



香港科技大學  
THE HONG KONG  
UNIVERSITY OF SCIENCE  
AND TECHNOLOGY

# The Web

**Prof. Zili Meng**

Fall 2024

ELEC 3120: Computer Networks

<https://www.foggynetwork.com>



# Warmup

- Identify four communications mediums: two may be computer communications, but two should be for other types of communication!
- Define the term *velocity factor*. Why is it relevant to communications networks? [velocity factor =  $\frac{\text{speed of light in the medium}}{\text{speed of light in the vacuum}}$ ]
- Bob transmits a 10000 byte message over a 10kbps link which is 10km long and has a velocity factor of 0.6.
  - How long does it take for Bob's message to reach the other side?
  - Bob wants to make his message arrive faster, so he switches to a link which is exactly the same, but only 5km long. Will Bob see a significant performance improvement?



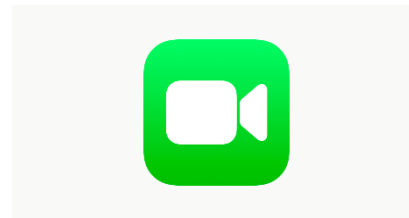
# Moving on to Applications!



# The Application Layer is what most users see.



facebook



zoom



Baidu 百度



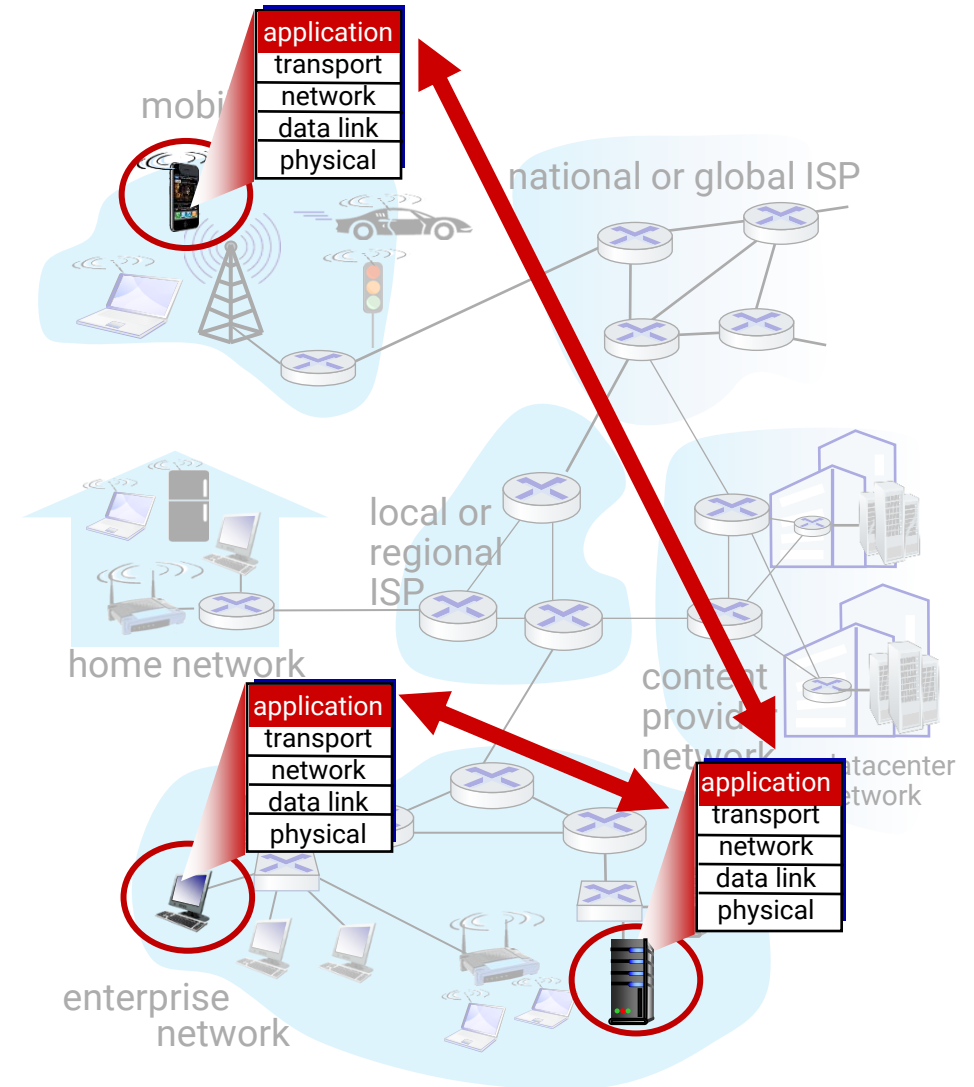
# Some network apps

- social networking
- Web
- text messaging
- e-mail
- multi-user network games
- P2P file sharing
- streaming stored video (YouTube, Hulu, Netflix)
- Real-time video conferencing (e.g., Zoom)
- Internet search
- remote login
- ...



# Creating a network app

- Write programs that:
  - run on (different) end systems
  - communicate over network
  - e.g., web server software communicates with browser software
- No need to write software for network-core devices
  - network-core devices do not run user applications
  - applications on end systems allows for rapid app development, propagation





# Client-server paradigm

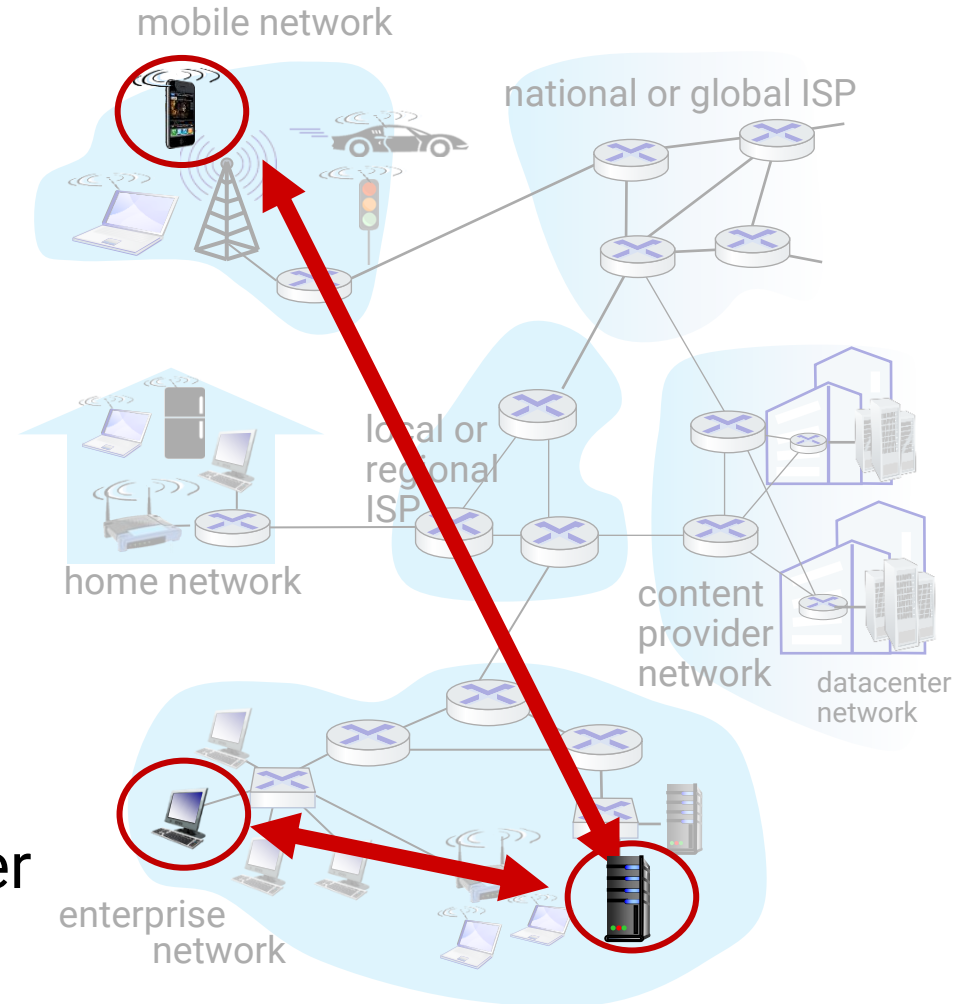
- **Server:**

- always-on host
- permanent IP address
- often in data centers, for scaling
- Think of it like a restaurant that has its door open waiting for customers to arrive.

- **Clients:**

- contact, communicate with server
- may be intermittently connected
- may have dynamic IP addresses
- do not communicate directly with each other

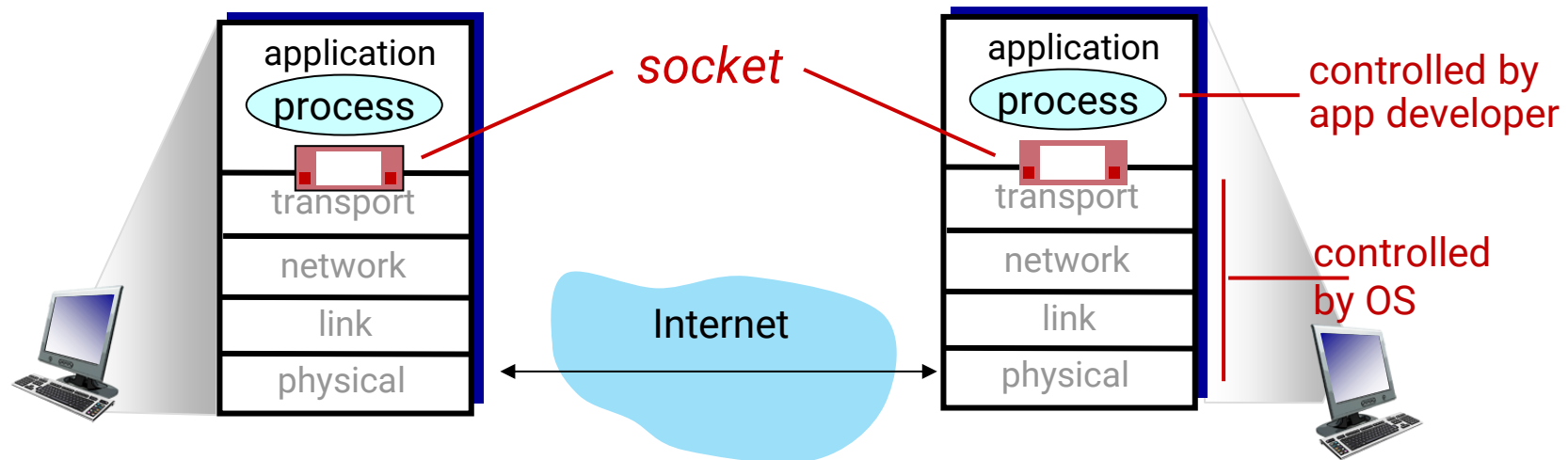
- **Examples: HTTP, IMAP, FTP**





# Sockets

- Process sends/receives messages to/from its socket
- Socket analogous to door
  - Sending process shoves message out door
  - Sending process relies on transport infrastructure on other side of door to deliver message to socket at receiving process
  - Two sockets involved: one on each side







Moving on to HTTP...



# 1945: Vannevar Bush

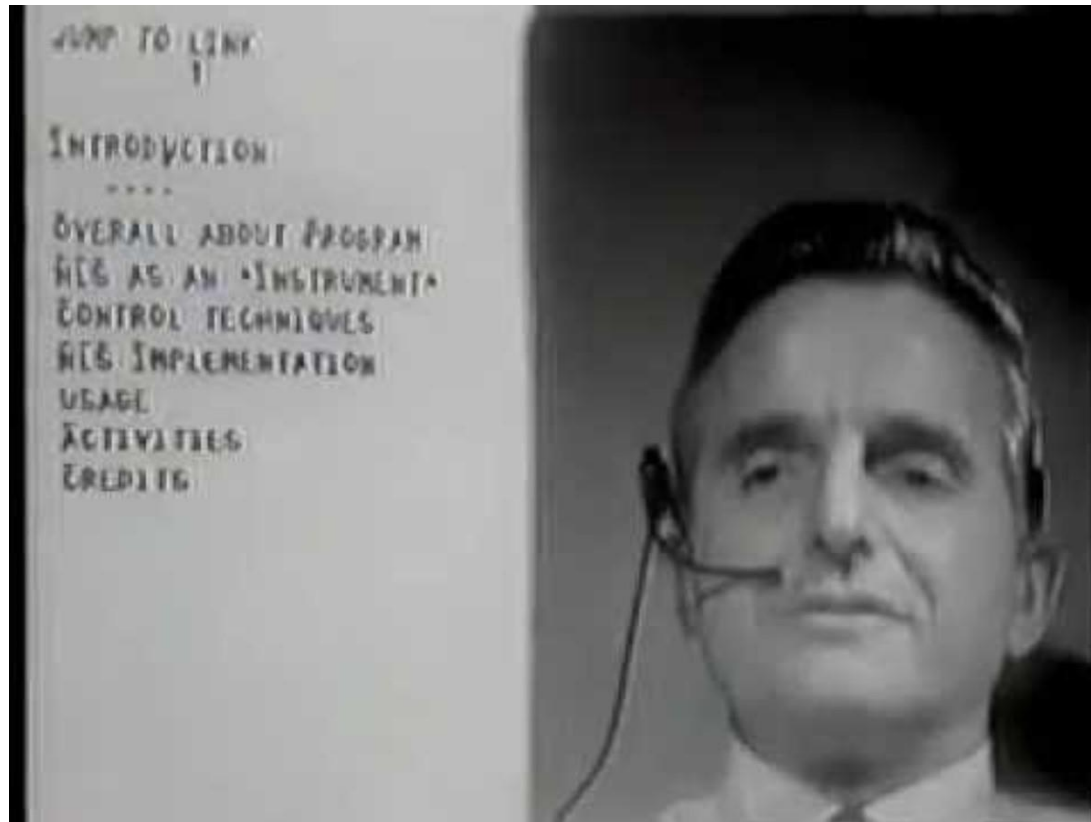
- “As we may think”, Atlantic Monthly, July, 1945.
- Describes the idea of a distributed hypertext system
- A “memex” that mimics the “web of trails” in our minds





# Dec 9, 1968: “The Mother of All Demos”

- First demonstration of Memex-inspired system
- Working prototype with hypertext, linking, use of a mouse...





# Many other hypertext systems before the World Wide Web

- MINITEL in France.
  - <https://en.wikipedia.org/wiki/Minitel>
- Project Xanadu.
  - [https://en.wikipedia.org/wiki/Project\\_Xanadu](https://en.wikipedia.org/wiki/Project_Xanadu)



# 1989: Tim Berners-Lee

- 1989: Tim Berners-Lee (CERN) writes internal proposal to develop a distributed hypertext system
  - Connects “a web of notes with links”.
  - Intended to help CERN physicists in large projects share and manage information
- 1990: TBL writes graphical browser for Next machines
- 1992-1994: NCSA/Mosaic/Netscape browser release



# Why is hypertext so powerful?

- Enables “browsing” through files on any web servers...
  - Exploring a “web” of connections not just on one server, but across servers.
- My opinion: transformed our mentality of Internet use from “point to point” communications (email, files on a single server) to one of *exploration* of all of the information online.



# Enabling Hypertext: The URL

- Uniform Resource Locator
- A reference to a resource on the network specifying
  - it's host's location in the network ("authority")
  - the resource's location on the host ("path")
  - the protocol to use to retrieve it ("scheme")



# Enabling Hypertext: The URL

- Uniform Resource Locator
- A reference to a resource on the network specifying
  - it's host's location in the network ("authority")
  - the resource's location on the host ("path")
  - the protocol to use to retrieve it ("scheme")

https://

ece.hkust.edu.hk

/index.php



This is a web page to access using the https protocol.  
Other protocols you might see here: "mailto:" "ftp:" "http:"





# Enabling Hypertext: The URL

- Uniform Resource Locator
- A reference to a resource on the network specifying
  - it's host's location in the network ("authority")
  - the resource's location on the host ("path")
  - the protocol to use to retrieve it ("scheme")

<https://>

[ece.hkust.edu.hk](https://ece.hkust.edu.hk)

[/index.php](https://ece.hkust.edu.hk/index.php)



Domain name or IP address. Optionally, you can put a : and manually specify what port to use, e.g. "ece.hkust.edu.hk:443"



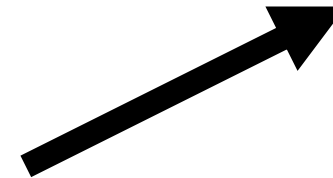
# Enabling Hypertext: The URL

- Uniform Resource Locator
- A reference to a resource on the network specifying
  - it's host's location in the network ("authority")
  - the resource's location on the host ("path")
  - the protocol to use to retrieve it ("scheme")

<https://>

[ece.hkust.edu.hk](https://ece.hkust.edu.hk)

[/index.php](https://ece.hkust.edu.hk/index.php)



Path: this is information on where to find the file on the server you are contacting



# Enabling Hypertext: The URL

- Uniform Resource Locator
- A reference to a resource on the network specifying
  - it's host's location in the network ("authority")
  - the resource's location on the host ("path")
  - the protocol to use to retrieve it ("scheme")
- URLs are a subclass of a more general "Uniform Resource Identifier" but this is all we need to know for now.



# Using Hypertext: HTML

- Originally:
  - A simple language for displaying text, images, and hypertext to be used in the web.


```
<html>
  <head>
    <title>This is a title</title>
  </head>
  <body>
    <div>
      <p>Hello world!</p>
      <a href="https://ece.ust.hk">This dept is awesome!</p>
    </div>
  </body>
</html>
```



# Using Hypertext: HTML

- Originally:
  - A simple language for displaying text, images, and hypertext to be used in the web.
- Today:
  - A much richer language that orchestrates multiple different languages and files to generate web applications.
    - Integrates with cascading style sheets, JavaScript, and other technologies to create interactive online services.

# Loading a "Single Web Page" Means Loading Many Files!



WIKIPEDIA  
The Free Encyclopedia

Main page  
Contents  
Current events  
Random article  
About Wikipedia  
Contact us  
Donate

Contribute

Help  
Learn to edit  
Community portal  
Recent changes  
Upload file

Tools

What links here  
Related changes  
Special pages  
Permanent link  
Page information  
Wikidata item

Print/export

Download as PDF  
Printable version

In other projects

Wikimedia Commons  
MediaWiki  
Meta-Wiki  
Multilingual Wikisource  
Wikispecies


Not logged in [Talk](#) [Contributions](#) [Create account](#) [Log in](#)

[Main Page](#) [Talk](#) [Read](#) [View source](#) [View history](#)

## Welcome to Wikipedia,

the [free encyclopedia](#) that [anyone can edit](#).  
6,567,166 articles in [English](#)

### From today's featured article




**Mini scule** is a [species](#) of [microhylid](#) frog [endemic](#) to Madagascar that was [described](#) in 2019. The [scientific name](#) of the species refers to its size, being a [pun](#) on the word *minuscule*. It is very small, measuring only 8.4 to 10.8 mm (0.33 to 0.43 in) in [snout–vent length](#). It has bronze underparts with a brown groin and back of the thigh, cream upperparts with brown flecking, a dark brown side of the head, and a red [iris](#). On the hind feet, the first toe is absent and the second and fifth toes are strongly reduced. The frog is known only from the [Sainte Luce Reserve](#), where it inhabits areas with deep [leaf litter](#) near semi-permanent water bodies. Specimens of frogs from Mandena, the [Vohimena](#) mountains, the southern Anosy Mountains, and [Tsitongambarika](#) may also be of this species. Along with *Mini mum* and *Mini ature*, the other two species in [its genus](#), it received media attention when first described due to the [wordplay](#) in its scientific name. ([Full article...](#))

Recently featured: [Mischief Makers](#) · [Second Battle of Cape Finisterre](#) · [Oryzomys antillarum](#)

[Archive](#) · [By email](#) · [More featured articles](#) · [About](#)

### Did you know ...


- ... that the relative rarity of the [radiodont](#) ***Titanokorys*** ([video featured](#)) in [Marble Canyon](#) suggests that the deposits in which it was found may represent the outermost edge of its distribution in life?
- ... that Iraqi poet **Isa Hasan al-Yasiri** ran away from school at 10 years old, before travelling with a camel



*Titanokorys* reconstruction

### In the news

- [Rishi Sunak](#) (*pictured*) **succeeds** [Liz Truss](#) as [Leader of the Conservative Party](#) and [Prime Minister of the United Kingdom](#).
- [Xi Jinping](#) is named [General Secretary of the Chinese Communist Party](#) for a third term after the conclusion of [the Party Congress](#).
- [Ulf Kristersson](#) is elected [Prime Minister of Sweden](#) following a [four-party agreement](#).
- [Hurricane Julia](#) leaves more than 90 people dead across [South](#) and [Central America](#).



Rishi Sunak


**Ongoing:** [Mahsa Amini protests](#) · [Russian invasion of Ukraine](#)  
**Recent deaths:** [John Jay Osborn Jr.](#) · [Ash Carter](#) · [Blanche Lemco van Ginkel](#) · [Galina Pisarenko](#) · [Rodney Graham](#) · [Libor Pešek](#)

[Nominate an article](#)

### On this day

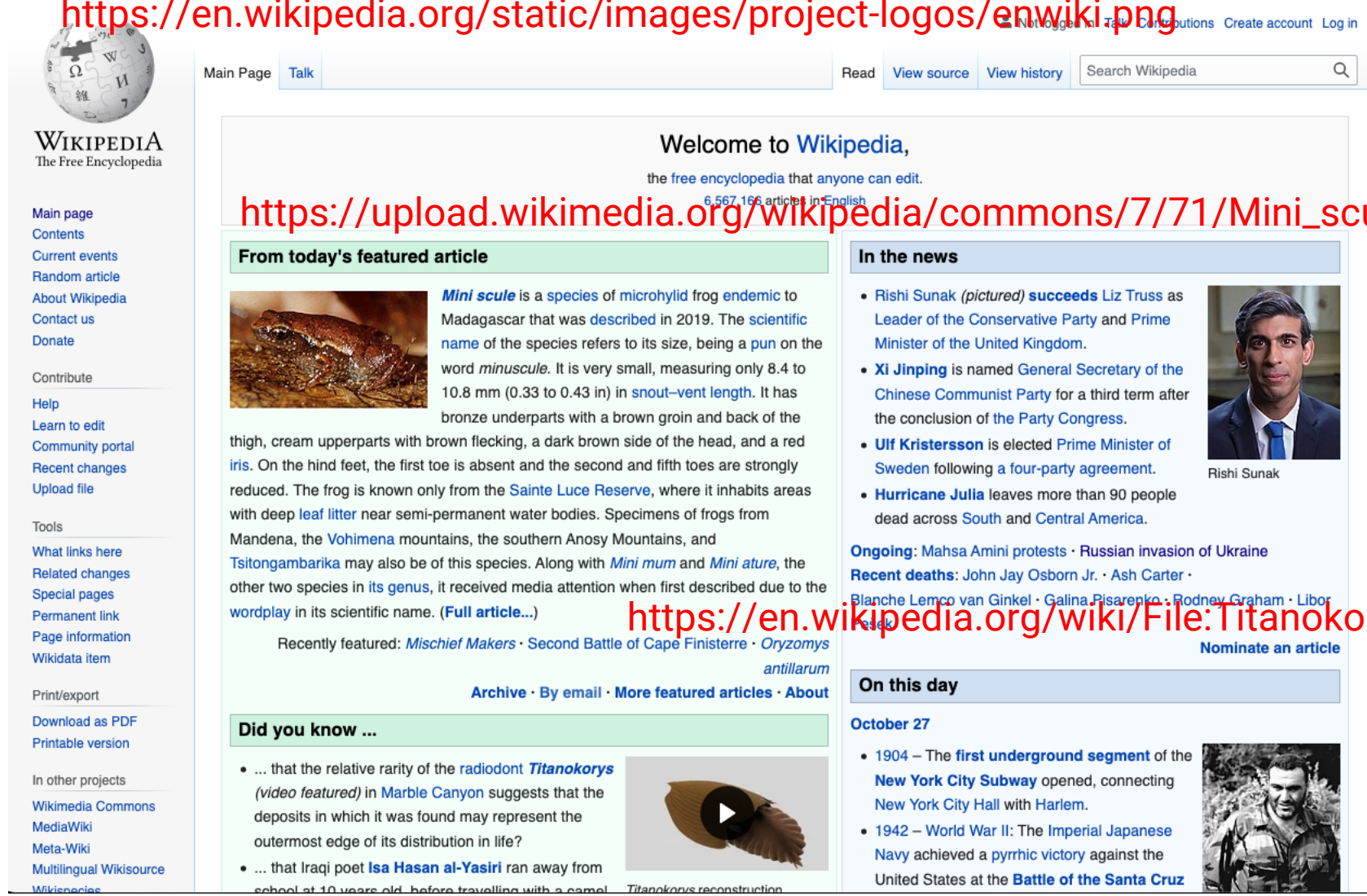
**October 27**

- 1904 – The [first underground segment](#) of the [New York City Subway](#) opened, connecting [New York City Hall](#) with [Harlem](#).
- 1942 – [World War II](#): The [Imperial Japanese Navy](#) achieved a [pyrrhic victory](#) against the United States at the [Battle of the Santa Cruz](#)



# Loading a "Single Web Page" Means Loading Many Files!

<https://en.wikipedia.org/static/images/project-logos/enwiki.png>



The screenshot shows the Wikipedia homepage with the following sections:

- Welcome to Wikipedia**: the free encyclopedia that anyone can edit. 6,567,166 articles in English.
- From today's featured article**: A featured article about the *Mini scule* frog, a species of microhylid frog endemic to Madagascar. It includes a photo of the frog and a detailed description of its physical characteristics and habitat.
- In the news**: A section listing recent news items, including Rishi Sunak's appointment as Prime Minister of the United Kingdom, Xi Jinping's re-election as General Secretary of the Chinese Communist Party, and the election of Ulf Kristersson as Prime Minister of Sweden.
- On this day**: A section for October 27, listing historical events such as the opening of the first underground segment of the New York City Subway and the Imperial Japanese Navy's victory against the United States at the Battle of the Santa Cruz.
- Did you know ...**: A section with trivia, including the relative rarity of the radiodont *Titanokorys* and the Iraqi poet Isa Hasan al-Yasiri.

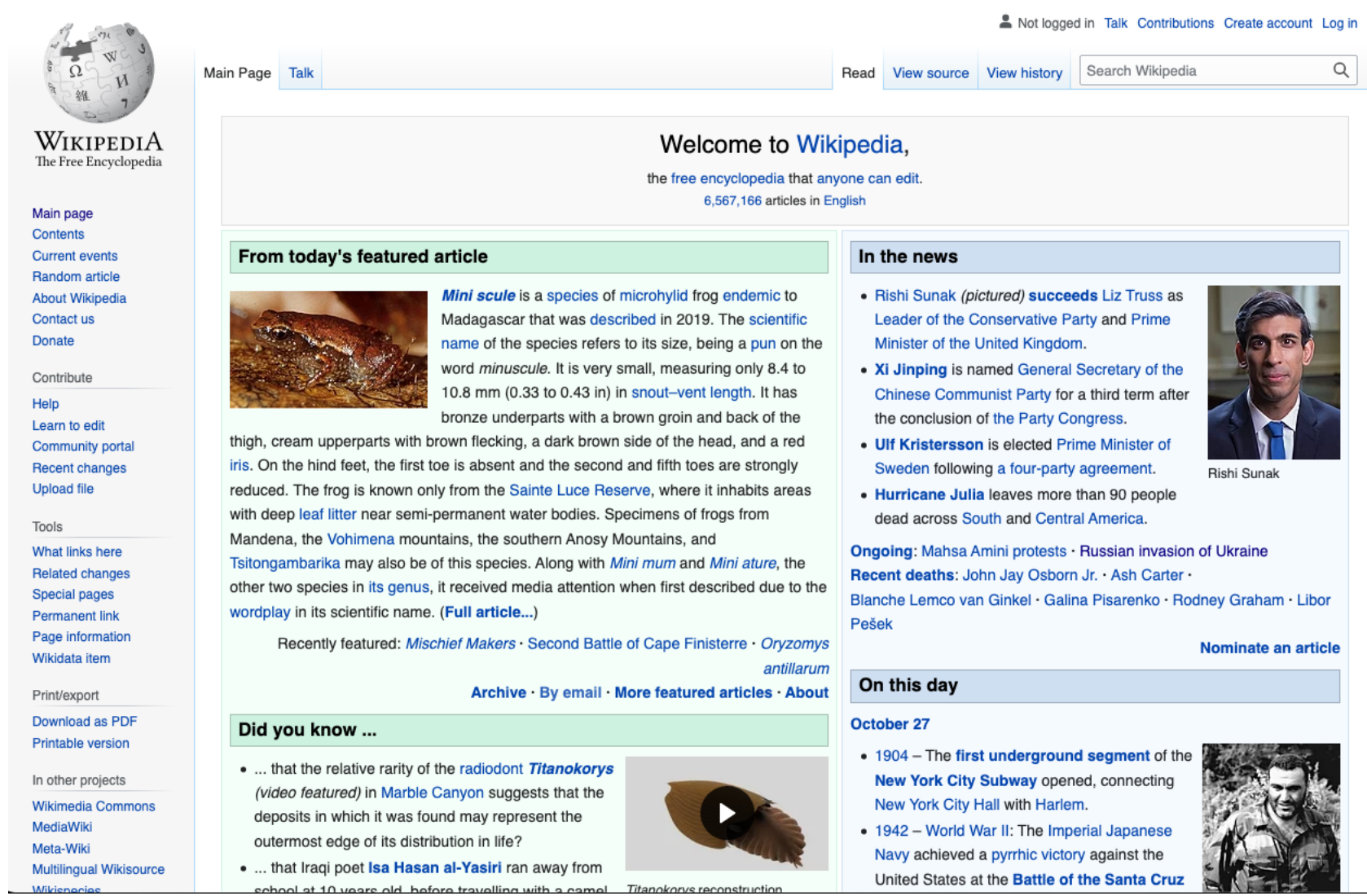
## Images and Videos

[https://upload.wikimedia.org/wikipedia/commons/thumb/a/af/%D5%8E%D5%A1%D5%A6%D5%A3%D5%A5%D5%B6\\_%D5%8D%D5%A1%D6%80%D5%A3%D5%BD%D5%B5%D5%A1%D5%B6.jpg/127px-%D5%8E%D5%A1%D5%A6%D5%A3%D5%A5%D5%B6\\_%D5%8D%D5%A1%D6%80%D5%A3%D5%BD%D5%B5%D5%A1%D5%B6.jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/a/af/%D5%8E%D5%A1%D5%A6%D5%A3%D5%A5%D5%B6_%D5%8D%D5%A1%D6%80%D5%A3%D5%BD%D5%B5%D5%A1%D5%B6.jpg/127px-%D5%8E%D5%A1%D5%A6%D5%A3%D5%A5%D5%B6_%D5%8D%D5%A1%D6%80%D5%A3%D5%BD%D5%B5%D5%A1%D5%B6.jpg)

<https://en.wikipedia.org/wiki/File:Titanokorys reconstruction.jpg>



# Loading a “Single Web Page” Means Loading Many Files!



The screenshot shows the Wikipedia Main Page. At the top, it says "Welcome to Wikipedia, the free encyclopedia that anyone can edit. 6,567,166 articles in English". Below this, there are two main sections: "From today's featured article" and "In the news".

**From today's featured article**

**Mini scule** is a species of microhylid frog endemic to Madagascar that was described in 2019. The scientific name of the species refers to its size, being a pun on the word *minuscule*. It is very small, measuring only 8.4 to 10.8 mm (0.33 to 0.43 in) in snout-vent length. It has bronze underparts with a brown groin and back of the thigh, cream upperparts with brown flecking, a dark brown side of the head, and a red iris. On the hind feet, the first toe is absent and the second and fifth toes are strongly reduced. The frog is known only from the Sainte Luce Reserve, where it inhabits areas with deep leaf litter near semi-permanent water bodies. Specimens of frogs from Mandena, the Vohimena mountains, the southern Anosy Mountains, and Tsitongambarika may also be of this species. Along with *Mini mum* and *Mini ature*, the other two species in its genus, it received media attention when first described due to the wordplay in its scientific name. (Full article...)

Recently featured: *Mischief Makers* · *Second Battle of Cape Finisterre* · *Oryzomys antillarum*

**Did you know ...**

- ... that the relative rarity of the radiodont *Titanokorys* (video featured) in Marble Canyon suggests that the deposits in which it was found may represent the outermost edge of its distribution in life?
- ... that Iraqi poet *Isa Hasan al-Yasiri* ran away from school at 10 years old, before travelling with a camel.

**In the news**

- Rishi Sunak** (pictured) **succeeds** Liz Truss as Leader of the Conservative Party and Prime Minister of the United Kingdom.
- Xi Jinping** is named General Secretary of the Chinese Communist Party for a third term after the conclusion of the Party Congress.
- Ulf Kristersson** is elected Prime Minister of Sweden following a four-party agreement.
- Hurricane Julia** leaves more than 90 people dead across South and Central America.

**Ongoing:** Mahsa Amini protests · Russian invasion of Ukraine  
**Recent deaths:** John Jay Osborn Jr. · Ash Carter · Blanche Lemco van Ginkel · Galina Pisarenko · Rodney Graham · Libor Pešek

**On this day**

**October 27**

- 1904 – The first underground segment of the New York City Subway opened, connecting New York City Hall with Harlem.
- 1942 – World War II: The Imperial Japanese Navy achieved a pyrrhic victory against the United States at the Battle of the Santa Cruz.

**Invisible: Scripts and Stylesheets**





# Loading a “Single Web Page” Means Loading Many Files!

The image shows a side-by-side comparison of a web browser displaying the Wikipedia Main Page and its corresponding network activity in Chrome DevTools.

**Left Panel: Wikipedia Main Page**

- Header:** "Welcome to Wikipedia, the free encyclopedia that anyone can edit. 6,567,166 articles in English."
- From today's featured article:** A section about the *Mini scule* frog, describing its microhylid characteristics and its status as an endemic species of Madagascar.
- In the news:** A list of recent news items, including Rishi Sunak succeeding Liz Truss as Prime Minister of the United Kingdom, and Xi Jinping being named General Secretary of the Chinese Communist Party.
- Did you know ...:** A section highlighting the relative rarity of the radiodont *Titanokorys* in Marble Canyon, suggesting that the deposits in which it was found may represent the outermost edge of its distribution in life.

**Right Panel: Chrome DevTools Network Tab**

- Filter:** Shows "Blocked Requests" and "3rd-party requests".
- Request List:** A list of 20 requests, including "Sainte\_Luce\_Reserve", "Mini\_scul", "Marble\_Canyon\_Canadian\_Rockies", "Ash\_Carter", "data:image/svg+xml...", "Libor\_Pe%C5%A1ek", "data:image/svg+xml...", "New\_York\_City\_Subway", "New\_York\_City\_Hall", "data:image/svg+xml...", "data:image/svg+xml...", "data:image/svg+xml...", "data:image/svg+xml...", "New\_York\_City\_Hall", "data:image/svg+xml...", "Pyrrhic\_victory", "Imperial\_Japanese\_Navy", and "data:image/svg+xml...".
- Headers:** Shows the "General" tab for the selected request, displaying the "Request URL" as "https://en.wikipedia.org/api/rest\_v1/page/summary/Sainte\_Luce\_Reserve" and the "Status Code" as "200".
- Response Headers:** Shows the "Response Headers" tab, displaying various headers such as "accept-encoding", "access-control-allow-headers", "accept-ranges", "access-control-allow-headers", "content-type", "content-length", "cache-control", "accept-language", "api-user-agent", "if-match", "if-modified-since", "if-none-match", "dnt", "accept-encoding", "access-control-allow-methods", "ET, HEAD", and "access-control-allow-origin".

Google Chrome showing  
me all the files it loaded  
to generate this page.



# How many domains are on one Web page?

- 10% of website home pages have more than 58 different domains!
  - Top domains in Top 200: dailymail.co.uk, cnn.com, ...
- <https://arxiv.org/abs/2310.18030>, Figure 1(a) – My measurement!



# Learning More about Web Applications

- Go to YouTube. That's easy.
- You can build your own website (homepage, blog, etc.) using the GitHub Pages easily.
  - And you will get a human-readable domain name: username.github.io
  - <https://pages.github.com/>
- This class: we are going to learn more about the underlying protocol beneath the web
  - **HTTP: The Hypertext Transfer Protocol**



# HTTP



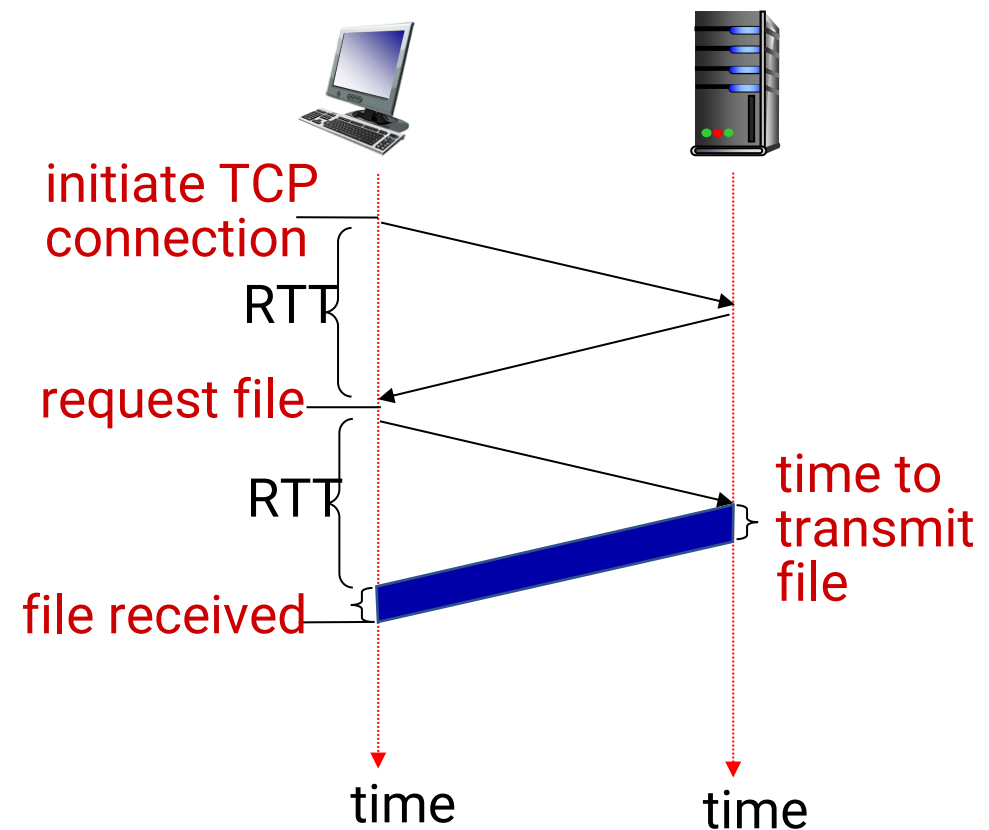
# Hyper Text Transfer Protocol (HTTP)

- Client-server architecture
  - Server is “always on” and “well known”
    - Recall the restaurant with the open door
  - Clients initiate contact to server
    - Instead of ordering dinner, they order up Wikipedia
- Synchronous request/reply protocol
  - Client asks sends a request, server replies



# HTTP 1.0: response time

- RTT (round-trip time)
  - Definition: time for a small packet to travel from client to server and back
- HTTP response time (per object):
  - One RTT to initiate TCP connection
  - One RTT for HTTP request and first few bytes of HTTP response to return
  - Object/file transmission time



*HTTP 1.0 response time = 2RTT + file transmission time*



# Client-to-Server Communication

- HTTP Request Message

- Request line: method, resource, and protocol version
- Request headers: provide information or modify request
- Body: optional data (e.g., to “POST” data to the server)

*request line*

*header lines*

GET /somedir/page.html HTTP/1.1

Host: www.someschool.edu

User-agent: Mozilla/4.0

Connection: close

Accept-language: fr

(blank line)

*carriage return line feed  
(CRLF) indicates end of  
message*



# Server-to-Client Communication

- HTTP Response Message

- Status line: protocol version, status code, status phrase
- Response headers: provide information
- Body: optional data

## *status line*

(protocol, status  
code, status  
phrase)

## *header lines*

## *data*

e.g., requested HTML  
file

HTTP/1.1 200 OK

Connection close

Date: Thu, 06 Aug 2006 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 2006 ...

Content-Length: 6821

Content-Type: text/html

(blank line)

data data data data data ...





# A Protocol Designed by a Physicist

- Wild ideas:
  - HTTP is plain-text.
    - You can read it, just by looking at it!
  - HTTP is variable-length.
    - There are no fixed byte-offsets.
- Leads to more complex parsing in software
  - But, easier for humans to reason about.
  - And, very extensible...
  - Maybe we should learn from physicists.



# One more note: HTTP is Stateless

- Each request-response treated independently
  - Servers not required to retain state
  - Abstraction is just “requesting and receiving files”
- Good: Improves scalability on the server-side
  - Failure handling is easier
  - Can handle higher rate of requests
  - Order of requests doesn't matter
- Bad: Some applications need persistent state
  - Need to uniquely identify user or store temporary info
  - e.g., Shopping cart, user profiles, usage tracking, ...



# One more note: HTTP is Stateless

- Solution: push state elsewhere
- To the client:
  - Cookies: web browser keeps some identifiers (e.g., an authentication token) or state and returns this information to the web server with its request.
- At the server side:
  - Database: store information relevant to client history here; when you receive a request from a client with a cookie, look up the right information from the database before assembling your response



# One more note: HTTP is Stateless

- Solution: push state elsewhere
- To the client:
  - Cookies: web browser keeps some identifiers (e.g., an authentication token) or state and returns this information to the web server with its request.
- At the server side:
  - Database: store information relevant to client history here; when you receive a request from a client with a cookie, look up the right information from the database before assembling your response

**Take note: some pages will take longer to assemble than others**



Can we make HTTP *fast*?

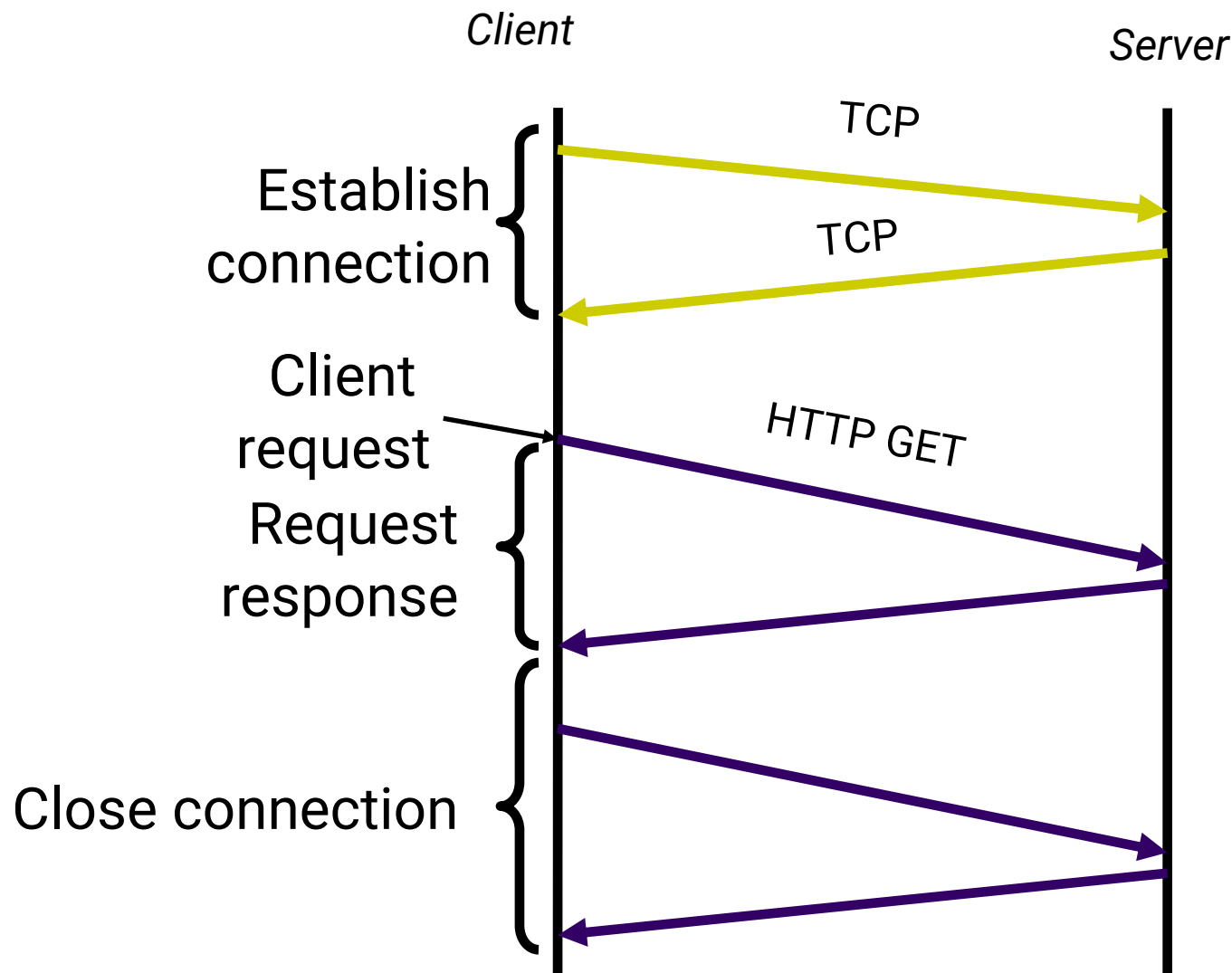


# Why make pages load faster?

- Back in the day, you would click and... wait!
- 100ms is typically cited as the “limit” of human perception
  - If you can get something to happen in 100ms or less, it feels “instant” to people.
- Oft-cited statistic from a study at Amazon that every additional 100ms of page load time (PLT) would cost them 1% in profit!
  - Sometimes you will see PLT called “latency” — I don’t like this because the word “latency” is used to mean many things. PLT is more precise.



# Let's start here with HTTP 1.0...





# HTTP Performance

- Most Web pages have multiple objects
  - e.g., HTML file and a bunch of embedded images
- How do you retrieve those objects (naively)?
  - One item at a time, i.e., one “GET” per TCP connection
  - Really limits the state on the server
  - Solution used in HTTP 0.9, and 1
- New TCP connection per (small) object!
  - Lots of handshakes



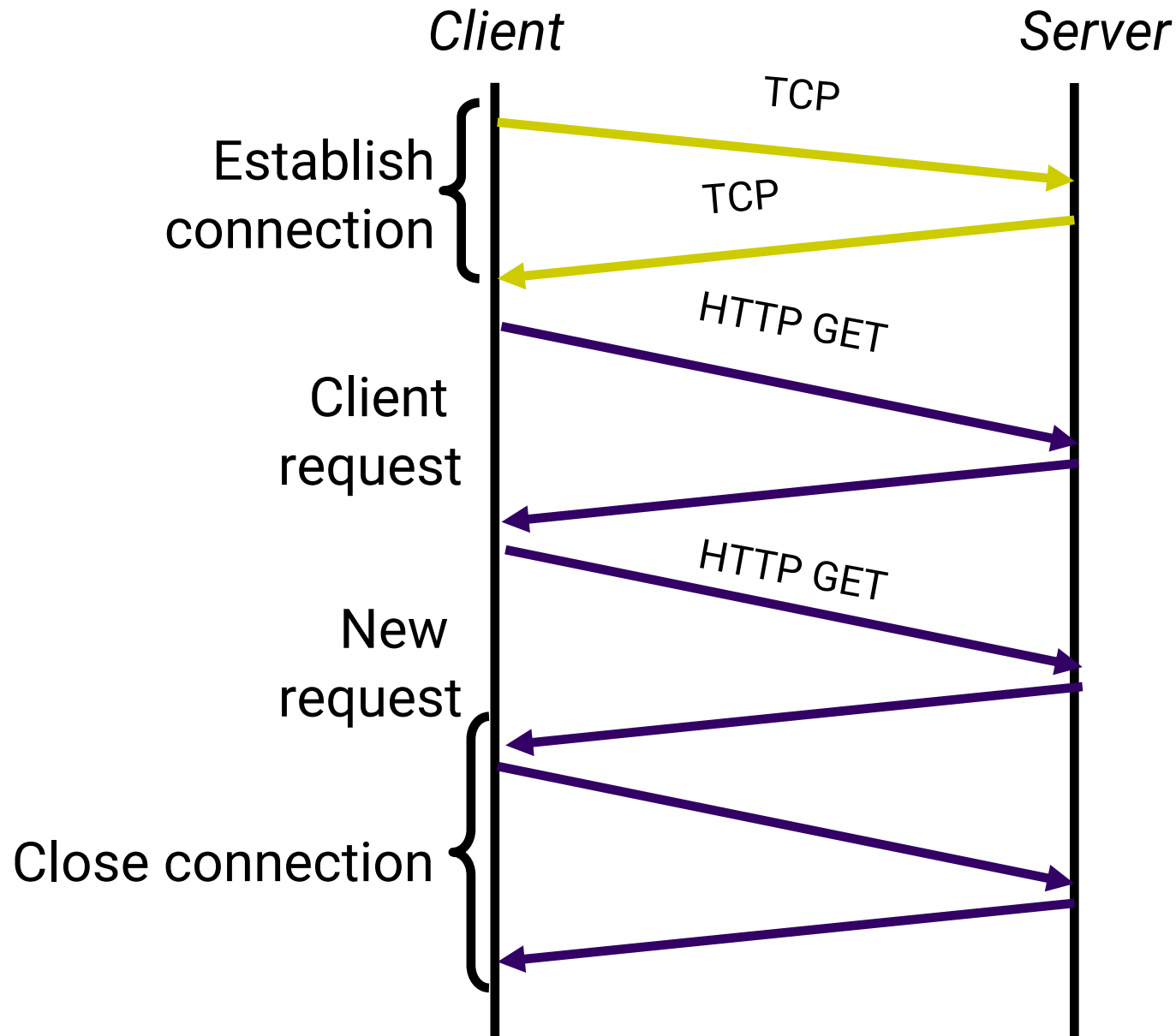


# Persistent Connections

- Maintain TCP connection across multiple requests
  - Including transfers subsequent to current page
  - Client or server can tear down connection
- Performance advantages:
  - Avoid overhead of connection set-up and tear-down
  - Allow network conditions to be reused
    - i.e., leverage previously discovered bandwidth



# HTTP with persistent connections





# Quick check...

- I have a web page with  $n$  files I need to request. Each file fits into one packet, but I need to make  $n$  requests.
- How many RTTs would it take for me to download these  $n$  files without persistent connections?
- How many RTTs would it take for me to download these  $n$  files with persistent connections?

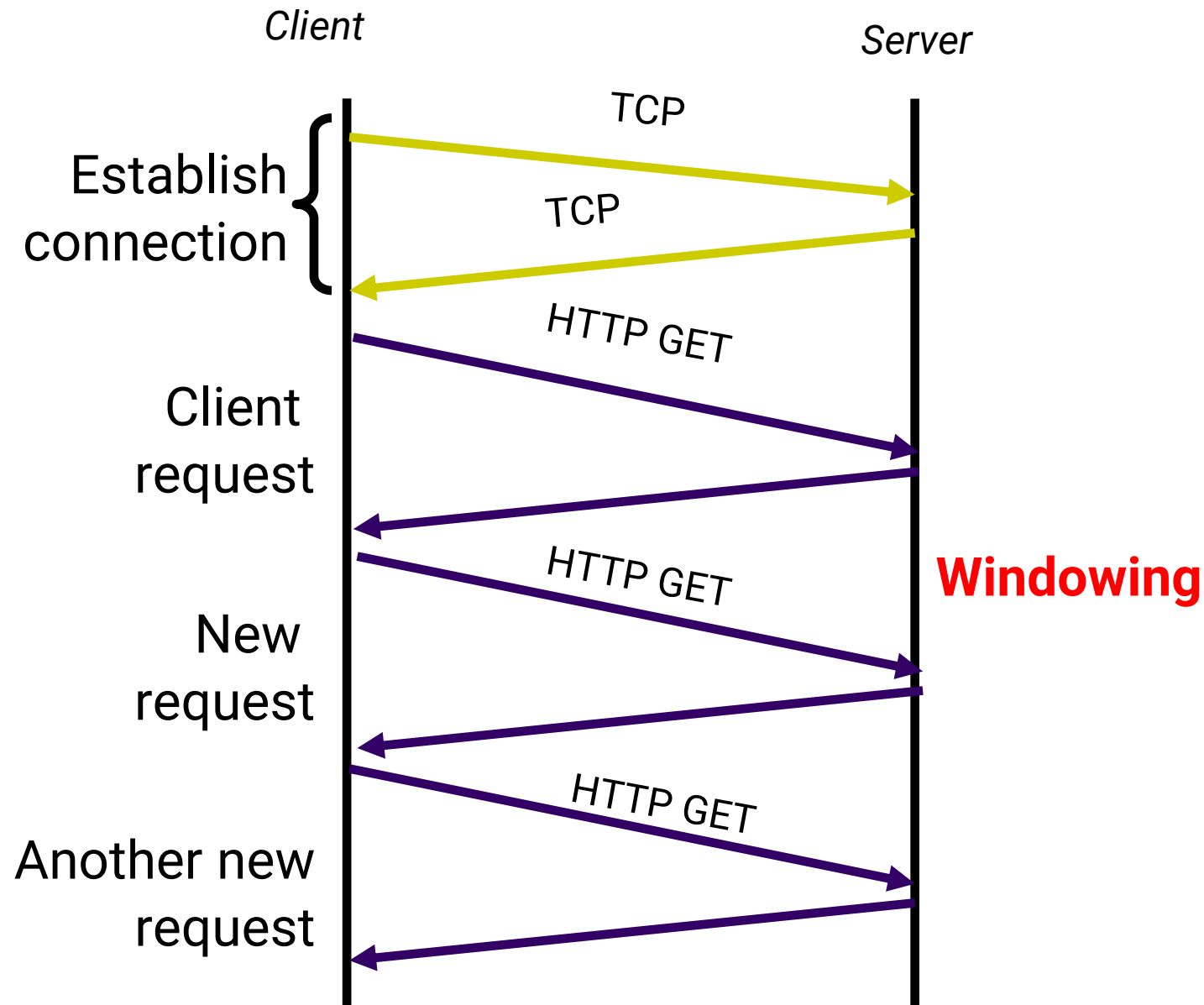


# Quick check...

- I have a web page with  $n$  files I need to request. Each file fits into one packet, but I need to make  $n$  requests.
- How many RTTs would it take for me to download these five files without persistent connections?  **$3n$**
- How many RTTs would it take for me to download these five files with persistent connections?  **$n + 2$**



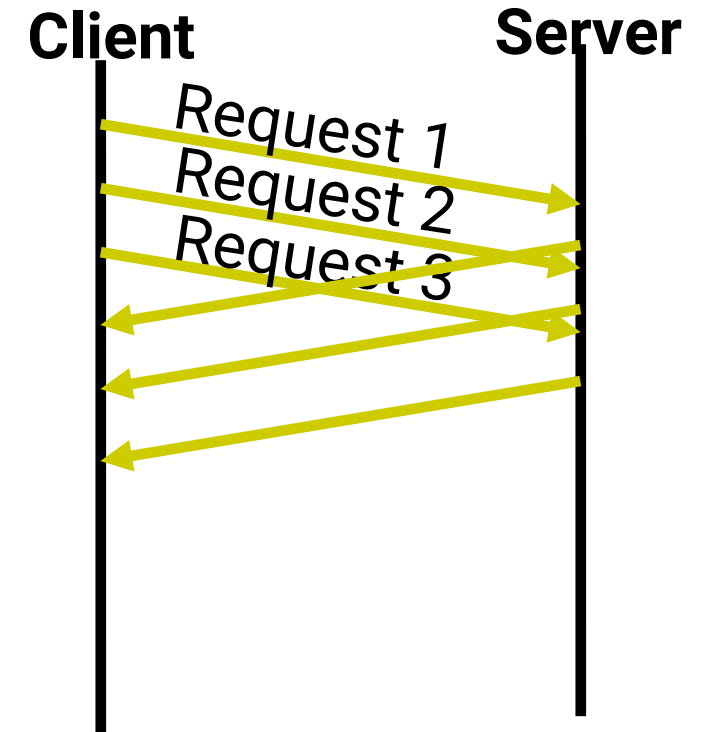
# Can we go faster?





# Pipelined Requests & Responses

- Key idea: don't need to wait until the previous request has received a response to send the next request.
- Send a series of requests, and server can respond one by one (in order) as they receive them.
- This approach more efficiently “fills the pipe.”





# Problems with Pipelining

- Pipelining works very well if all of the requested objects are immediately available and “ready to serve” at the time of request
- What if some objects take a long time for the server to produce?
  - Example: *I need to load a small image from the web server AND I need to load a complex data visualization that requires that the web server fetch information from an external database...*



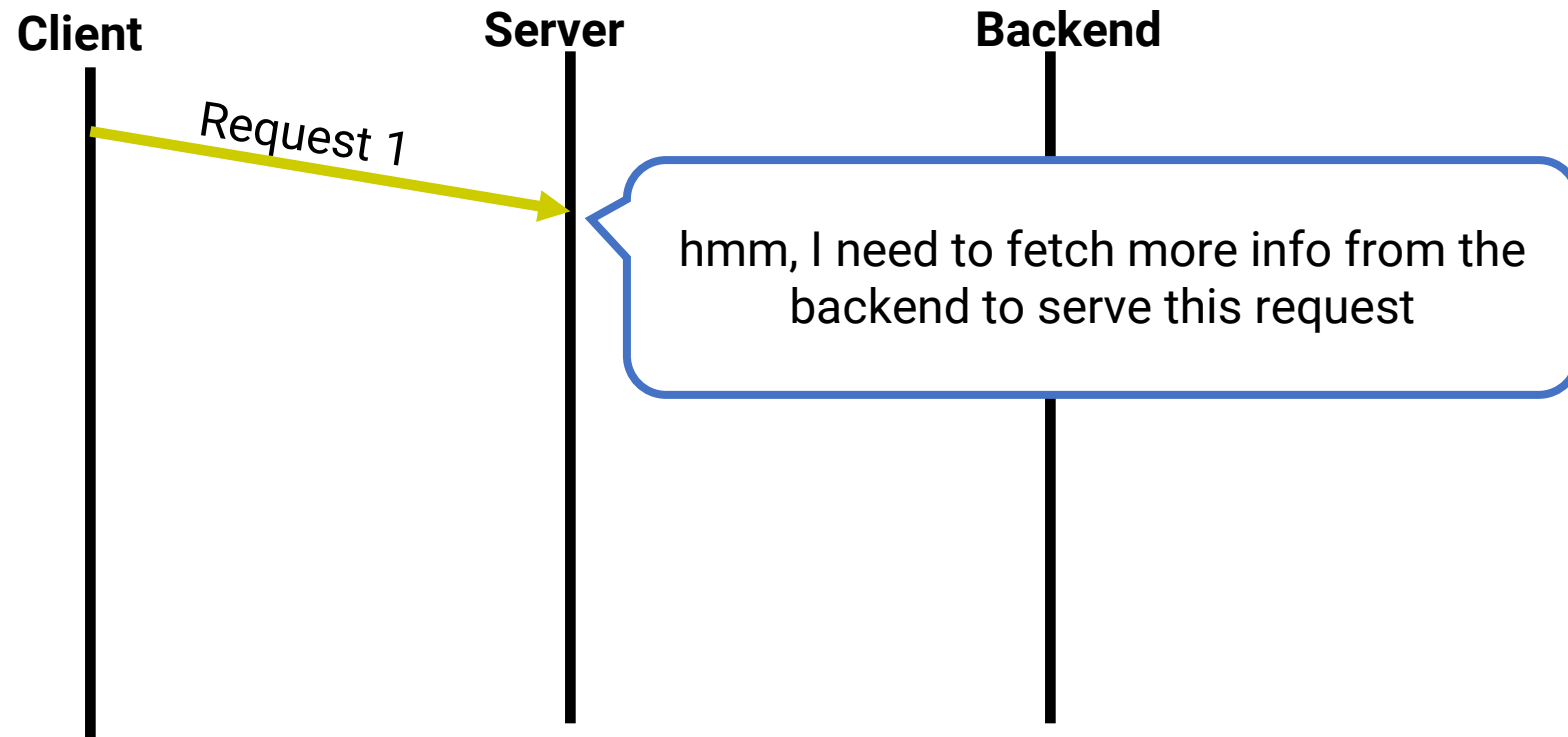
# Head of Line (HoL) Blocking

- When multiple tasks are queued to be performed by a system, and one task at the “head” of the queue cannot be served, the other tasks have to wait.
- This means that time goes by where the system is not doing work!
  - It is idle, because it cannot serve the first task.
  - Even though there are other tasks in the system, since they are queued behind the first task, they can't be served either.
- This means that the systems not being work-conserving which is inefficient.



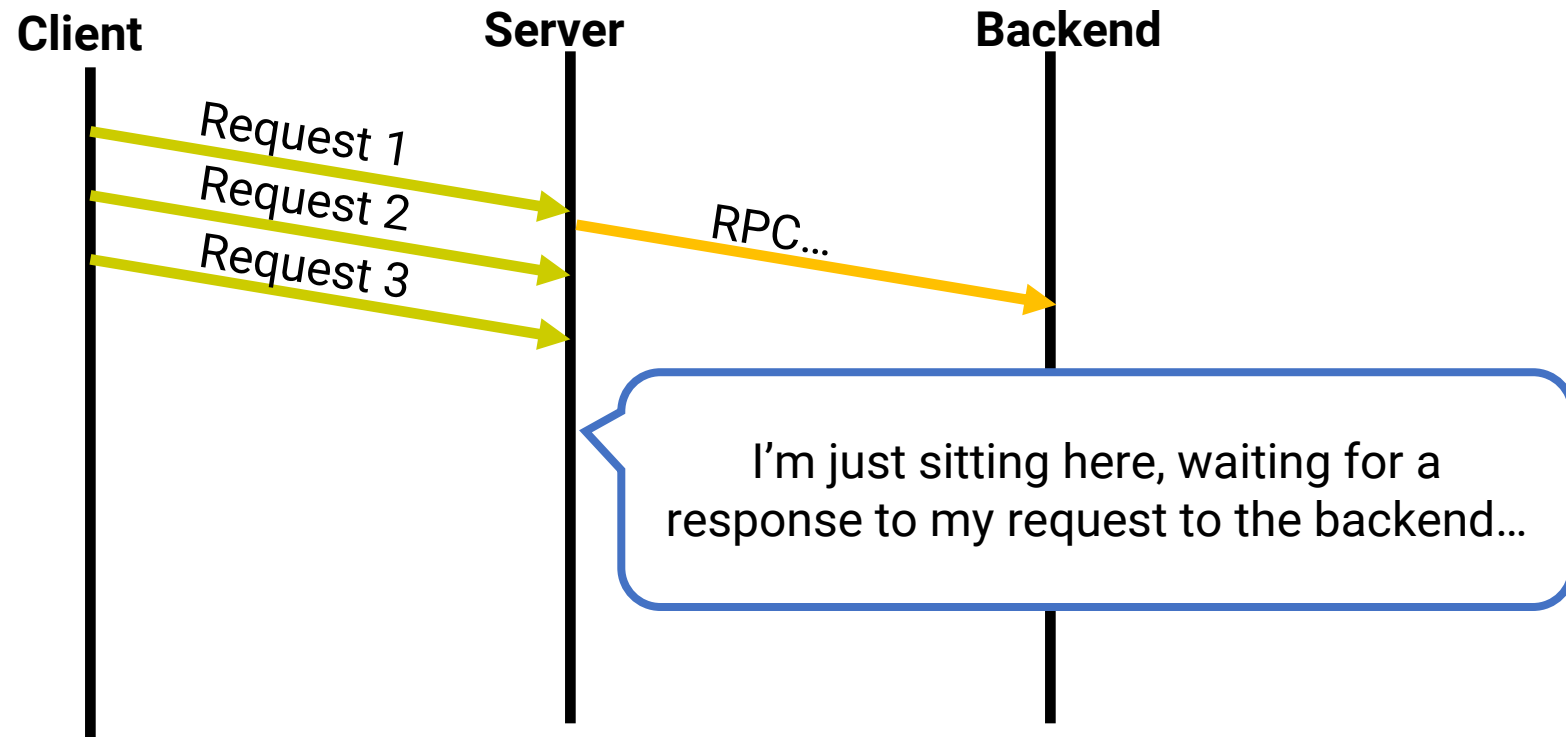


# Head of Line Blocking: Example



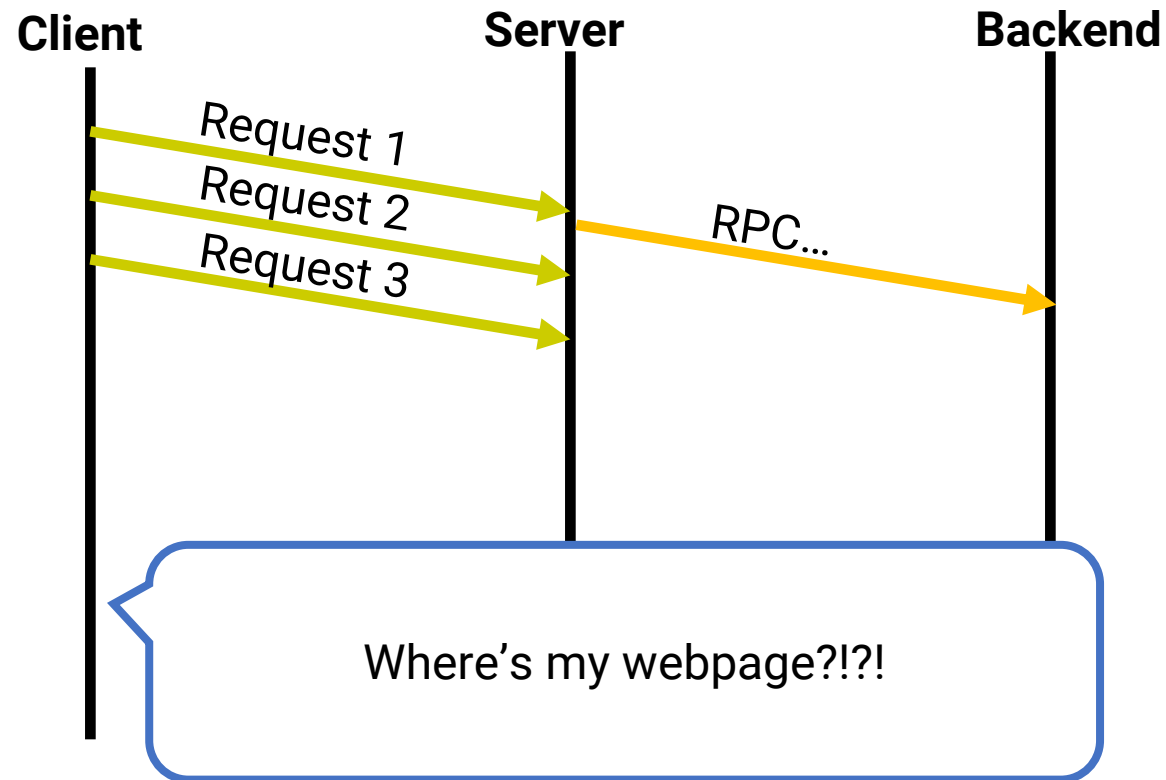


# Head of Line Blocking: Example



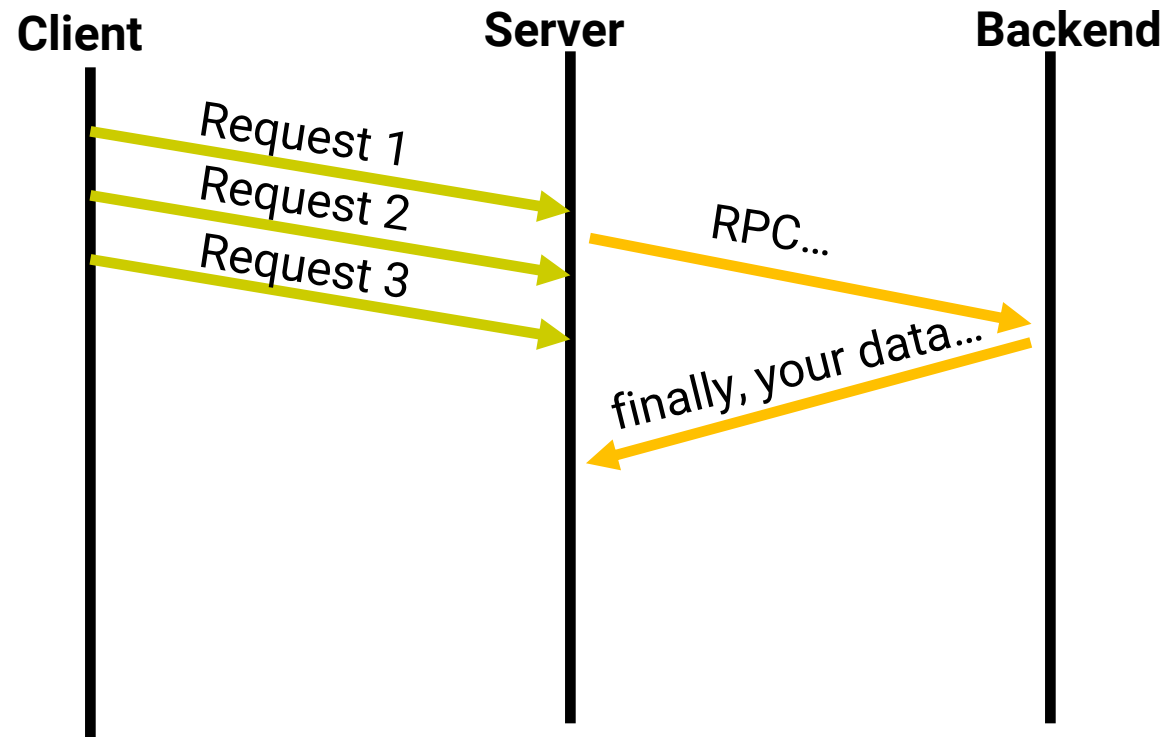


# Head of Line Blocking: Example



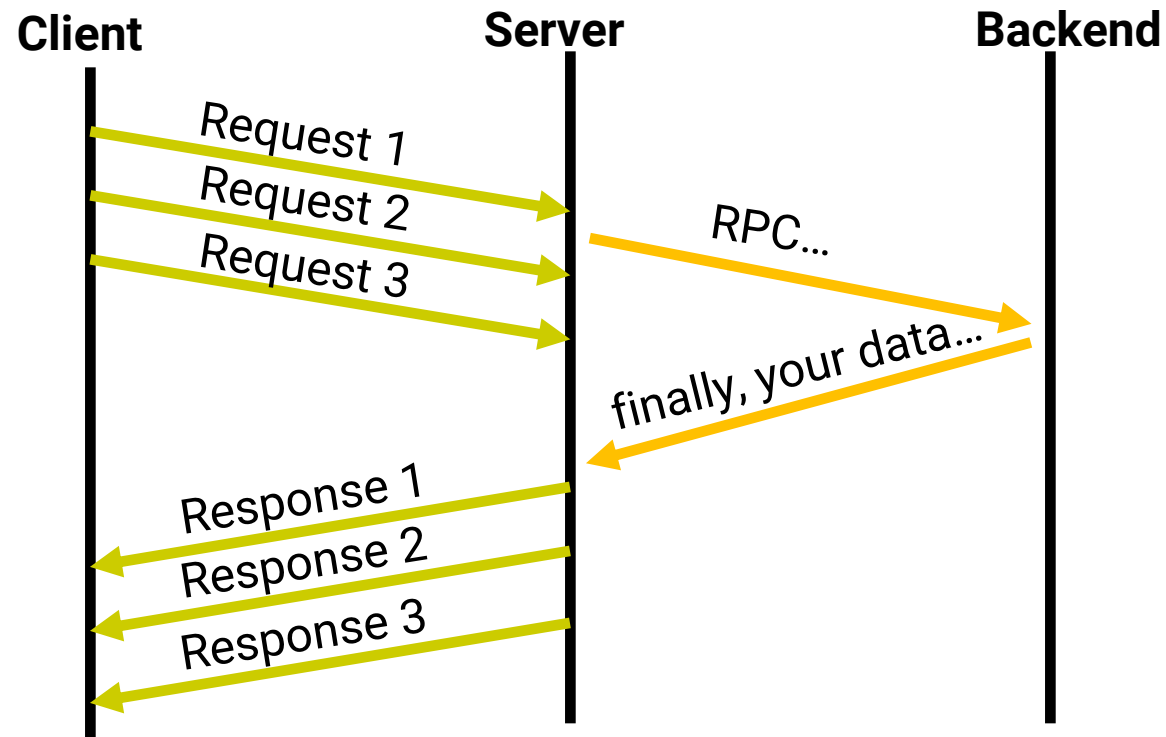


# Head of Line Blocking: Example



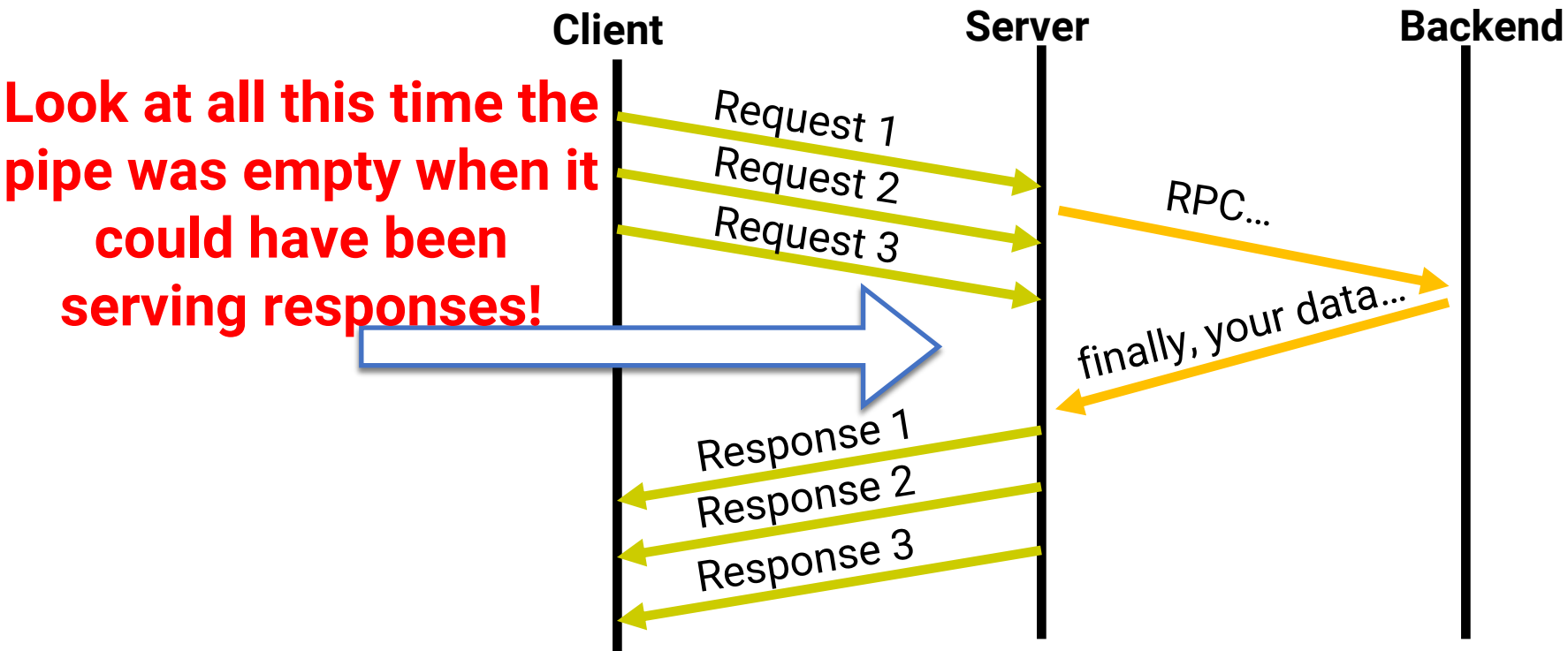


# Head of Line Blocking: Example





# Head of Line Blocking: Example








# Three Generations of “Fixes” to HOL Blocking

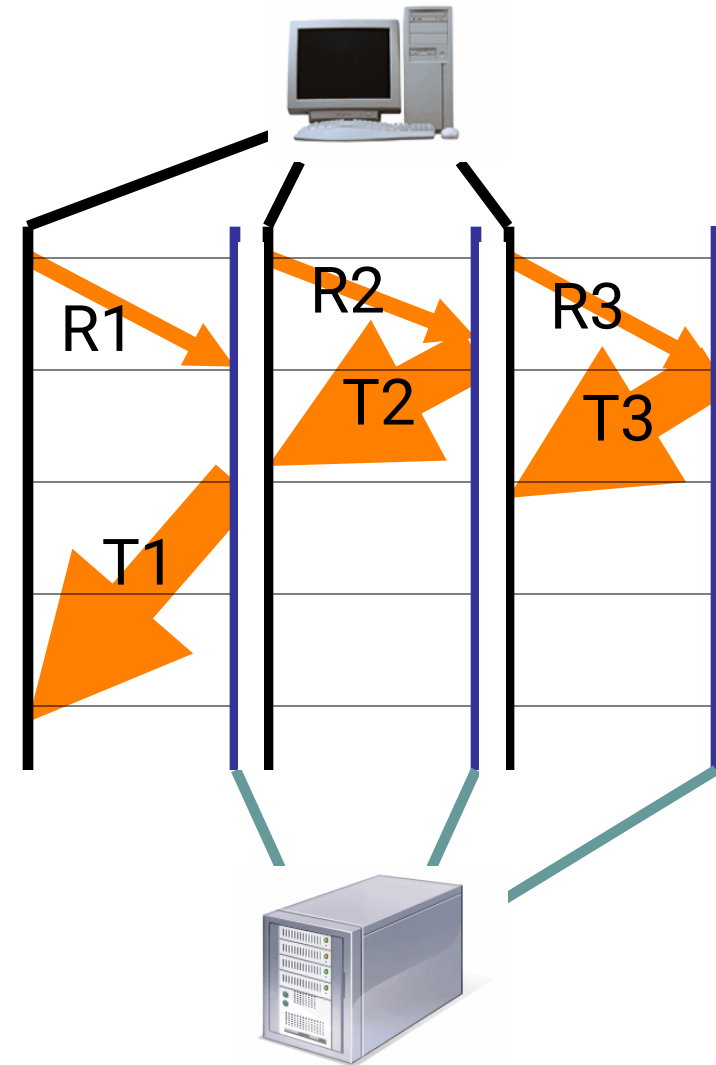
- Generation #1: Concurrent/Parallel Requests
  - Not ideal
- Generation #2: HTTP 2.0 Streams
  - Fixes part of the problem, but not all of it...
- Generation #3: HTTP 3.0 with UDP (QUIC)
  - State of the art, but not yet widely used...

*Quick note: older versions of this course, and lots of materials online will also talk about HTTP 2.0 PUSH. It is now deprecated — it was a big flop, so I’m not teaching it anymore!*



# Concurrent Requests & Responses Over Parallel TCP Sessions

- Use multiple connections in parallel
  - Speeds up retrieval by  $\sim m$
- Does not necessarily maintain order of responses
- Partially deals with HOL blocking
  - Client = 
  - Content provider = 
  - Network =  Why?





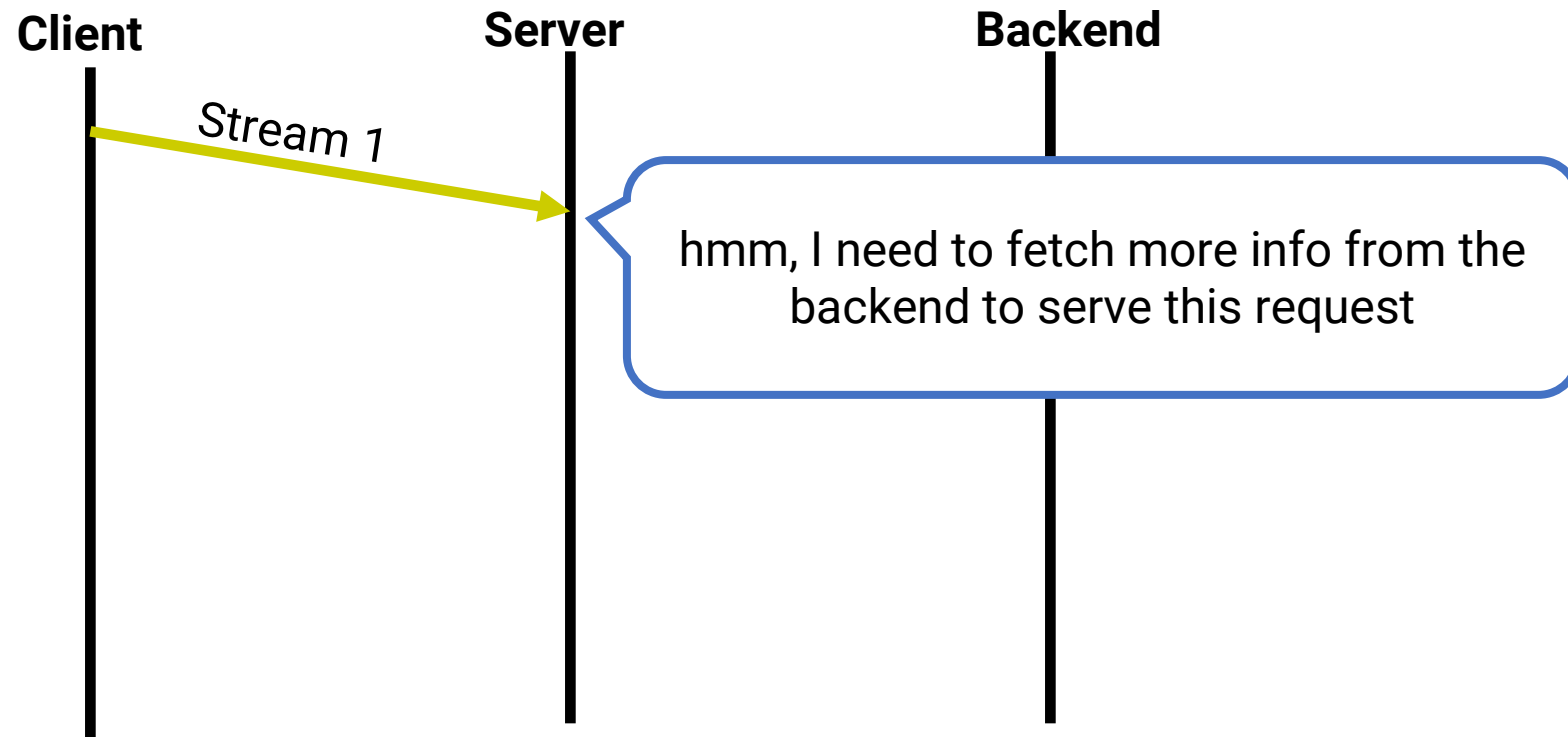


# HTTP 2.0 to the Rescue

- Still uses one persistent connection for multiple requests
  - Fewer handshakes, more traffic allowed in the pipe (let underlying congestion window grow)
- Request/Response pairs now abstracted into the idea of a “stream”
  - Each request/response is labeled with a Stream ID
  - Server is free to send responses for different streams out of order
  - This avoids HOL blocking since “fast” objects can be sent right away while processing continues for “slow” objects
- Another benefit in HTTP 2.0: HTTP headers are compressed to make header overhead smaller

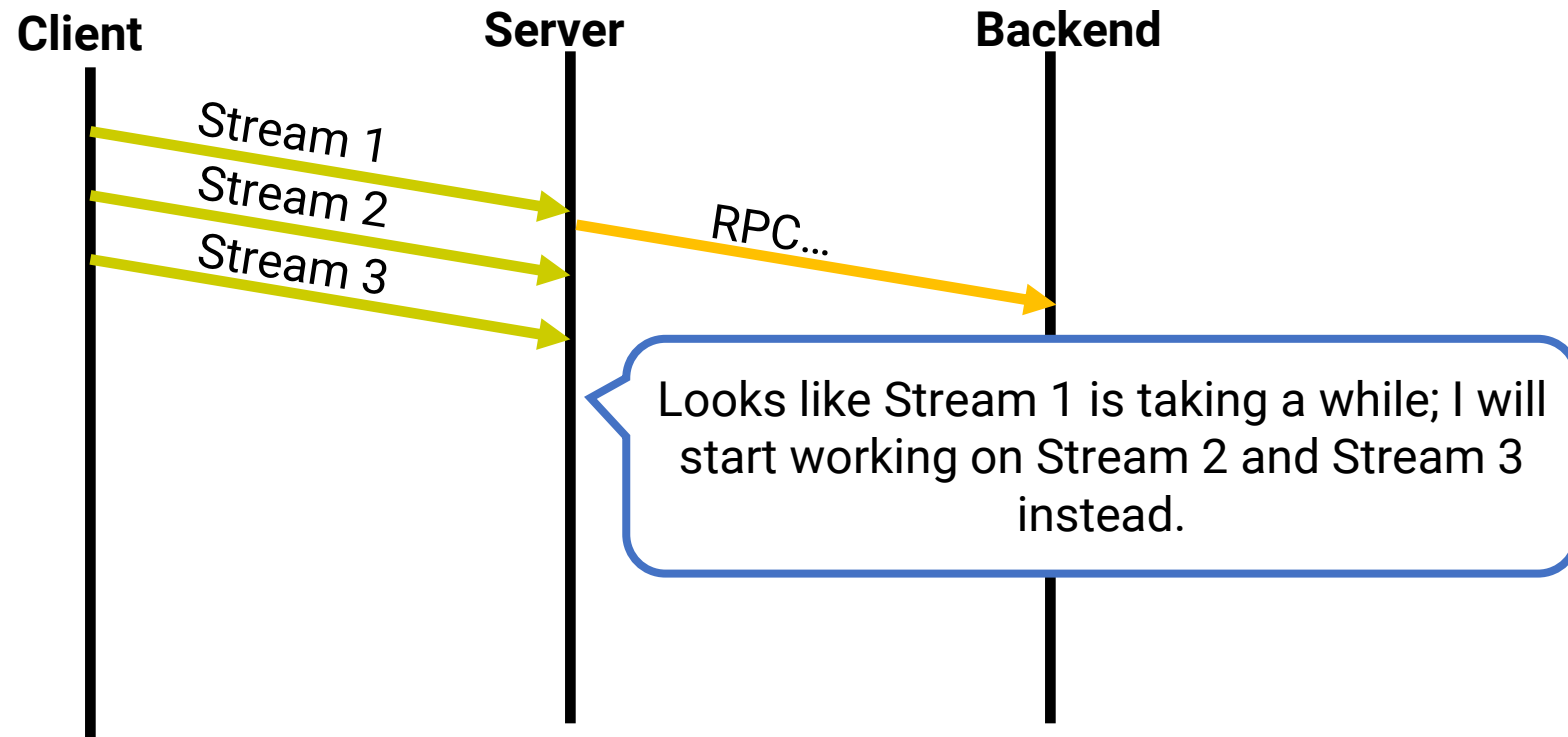


# Avoiding HOL Blocking in HTTP 2.0



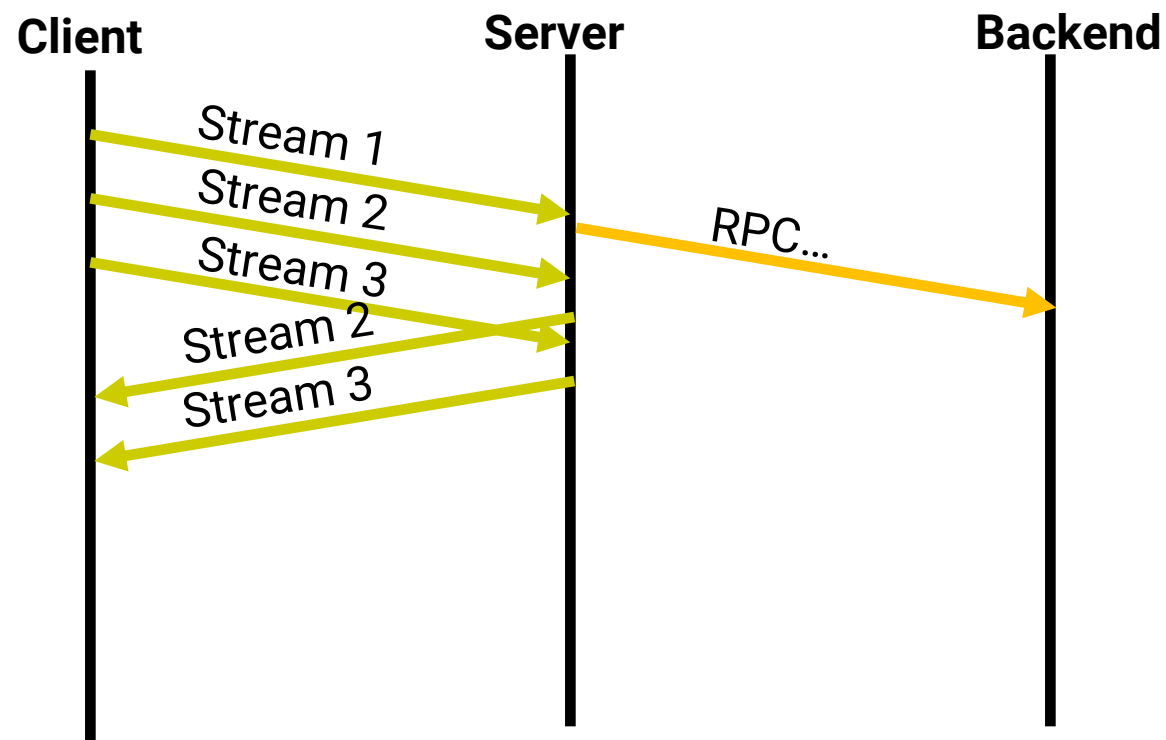


# Head of Line Blocking: Example



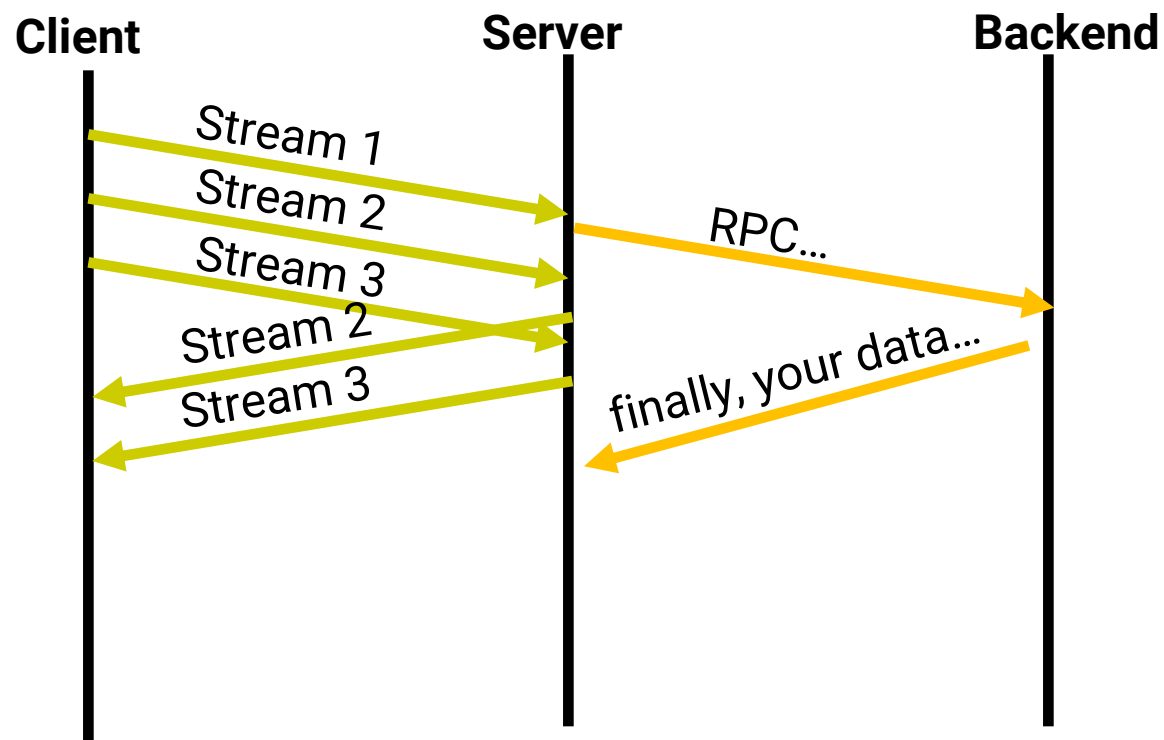


# Head of Line Blocking: Example



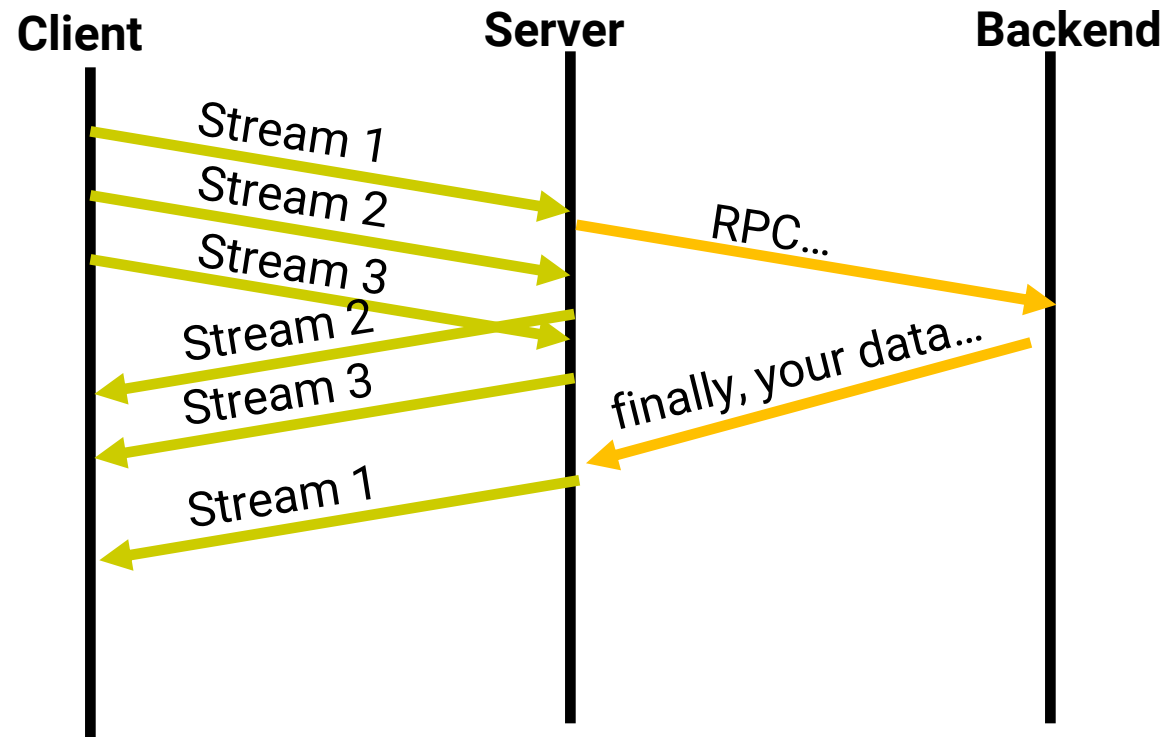


# Head of Line Blocking: Example





# Head of Line Blocking: Example

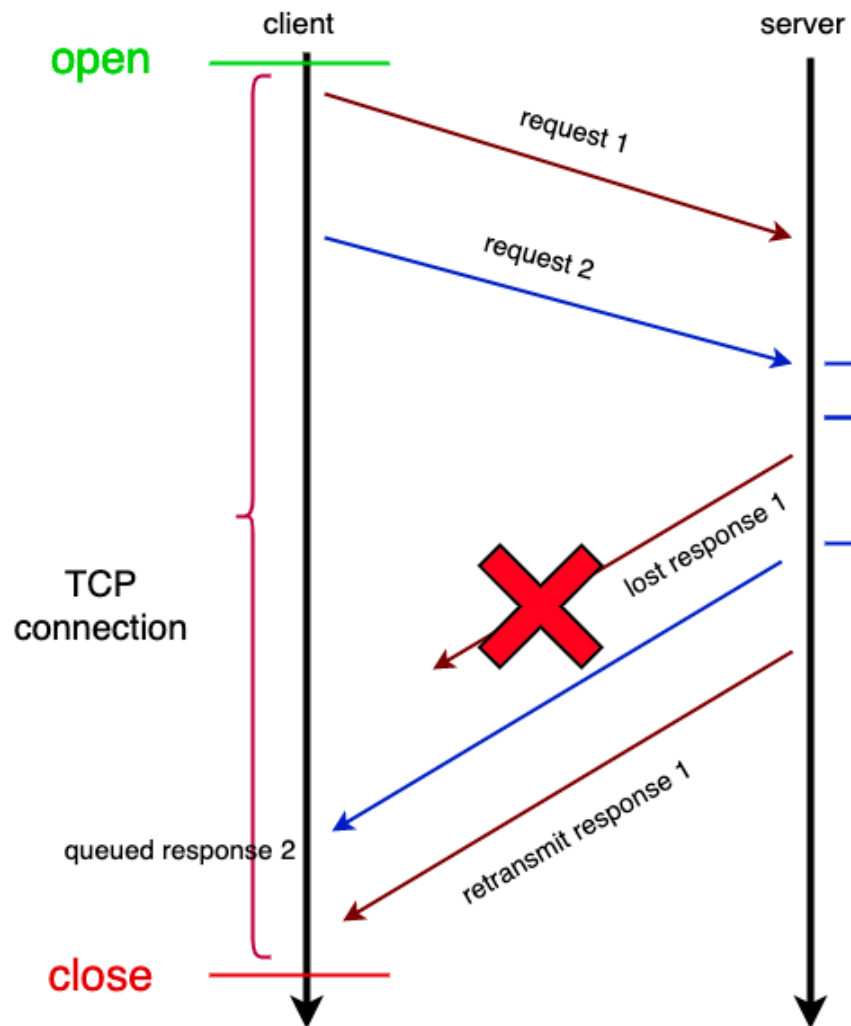




Problem solved?



# HTTP 2.0 fixes HoL at Layer 7, but not at Layer 4





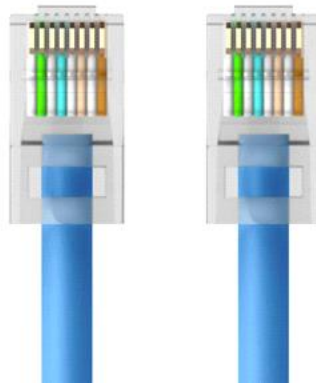


# Receive Buffer

HTTP Server

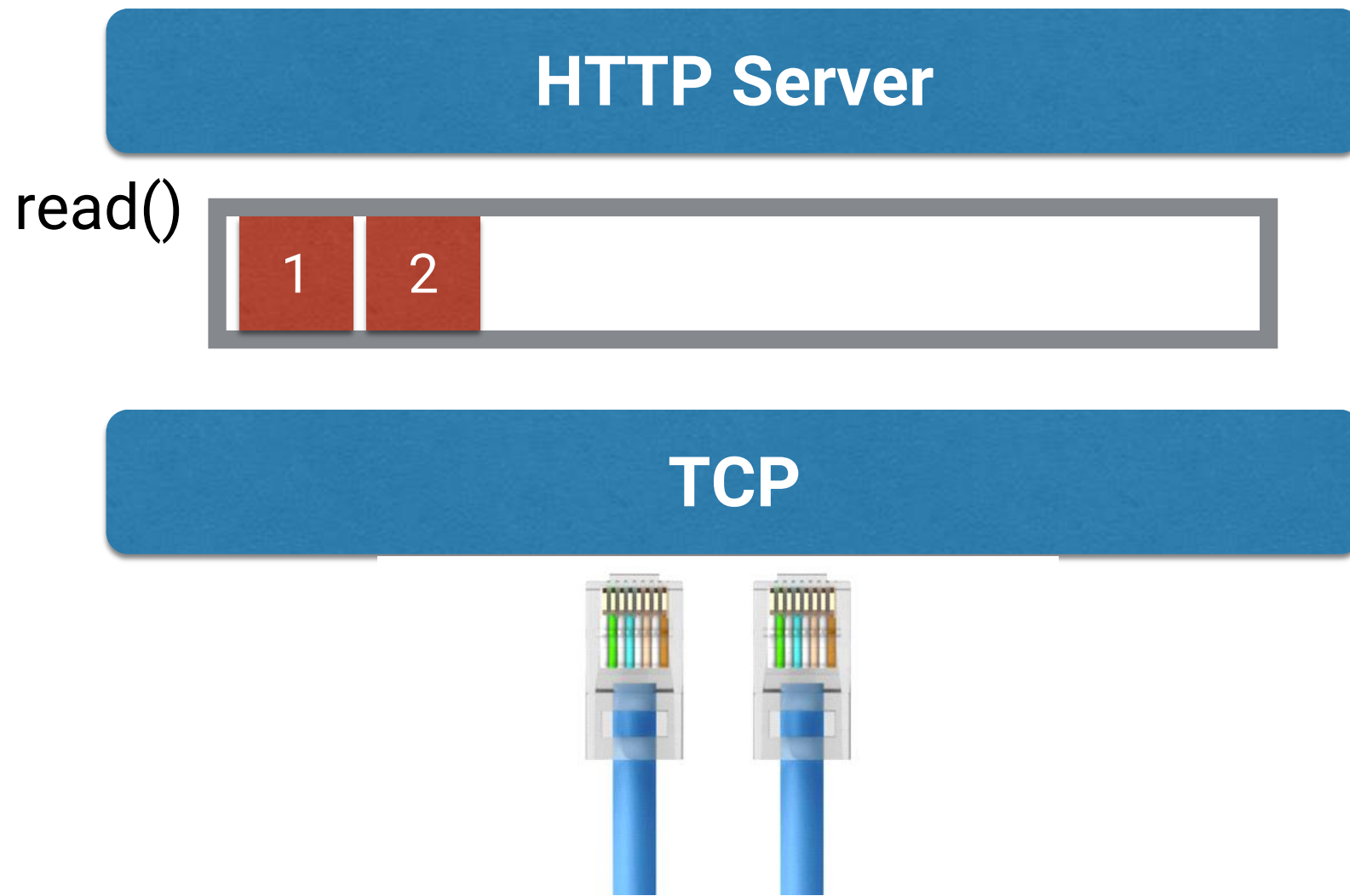


TCP



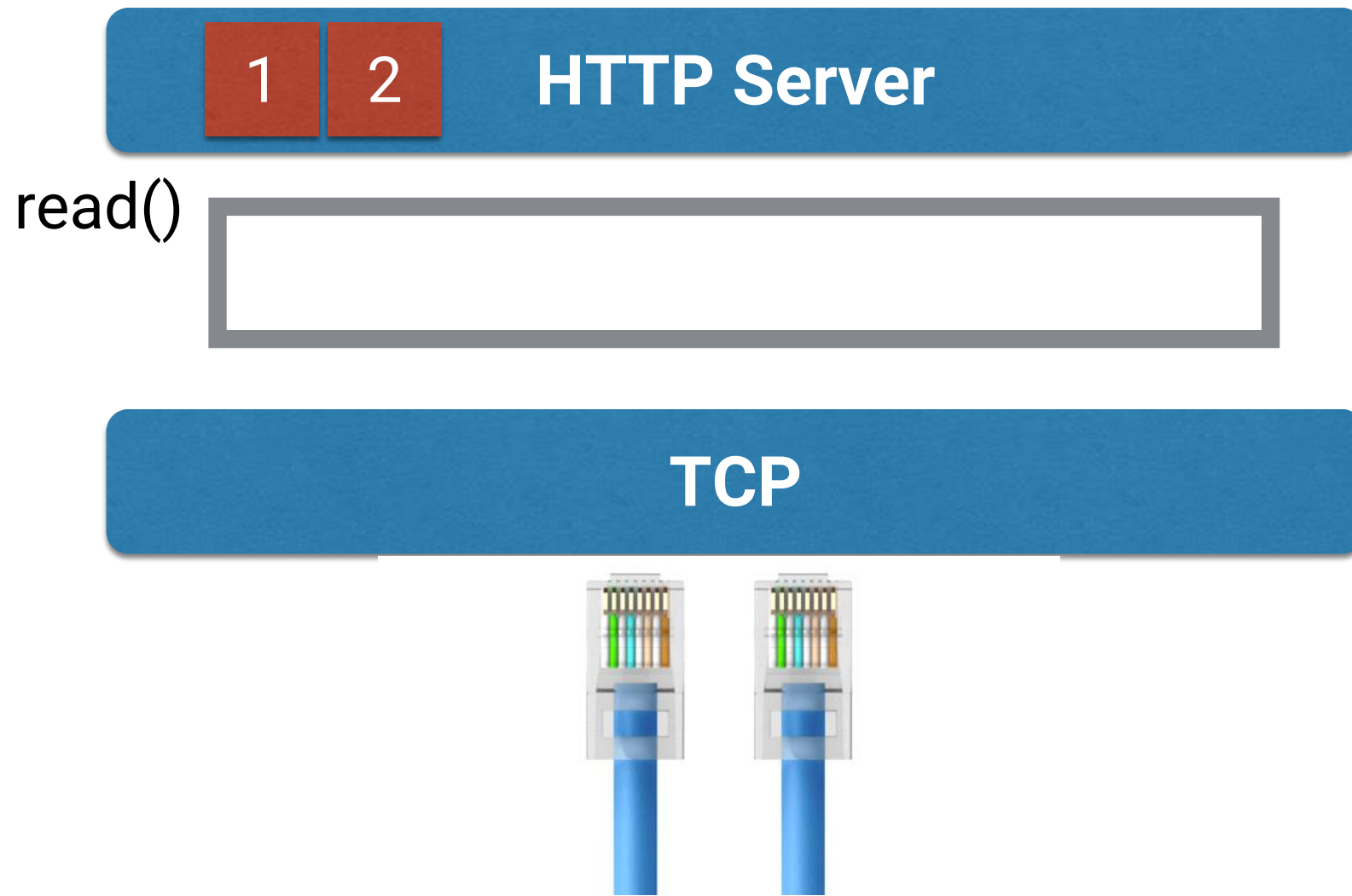


# Receive Buffer





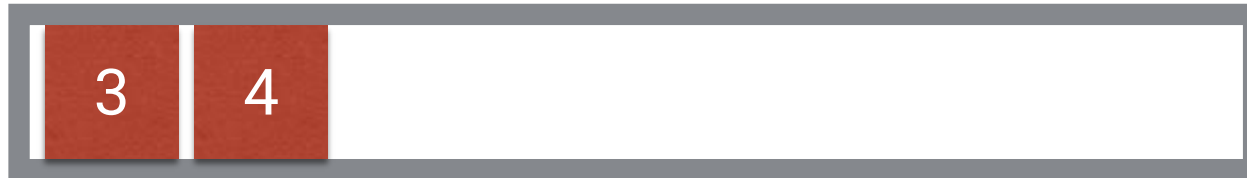
# Receive Buffer



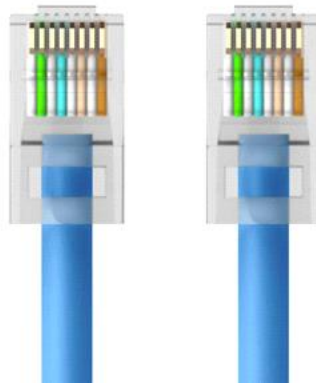


# Receive Buffer

HTTP Server



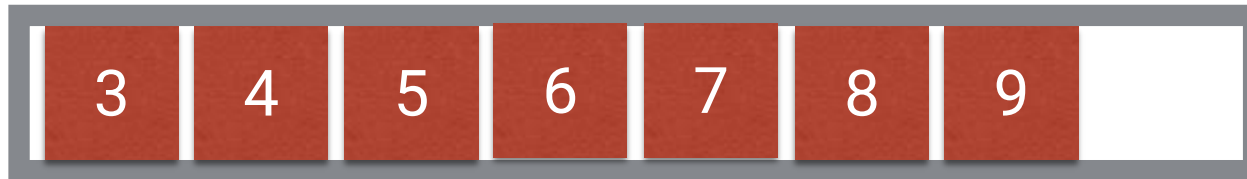
TCP



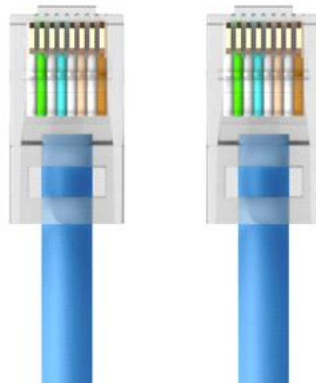


# Receive Buffer

HTTP Server



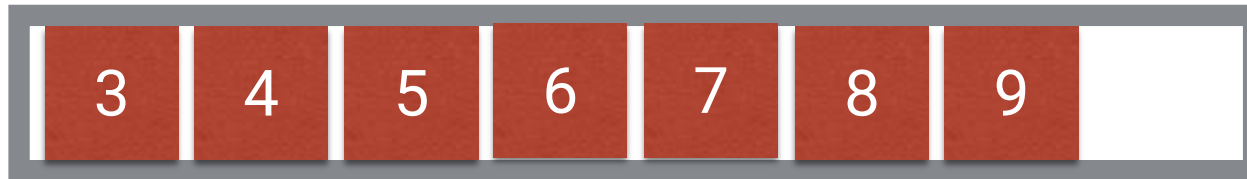
TCP





# Receive Buffer

HTTP Server

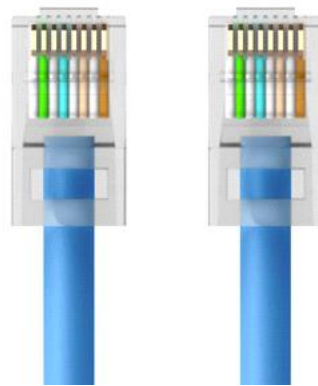


TCP

10

11

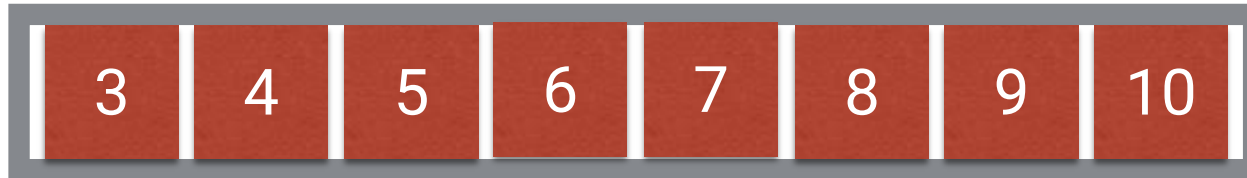
12



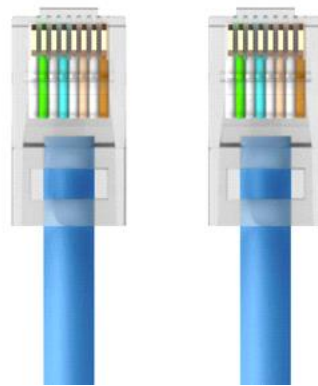


# Receive Buffer

HTTP Server

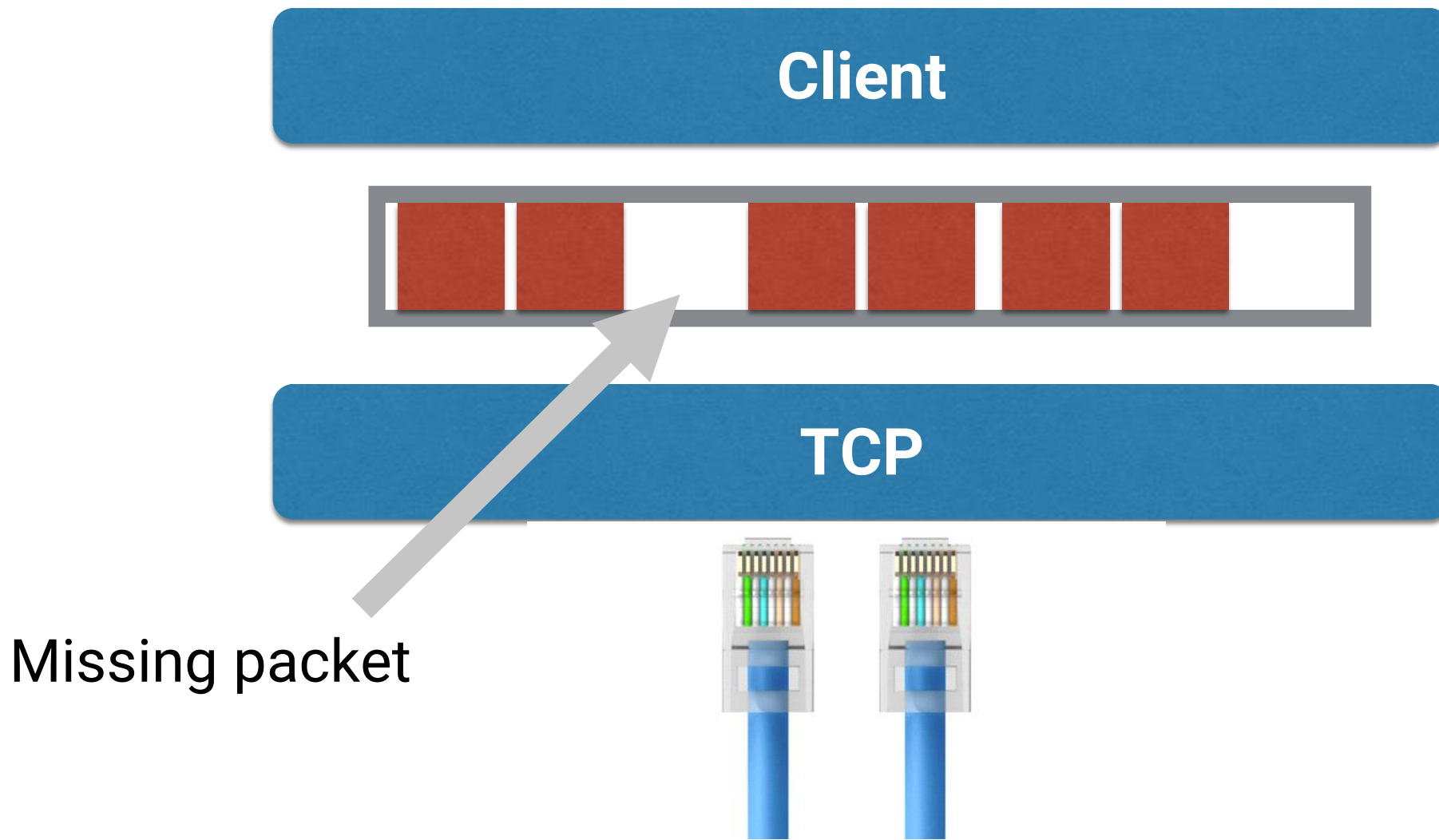


TCP





# Recall: Receive Buffer



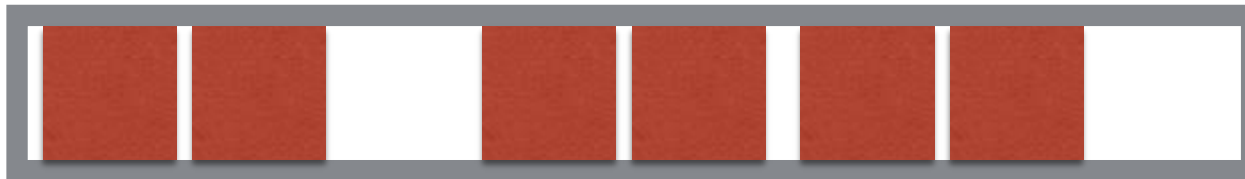




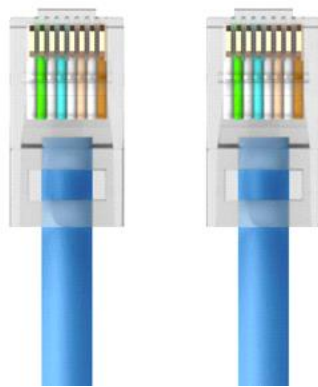
# Recall: Receive Buffer

read(...)

Client



TCP

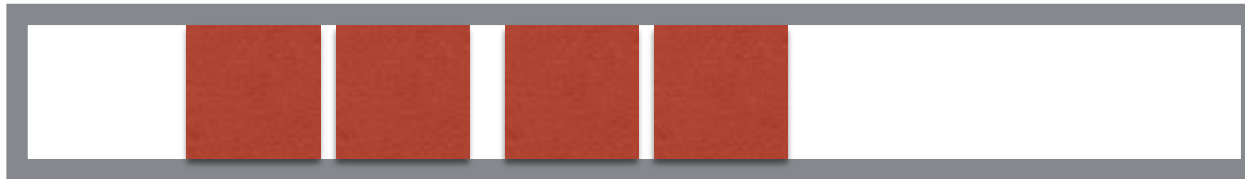




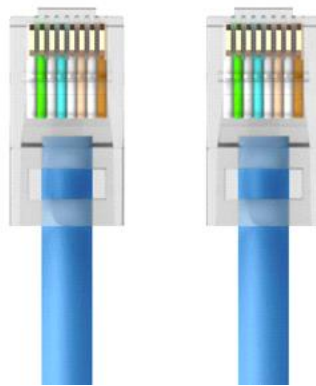
# Recall: Receive Buffer

read(...)

Client



TCP

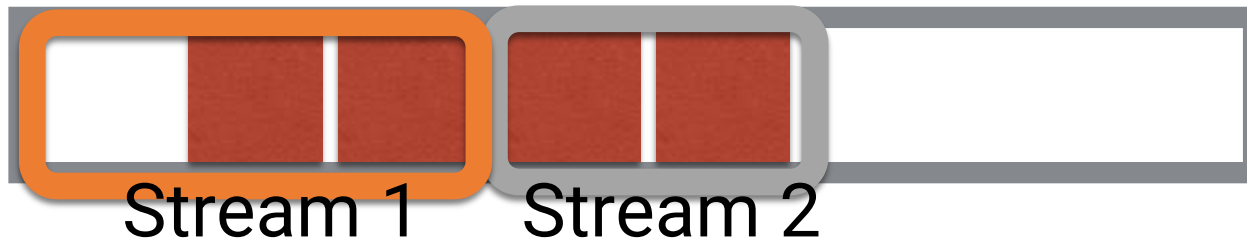




# Recall: Receive Buffer

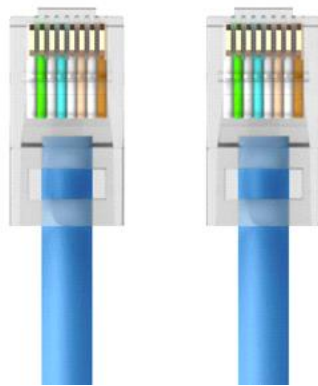
read(...)

Client



TCP

**Missing packet for Stream 1  
prevents client from reading  
in data for Stream 2!**





# HTTP 3.0: Coming

- Persistent connections + pipelining + re-orderable streams...
- Now all over UDP!
  - Allows application to pull in data for streams as soon as data for that stream is complete, even if data for other streams has been lost in the network.
- Quick UDP Internet Connections (QUIC)
  - Standardized in 2021
  - The QUIC Transport Protocol: Design and Internet-Scale Deployment
  - <https://dl.acm.org/doi/pdf/10.1145/3098822.3098842> [SIGCOMM'17]
  - <https://datatracker.ietf.org/doc/rfc9000/> IETF RFC 9000



# Recap: HTTP Evolution

- Persistent connections to avoid setup/teardown overheads
- Pipelining/windowing to keep lots of requests/responses “in-flight”
- Concurrent connections can help with re-ordering, but come with their own drawbacks
- Out-of-order processing solves head-of-line blocking and improves work conservation.



# Recap: HTTP Evolution

- The previous few slides are just a “highlights reel” of big ideas in the evolution of HTTP to make pages load faster.
- Why teach these ideas, and not other things in HTTP?
- Turns out that these concepts highlighted — pipelining, parallelism, out-of-order processing — all appear throughout computer systems.
- You might see versions of these ideas in this semester and other areas (e.g., databases).

# Recap: Study Cheat Sheet



# My Course Roadmap

HTTP Protocol

Cookies

Databases

JavaScript and Web Applications

Persistent Connections

Pipelining

Parallel Connections





# Tutorial

- The TA will go through the checkpoint 1 in the next tutorial.
- Tutorial T1      Mo 9:30AM - 10:20AM Rm 1034, LSK Bldg
- Tutorial T2      Mo 6:00PM - 6:50PM    Rm 6573, Lift 29-30
- <https://www.foggynetwork.com>