

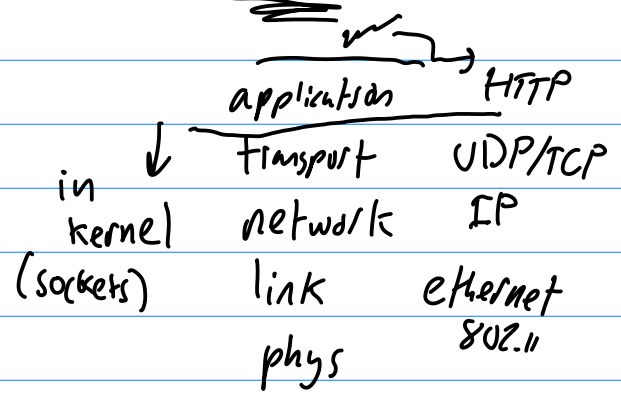
CS 3214

lecture # 25

"HTTP"

SPOT surveys 5%

layers



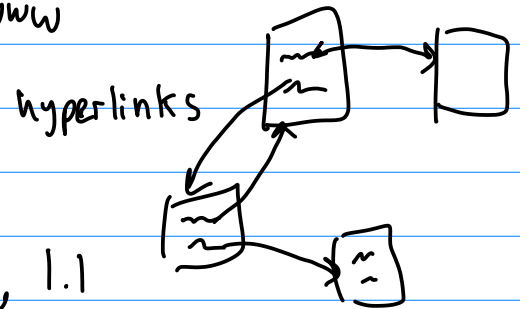
Hypertext transfer protocol

- world wide web

www

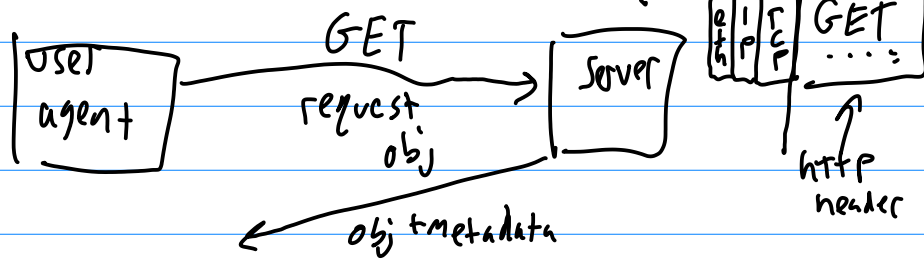
Tim Berners-Lee 1991

HTTP 0.9
 1996, 1997 HTTP 1.0, 1.1
 2015 HTTP 2.0
 HTTP 3.0

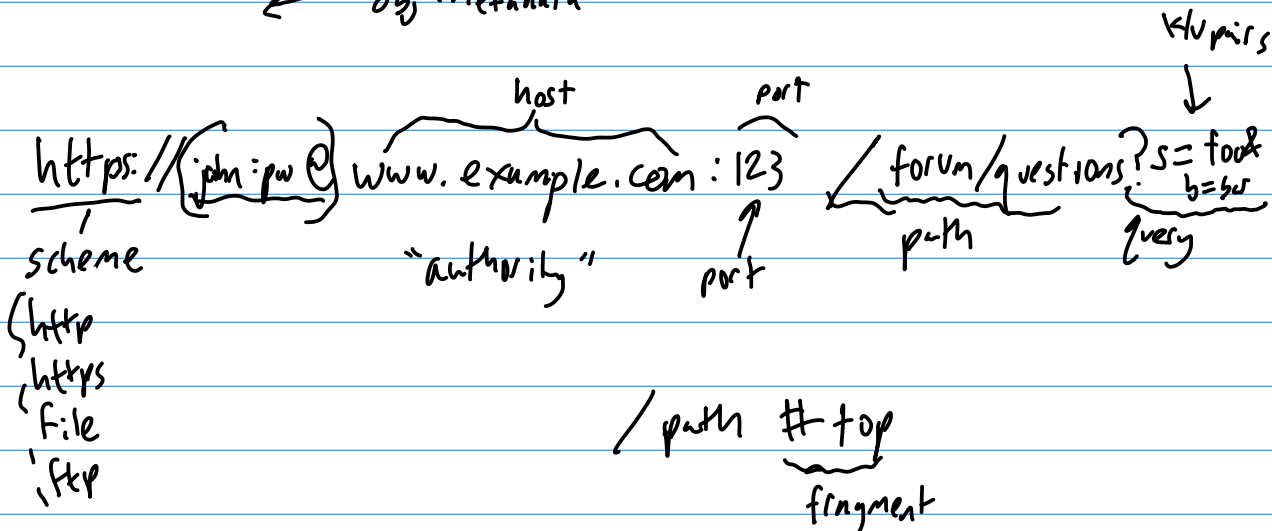


HTTP

Request / Response



URL

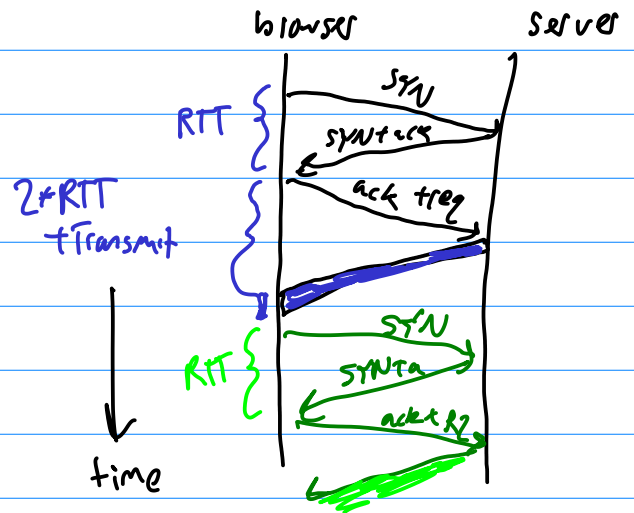


Fetch a webpage

- - multiple request HTML, JS, fonts, CSS, images, ...
- - reliable ← TCP

HTTP 1.0

- new TCP connection for every req.
- inefficient (pay for handshakes)
 - (small resource)
 - WAN



Solution HTTP 1.1

- persistent connections
- http servers
 - ↳ high concurrency



- forcing sequential behavior
- clients use multiple connections to pipeline requests

- server: that's not fair
- network: congestion control

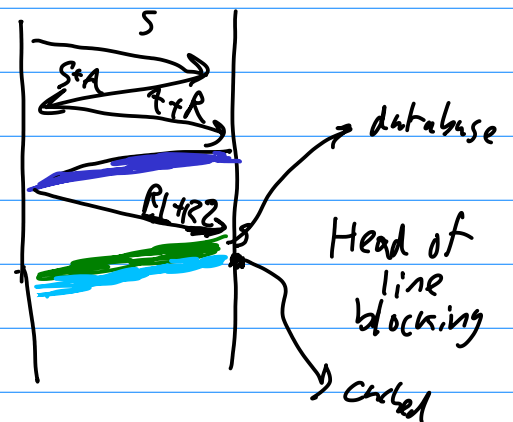
HTTP 1.1 Pipelining

never widely used

RFC 2616
"2 per server"

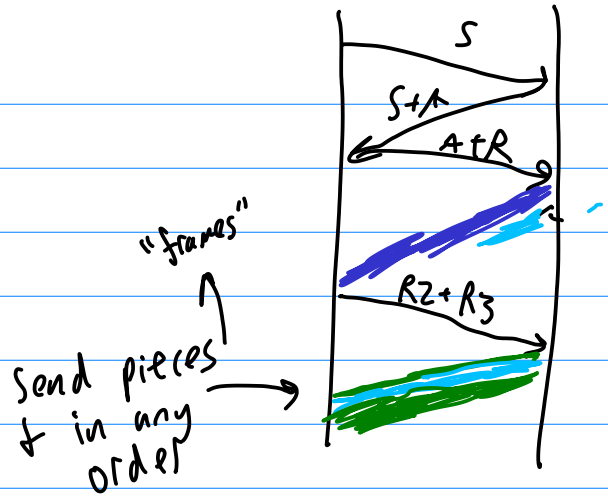
RFC 7230

"be conservative when opening new 11 conns"



HTTP 2.0

- frames
- any order
- server push
- headers were compressed



→ remove incentive for multiple connections

HTTP
TLS
TCP

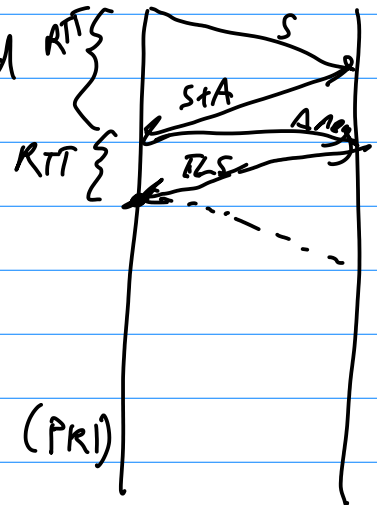
Issues:

- what if packet gets dropped?
 - TCP retransmit, all objects delayed
- handshake has become more expensive

"HTTPS everywhere"

TLS "transport layer security"

- encryption
- server authentication certificate
cryptokeys (PKI)



Fix TCP!

HTTP/3 over QUIC (chrome)

quick udp internet connection

UDP ← IP

- reimpl. reliability (per-stream)
- reimpl. congestion control
- combine handshake "connection" + "TLS"
- forward error correction
- connection migration
- already used in mobile video

NAT
≡

Layers?