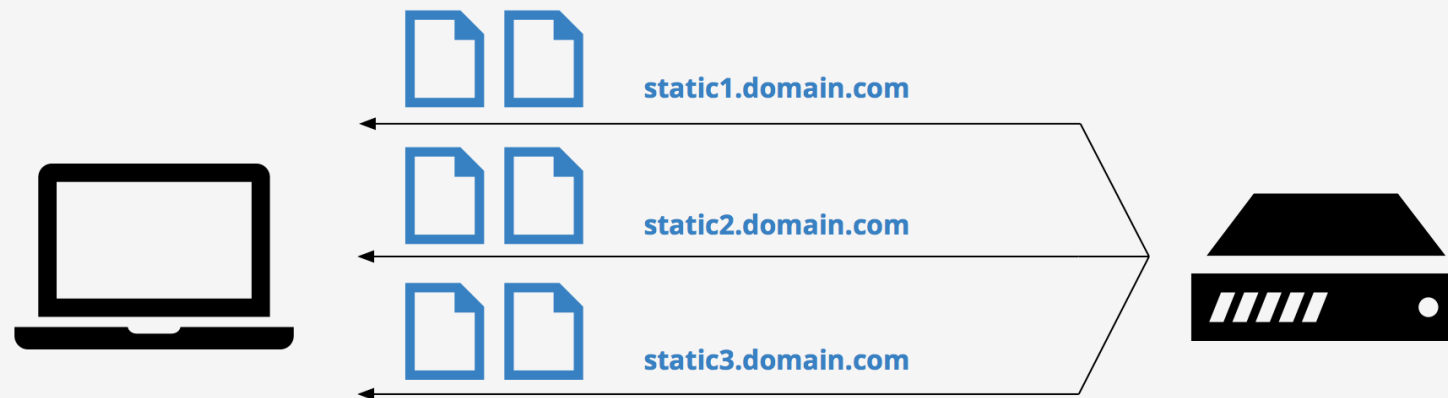
The pipelining of requests results in **a dramatic improvement] in the loading times of HTML pages**, especially over high latency connections such as satellite Internet connections.

The speedup is less apparent on broadband connections, as the limitation of HTTP 1.1 still applies: **the server must send its responses in the same order that the requests were received — so the entire connection remains first-in-first-out and Head Of Line blocking can occur.**

# Domain sharding

- Web browsers traditionally place limits on the amount of concurrent downloads allowed for each domain (between 2-16). This limit was put in place by the Internet Engineering Task Force and is mentioned in the HTTP/1.1 specification. It was recommended in order to reduce Internet congestion and web server overloading.

# Domain sharding

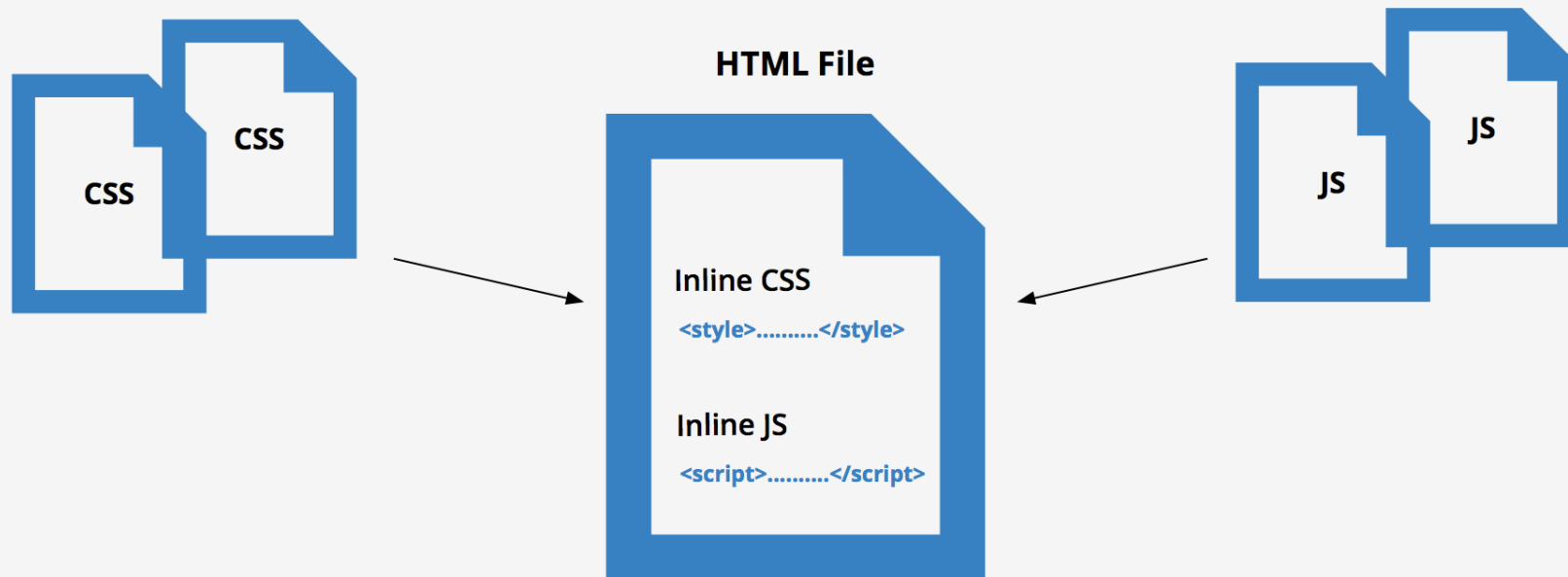


**Domain Sharding**



# Inlining

- Inline Small CSS and Javascript



**Inline Small CSS and Javascript**

# Spriting

- Multiple images are combined into a larger, composite image.

## **Concatenating files (JavaScript, CSS)**

- Reduces number of downloads and latency overhead
- Less modular code and expensive cache invalidations (e.g. app.js)
- Slower execution (must wait for entire file to arrive)

## **• Spriting images**

- Reduces number of downloads and latency overhead
- Painful and annoying preprocessing and expensive cache invalidations
- Have to decode entire sprite bitmap - CPU time and memory

## **• Domain sharding**

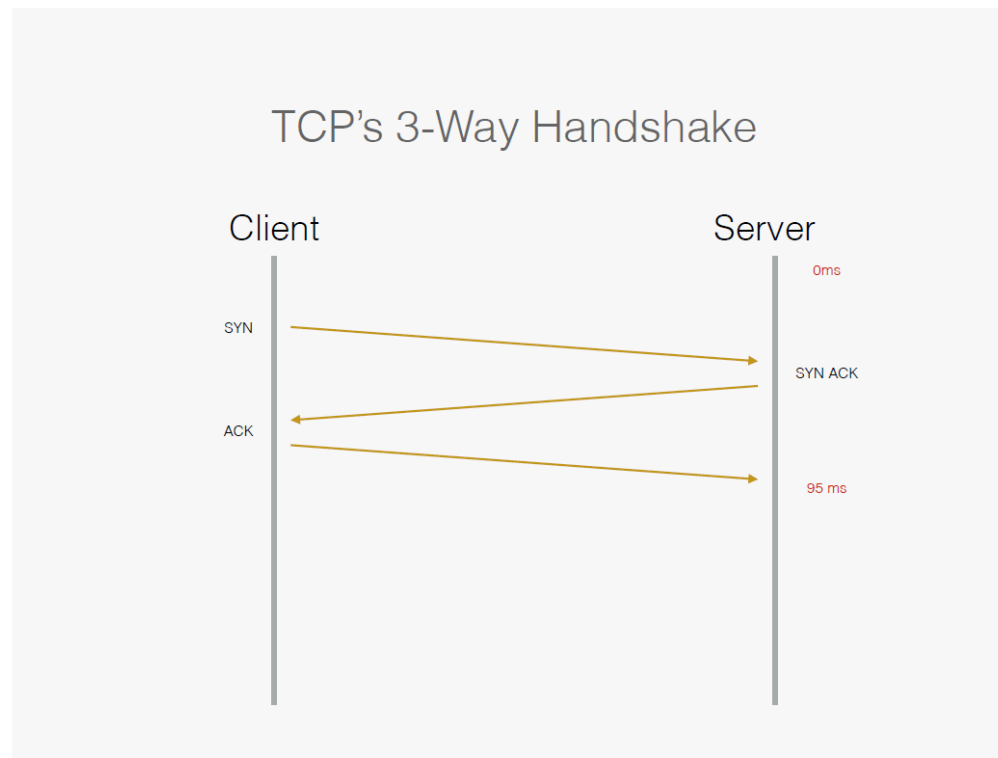
- TCP Slow Start? Browser limits, Nah... 15+ parallel requests -- *Yeehaw!!!*
- Causes congestion and unnecessary latency and retransmissions

## **• Resource inlining**

- Eliminates the request for small resources
- Resource can't be cached, inflates parent document
- 30% overhead on base64 encoding

# The culprit is HTTP on TCP

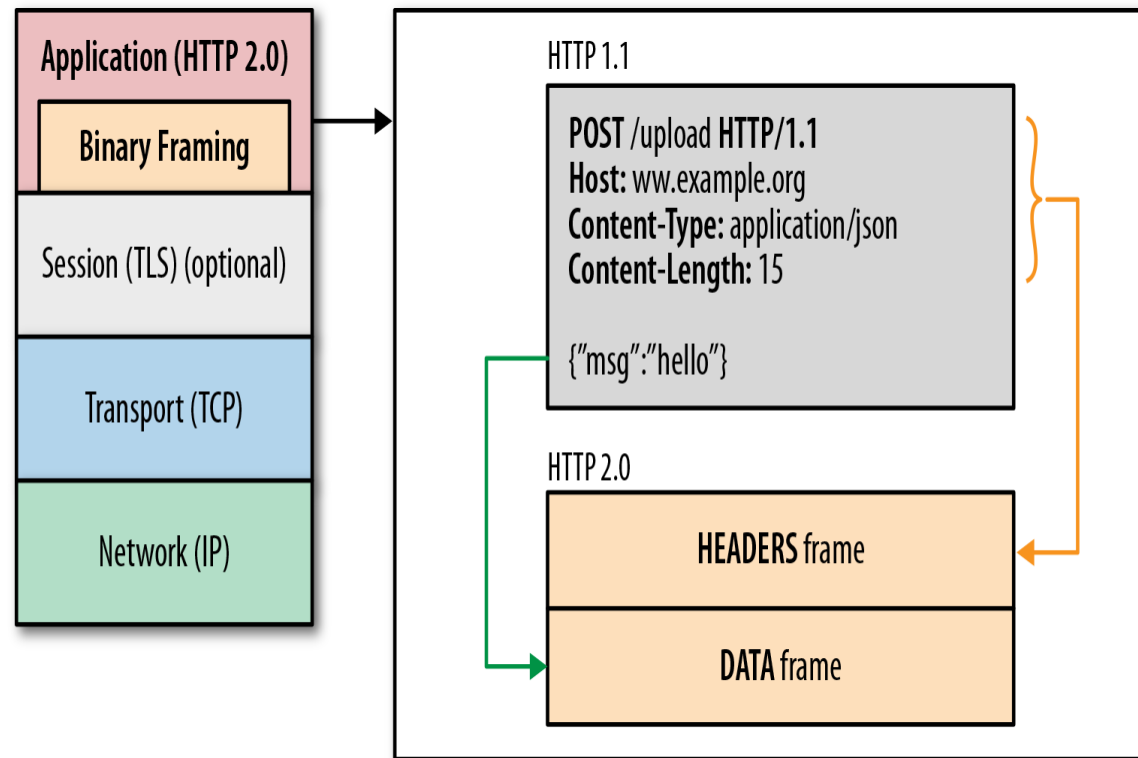
Http 1.1 chatty - TCP is not made for chatty protocols  
TCP has slow start and head of line blocking



Most HTTP transfers are short and bursty, whereas TCP is optimized for long-lived, bulk data transfers

# HTTP 2.0 in a nutshell

- One TCP connection
- Request = Stream
  - Streams are multiplexed
  - Streams are prioritized
- (New) binary framing layer
  - Prioritization
  - Flow control
  - Server push
- Header compression



*“... **we’re not replacing all of HTTP** — the methods, status codes, and most of the headers you use today will be the same. Instead, **we’re re-defining how it gets used “on the wire” so it’s more efficient**, and so that it is more gentle to the Internet itself ....”*

- Mark Nottingham  
(chair)



# All frames have a common 8-byte header

Bit	+0..7		+8..15	+16..23	+24..31
0	Length			Type	Flags
32	R	Stream Identifier			
...	Frame Payload				

- Length-prefixed frames
- **Type** indicates ...type of frame
  - *DATA, HEADERS, PRIORITY, PUSH\_PROMISE, ...*
- Each frame may have custom **flags**
  - e.g. *END\_STREAM*
- Each frame carries a **31-bit stream identifier**
  - After that, it's frame specific payload...

```
frame =  
buf.read(8)  
if  
    frame_i_care  
    _about  
    do_someth  
    ing_smart  
else  
    buf.skip(fram
```

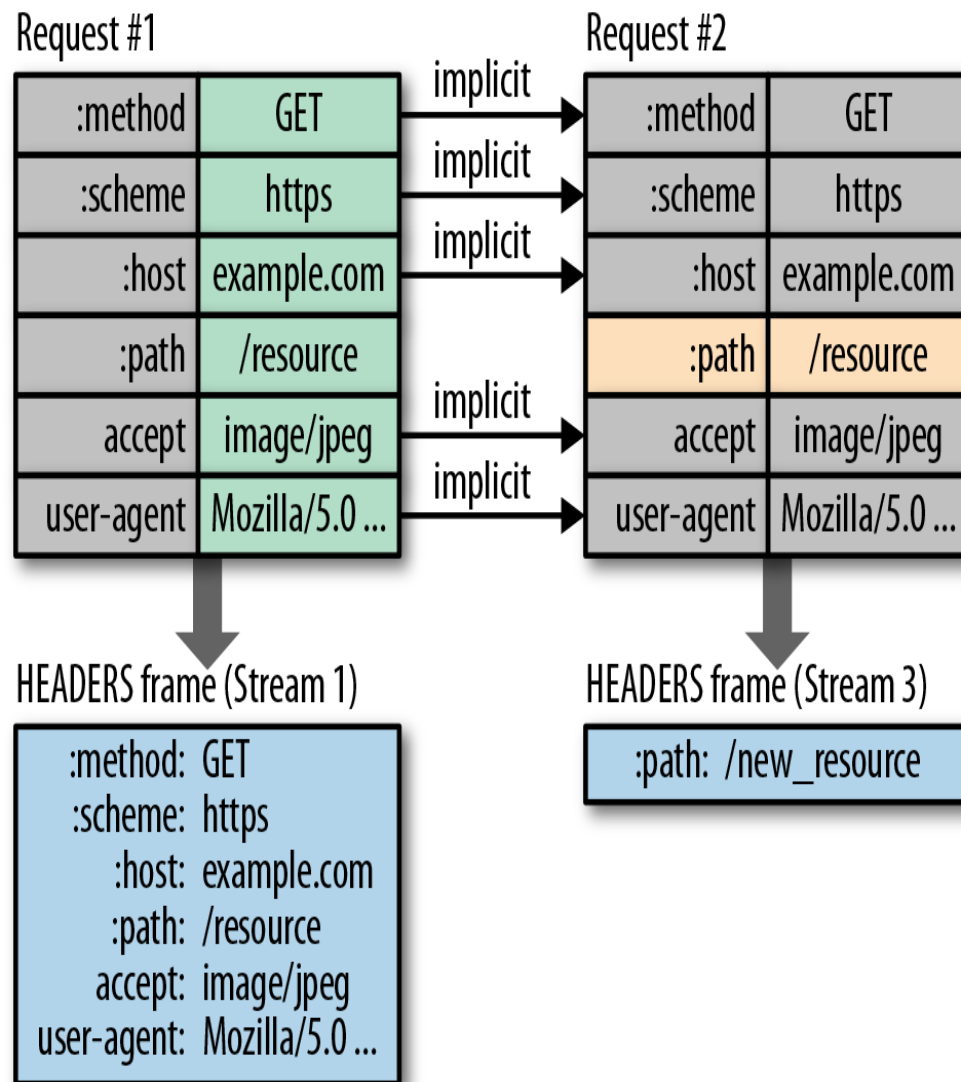
# Opening a new stream with HTTP 2.0 (HEADERS)

Bit	+0..7		+8..15	+16..23	+24..31
0	Length			Type (1)	Flags
32	R	Stream Identifier			
64	X	Priority			
...	Header Block				

- Common 8-byte header
- Client / server allocate new stream ID
  - *client: odd, server: even*
- Optional 31-bit stream priority field
  - *Flags indicates if priority is present*
  - *2<sup>31</sup> is lowest priority*
- HTTP header payload
  - see [header-compression-01](#)



# HTTP 2.0 header compression (in a nutshell)



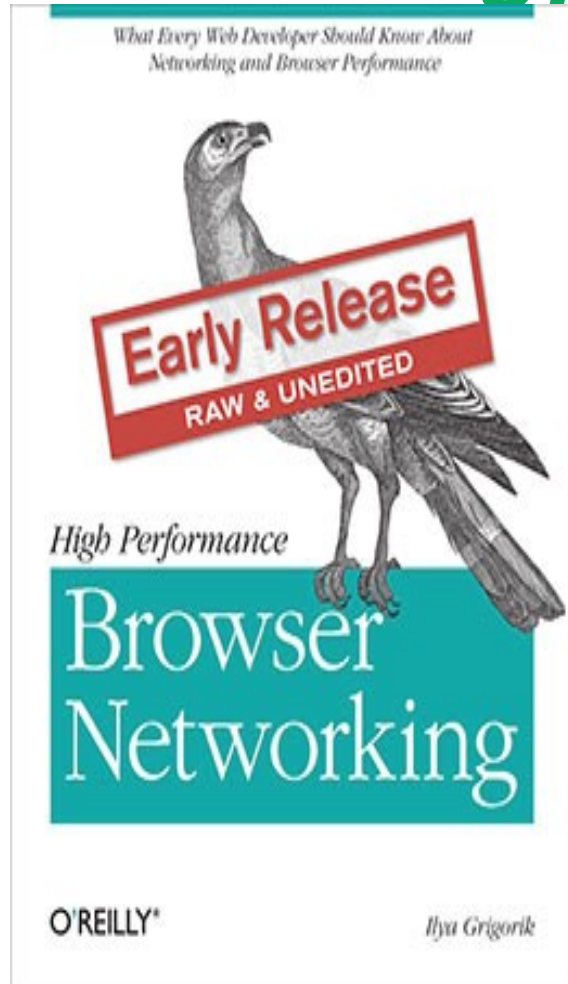
- Each side maintains “header tables”
- Header tables are initialized with common header key-value pairs
- New requests “toggle” or “insert” new values into the table
- New header set is a “diff” of the previous set of headers
- *E.g. Repeat request (polling) with exact same headers incurs no overhead (sans frame header)*

# Sending application data with ...DATA frames.

Bit	+0..7		+8..15	+16..23	+24..31
0	Length			Type (0)	Flags
32	R	Stream Identifier			
...	HTTP payload				

- Common 8-byte header
- Followed by application data...
- In theory, max-length =  $2^{16}-1$
- To reduce head-of-line blocking: **max frame size is  $2^{14}-1$**  (~16KB)
  - Larger payloads are split into multiple DATA frames, last frame carries "END\_STREAM" flag

*For an in-depth discussion on all  
of the above...*



<http://hpbn.co>

- Optimizing **TCP** server stacks
- Optimizing **TLS** deployments
- Optimizing for **mobile** networks
- HTTP 2.0 features, framing, deployment...
- XHR, SSE, WebSocket, WebRTC, ...