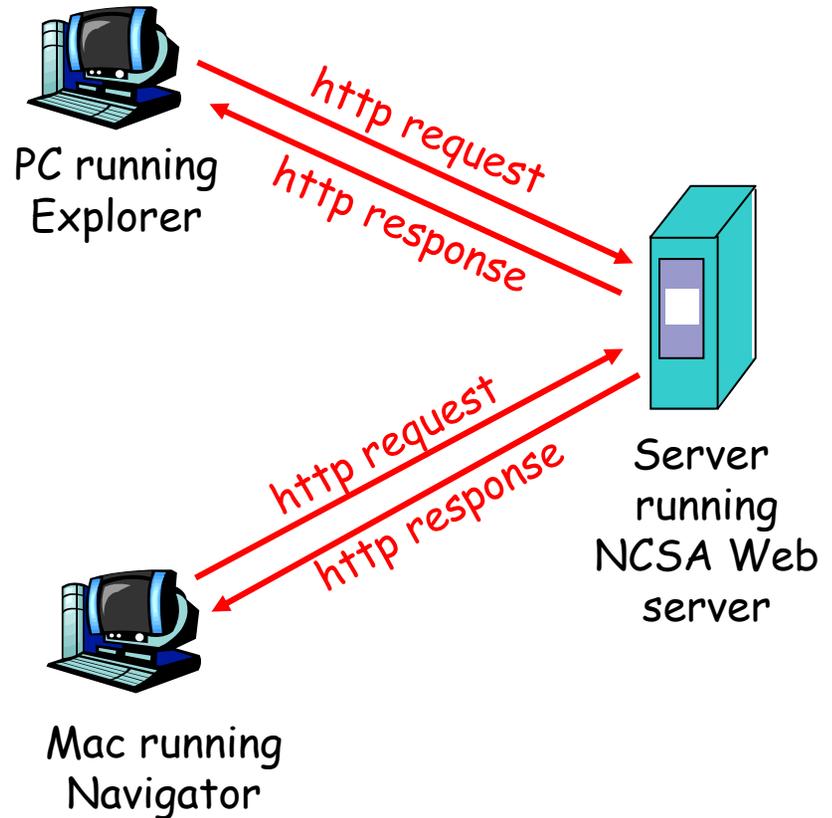


HTTP – HyperText Transfer Protocol

The Web: the http protocol

http: hypertext transfer protocol

- Web's application layer protocol
- client/server model
 - *client*: browser that requests, receives, "displays" Web objects
 - *server*: Web server sends objects in response to requests
- http1.0: RFC 1945
- http1.1: RFC 2068



Universal Resource Locator

protocol://host:port/path#anchor?parameters

http://www.google.com/search?hl=en&g=blabla

- There are other types of URL's
 - mailto:<account@site>
 - news:<newsgroup-name>

The http protocol: more

http: TCP transport service:

- client initiates TCP connection (creates socket to server, port 80)
- server accepts TCP connection from client
- http messages (application-layer protocol messages) exchanged between browser (http client) and Web server (http server)
- TCP connection closed

http is "stateless"

- server maintains no information about past client requests

Protocols that maintain "state" are complex!

- past history (state) must be maintained
- if server/client crashes, their views of "state" may be inconsistent, must be reconciled

Persistent vs. Non-Persistent Connection

- A page that we see on the browser can include more than one resource
- The resources are sent from the server to the client one after the other
- Sending the resources to the browser can be by using a persistent connection or by using a non-persistent connection

Non-Persistent Connection

- 
1. Browser opens TCP connection to port 80 of server (handshake)
 2. Browser sends http request message
 3. Server receives request, locates object, sends response
 4. Server closes TCP connection
 5. Browser receives response, parses object
 6. Browser repeats steps 1-5 for each embedded object

Persistent Connection

1. Browser opens TCP connection to port 80 of server (handshake)

2. Browser sends http request message

3. Server receives request, locates object, sends response

4. Browser receives response, parses object

5. Browser repeats steps 2-4 for each embedded object

6. TCP connection closes on demand or timeout

Advantages of Persistent Connection

- **CPU time** saved in routers and hosts
- HTTP requests and responses can be **pipelined** on a connection
- **Network congestion** is reduced
- **Latency** on subsequent requests is reduced

What are the disadvantages of persistent connection?

Pipelines

- 2 types of persistent connections
 - without pipelining
 - the client issues a new request only after the previous response has arrived
 - with pipelining
 - client sends the request as soon as it encounters a reference
 - multiple requests/responses

http message format: request

- two types of http messages: *request, response*
- **http request message:**

- ASCII (human-readable format)

request line
(GET, POST,
HEAD
commands)

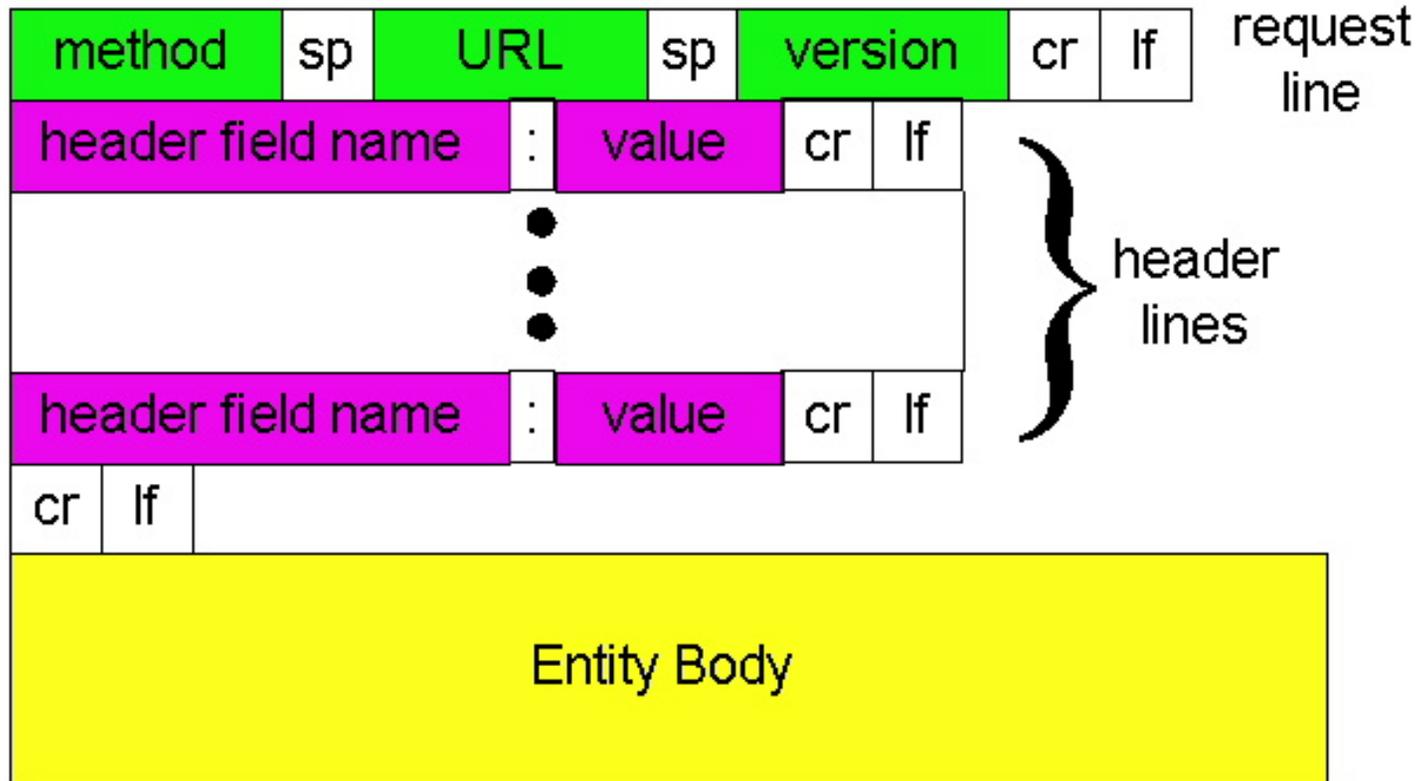
header
lines

```
GET /somedir/page.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Accept-language: fr
```

Carriage return,
line feed
indicates end
of message

(extra carriage return, line feed)

http request message: general format



http request message: more info

- http/1.0 has only three request *methods*
 - **GET:**
 - **POST:** for forms. Uses *Entity Body* to transfer form info
 - **HEAD:** Like *GET* but response does not actually return any info. This is used for debugging/test purposes
- http/1.1 has two additional request *methods*
 - **PUT:** Allows uploading object to web server
 - **DELETE:** Allows deleting object from web server

Trying out http (client side) for yourself

1. Telnet to your favorite Web server:

```
telnet cis.poly.edu 80
```

Opens TCP connection to port 80 (default http server port) at cis.poly.edu. Anything typed in sent to port 80 at cis.poly.edu.

2. Type in a GET http request:

```
GET /~ross/index.html HTTP/1.0
```

By typing this in (hit carriage return twice), you send this minimal (but complete) GET request to http server

3. Look at response message sent by http server!

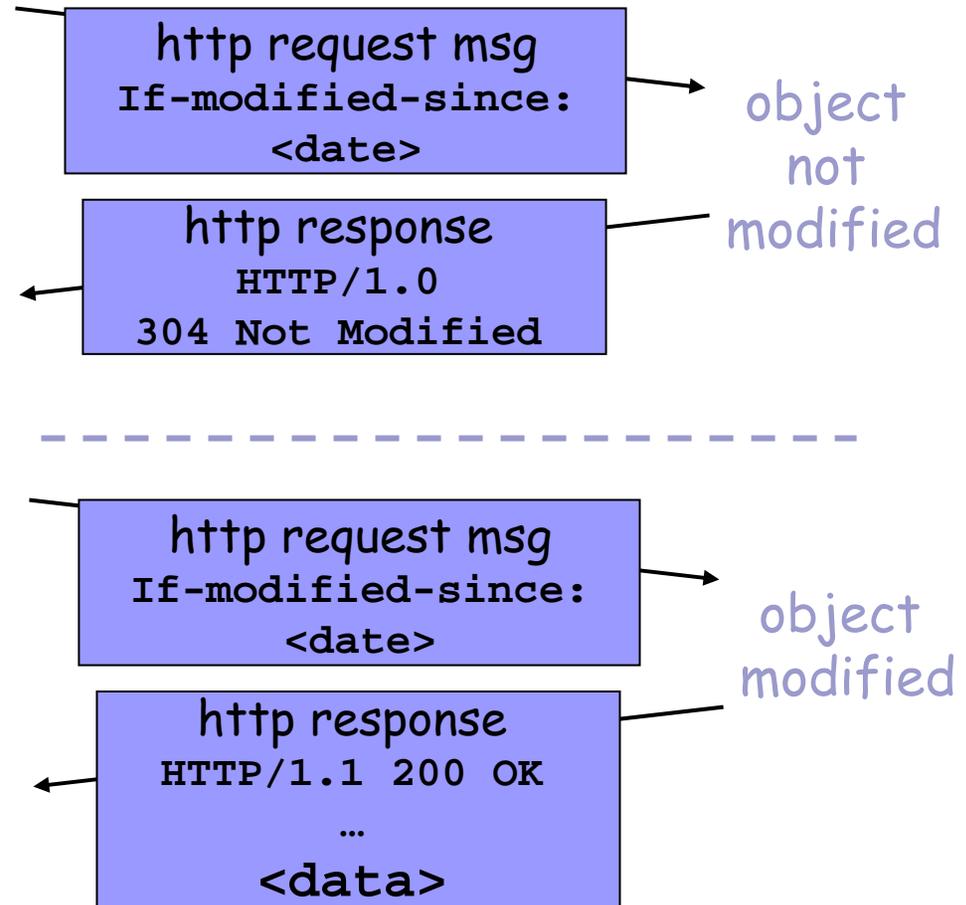
Try telnet www.cs.wmich.edu 80

User-server interaction: conditional GET

- **Goal:** don't send object if client has up-to-date stored (cached) version
- client: specify date of cached copy in http request
`If-modified-since: <date>`
- server: response contains no object if cached copy up-to-date:
`HTTP/1.0 304 Not Modified`

client

server



Post Example

- Here's a typical form submission, using POST:

```
POST /path/register.cgi HTTP/1.0
```

```
From: frog@cs.huji.ac.il
```

```
User-Agent: HTTPTool/1.0
```

```
Content-Type: application/x-www-form-urlencoded
```

```
Content-Length: 35
```

```
home=Ross+109&favorite+flavor=flies
```

Virtual Hosts

- With HTTP 1.1, one server at one IP address can be multi-homed:
 - “www.cs.wmich.edu” and “www.ece.wmich.edu” can live on the same server
 - These are called **virtual hosts**
 - Without this mechanism, we have to use 2 different IP addresses
- It is like several people sharing one phone
- An HTTP request must specify the host name (and possibly port) for which the request is intended (this is done using the Host header).

Virtual Hosting (cont.)

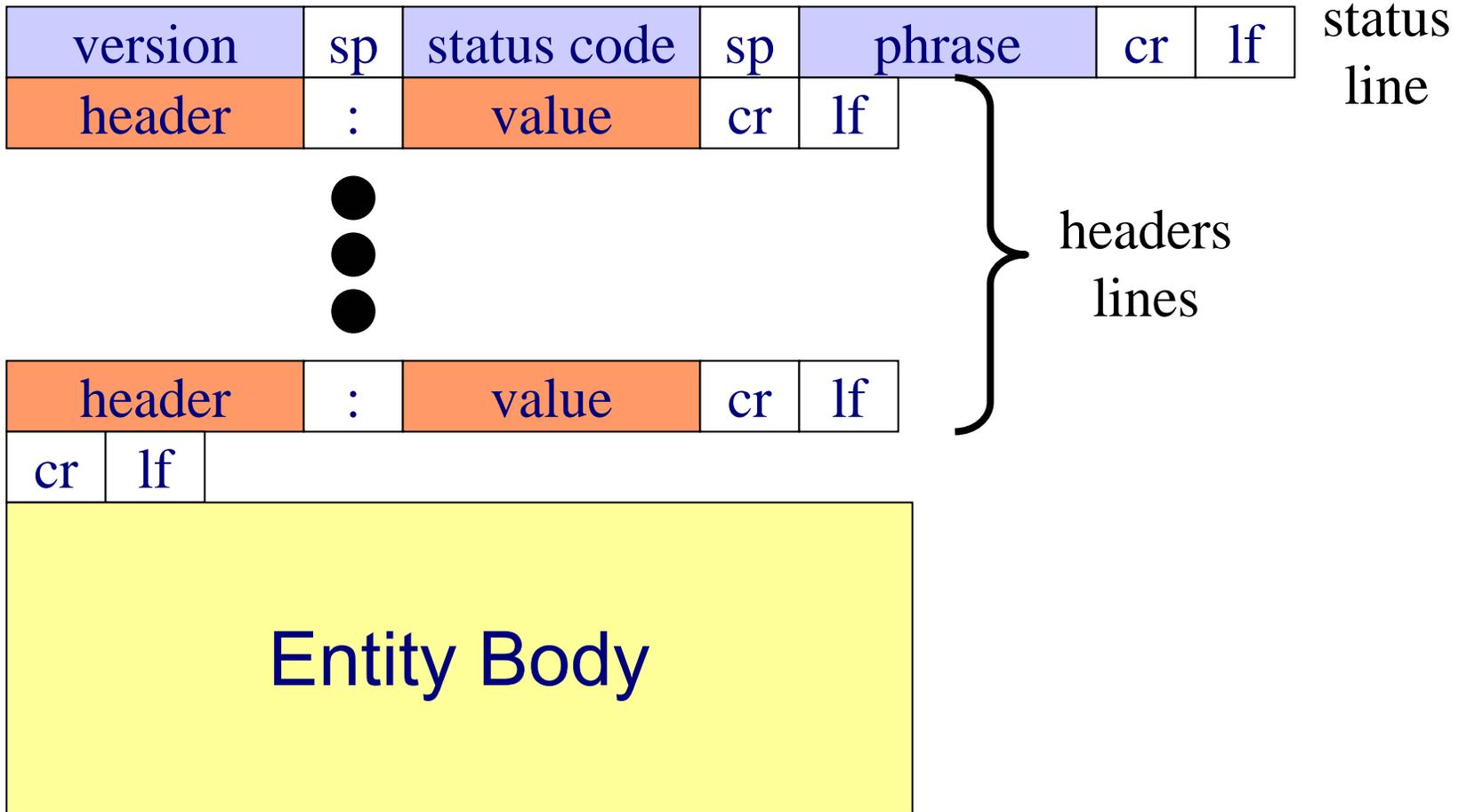
- Virtual hosting

- reduces hardware expenditures
- extends the ability to support additional servers
- makes load balancing and capacity planning much easier

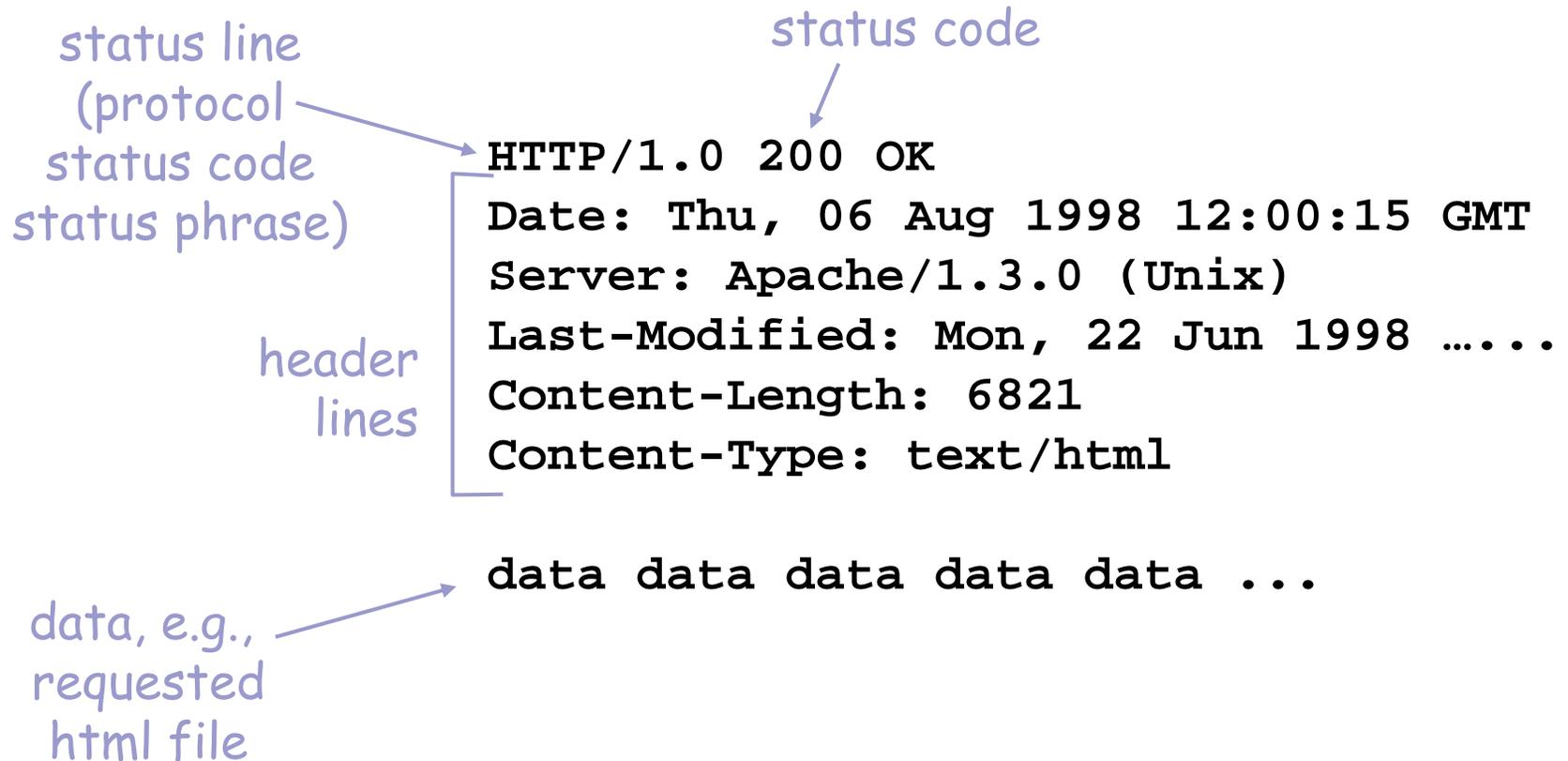
- Without it

- each host name requires a unique IP address, and we are quickly running out of IP addresses with the explosion of new domains

Format of Response



http message format: response



Status Code

- The status code is a three-digit integer, and the first digit identifies the general category of response:
 - 1xx indicates an informational message
 - 2xx indicates success of some kind
 - 3xx redirects the client to another URL
 - 4xx indicates an error on the client's part
 - Yes, the system blames it on the client if a resource is not found (i.e., 404)
 - 5xx indicates an error on the server's part

Redirection Process

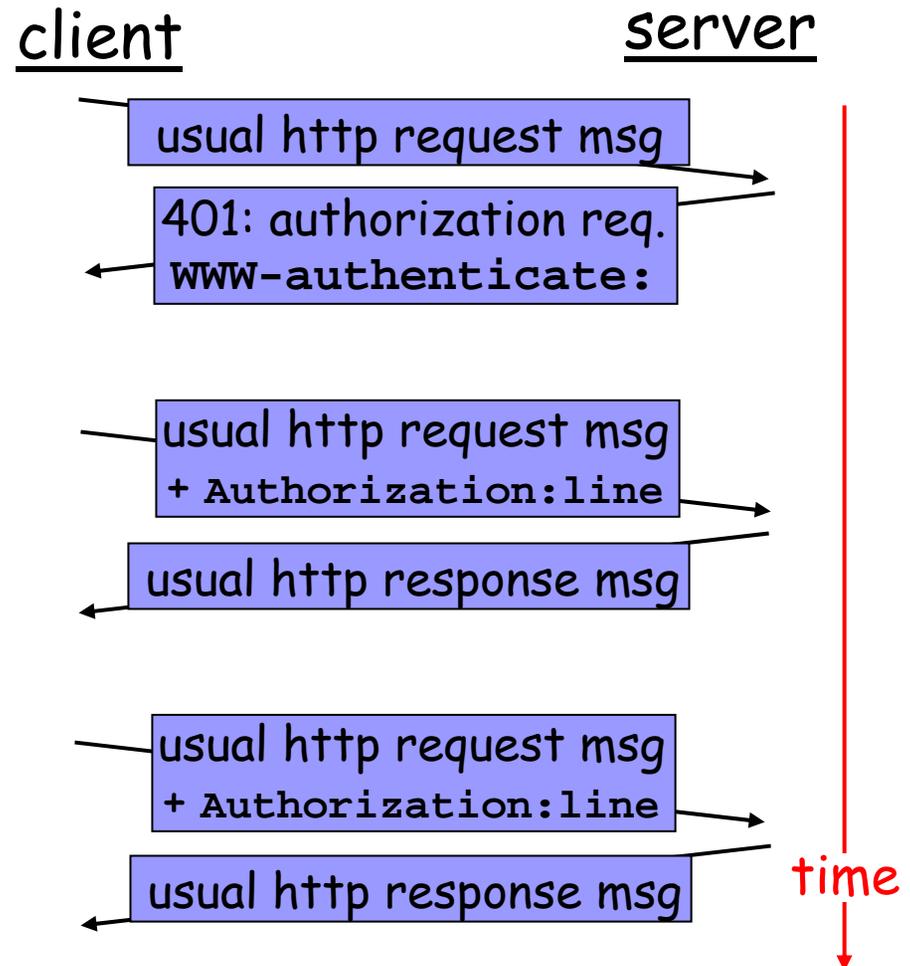
- Client asks for /foo, which is really a directory
- Server guesses that client meant /foo/ and so it replies with
 - 302 Moved
 - Location: /foo/
- Most browsers retry the new location automatically

Advantages of Redirection

- Simple Uses: Fix clients naming errors
- Complex Uses: Server can send client dynamically to a different page depending on
 - Who they are
 - What server is managing their session, etc.
- Note the changing URL in the browser

User-server interaction: authentication

- Authentication goal:** control access to server documents
- **stateless:** client must present authorization in each request
 - authorization: typically name, password
 - **authorization:** header line in request
 - if no authorization presented, server refuses access, sends **WWW authenticate:** header line in response



Browser caches name & password so that user does not have to repeatedly enter it.

References

- V22.0480-001, Sana` Odeh , Computer Science Department, New York University.
- Representation and Management of Data on the Internet (67633), Yehoshua Sagiv, The Hebrew University - Institute of Computer Science.
- Java Network Programming and Distributed Computing, Reilly & Reilly.
- *Computer Networking: A Top-Down Approach Featuring the Internet*, Kurose & Rose, Pearson Addison Wesley.