

# 深入理解 HTTP

从 HTTP/1.x 到 HTTP/3

# HTTP 的发展史

HTTP/0.9

HTTP/1.0

HTTP/1.1

SPDY

SPDY/2.0

SPDY/3.0

SPDY/4.0

HTTP/2.0

HTTP/3.0

1991 1996 1997 1999 2006 2008 2009 2012 2013 2014 2015 2016

TLS1.0

TLS1.1

TLS1.2

# HTTP/0.9

1. 纯文本传输

2. 只支持 GET

# HTTP/1.0

1. 增加了 HEAD、POST 等方法
2. 增加了相应状态码，标记可能的错误原因
3. 引入了协议版本号的概念
4. 引入了 HTTP Header 的概念，让 HTTP 处理请求和响应更加灵活
5. 传输的数据不再仅限于文本

# HTTP/1.1

1. 增加了 PUT、DELETE 等新方法
2. 增加了缓存管理和控制
3. 长链接
4. 允许响应数据分块 (chunked) , 利于传输大文件
5. 强制要求 Host 头, 让互联网主机托管成为可能

# HTTP/2

1. 多路复用
2. 二进制分帧层，不再是纯文本
3. 头部压缩 HPACK 算法
4. 请求优先级
5. 服务器推送
6. 增强了安全性，“事实上”的加密通信

# HTTP/3

1. 减少了 TCP 三次握手及 TLS 握手时间
2. 改进了拥塞控制
3. 避免队头阻塞的多路复用
4. 连接迁移
5. 前向冗余纠错

# HTTP/1.x 基础知识

1. HTTP 优缺点概述
2. 报文结构
3. 请求方法以及状态码
4. 重点特性详解

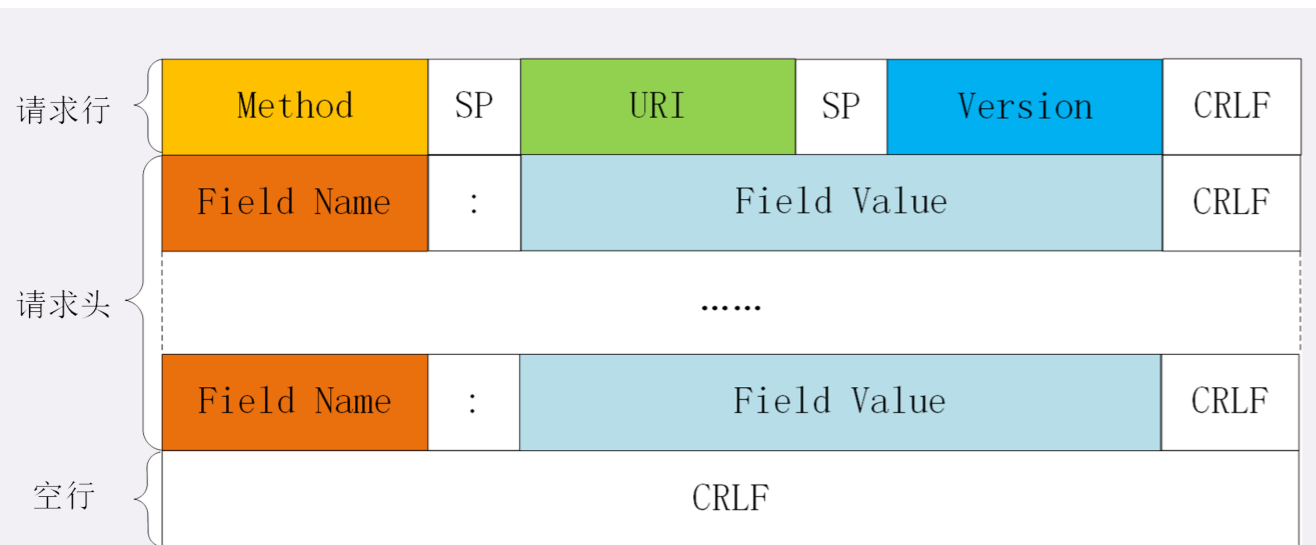


# HTTP 优缺点概述

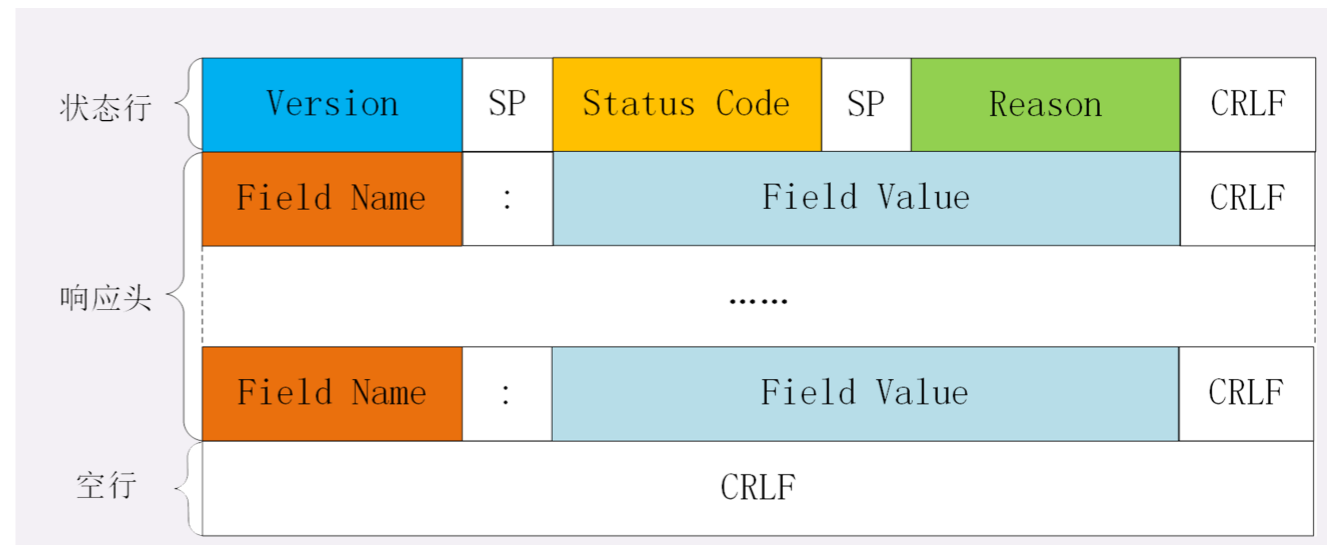
1. 简单灵活易扩展
2. 无状态
3. 明文传输
4. 不安全
5. 性能问题

# 报文结构

## 请求报文



## 响应报文



# 请求报文结构

The image shows a Wireshark network traffic analysis window. The top toolbar includes various icons for file operations, search, and zooming. The main display area is divided into several sections:

- Packet List:** A table showing captured packets. The selected packet (607) is highlighted in green. The table has columns for Source, Destination, Protocol, Length, and Info.
- Packet Bytes:** A detailed view of the selected packet, showing the Hypertext Transfer Protocol (HTTP) request structure.
- Packet Hex:** A hex dump of the selected packet's bytes, with the corresponding ASCII text shown to the right.

**Packet List:**

	Source	Destination	Protocol	Length	Info
607	172.17.101.64	220.181.38.156	HTTP	828	GET /su?p=3&wd=2&ie=UTF-8&cb= HTTP/1.1
615	220.181.38.156	172.17.101.64	HTTP	60	HTTP/1.1 200 OK (text/javascript)
622	172.17.101.64	220.181.38.156	HTTP	829	GET /su?p=3&wd=23&ie=UTF-8&cb= HTTP/1.1
633	220.181.38.156	172.17.101.64	HTTP	60	HTTP/1.1 200 OK (text/javascript)

**Packet Bytes (Frame 607):**

- ▶ Frame 607: 828 bytes on wire (6624 bits), 828 bytes captured (6624 bits) on interface 0
- ▶ Ethernet II, Src: Apple\_37:39:69 (38:f9:d3:37:39:69), Dst: HuaweiTe\_a6:d4:b6 (c8:8d:83:a6:d4:b6)
- ▶ Internet Protocol Version 4, Src: 172.17.101.64, Dst: 220.181.38.156
- ▶ Transmission Control Protocol, Src Port: 51576, Dst Port: 80, Seq: 1, Ack: 1, Len: 774
- ▼ **Hypertext Transfer Protocol**
  - ▶ GET /su?p=3&wd=2&ie=UTF-8&cb= HTTP/1.1\r\n
  - Host: suggestion.baidu.com\r\n
  - Connection: keep-alive\r\n
  - Accept: application/json, text/plain, \*/\*\r\n
  - User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_14\_1) AppleWebKit/537.36 (KHTML, like Gecko) ...
  - Accept-Encoding: gzip, deflate\r\n
  - Accept-Language: zh-CN,zh;q=0.9\r\n
  - ▶ [truncated]Cookie: BAIDUID=CA195640FBE49FD94C428E745ED3A86D:FG=1; BIDUPSID=CA195640FBE49FD94C428E7... \r\n
  - [\[Full request URI: http://suggestion.baidu.com/su?p=3&wd=2&ie=UTF-8&cb=\]](http://suggestion.baidu.com/su?p=3&wd=2&ie=UTF-8&cb=)
  - [HTTP request 1/4]
  - [\[Response in frame: 615\]](#)
  - [\[Next request in frame: 622\]](#)

**Packet Hex:**

0030	10 00 75 f5 00 00 47 45 54 20 2f 73 75 3f 70 3d	..u...GE T /su?p=
0040	33 26 77 64 3d 32 26 69 65 3d 55 54 46 2d 38 26	3&wd=2&ie=UTF-8&
0050	63 62 3d 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f	cb= HTTP /1.1..Ho

**Status Bar:** Hypertext Transfer Protocol (http), 774 bytes | Packets: 5185 · Displayed: 16 (0.3%) | Profile: Default

# 响应报文结构

The image shows a Wireshark network traffic analysis window. The top toolbar includes various icons for navigation and analysis. The main display area is divided into several sections:

- Packet List:** A table with columns for No., Source, Destination, Protocol, Length, and Info. It shows three packets: packet 607 (GET request), packet 615 (200 OK response), and packet 622 (another GET request).
- Packet Details:** A tree view showing the structure of the selected packet (615). It includes:
  - Frame 615: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
  - Ethernet II, Src: HuaweiTe\_a6:d4:b6 (c8:8d:83:a6:d4:b6), Dst: Apple\_37:39:69 (38:f9:d3:37:39:69)
  - Internet Protocol Version 4, Src: 220.181.38.156, Dst: 172.17.101.64
  - Transmission Control Protocol, Src Port: 80, Dst Port: 51576, Seq: 490, Ack: 775, Len: 5
  - [3 Reassembled TCP Segments (494 bytes): #613(185), #614(304), #615(5)]
  - Hypertext Transfer Protocol**
    - HTTP/1.1 200 OK\r\n
    - Content-Type: text/javascript; charset=UTF-8\r\n
    - Date: Thu, 14 Nov 2019 01:33:53 GMT\r\n
    - Server: Apache\r\n
    - Tracecode: 20331571752422859018111409\r\n
    - Transfer-Encoding: chunked\r\n
    - \r\n
    - [HTTP response 1/4]
    - [Time since request: 0.089781000 seconds]
    - [\[Request in frame: 607\]](#)
    - [\[Next request in frame: 622\]](#)
    - [\[Next response in frame: 633\]](#)
    - [Request URI: http://suggestion.baidu.com/su?p=3&wd=2333&ie=UTF-8&cb=]
    - HTTP chunked response
    - File Data: 297 bytes
  - Line-based text data: text/javascript (1 lines)
    - [truncated]({q:"2",p:false,s:["246+\345\244\251\345\244\251\345\245\275\345\275\251944\345\205\215\3
- Packet Bytes:** A hex dump showing the raw bytes of the selected packet, with corresponding ASCII text on the right. The first line shows "HTTP/1.1 200 OK" and the second line shows "Content-Type: t".

At the bottom, the status bar indicates: Hypertext Transfer Protocol (http), 185 bytes. Packets: 74690 · Displayed: 344 (0.5%) · Profile: Default

# 请求方法以及状态码

状态码：

请求方法（安全、幂等、可缓存）：

1. GET
2. POST
3. HEAD
4. PUT
5. DELETE
6. ...

1. 1xx 提示信息
  - 101 Switching Protocol
2. 2xx 成功
  - 200 OK
  - 204 No Content
  - 206 Partial Content
3. 3xx 重定向
  - 301 永久重定向
  - 302 临时重定向
  - 304 Not Modified
4. 4xx 客户端错误
  - 400 Bad Request
  - 403 Forbidden
  - 404 Not Found
  - 408 Request Timeout
5. 5xx 服务端错误
  - 500 Internal Server Error
  - 502 Bad Gateway
  - 503 Service Unavailable

# 重点特性详解

1. 如何传输大文件
2. 连接管理
3. Cookie 机制
4. 缓存控制
5. 代理服务

# HTTP 如何传输大文件

1. 数据压缩
2. 分块传输
3. 范围请求
4. 多段请求

# 数据压缩

Request:

Accept-Encoding: gzip\r\n

Response:

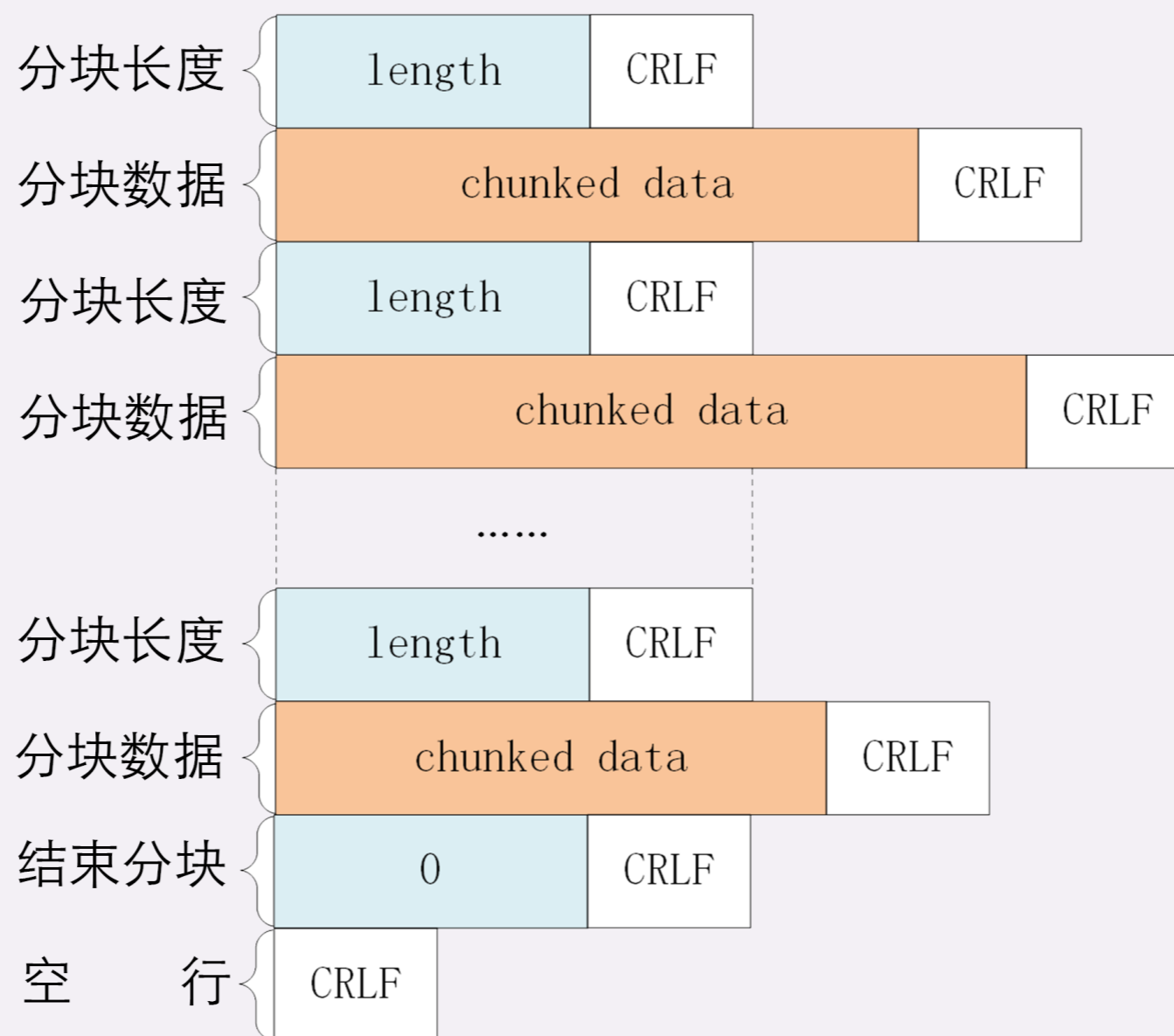
Content-Encoding: gzip\r\n



# 分块传输

Response:

Transfer-Encoding: chunked



# 分块传输

The image shows a Wireshark network traffic analysis window. The top toolbar includes icons for file operations, search, and zooming. The main pane displays a list of network packets. Packet 655 is selected, showing an HTTP response from 220.181.38.156 to 172.17.101.64. The response details are expanded to show an 'HTTP chunked response' with a data chunk of 233 octets. The chunked response structure is as follows:

- [HTTP response 3/4]
- [Time since request: 0.081097000 seconds]
- [Prev request in frame: 622]
- [Prev response in frame: 633]
- [Request in frame: 647]
- [Next request in frame: 661]
- [Next response in frame: 667]
- [Request URI: http://suggestion.baidu.com/su?p=3&wd=2333&ie=UTF-8&cb=]
- ▼ HTTP chunked response
  - ▼ Data chunk (233 octets)
    - Chunk size: 233 octets
    - ▼ Data (233 bytes)
      - Data: 287b713a22323333222c703a66616c73652c733a5b223233...
      - [Length: 233]
      - Chunk boundary: 0d0a
    - ▼ End of chunked encoding
      - Chunk size: 0 octets
  - \r\n
  - File Data: 233 bytes
  - ▼ Line-based text data: text/javascript (1 lines)
    - [truncated]({q:"233",p:false,s:["233\345\260\217\346\270\270\346\210\217\344\270\213\350\275\275","2...

The bottom pane shows the raw bytes of the selected packet, with a hex dump and ASCII representation. The ASCII part shows the start of a JavaScript object: `{q:"233",p:false,s:["233\345\260\217\346\270\270\346\210\217\344\270\213\350\275\275","2...`

At the bottom, the status bar indicates: Frame (60 bytes), Reassembled TCP (429 bytes), De-chunked entity body (233 bytes), Chunk size (http.chunk\_size), 3 bytes, Packets: 270063 · Displayed: 1348 (0.5%), Profile: Default.

# 范围请求

单独请求一段数据:

Request:

Range: bytes=0-31

Response:

Accept-Ranges: bytes

Content-Ranges: bytes 0-31/96

# 多段请求

Request:

Range: bytes=0-31, 20-30

Response:

Accept-Ranges: bytes

--000000000001

Content-Type: text/plain

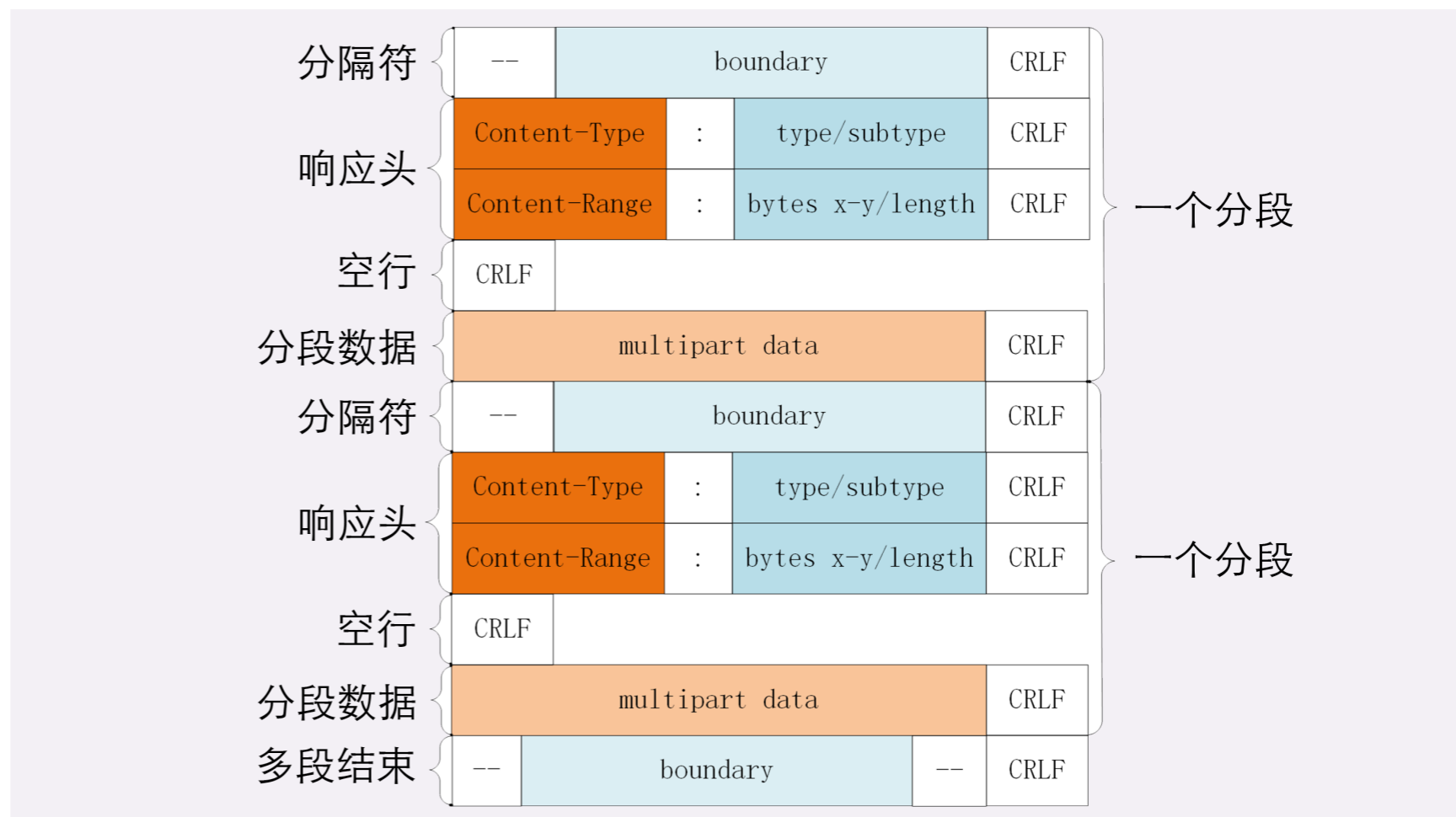
Content-Ranges: bytes 0-31/96

--000000000001

Content-Type: text/plain

Content-Ranges: bytes 0-31/96

ext json d

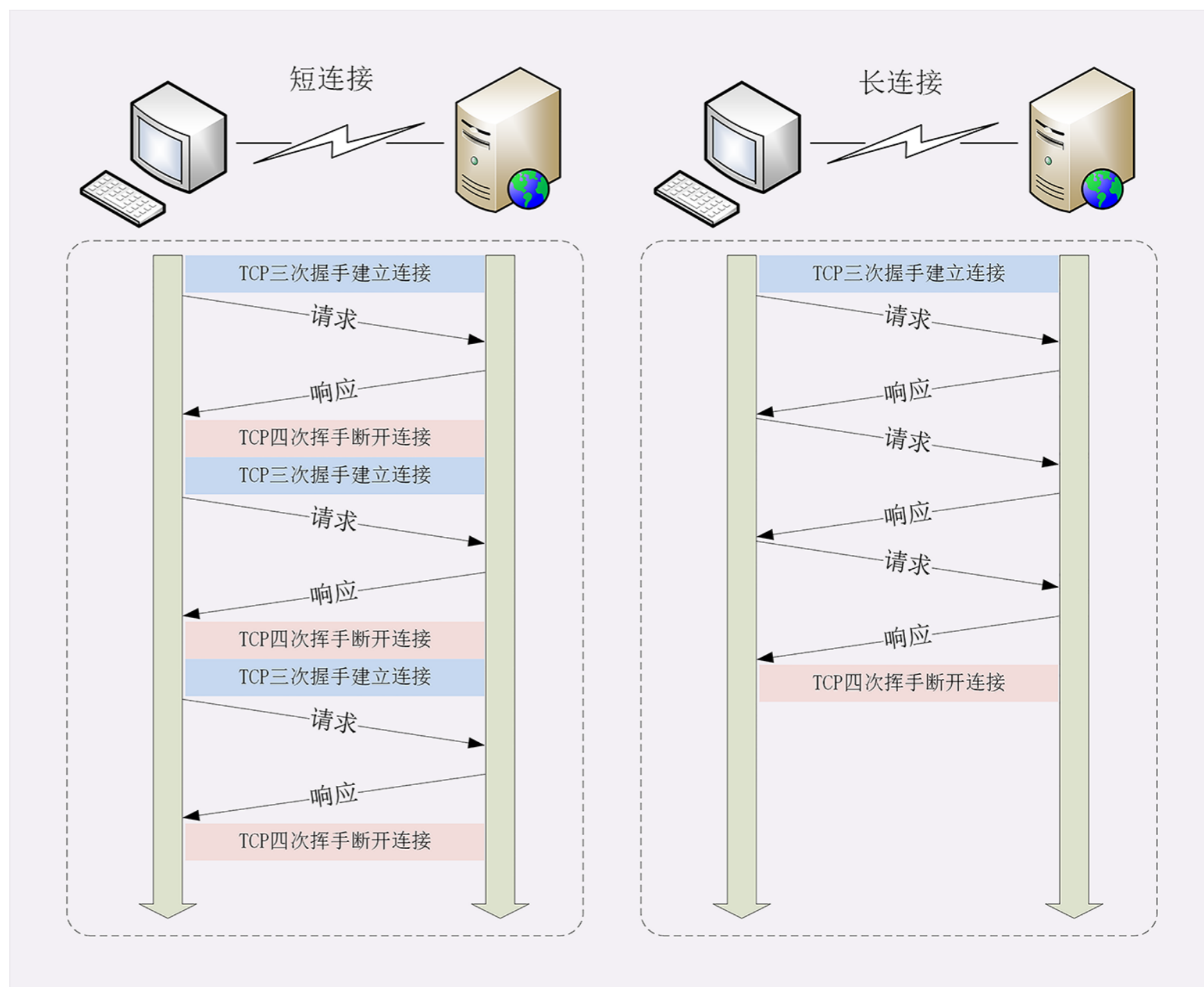


# HTTP 连接管理

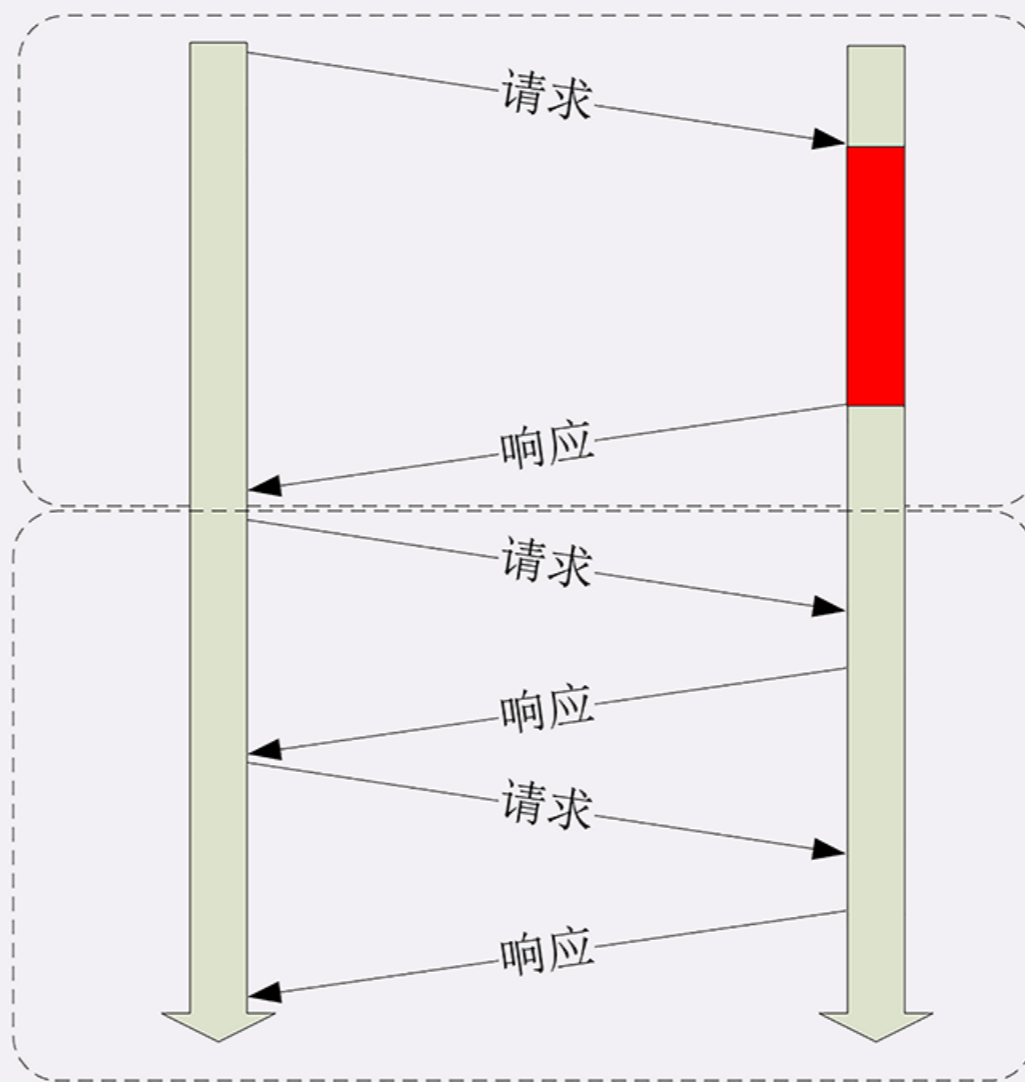
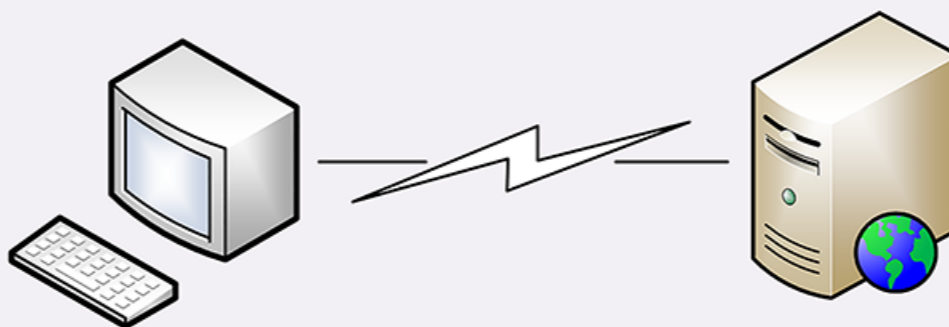
1. 长连接
2. 队头阻塞
3. 连接优化

# 长连接

Request / Response:  
Connection: keep-alive



# 队头阻塞



队头阻塞

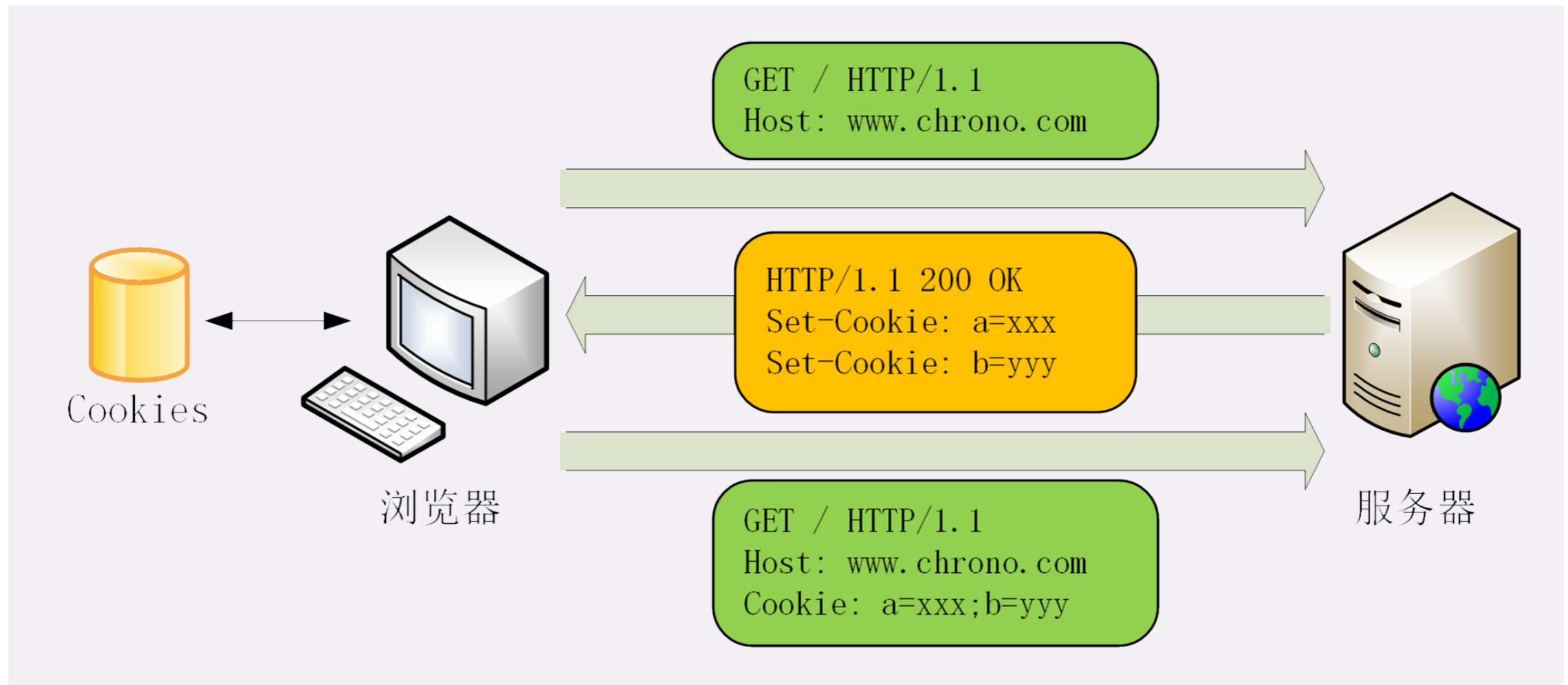
后续请求  
也被阻塞

# 连接优化

1. 并发连接
2. 域名分片



# HTTP Cookie 机制



# Cookie 属性

## 1. 生命周期

1. Expires 绝对时间
2. Max-Age 相对时间 (单位为秒)

## 2. 作用域

1. Path
2. Domain

## 3. 安全性

1. HttpOnly (XSS)
2. SameSite=Strict (XSRF)
3. Secure (HTTPS)

Elements Console Sources **Network** Performance Memory Application Security Audits

Filter  Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Manifest Other

Name **Headers** Preview Response Cookies Timing

juejin.im  
 icon.a87e5ae.svg  
 icons.min.css  
 iconfont.css  
 0.e3d5abc7536595b3152..  
 runtime.17cb348ef7939a8..

```

Date: THU, 14 NOV 2019 07:58:07 GMT
Server: nginx
Set-Cookie: ab={}; path=/; expires=Fri, 13 Nov 2020 07:58:07 GMT; secure; httponly
Set-Cookie: auth=eyJ0b2tlbiI6ImV5SmhZMk5sYzNOZmRHOXJaVzRPT2lKeFVF0U9VWGXxV1V0eGRHUhVM2xZSWl3aWNTVm1jbVZ6YUY5MGlydGxiaUk2SWxGTuZVhCbElqb2liV0ZqSWl3aVpYaHdhWEpsWDJsdUlpb3l0VGt5TURBd2ZRPT0iLCJjbGllbnRjZCI6MTU3MzcxODI4NzQzNiwiOiNjZCJmIjoiaWNTgXNzA0Y2ZkMjA4ZmUu, 21 Nov 2019 07:58:07 GMT; secure; httponly
Set-Cookie: auth.sig=7GQVCCr66uulecngEdNKav8oSCM; path=/; expires=Thu, 21 Nov 2019 07:58:07 GMT; secure; httponly
Set-Cookie: QINGCLOUDELB=cfbefc7e557c04ac9b28122c52f13076ab7e48fb3972913efd95d72fe838c4fb|Xc0JT|Xc0JL; path=/; HttpOnly
Strict-Transport-Security: max-age=31536000
Transfer-Encoding: chunked
Vary: Accept-Encoding
X-Frame-Options: SAMEORIGIN
  
```

Elements Console Sources **Network** Performance Memory Application Security Audits

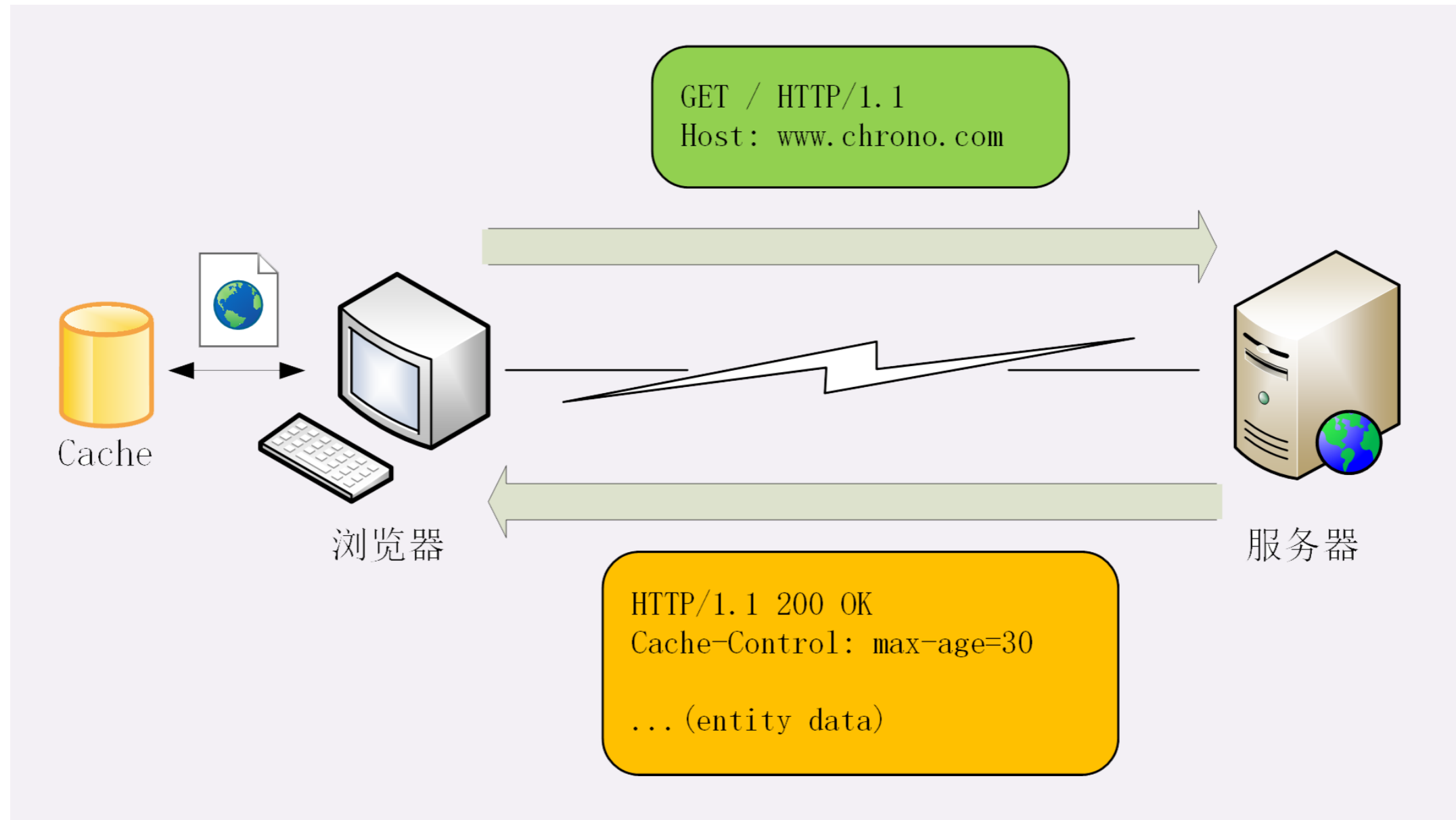
Filter  Hide data URLs **All** XHR JS CSS Img Media Font Doc WS Manifest Other

Name **Cookies** Timing

Name	Value	Domain	Path	Expires / ...	Size	HttpOnly	Secure	SameSite
<b>Request Cookies</b>								
Hm_ipvt_93bbd335a208870aa1f296bcd6842e5e	1573723649	N/A	N/A	N/A	53			
Hm_lvt_93bbd335a208870aa1f296bcd6842e5e	1573547365,1573550477,1573647757,1573647761	N/A	N/A	N/A	85			
QINGCLOUDELB	7c5122b6c6517c59163563fe189d391bab7e48fb3972913efd95d72f...	N/A	N/A	N/A	89			
_ga	GA1.2.934330644.1552783807	N/A	N/A	N/A	32			
_gat	1	N/A	N/A	N/A	8			
_gid	GA1.2.1816484886.1573647757	N/A	N/A	N/A	34			
ab	{}	N/A	N/A	N/A	7			
auth	eyJ0b2tlbil6ImV5SmhZMk5sYzNOZmRHOXJaVzRPT2lKQlpFMDJib...	N/A	N/A	N/A	303			
auth.sig	_jZh9wu45uLWDr19nu0tPKDKIhA	N/A	N/A	N/A	38			
gr_cs1_81ff0cdf-1923-41bf-8db2-a1b657b570ba	objectId%3A581704cfd203090055c9a67d	N/A	N/A	N/A	81			
gr_session_id_89669d96c88aefbc	81ff0cdf-1923-41bf-8db2-a1b657b570ba	N/A	N/A	N/A	69			
gr_session_id_89669d96c88aefbc_81ff0cdf-19...	true	N/A	N/A	N/A	74			
gr_user_id	f433e280-b5a9-4c56-846c-7bf15cb33c52	N/A	N/A	N/A	49			
<b>Response Cookies</b>								
ab	{}		/	2020-11-1...	71	✓	✓	
auth	eyJ0b2tlbil6ImV5SmhZMk5sYzNOZmRHOXJaVzRPT2lKQlpFMDJib...		/	2019-11-2...	367	✓	✓	
auth.sig	_jZh9wu45uLWDr19nu0tPKDKIhA		/	2019-11-2...	101	✓	✓	

65 requests 71.9 KB transferred

# HTTP 缓存控制



# Cache 属性

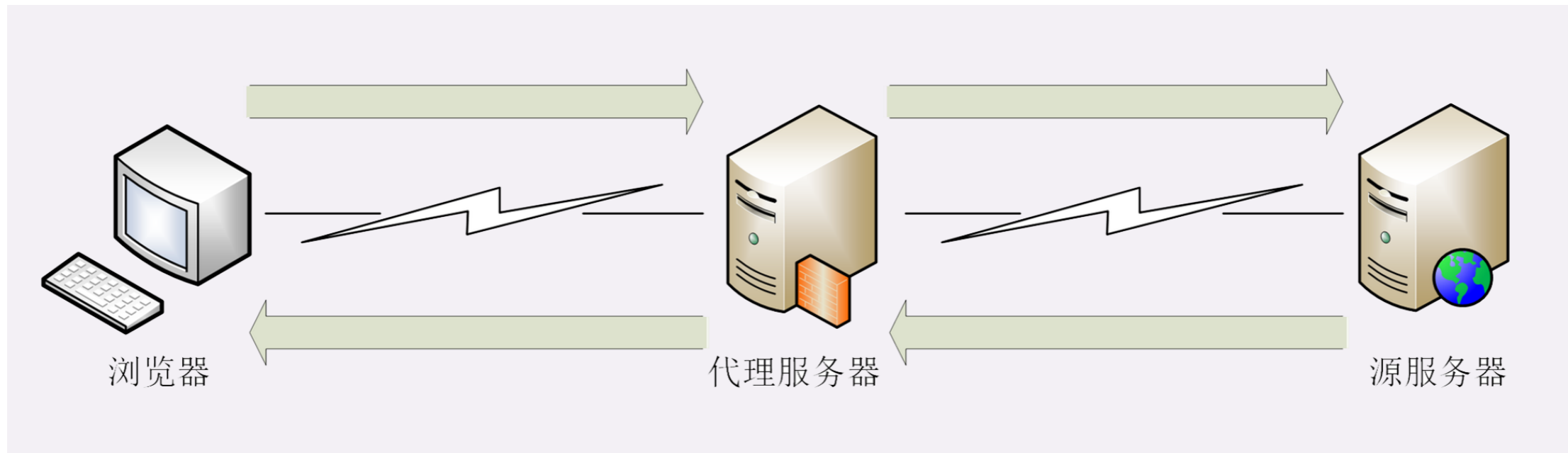
1. max-age
2. no-store
3. no-cache
4. must-revalidate

# Cache-control

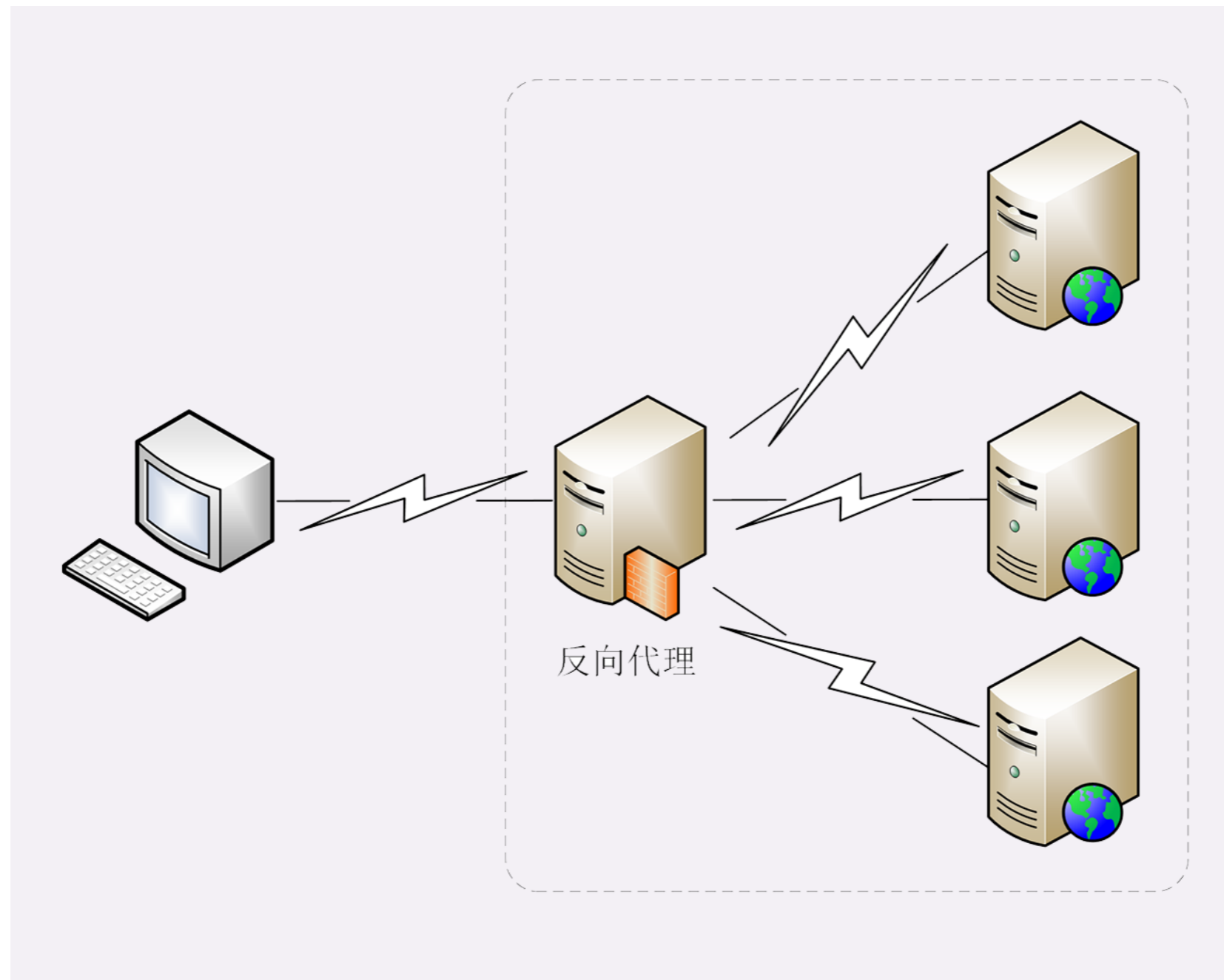
The image shows a screenshot of the Chrome DevTools Network panel. The top bar displays the page URL: DevTools - juejin.im/timeline. The Network tab is active, showing a list of requests on the left and the details of the selected request on the right. The selected request is for a 'timeline' resource. The 'Headers' tab is open, displaying the following response headers:

- Referrer Policy:** no-referrer-when-downgrade
- Response Headers:**
  - Access-Control-Max-Age:** 86400
  - Cache-control:** private
  - Connection:** close
  - Content-Encoding:** gzip
  - Content-Security-Policy:** default-src \* blob: data:;script-src 'self' 'unsafe-eval' \*.xitu.io \*.juejin.im \*.baidu.com \*.google-analytics.com \*.meiqia.com \*.meiqiauserconte nt.com dn-growing.qbox.me \*.growingio.com \*.guard.qcloud.com \*.gtimg.com \*.play.itdks.com g.alicdn.com;style-src 'self' 'unsafe-inline' \*.xitu.io \*.juejin.im \*.growi ngio.com \*.guard.qcloud.com \*.gtimg.com \*.play.itdks.com g.alicdn.com;
  - Content-Type:** text/html; charset=utf-8
  - Date:** Fri, 22 Nov 2019 07:49:28 GMT
  - Server:** nginx
  - Set-Cookie:** ab={}; path=/; expires=Sat, 21 Nov 2020 07:49:28 GMT; secure; httponly
  - Set-Cookie:** auth=eyJ0b2tlbiI6ImV5SmhZMk5sYzN0ZmRH0XJaVzRpT2lJm1xWnZVbTVsUmx0aFYySTVWSE5NSWl3aWNTVm1jbVZ6YUUY5MGYydGxiaUk2SWxWcU5WRlBaMjkzZGtKNE9HcEzabFlpTENKMGIydGxibDk wZVhCbElqb2liV0ZqSWl3aVpYaHdhWEpsWDJsduU1qb3l0VGt5TURBd2ZRPT0iLCJjbGlbnRjZCI6MTU3NDQwODkwTUyMCwidXNlcklIjoINTGxNzA0Y2ZkMjAzMDkwMDU1YzlhNjdkIn0=; path=/; expires=Fr i, 29 Nov 2019 07:49:28 GMT; secure; httponly
  - Set-Cookie:** auth.sig=RIzKLl\_u0lAWuaY2Pbx0Ffgdp5I; path=/; expires=Fri, 29 Nov 2019 07:49:28 GMT; secure; httponly
  - Set-Cookie:** QINGCLOUDELb=0b1f7c3cf8ed56a5d8245b286e7f9983a4db71eff5b47ab5974a1647066247c5|XdeTC|XdeTC; path=/; HttpOnly
  - Strict-Transport-Security:** max-age=31536000
  - Transfer-Encoding:** chunked
  - Vary:** Accept-Encoding
  - X-Frame-Options:** SAMEORIGIN
- Request Headers:**
  - Accept:** text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3
  - Accept-Encoding:** gzip, deflate, br
  - Accept-Language:** zh-CN,zh;q=0.9
  - Cache-Control:** max-age=0
  - Connection:** keep-alive
  - Cookie:** ab={}; gr\_user\_id=f433e280-b5a9-4c56-846c-7bf15cb33c52; \_ga=GA1.2.934330644.1552783807; Hm\_lvt\_93bbd335a208870aa1f296bcd6842e5e=1573647761,1574045815,157424302 8,1574243122; gr\_session\_id\_89669d96c88aefbc=c565beee-ab3e-41ca-adca-61a40bf9124d; gr\_cs1\_c565beee-ab3e-41ca-adca-61a40bf9124d=objectId%3A581704cfd203090055c9a67d; \_g id=GA1.2.306648674.1574408819; gr\_session\_id\_89669d96c88aefbc\_c565beee-ab3e-41ca-adca-61a40bf9124d=true; auth=eyJ0b2tlbiI6ImV5SmhZMk5sYzN0ZmRH0XJaVzRpT2lJm1xWnZVbTVs

# HTTP 代理服务



# 反向代理



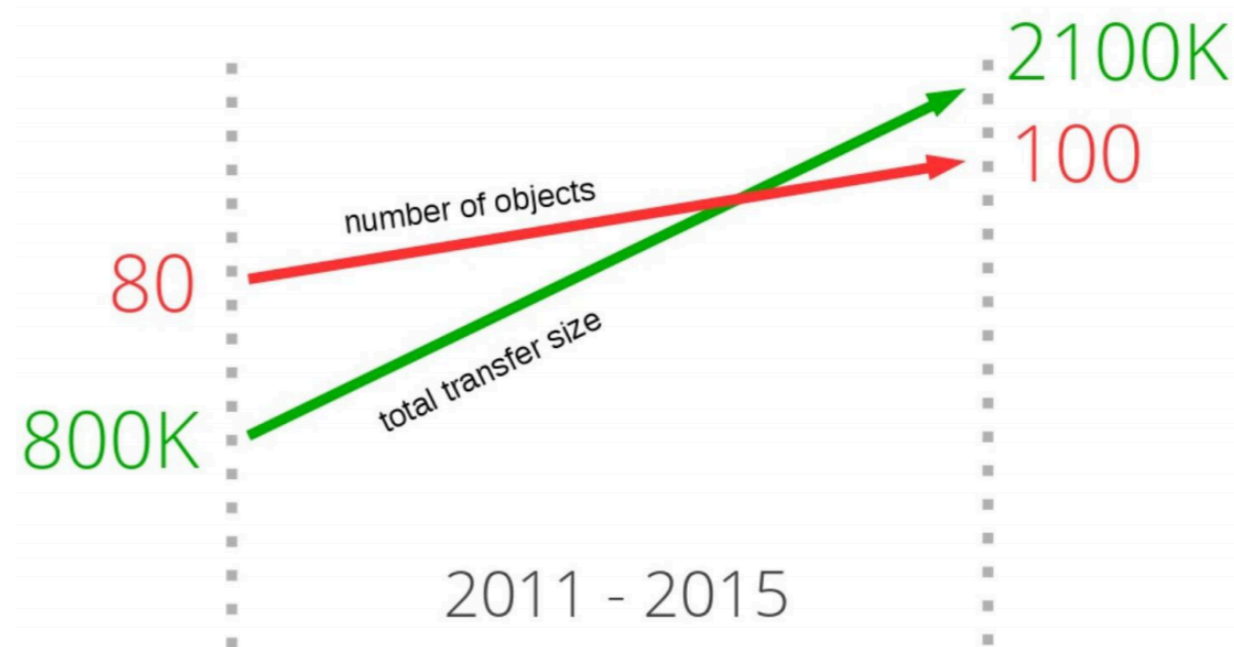


# HTTP/2

# HTTP/1.1 发展中遇到的问题

# HTTP/1.1 发明以来发生了哪些变化

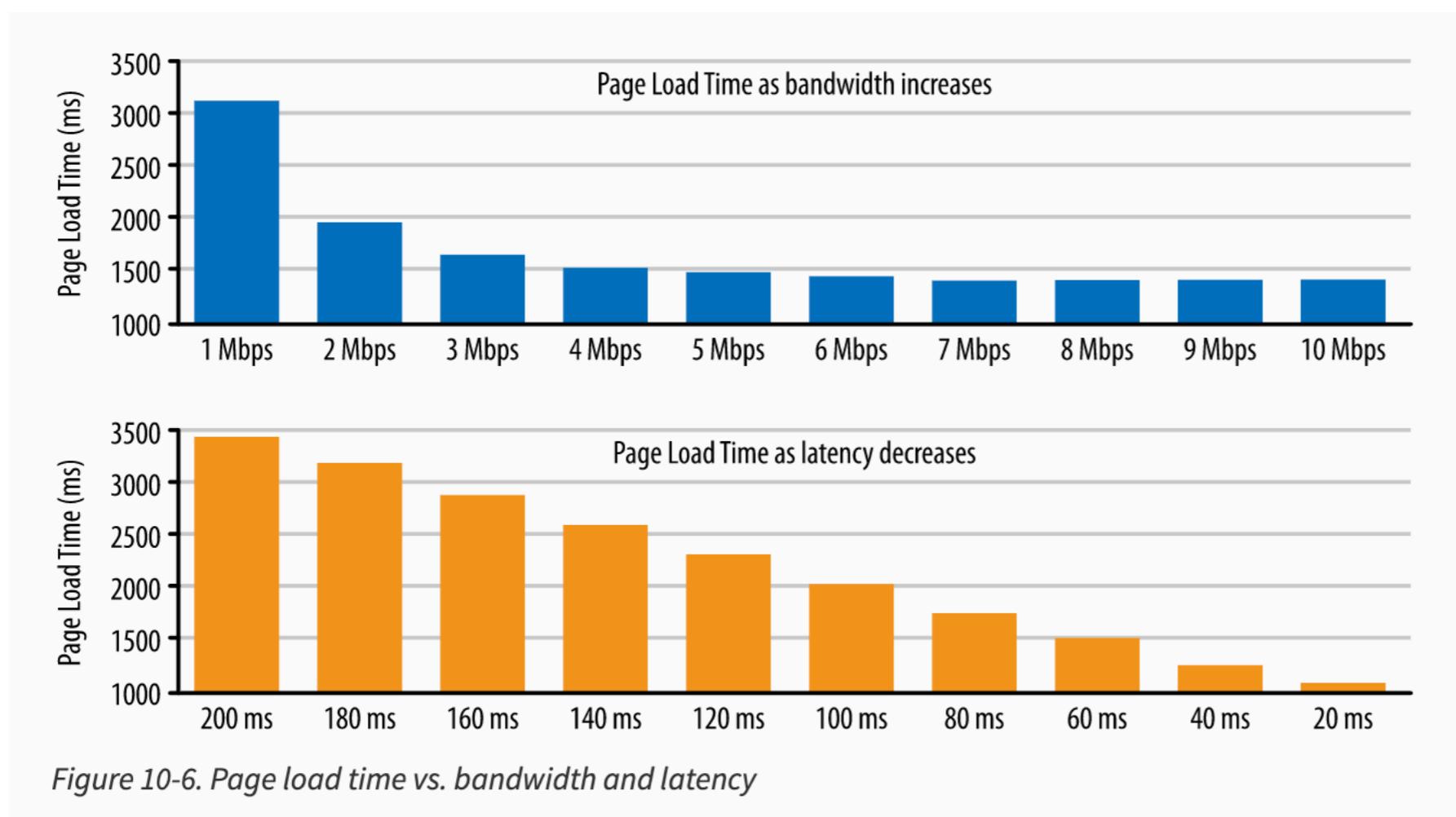
- 消息从几 KB 到几 MB
- 每个页面的资源从小于10 到 大于100
- 以文本为主到富媒体
- 对页面内容实时性要求越来越高



# HTTP/1.1 高延迟问题

高延迟带来的页面加载速度降低

- 随着带宽的增加，延迟并没有明显下降
- 并发连接有限
- 同一连接同时只能完成一个 HTTP 事务才能处理下一个事务



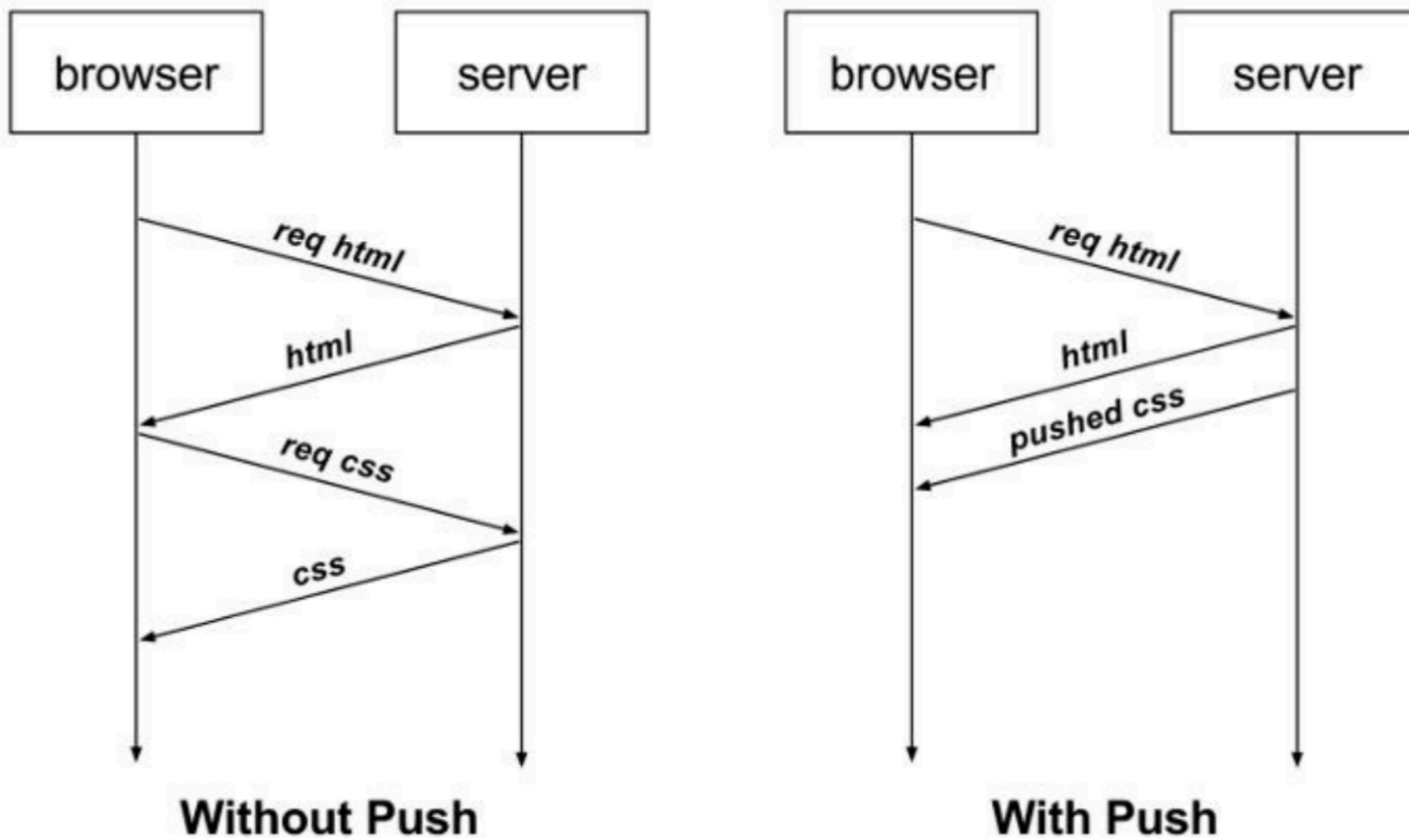
# 无状态带来的巨大 HTTP 头部

重复传输巨大的 HTTP 头部

# HTTP/1.1 为了解决性能问题作出的努力

- 精灵图
- 资源内联
- 资源拼接
- 分片

# HTTP/1.1 不支持服务器推送

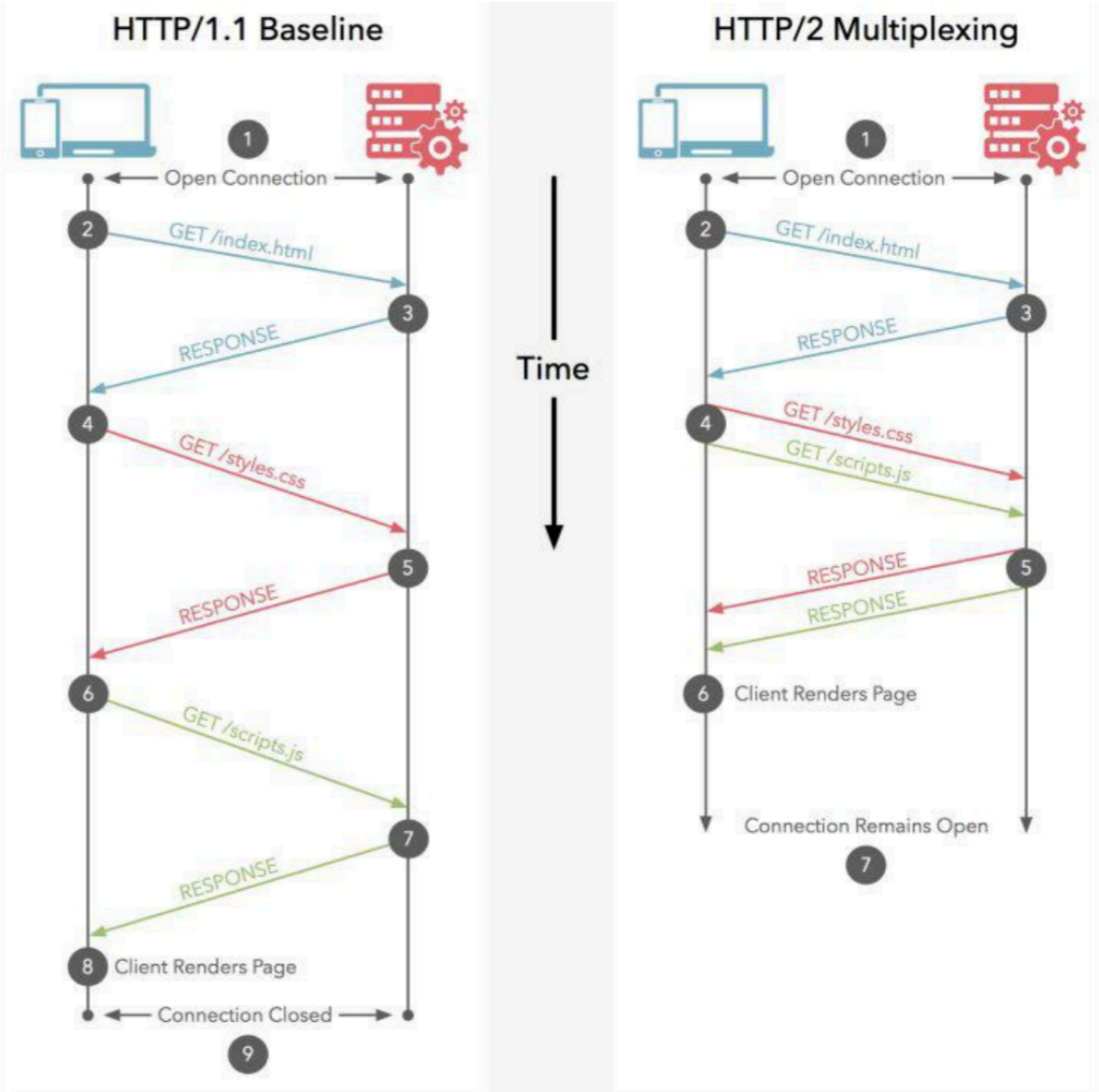
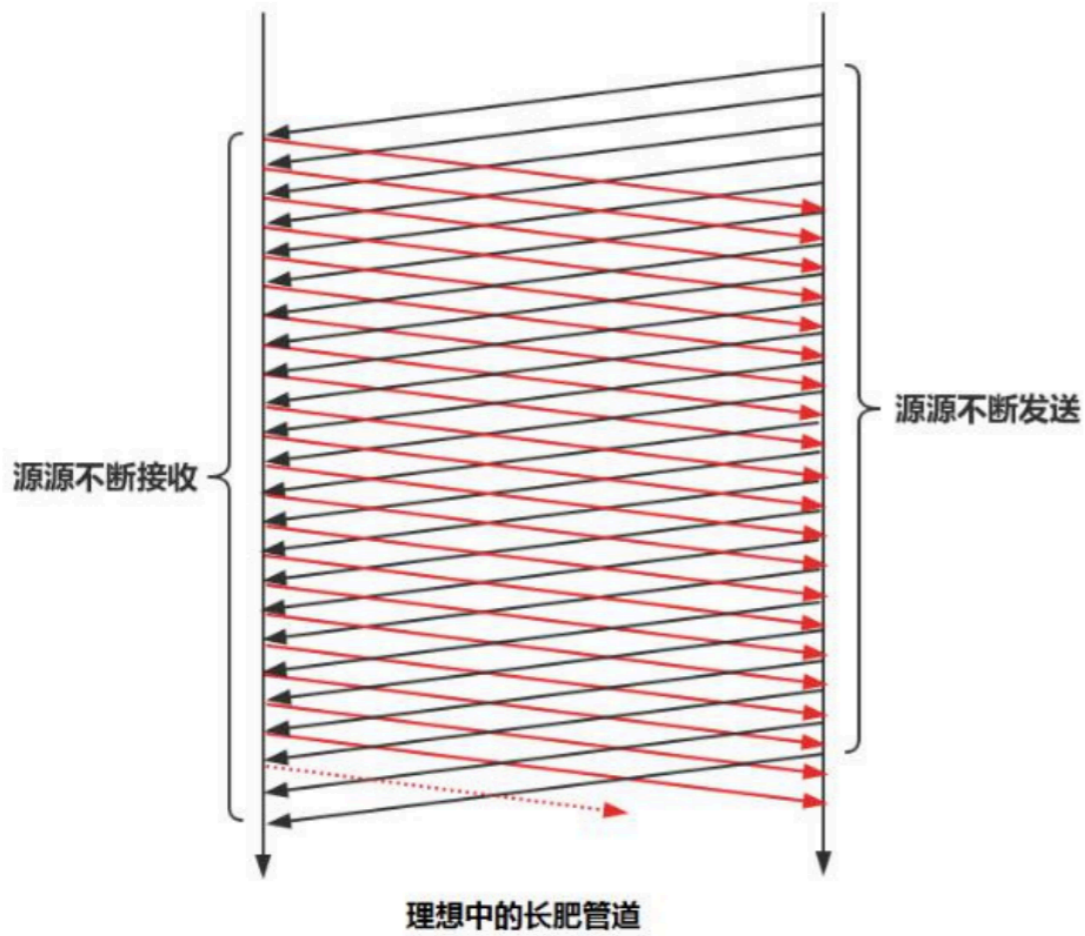


# HTTP/2 带来的特性

1. 传输的数据量大幅减少
  1. 二进制传输，而非文本
  2. HPACK
2. 多路复用及相关功能
  1. 消息优先级
3. 服务器推送



# 多路复用的带来的提升

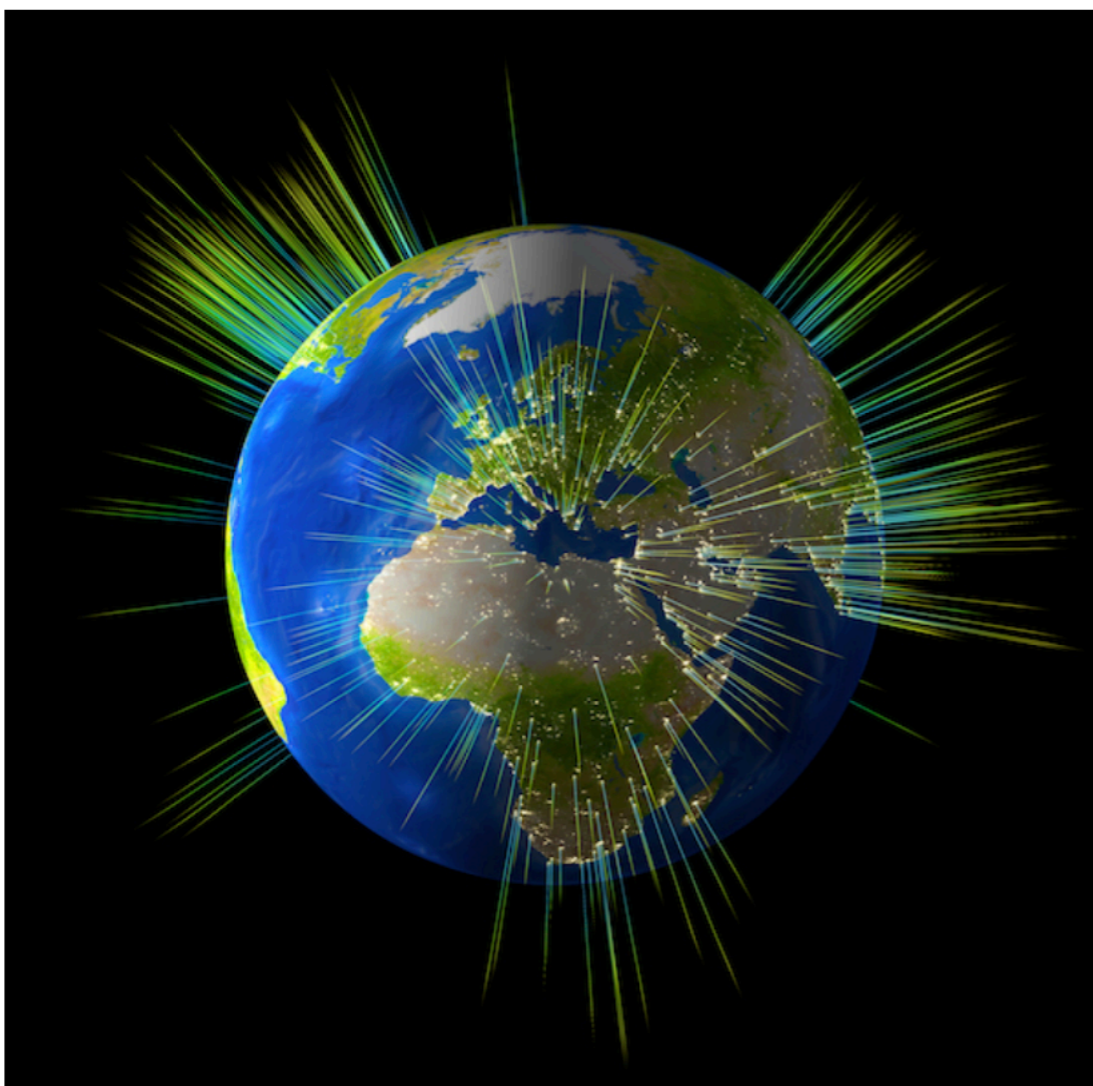


# HTTP/2 的强大之处

<https://http2.akamai.com/demo>

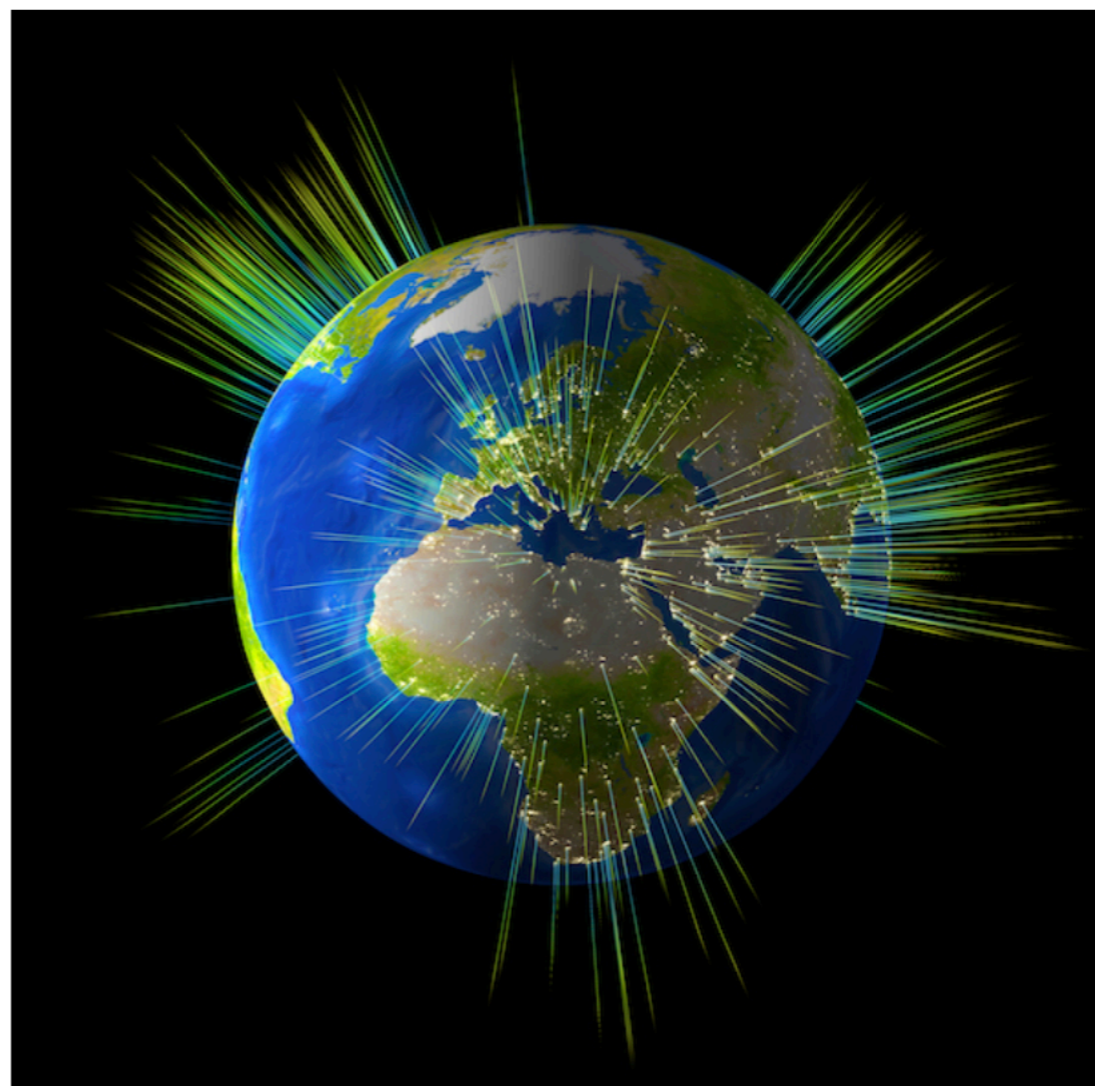
HTTP/1.1

Latency: **6ms**  
Load time: **9.47s**



HTTP/2

Latency: **9ms**  
Load time: **1.87s**

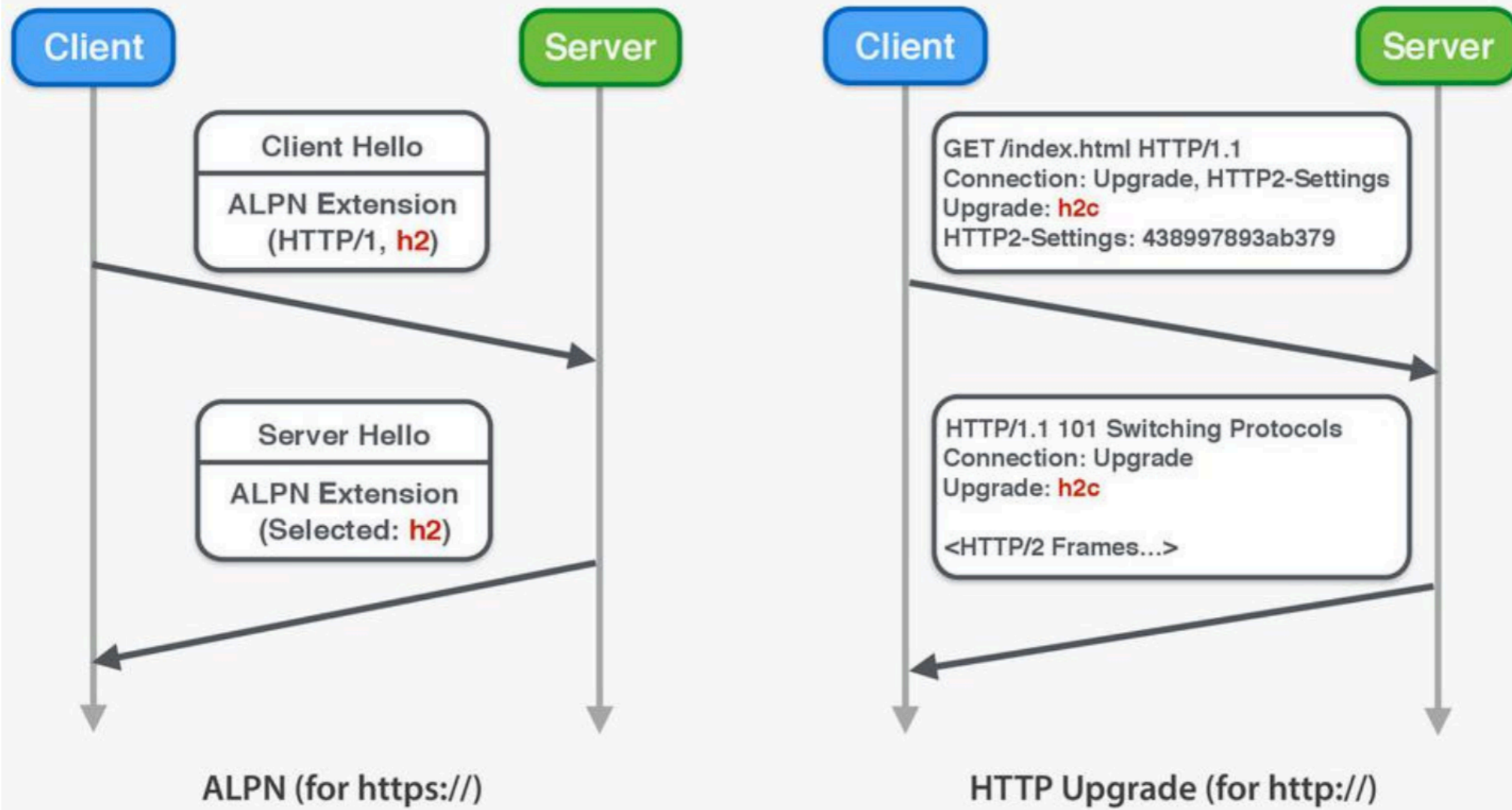


# HTTP/2 是不是必须基于 TLS 协议?

- IETF 标准并不要求一定要基于 TLS
- 浏览器必须要求使用 TLS
- h2: 基于 TLS 协议运行的 HTTP/2
- h2c: 直接在 TCP 协议之上运行的 HTTP/2

# h2 and h2c

## Protocol Negotiation



# h2c

```
curl http://nghttp2.org --http2 -v
```

```
omooo — -bash — 80x24
[OmooodeMacBook-Pro:~ omooo$ curl http://nghttp2.org --http2 -v
* Rebuilt URL to: http://nghttp2.org/
*   Trying 127.0.0.1...
* TCP_NODELAY set
* Connected to 127.0.0.1 (127.0.0.1) port 8888 (#0)
> GET http://nghttp2.org/ HTTP/1.1
> Host: nghttp2.org
> User-Agent: curl/7.54.0
> Accept: */*
> Proxy-Connection: Keep-Alive
> Connection: Upgrade, HTTP2-Settings
> Upgrade: h2c
> HTTP2-Settings: AAMAAABkAARAAAAAAAAIAAAAA
>
< HTTP/1.1 101 Switching Protocols
< Connection: Upgrade
< Upgrade: h2c
* Received 101
* Using HTTP2, server supports multi-use
* Connection state changed (HTTP/2 confirmed)
* Copying HTTP/2 data in stream buffer to connection buffer after upgrade: len=2769
* Connection state changed (MAX_CONCURRENT_STREAMS updated)!
< HTTP/2 200
```

# Magic 帧

The image shows a Wireshark network traffic capture window. The top toolbar includes various icons for file operations, search, and zooming. The main display area is divided into three sections: a packet list, a packet details pane, and a packet bytes pane.

No.	Source	Destination	Protocol	Length	Info
155...	58.211.137.91	172.17.101.64	HTTP2	1393	HEADERS[23]: 200 OK
155...	58.211.137.91	172.17.101.64	HTTP2	390	DATA[23]
155...	58.211.137.91	172.17.101.64	HTTP2	86	DATA[23]
155...	58.211.137.91	172.17.101.64	HTTP2	85	DATA[23] (text/html)
155...	172.17.101.64	58.211.137.91	HTTP2	140	Magic, SETTINGS[0], WINDOW_UPDATE[0]
155...	172.17.101.64	58.211.137.91	HTTP2	335	HEADERS[1]: GET /apps?page=1
155...	58.211.137.91	172.17.101.64	HTTP2	575	SETTINGS[0], WINDOW_UPDATE[0], SETTINGS[0]
155...	172.17.101.64	58.211.137.91	HTTP2	85	SETTINGS[0]
155...	172.17.101.64	58.211.137.91	HTTP2	89	RST_STREAM[1]
155...	58.211.137.91	172.17.101.64	HTTP2	685	HEADERS[1]: 401 Unauthorized, DATA[1]
155...	58.211.137.91	172.17.101.64	HTTP2	85	DATA[1] (application/json)
155...	172.17.101.64	58.211.137.91	HTTP2	653	HEADERS[25]: GET /favicon.ico
155...	58.211.137.91	172.17.101.64	HTTP2	882	HEADERS[25]: 401 Unauthorized, DATA[25]
155...	58.211.137.91	172.17.101.64	HTTP2	85	DATA[25]

**Packet Details:**

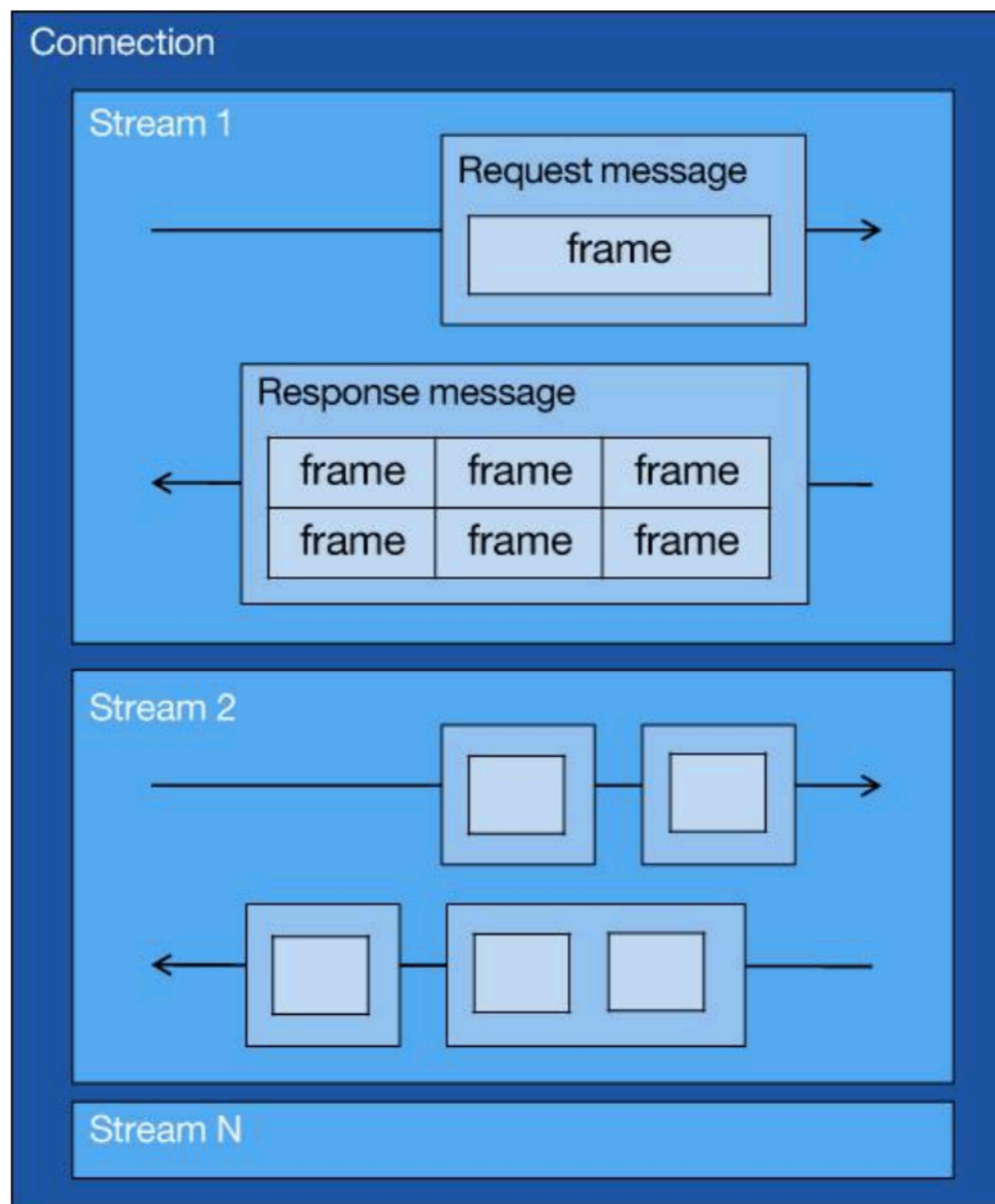
- Transmission Control Protocol, Src Port: 49299, Dst Port: 443, Seq: 582, Ack: 1984, Len: 86
- Transport Layer Security
- HyperText Transfer Protocol 2
  - Stream: Magic
    - Magic: PRI \* HTTP/2.0\r\n\r\nSM\r\n\r\n
  - Stream: SETTINGS, Stream ID: 0, Length 18
    - Length: 18
    - Type: SETTINGS (4)
    - Flags: 0x00
      - 0... .. = Reserved: 0x0
      - .000 0000 0000 0000 0000 0000 0000 0000 = Stream Identifier: 0
    - Settings - Header table size : 65536
    - Settings - Max concurrent streams : 1000
    - Settings - Initial Windows size : 6291456
  - Stream: WINDOW\_UPDATE, Stream ID: 0, Length 4

**Packet Bytes:** Frame (140 bytes) | Decrypted TLS (64 bytes)

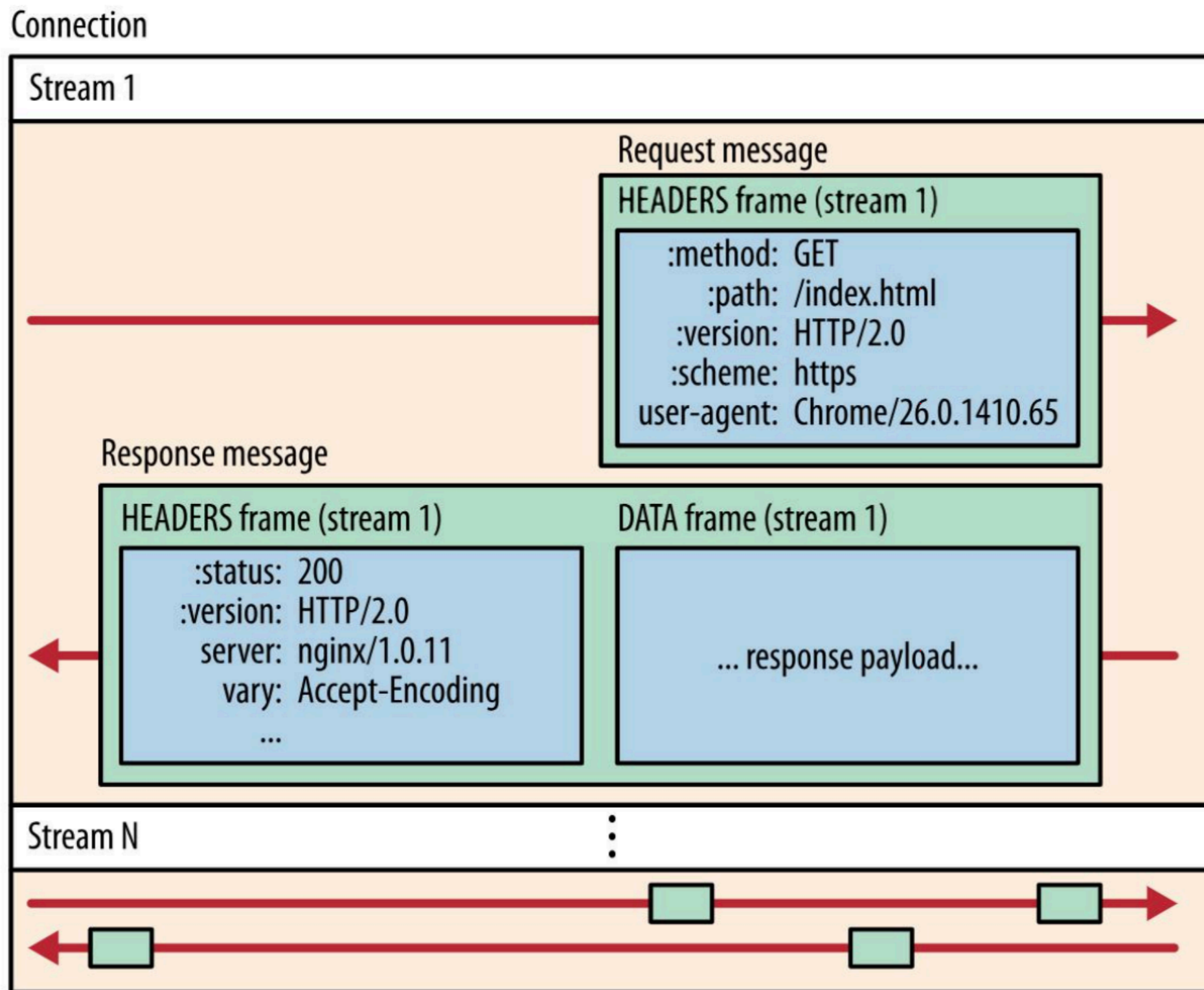
**Status Bar:** Magic (http2.magic) 24 bytes | Packets: 157520 · Displayed: 2424 (1.5%) | Profile: Default

# HTTP/2 核心概念

1. Stream 流
2. Message 消息
3. Frame 帧

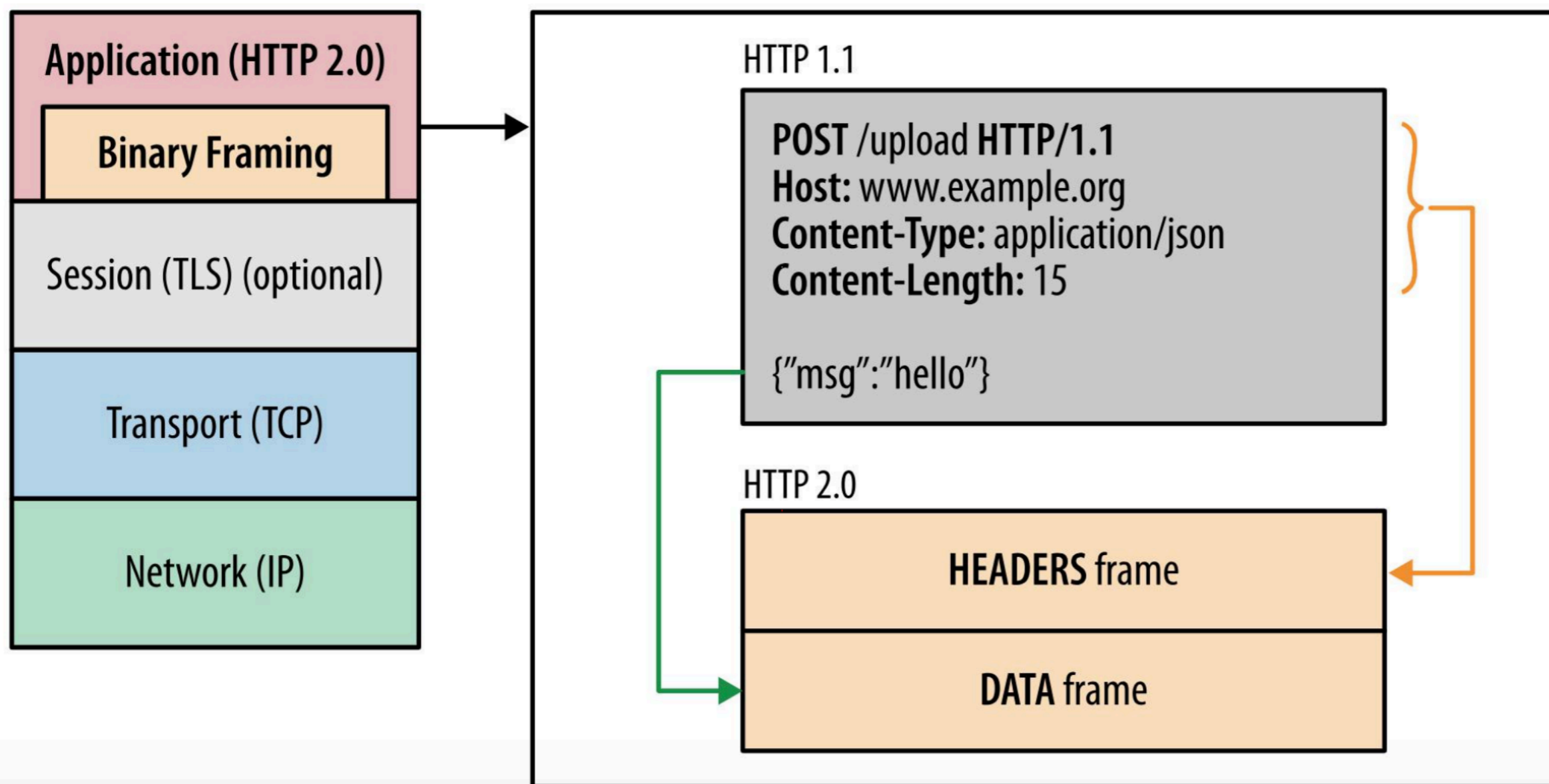


# Stream、Message、Frame 之间的关系

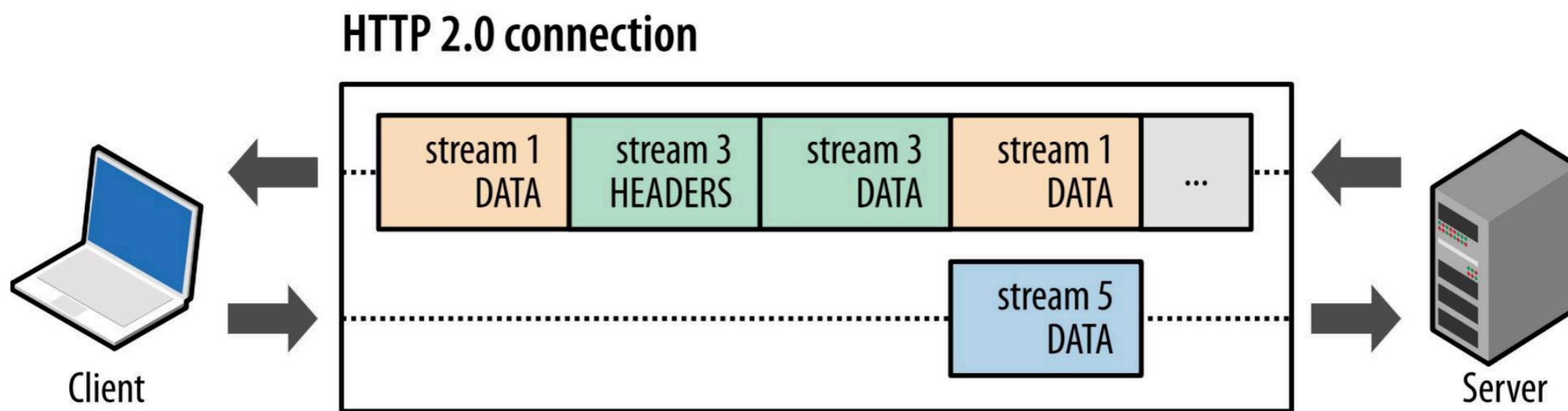




# 消息的组成：HEADER 帧和 DATA 帧



# 传输时无序，接收时组装

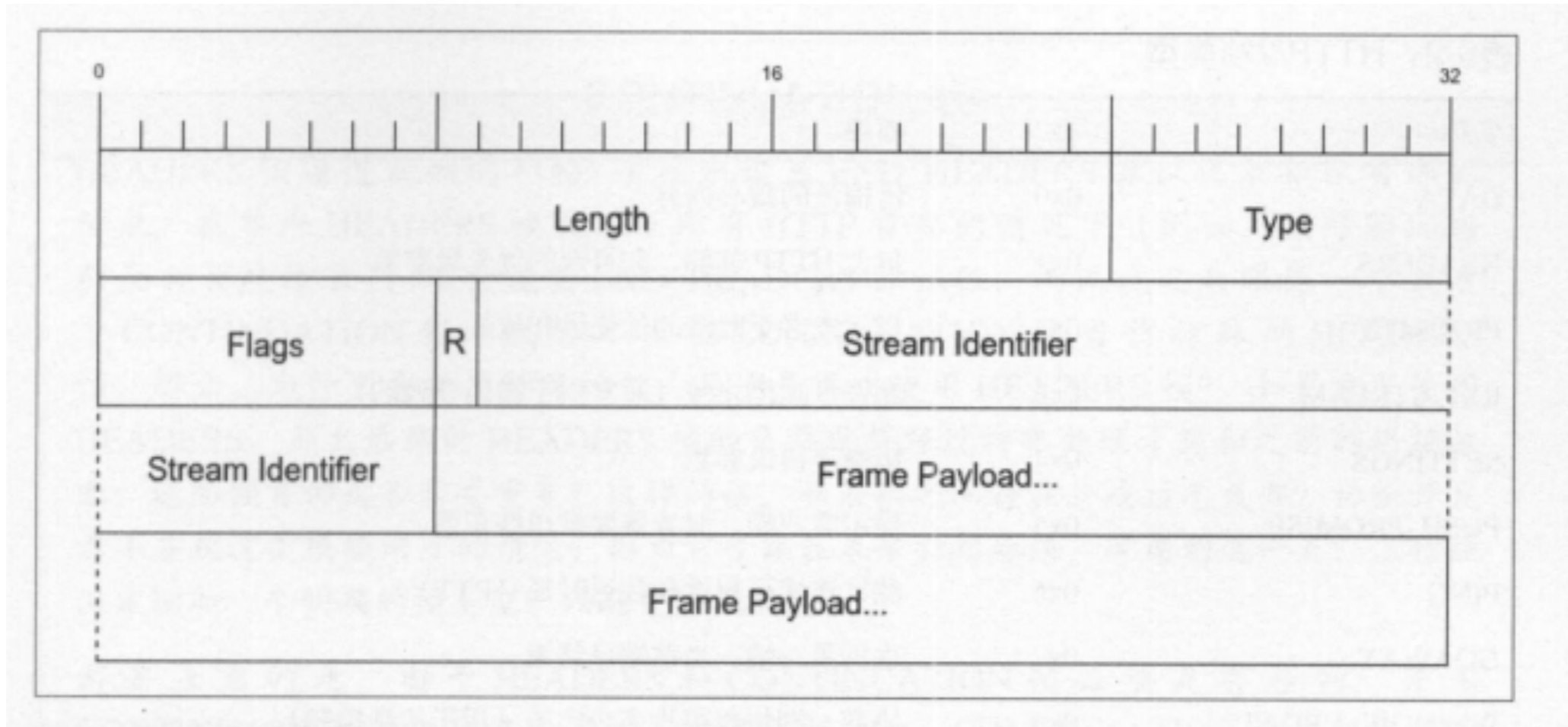


# Stream 流

流是二进制帧的双向传输序列。

1. 流是可以并发的，一个 HTTP/2 连接上可以同时发出多个传输流数据，也就是并发多请求，实现多路复用
2. 客户端和服务端都可以创建流，互不干扰
3. 流之间没有固定关系，但流内部的帧有严格顺序
4. 流可以设置优先级，让服务器优先处理
5. 流 ID 不能复用，只能顺序递增，客户端发的流 ID 是奇数，服务器发送的流 ID 是偶数
6. 第 0 号流比较特殊，不能关闭，也不能发送数据帧，只能发送控制帧，用于流量控制

# Frame 帧



# 帧类型

名称	ID	描述
DATA	0x0	传输流的核心内容
HEADERS	0x1	包含 HTTP 首部和可选的优先级参数
PRIORITY	0x2	指示或者更改流的优先级和依赖
RST_STREAM	0x3	允许一端停止流
SETTINGS	0x4	协商连接时参数
PUSH_PROMISE	0x5	提示客户端，服务器要推送些东西
PING	0x6	测试连接可用性和往返时延
GOAWAY	0x7	告诉另一端，当前端已结束
WINDOWS_UPDATE	0x8	协商一端将要接收多少字节，用于流量控制
CONTINUATION	0x9	用以扩展 HEADER 数据

# 帧类型

Wi-Fi: en0

http2

No.	Source	Destination	Protocol	Length	Info
1194	172.17.101.201	54.222.230.156	HTTP2	159	Magic, SETTINGS[0], WINDOW_UPDATE[0]
1195	172.17.101.201	54.222.230.156	HTTP2	385	HEADERS[1]: GET /products/89669d96c88aefb
1208	203.208.41.46	172.17.101.201	HTTP2	395	HEADERS[3]: 302 Found
1209	203.208.41.46	172.17.101.201	HTTP2	513	DATA[3]
1210	203.208.41.46	172.17.101.201	HTTP2	97	DATA[3] (text/html)
1211	203.208.41.46	172.17.101.201	HTTP2	105	PING[0]
1216	172.17.101.201	203.208.41.46	HTTP2	105	PING[0]
1281	54.222.230.156	172.17.101.201	HTTP2	135	SETTINGS[0], WINDOW_UPDATE[0]
1282	54.222.230.156	172.17.101.201	HTTP2	104	SETTINGS[0]
1283	54.222.230.156	172.17.101.201	HTTP2	253	HEADERS[1]: 200 OK, DATA[1]
1284	54.222.230.156	172.17.101.201	HTTP2	104	DATA[1] (application/json)
1292	172.17.101.201	54.222.230.156	HTTP2	104	SETTINGS[0]
1457	172.17.101.201	101.89.125.227	HTTP2	147	Magic, SETTINGS[0], WINDOW_UPDATE[0]
1458	172.17.101.201	101.89.125.227	HTTP2	520	HEADERS[3]: GET /2018/10/23/166a0387b91066b

Length: 0  
Type: DATA (0)  
▶ Flags: 0x01  
0... .. = Reserved: 0x0  
.000 0000 0000 0000 0000 0000 0000 0011 = Stream Identifier: 3  
[Pad Length: 0]  
▶ [1 Body fragment (416 bytes): #1209(416)]  
Data: 3c48544d4c3e3c484541443e3c6d65746120687474702d65...  
▼ Line-based text data: text/html (6 lines)  
<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">\n  
<TITLE>302 Moved</TITLE></HEAD><BODY>\n  
<H1>302 Moved</H1>\n  
The document has moved\n  
[truncated]<A HREF="https://stats.g.doubleclick.net/r/collect?v=1&aip=1&t=dc&\_r=3&tic  
</BODY></HTML>\r\n

Frame (97 bytes) | Decrypted TLS (9 bytes) | Reassembled body (416 bytes)

wireshark\_Wi-Fi\_20191127105826\_8noyCs.pcapng | Packets: 286410 · Displayed: 791 (0.3%) | Profile: Default

# HPACK 头部压缩

- 静态字典
- 动态字典
- Huffman 编码

```
h2load https://www.sina.com.cn -n 1 |tail -6 |head -1
```



```
omooo — -bash — 103x17
[OmooodeMacBook-Pro:~ omooo$ h2load https://www.sina.com.cn -n 1 |tail -6 |head -1
traffic: 534.97KB (547811) total, 510B (510) headers (space savings 24.22%), 533.23KB (546028) data
[OmooodeMacBook-Pro:~ omooo$ h2load https://www.sina.com.cn -n 2 |tail -6 |head -1
traffic: 1.04MB (1095708) total, 1.12KB (1146) headers (space savings 24.90%), 1.04MB (1092056) data
[OmooodeMacBook-Pro:~ omooo$ h2load https://www.sina.com.cn -n 4 |tail -6 |head -1
traffic: 2.09MB (2191101) total, 1.99KB (2035) headers (space savings 24.41%), 2.08MB (2184112) data
[OmooodeMacBook-Pro:~ omooo$ h2load https://www.sina.com.cn -n 8 |tail -6 |head -1
traffic: 4.18MB (4382171) total, 3.99KB (4088) headers (space savings 24.23%), 4.17MB (4368224) data
OmooodeMacBook-Pro:~ omooo$
```

# HPACK 头部压缩示意

*Request headers*

:method	GET
:scheme	https
:host	example.com
:path	/resource
user-agent	Mozilla/5.0 ...
custom-hdr	some-value



*Static table*

1	:authority	
2	:method	GET
...	...	...
51	referer	
...	...	...
62	user-agent	Mozilla/5.0 ...
63	:host	example.com
...	...	...

*Dynamic table*



*Encoded headers*

2	
7	
63	
19	Huffman("/resource")
62	
	Huffman("custom-hdr")
	Huffman("some-value")



# Huffman 编码

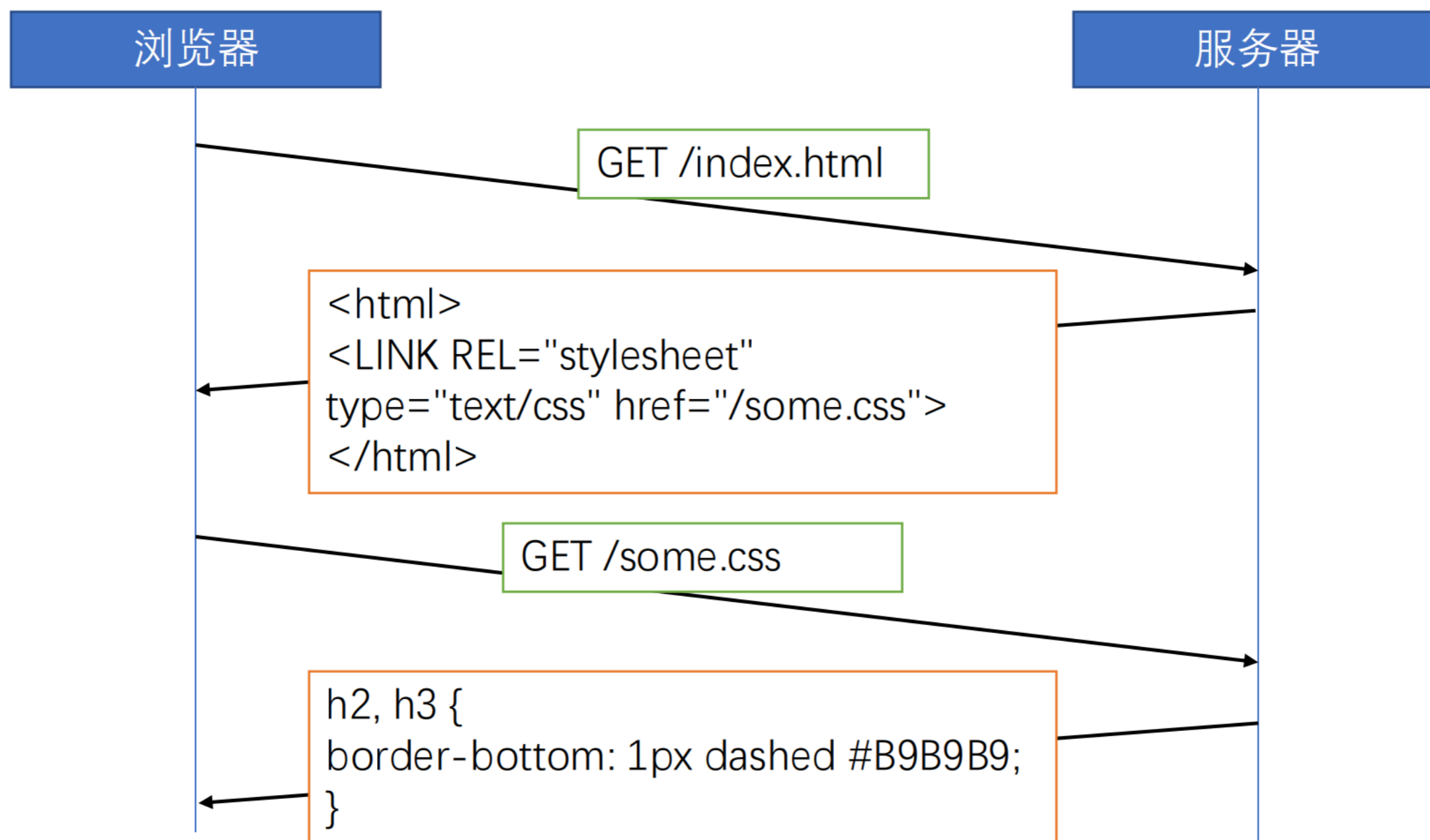
- 原理：出现概率较大的符号采用较短的编码，概率较小的符号采用较长的编码
- [静态 Huffman 编码](#)
- 动态 Huffman 编码

# 服务器推送

- 提前将资源推送至浏览器缓存
- 特性
  - 推送可以基于已发送的请求，例如客户端请求 html，主动推送 js 文件
- 实现方式
  - 推送资源必须对应一个请求
  - 请求由服务端 PUSH\_PROMISE 帧发送
  - 响应在 ID 为偶数的 STREAM 中发送

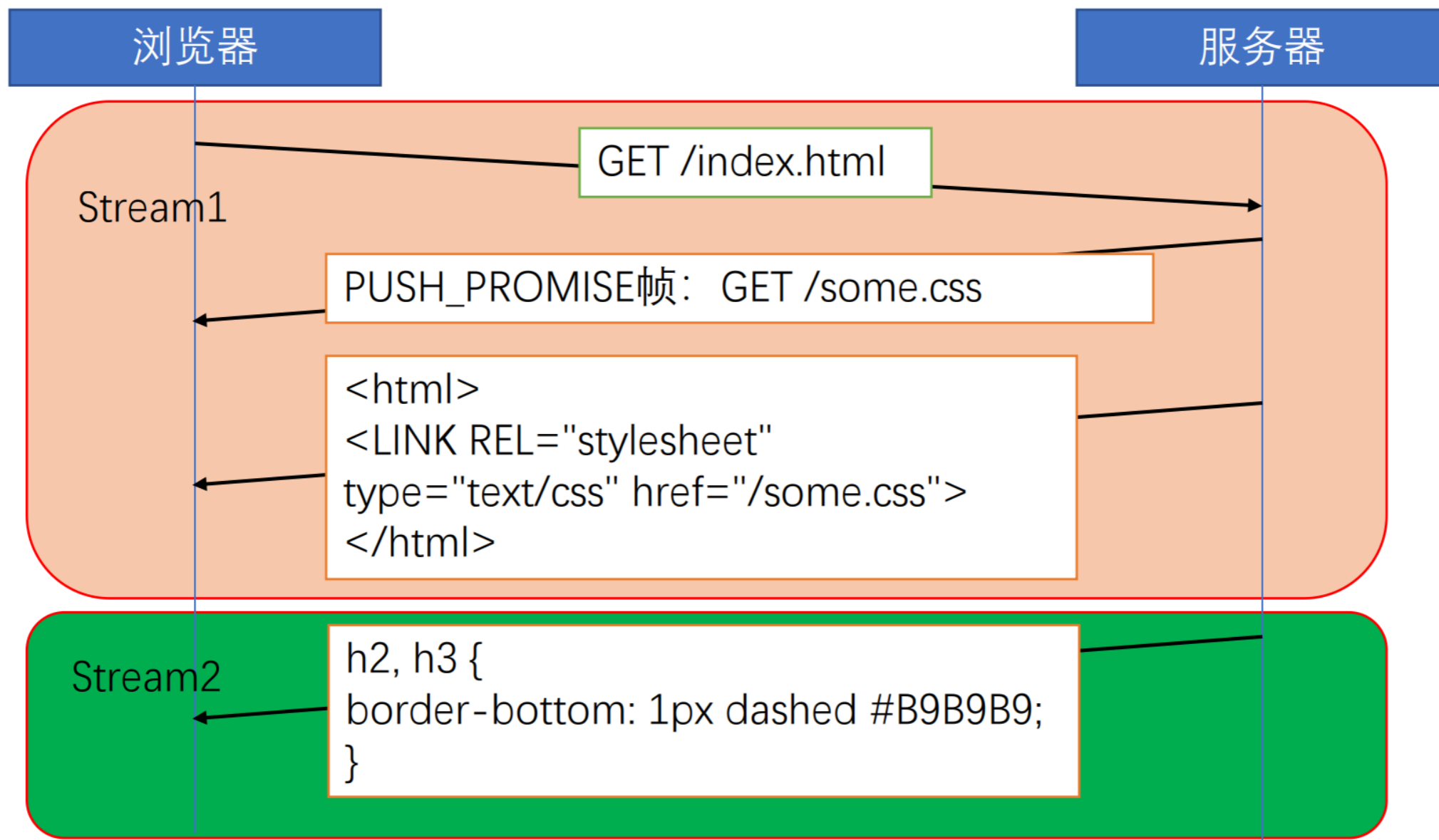
# 当获取 HTML 后，需要 CSS 资源时

- 浏览器触发方式：



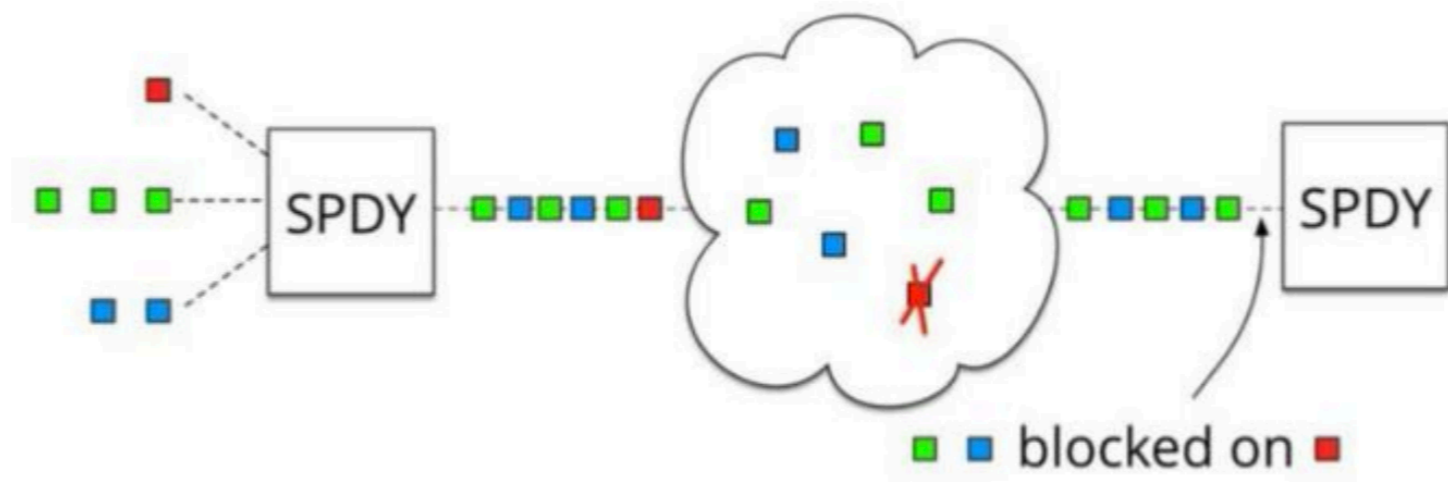
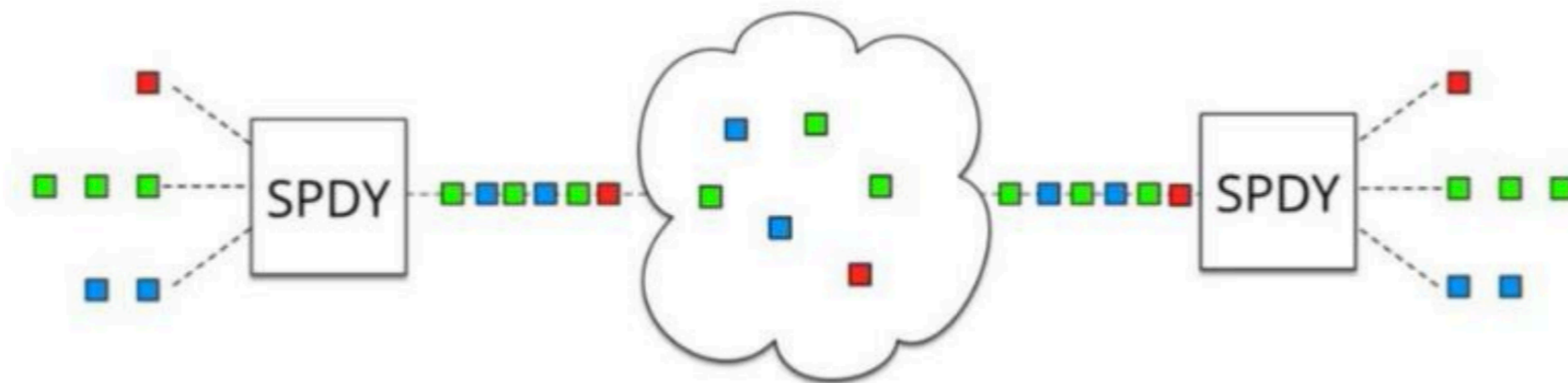
- PUSH\_PROMISE 方式

- 在 Stream1 中通知客户端 CSS 资源即将到来
- 在 Stream2 中发送 CSS 资源 (Stream1 和 2 可以并发)



**HTTP/3**

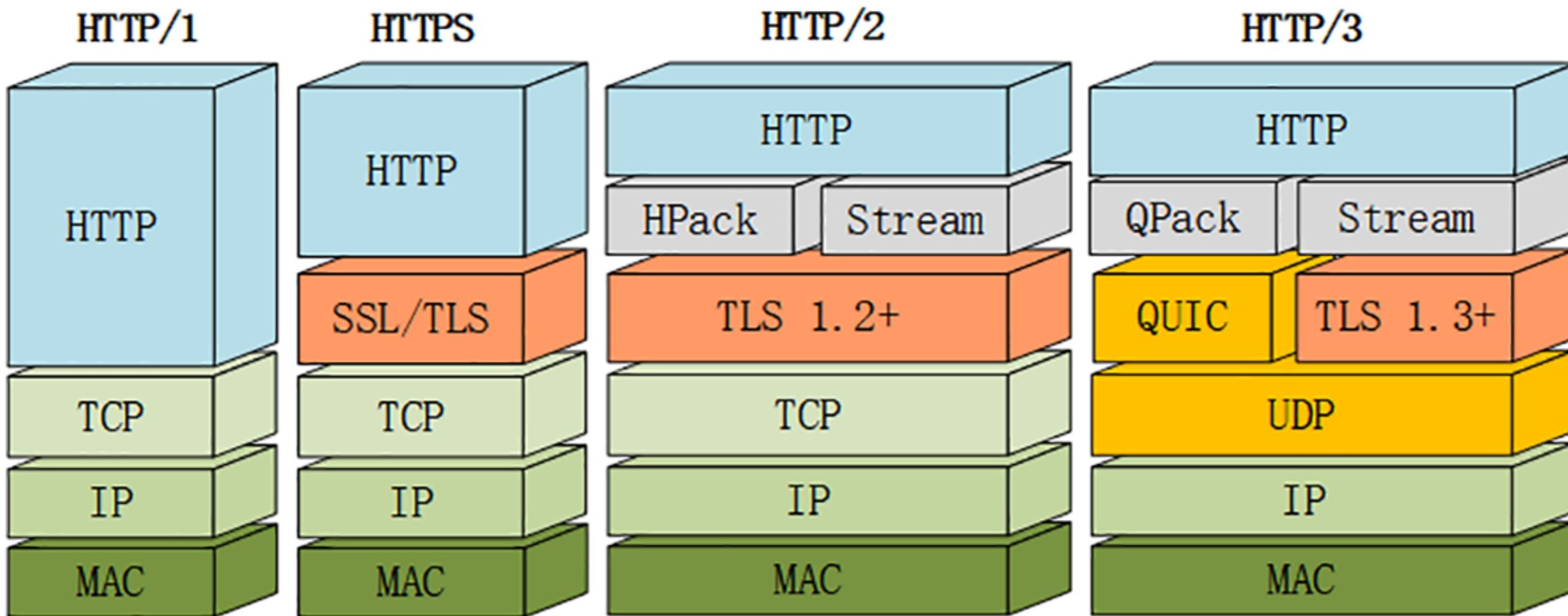
# TCP 的队头阻塞



# TCP 的问题

由操作系统内核实现，更新缓慢

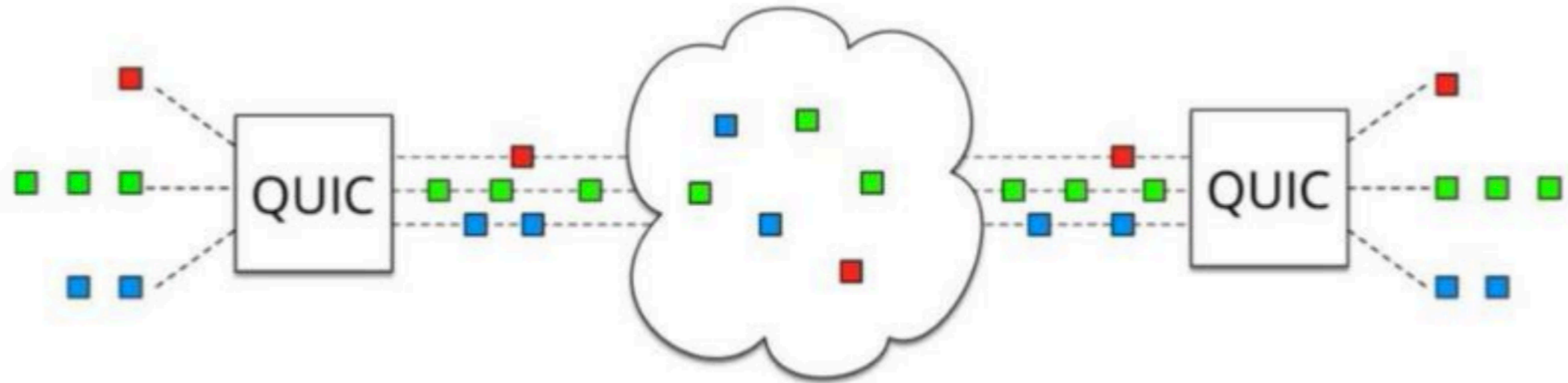
# QUIC (Quick UDP Internet Connection)





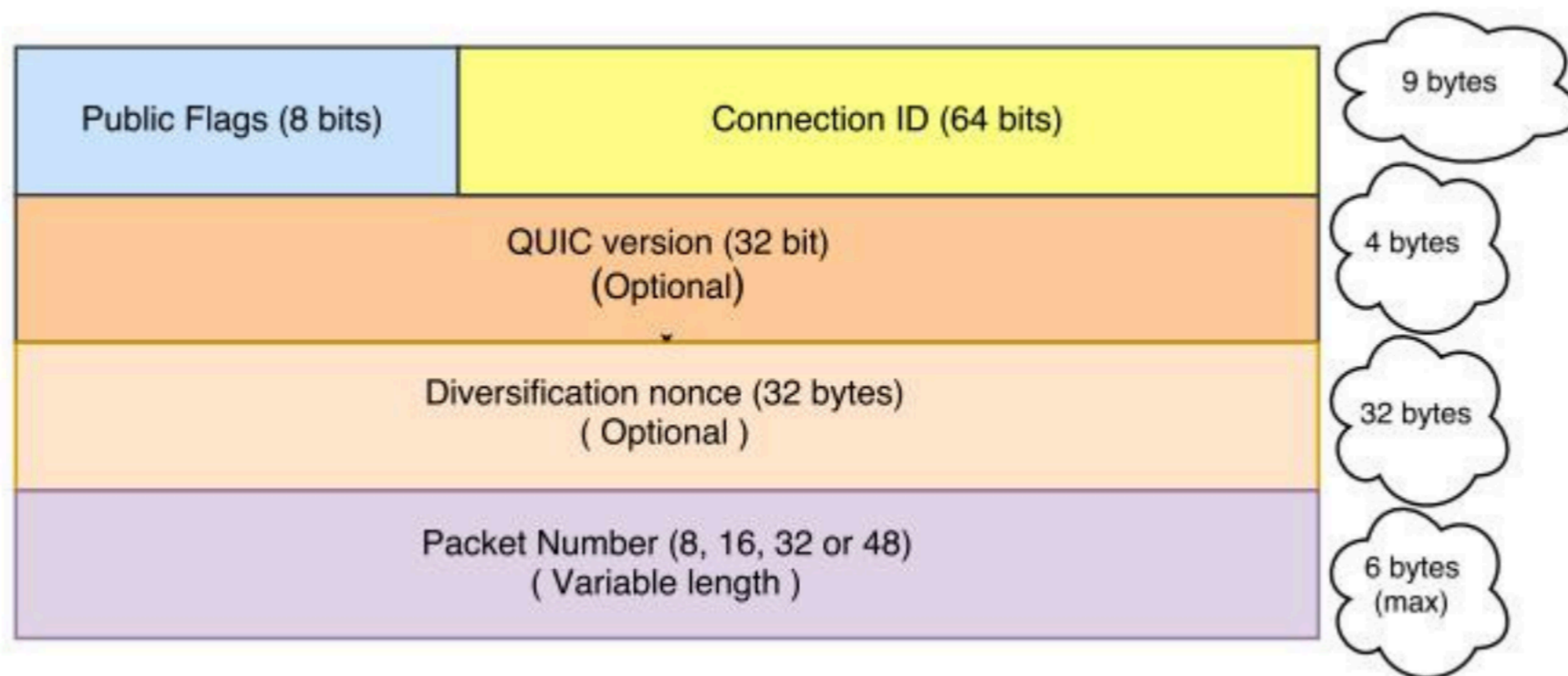
# 解决了队头阻塞

UDP 报文：先天没有队列概念



# HTTP/3 连接迁移

允许客户端更换 IP、端口号后，仍然复用以前的连接



# 参考资料

## 书籍：

1. 《Web 性能权威指南》
2. 《HTTP/2 基础教程》
3. 《HTTP 权威指南》

## 专栏：

1. [透视 HTTP 协议](#)
2. [Web协议详解与抓包实战](#)

## 链接：

1. [HTTP/2 简介](#)
2. [HTTP: HTTP / 2-高性能浏览器网络 \(O'Reilly\)](#)
3. [如何看待 HTTP/3 ? - 知乎](#)
4. [科普：QUIC协议原理分析](#)