

The Web (part 2)

Lecture 7

<http://www.cs.rutgers.edu/~sn624/352-F24>

Srinivas Narayana

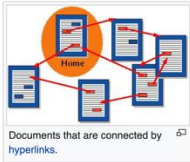
Review of concepts

Hypertext

From Wikipedia, the free encyclopedia

*For the concept in semiotics, see Hypertext (semiotics).
"Metatext" redirects here. For the literary concept, see Metafiction.*

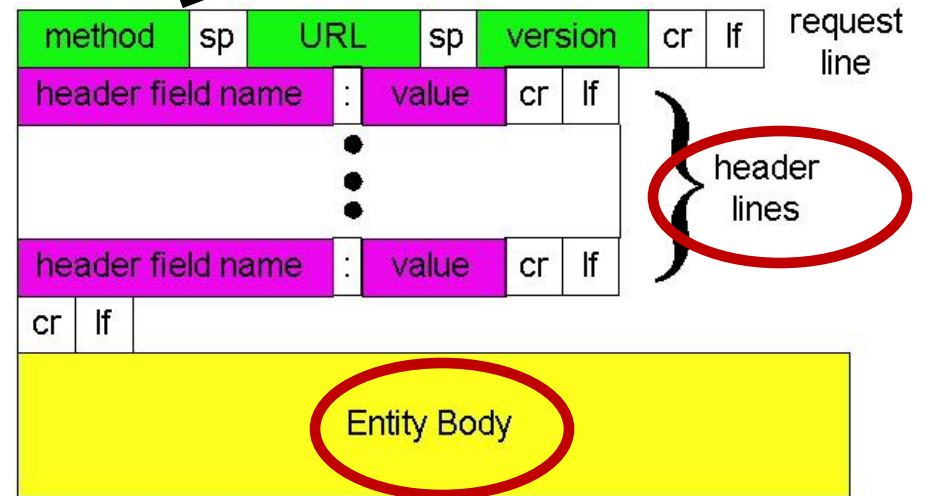
Hypertext is text displayed on a computer display or other electronic devices with references (hyperlinks) to other text that the reader can immediately access.^[1] Hypertext documents are interconnected by hyperlinks, which are typically activated by a mouse click, keypress set, or screen touch. Apart from text, the term "hypertext" is also sometimes used to describe tables, images, and other presentational content formats with integrated hyperlinks. Hypertext is one of the key underlying concepts of the World Wide Web,^[2] where Web pages are often written in the Hypertext Markup Language (HTML). As implemented on the Web, hypertext enables the easy-to-use publication of information over the Internet.



Contents [hide]

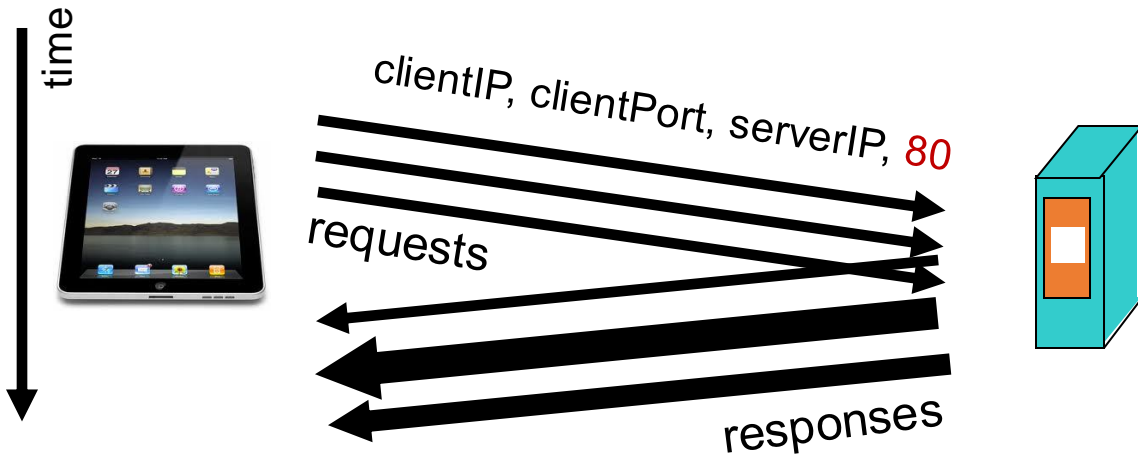
- 1 Etymology
- 2 Types and uses of hypertext
- 3 History
- 4 Implementations
- 5 Academic conferences

Request → GET, POST, ...

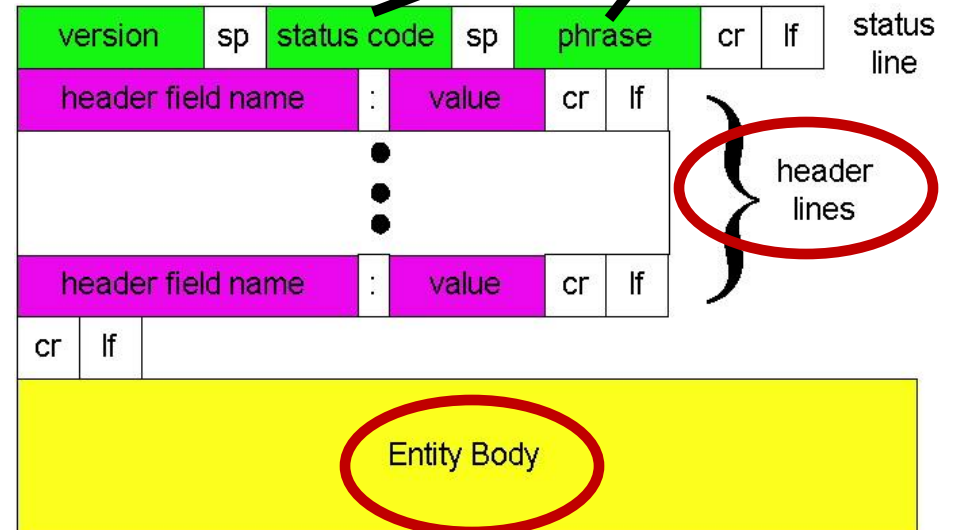


HyperText Transfer Protocol (HTTP)

Client-Server Protocol



Response → 200 OK, 301 Moved, etc.



HTTP Persistence

Two types of HTTP connectivity

Non-persistent HTTP

- At most one object is sent over a TCP connection.
- HTTP/1.0 uses non-persistent connections

Persistent HTTP

- Multiple objects can be sent over single TCP connection between client and server.
- HTTP/1.1 uses persistent connections in default mode

TCP is a reliable communication protocol provided by the transport layer. It requires setting up some resources (e.g., memory regions) for the connection to be set up at the endpoints before data communication.

Non-persistent HTTP (HTTP/1.0)



Web Server

1a. HTTP client initiates TCP connection to HTTP server

1b. HTTP server at host "accepts" connection, notifying client

2. HTTP client sends HTTP request message

3. HTTP server receives request message, replies with response message containing requested object

time

Suppose a user visits a page with text and 10 embedded images.

Non-persistent HTTP (HTTP/1.0)



Web Server

time
↓

5. HTTP client receives response message containing HTML file, displays HTML. Parsing HTML file, finds 10 referenced image objects

6. Steps 1-5 repeated for each of 10 image objects

4. HTTP server closes TCP connection.

Single connection per object

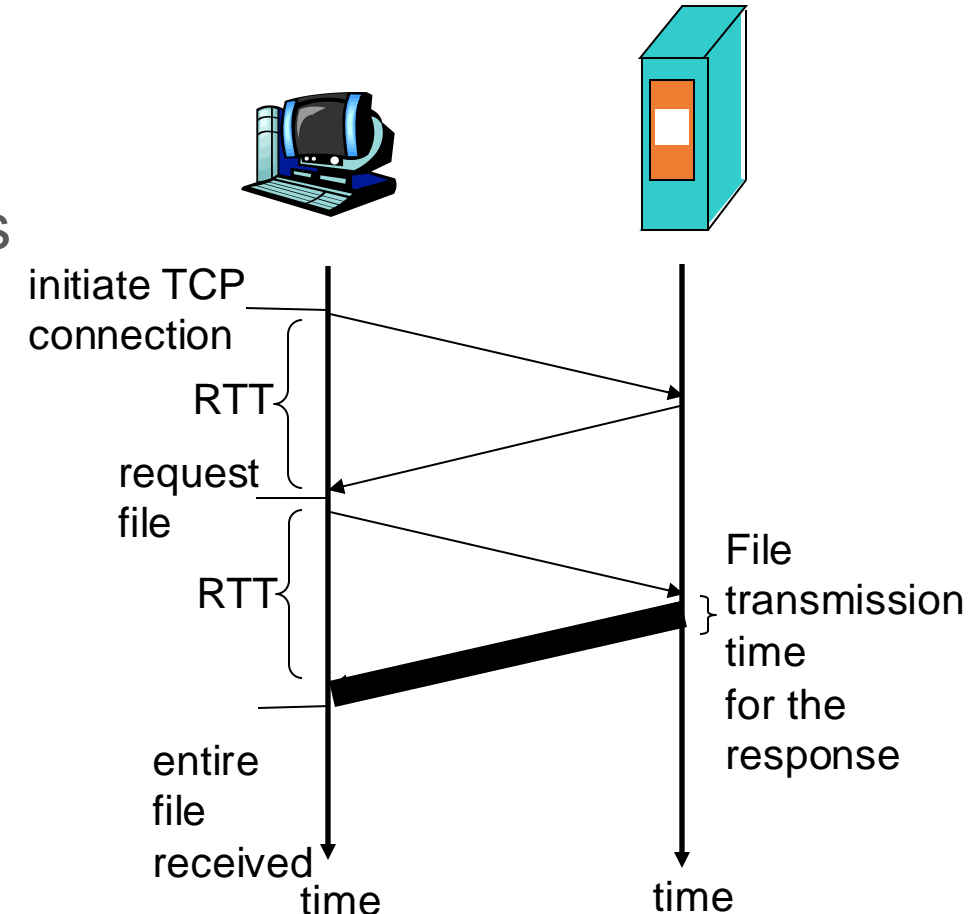
Useful at a time when web pages contained 1 object: the base HTML file.

How long does it take to download
an entire web page with non-
persistent HTTP?

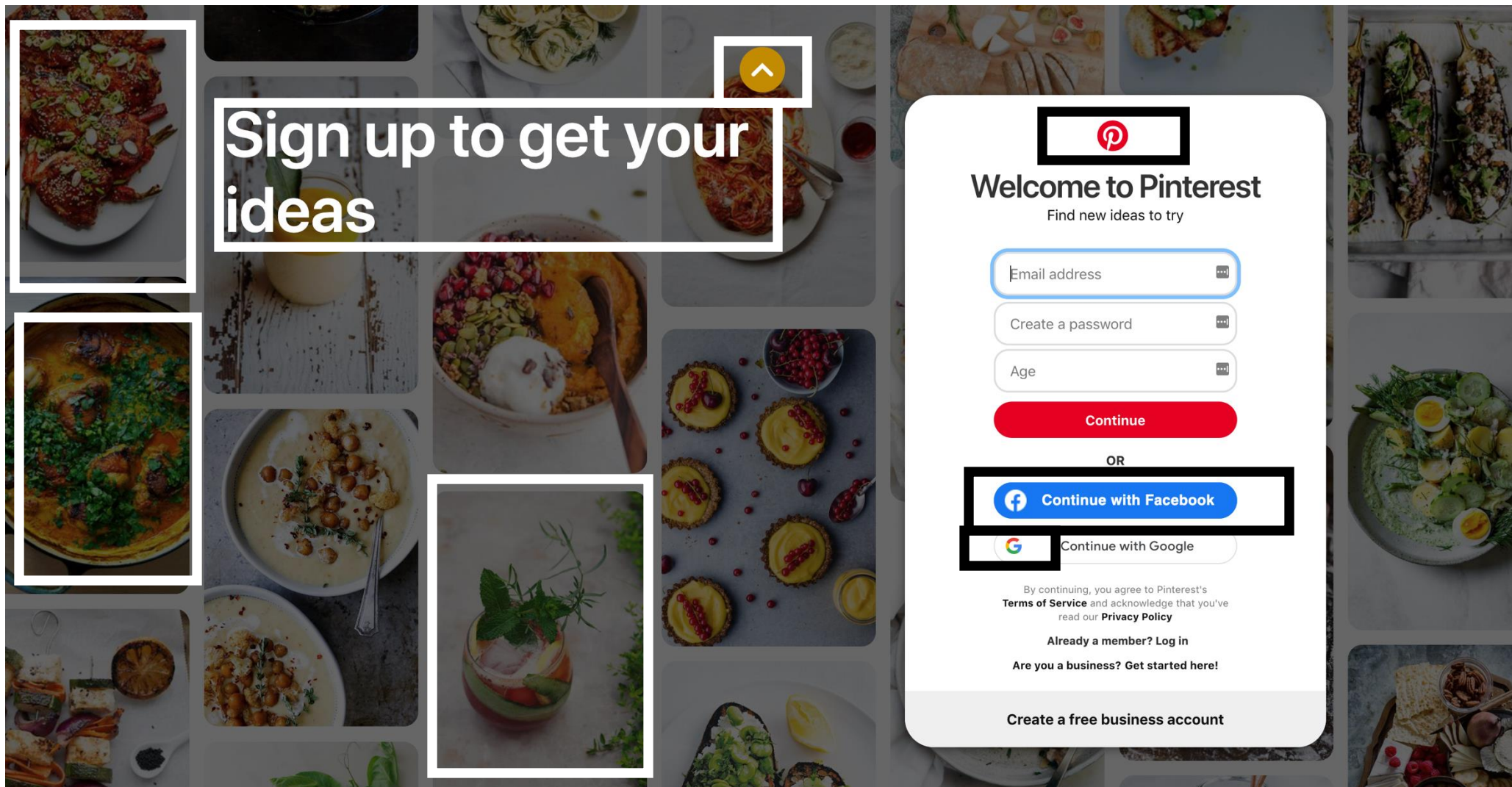
i.e.: before your browser can load
the (entire) web page?

Non-persistent HTTP user response time

- Total delay = propagation + queueing + transmission
- Response time for the user
 - = sum of forward and backward total delays
- **Round-Trip Time (RTT)**: total forward + backward delay for a “small” packet
 - Zero transmission delay
- Assumptions:
 - TCP initiation packet, response, HTTP requests are all “small” packets
 - No processing delays at the server
 - RTT is stable over time
- **$(2RTT + \text{file transmission time}) * \#objects$**



Per-object overheads quickly add up



The image shows a Pinterest sign-up page overlaid on a background of various food images. Several elements are highlighted with white boxes:

- A white box containing the text "Sign up to get your ideas" is positioned in the upper left quadrant.
- A white box with a yellow upward-pointing arrow is located in the upper right quadrant.
- A white box highlights the Pinterest logo at the top of the sign-up form.
- A white box highlights the "Continue with Facebook" button.
- A white box highlights the "Continue with Google" button.

The sign-up form itself includes the following elements:

- The Pinterest logo.
- The text "Welcome to Pinterest" and "Find new ideas to try".
- Input fields for "Email address", "Create a password", and "Age".
- A red "Continue" button.
- The word "OR" centered below the "Continue" button.
- Buttons for "Continue with Facebook" and "Continue with Google".
- Small text: "By continuing, you agree to Pinterest's Terms of Service and acknowledge that you've read our Privacy Policy".
- Links: "Already a member? Log in" and "Are you a business? Get started here!".
- A link at the bottom: "Create a free business account".

Modern web pages have 100s of objects in them.

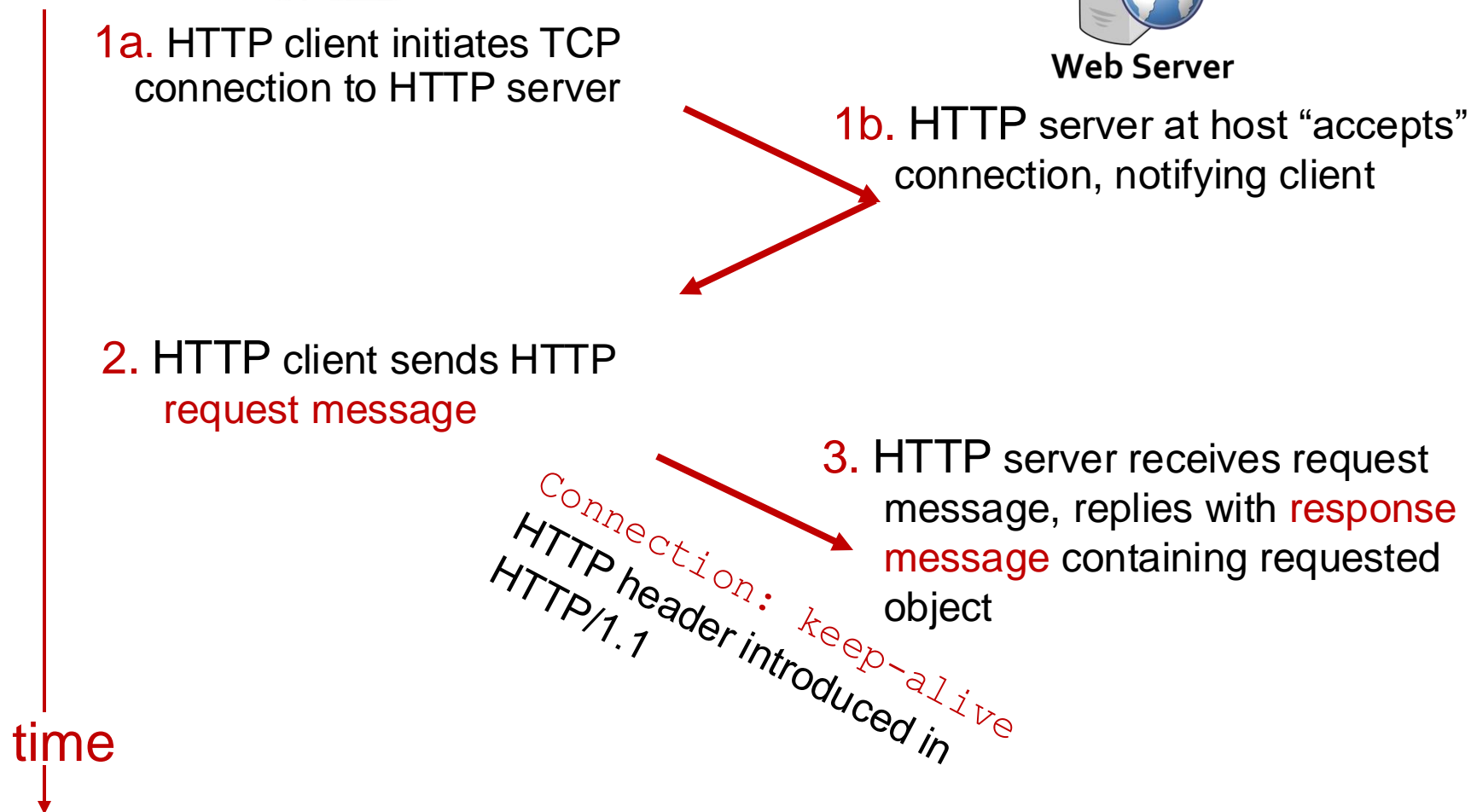
Objects (e.g. images) may not be small.

Persistent HTTP (HTTP/1.1)



Web Server

Suppose user visits a page with text and 10 images.



Persistent HTTP (HTTP/1.1)



Web Server

4. HTTP server sends a response.

Server keeps the TCP connection alive.

5. HTTP client receives response message containing HTML file, displays HTML. Parsing HTML file, finds 10 referenced image objects

The 10 objects can be requested over the **same** TCP connection.

i.e., save an RTT per object (otherwise spent opening a new TCP connection in HTTP/1.0)



Persistent HTTP user response time

- Assume requests made one at a time (separate RTT per req)
- $RTT + (RTT + \text{file transmission time}) * \#objects$
- **Pipelining**: send more than one HTTP request at a time
 - Extreme case: all requests in one (small) packet
 - $2RTT + (\text{file transmission time}) * \#objects$
 - In practice, dependencies between objects
- Compare with non-persistent:
 - $(2RTT + \text{file transmission time}) * \#objects$
- Persistence (& pipelining) can save significant time, especially on high-RTT connections
- Other advantages of persistence: CPU savings, reduced network congestion, less memory (fewer connections)

Persistence vs. # of connections

- Persistence is distinct from the **number of concurrent connections** made by a client
- Your browser has the choice to open multiple connections to a server
 - HTTP spec suggests to limit this to a small number (2)
- Further, a single connection can have multiple HTTP requests in flight (pipelining) with persistent HTTP

Clients that use persistent connections SHOULD limit the number of simultaneous connections that they maintain to a given server. A single-user client SHOULD NOT maintain more than 2 connections with any server or proxy. A proxy SHOULD use up to 2*N connections to another server or proxy, where N is the number of simultaneously active users. These guidelines are intended to improve HTTP response times and avoid congestion.

Remembering Users On the Web

HTTP: Remembering users

So far, HTTP mechanisms considered **stateless**

- Each request processed independently at the server
- The server maintains no memory about past client requests

However, **state**, i.e., memory, about the user at the server, is very useful!

- User authentication (e.g., gmail)
- Shopping carts (e.g., Amazon)
- Video recommendations (e.g., Netflix)
- Any user session state in general

Familiar with these?

Your Privacy

We use cookies to make sure that our website works properly, as well as some 'optional' cookies to personalise content and advertising, provide social media features and analyse how people use our site. By accepting some or all optional cookies you give consent to the processing of your personal data, including transfer to third parties, some in countries outside of the European Economic Area that do not offer the same data protection standards as the country where you live. You can decide which optional cookies to accept by clicking on 'Manage Settings', where you can also find more information about how your personal data is processed. Further information can be found in our [privacy policy](#).

Accept all cookies

[Manage preferences](#)

This website uses cookies

We use cookies to personalise content and ads, to provide social media features and to analyse our traffic. We also share information about your use of our site with our social media, advertising and analytics partners who may combine it with other information that you've provided to them or that they've collected from your use of their services

Use necessary cookies only

Allow selection

Allow all cookies

Necessary Preferences Statistics Marketing

Show details 

Cookies: Keeping user memory



Cookie is typically opaque to client.

