The Web (HTTP)

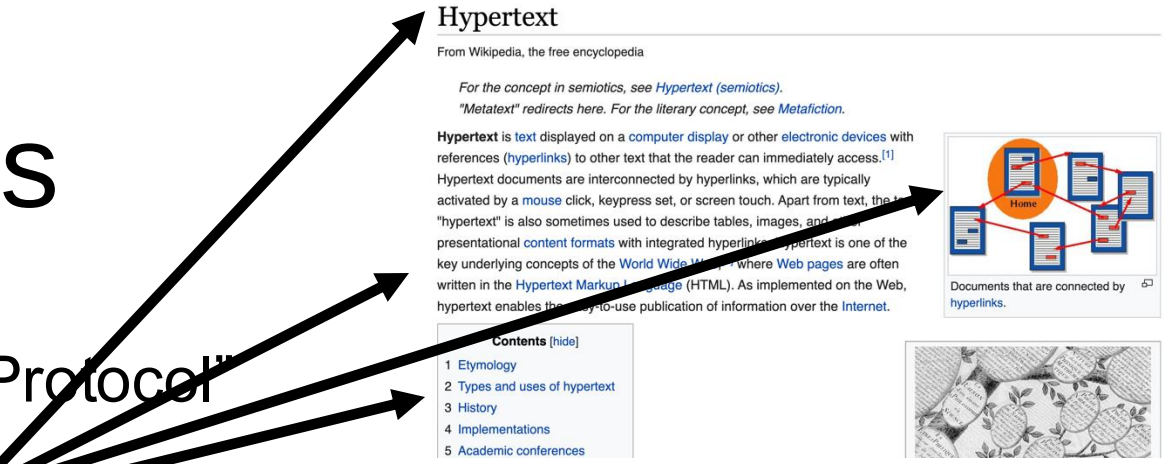
Lecture 6

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

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Web and HTTP: Terms


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 ... "hypertext" is also sometimes used to describe tables, images, and ... presentational content formats with integrated hyperlinks. ... ypertext is one of the key underlying concepts of the World Wide ... where Web pages are often written in the Hypertext Markup ... guage (HTML). As implemented on the Web, hypertext enables the ... y-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

Home

Documents that are connected by hyperlinks.

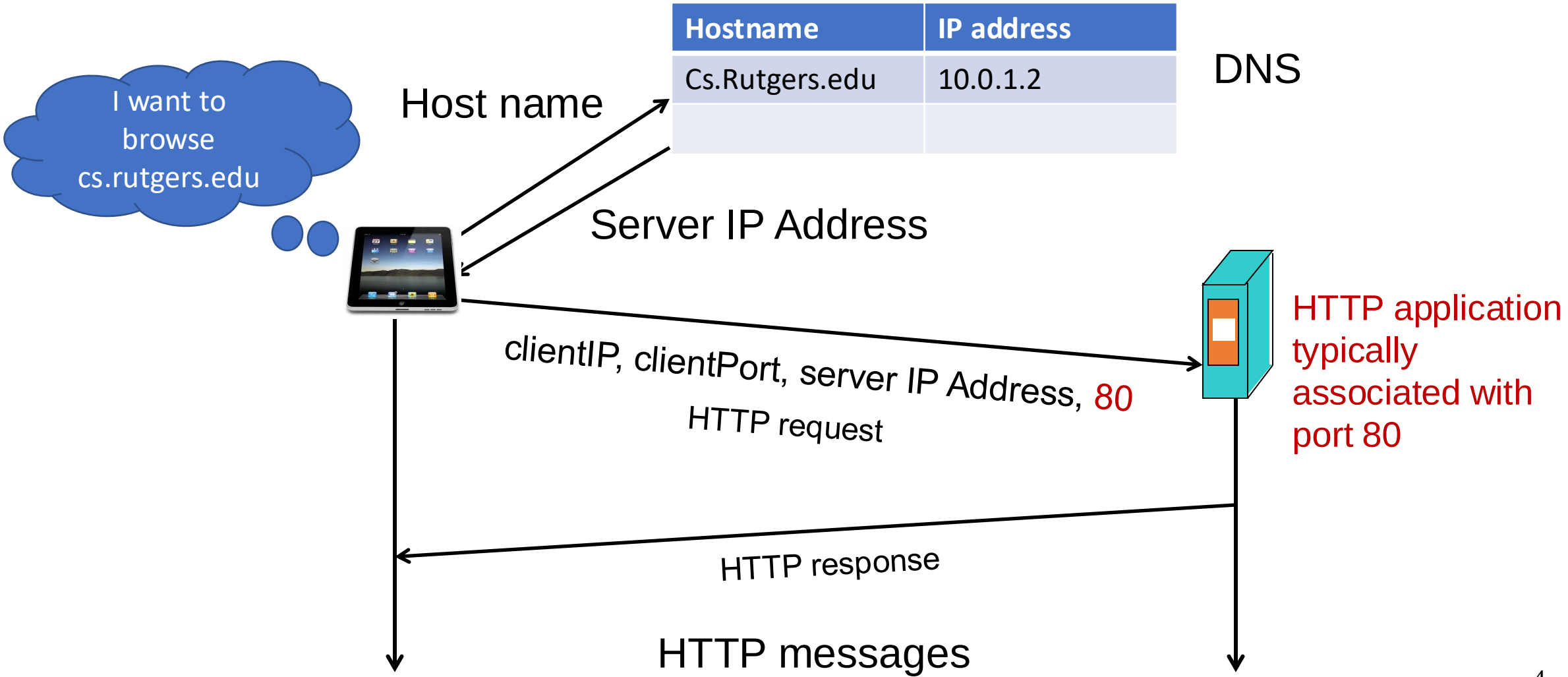
- HTTP stands for “HyperText Transfer Protocol”
- A web page consists of many objects
- Object can be HTML file, JPEG image, video stream chunk, audio file,…
- Web page consists of base HTML-file which embeds several objects
- Each object is addressable by a uniform resource locator (URL)
  - sometimes also referred to as uniform resource identifier (URI)
- Example URL:

`www.cs.rutgers.edu/~sn624/index.html`

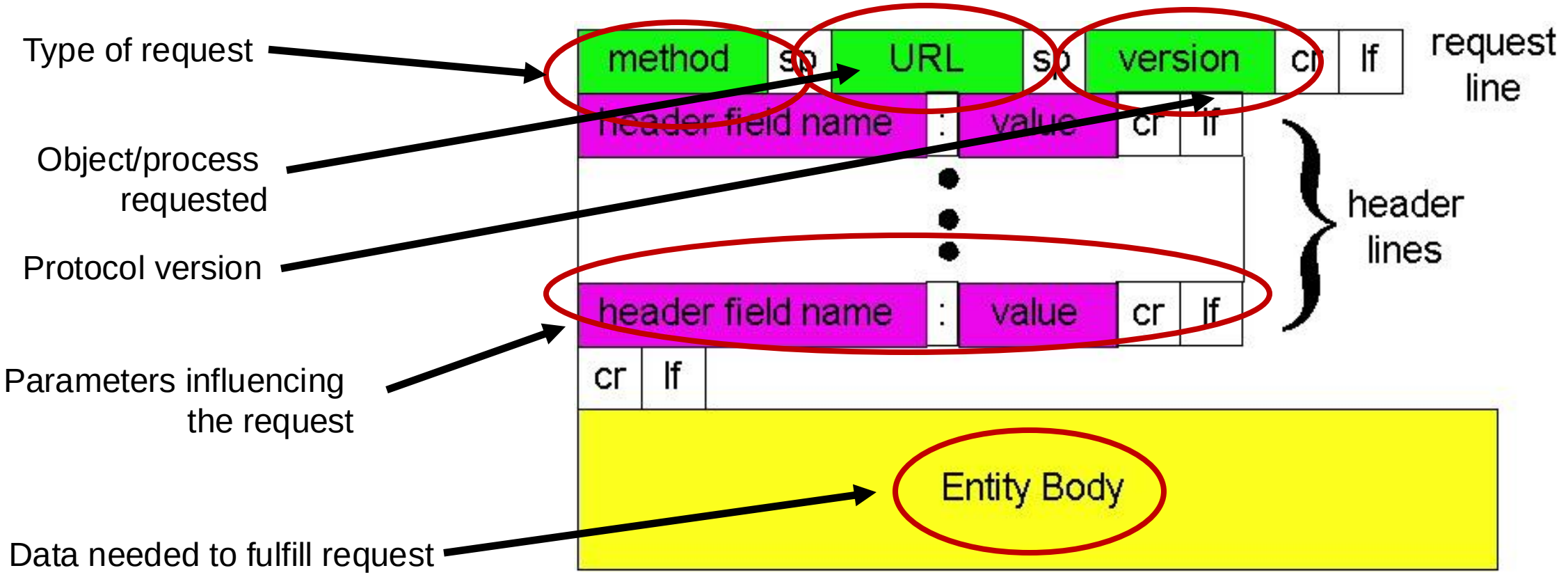
Domain/host name | path

HTTP Protocol

Client server protocol



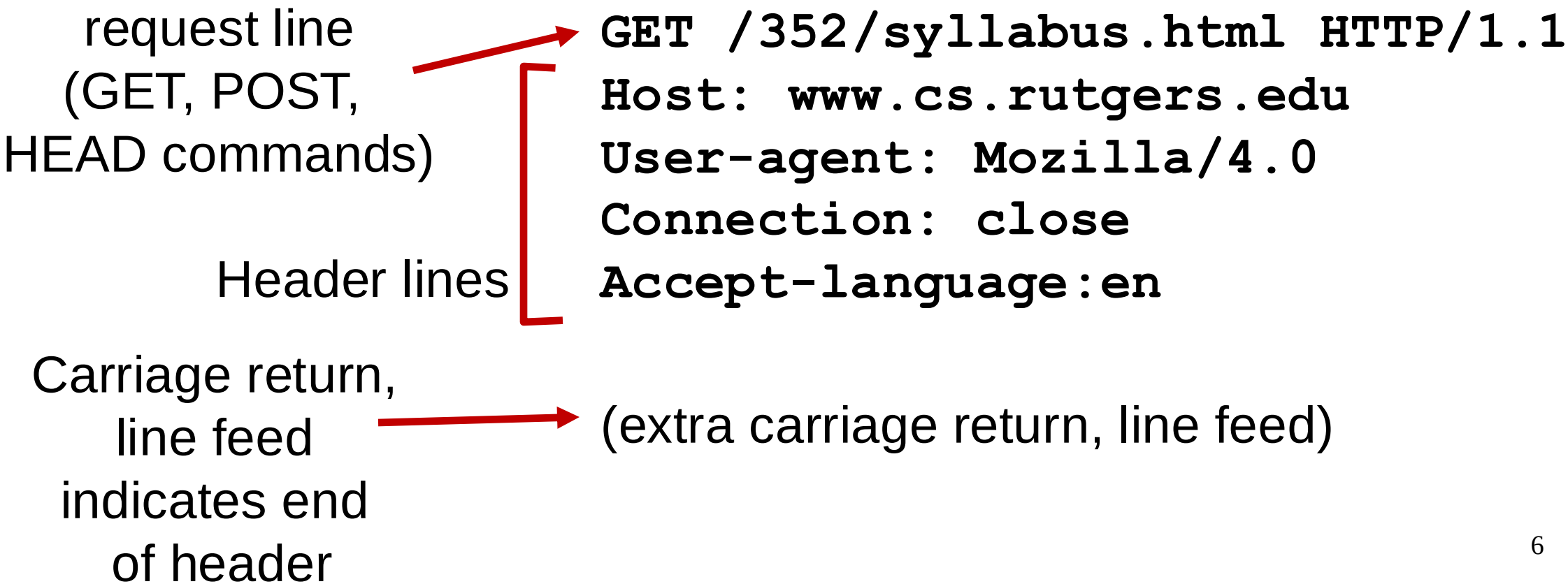
HTTP Request: Message Format



http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14

HTTP messages: request message

- ASCII (human-readable format)



The URL

- Universal Resource Locator: a way to name objects on server
- But can also name an application process on the server!
- Examples:
  - Data storage from data entered in web forms
  - Login pages
  - Web carts
- Providing almost any service requires data handling by running code at the server
  - Not just rendering “static” resources

HTTP method types

- GET
  - Get the resource specified in the requested URL (could be a process)
- POST
  - Send entities (specified in the entity body) to a data-handling process at the requested URL
- HEAD
  - Asks server to leave requested object out of response, but send the rest of the response
  - Useful for debugging
- PUT
  - Update a resource at the requested URL with the new entity specified in the entity body
- DELETE
  - Deletes file specified in the URL
- and other methods

https://httpwg.org/specs/rfc9110.html#method.definitions

Uploading form input: GET and POST

POST method:

- Web page often includes form input
- Input is uploaded to server in entity body

- Posted content not visible in the URL
  - Free form content (ex: images) can be posted since entity body interpreted as data bytes

GET method:

- Entity body is empty
- Input is uploaded in URL field of request line

- URL must contain a restricted set of characters

- Example:
  - http://site.com/form?first=jane&last=austen

Difference between POST and PUT

- POST: the URL of the request identifies the resource that processes the entity body
- PUT: the URL of the request identifies the resource that is contained in the entity body

https://tools.ietf.org/html/rfc2616

Difference between HEAD and GET

- GET: return the requested resource in the entity body of the response along with response headers (we'll see these shortly)
- HEAD: return all the response headers in the GET response, but without the resource in the entity body

https://tools.ietf.org/html/rfc2616

Observing HTTP GET and POST

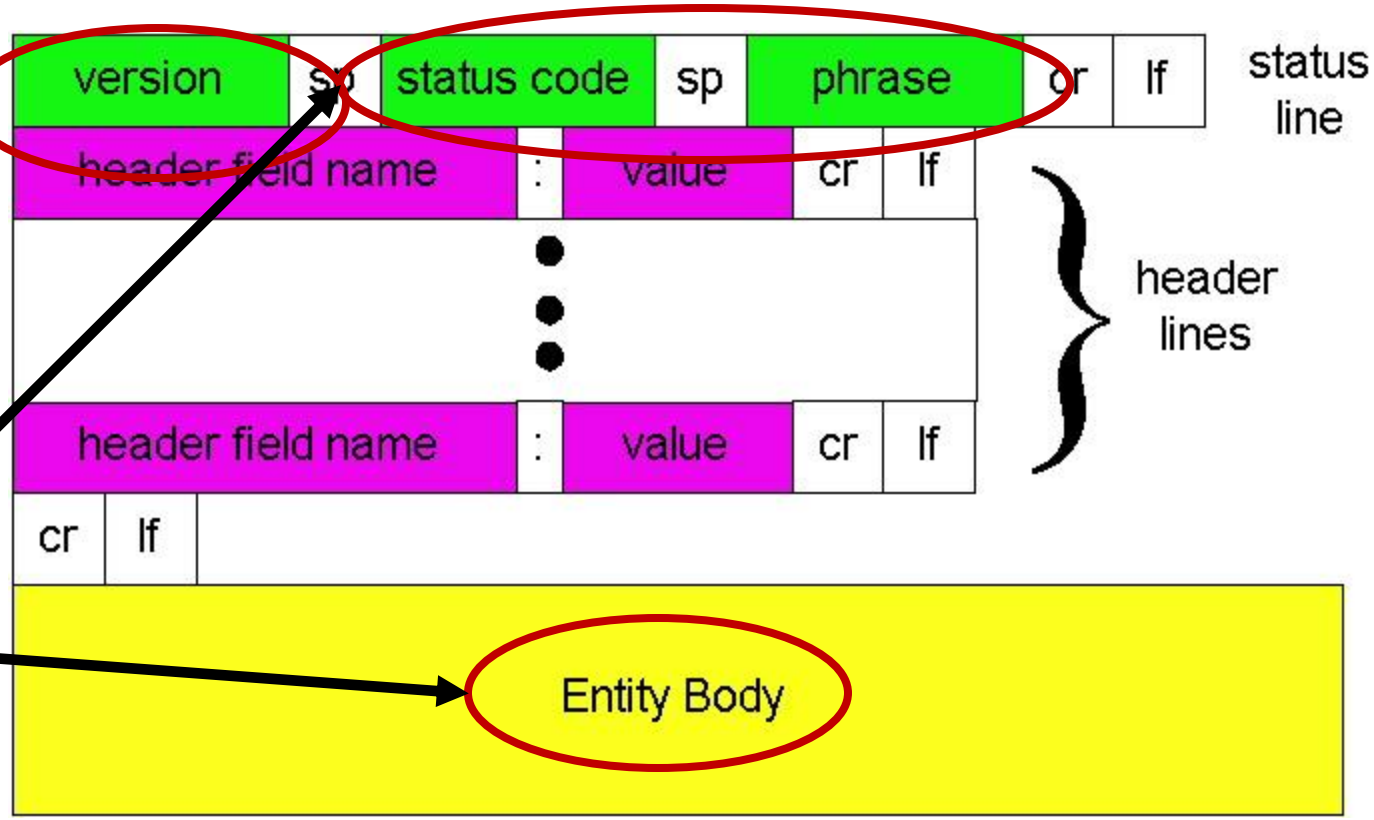
HTTP Response: General format

Unlike HTTP request,
No method name

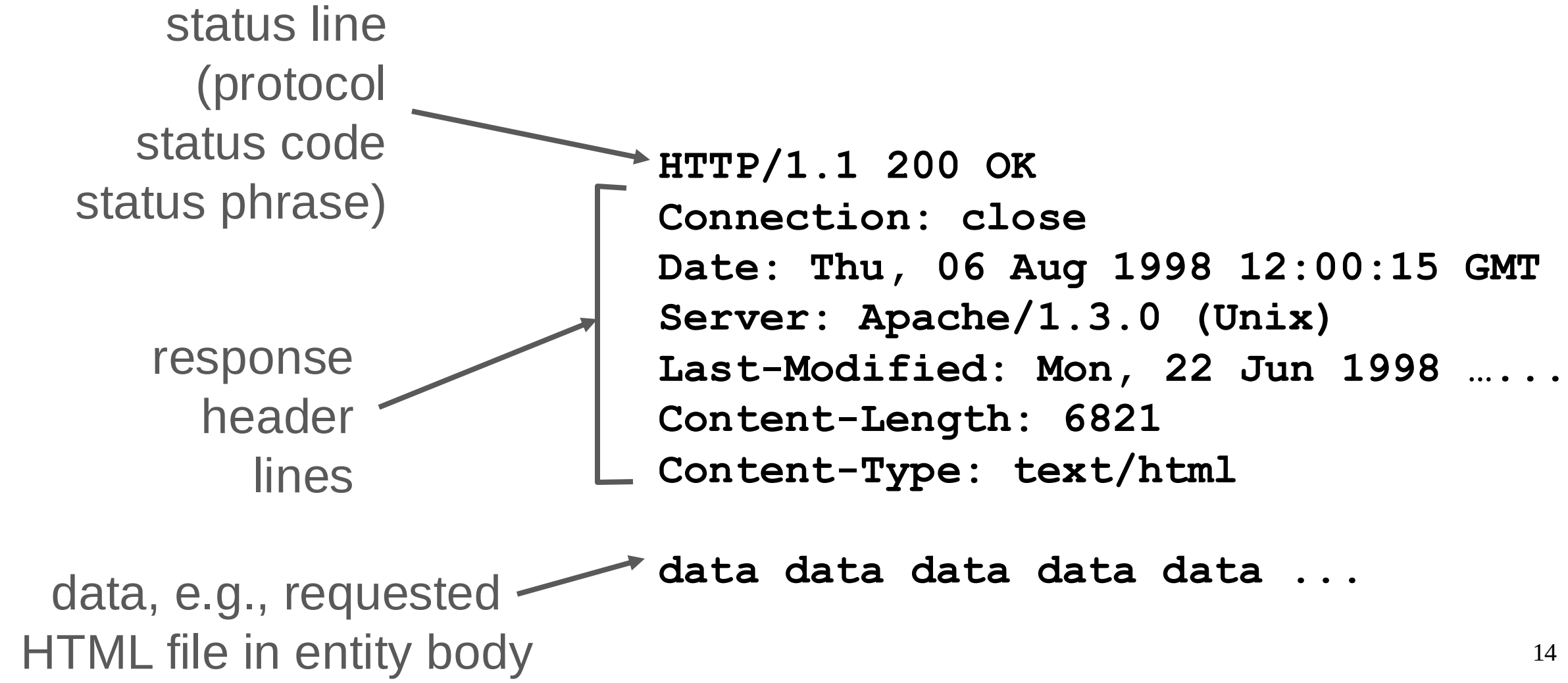
HTTP protocol version
used by server

Was request successful?
(or error condition)

Returned object data



HTTP message: response message



HTTP response status codes

In first line in server->client response message.

A few sample codes:

200 OK

- request succeeded, requested object later in this message

301 Moved Permanently

- requested object moved, new location specified later in this message (Location:)

403 Forbidden

- Insufficient permissions to access the resource

404 Not Found

- requested document not found on this server

505 HTTP Version Not Supported

Observing HTTP behaviors

- `wget google.com (or) curl google.com`

- `telnet example.com 80`
  - `GET / HTTP/1.1`
  - `Host: example.com`

(followed by two enter's)

- Exercise: try
  - `telnet google.com 80`
  - `telnet web.mit.edu 80`