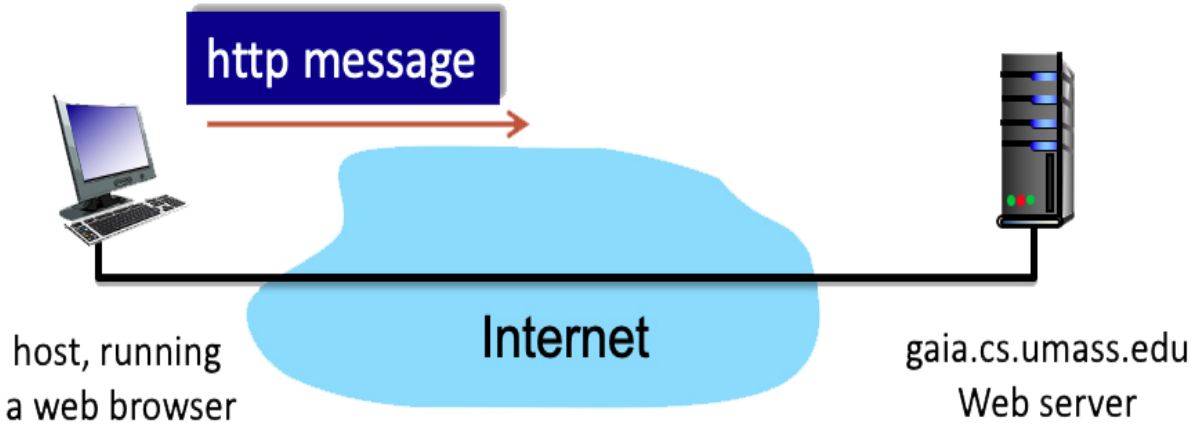
The HTTP GET message

Consider the figure below, where a client is sending an HTTP GET message to a web server, gaia.cs.umass.edu



Suppose the client-to-server HTTP GET message is the following:

GET /kurose_ross_sandbox/interactive/quotation9.htm HTTP/1.1
Host: gaia.cs.umass.edu
Accept: text/plain, text/html, image/gif, image/png, audio/basic, audio/mpeg, video/mpeg, video/mp4,
Accept-Language: en-us, en-gb;q=0.6, en;q=0.3, fr, fr-ch, ar
If-Modified-Since: Mon, 25 Aug 2025 03:08:41 -0700
User Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0)

1. What is the name of the file that is being retrieved in this GET message?
2. What version of HTTP is the client running?
3. True or False: The client will accept html files
4. True or False: The client will accept jpeg images
5. What is the client's preferred version of English?
6. What is the client's least preferred version of English?
7. True or False: The client will accept the German language
8. True or False: The client already has a cached copy of the file

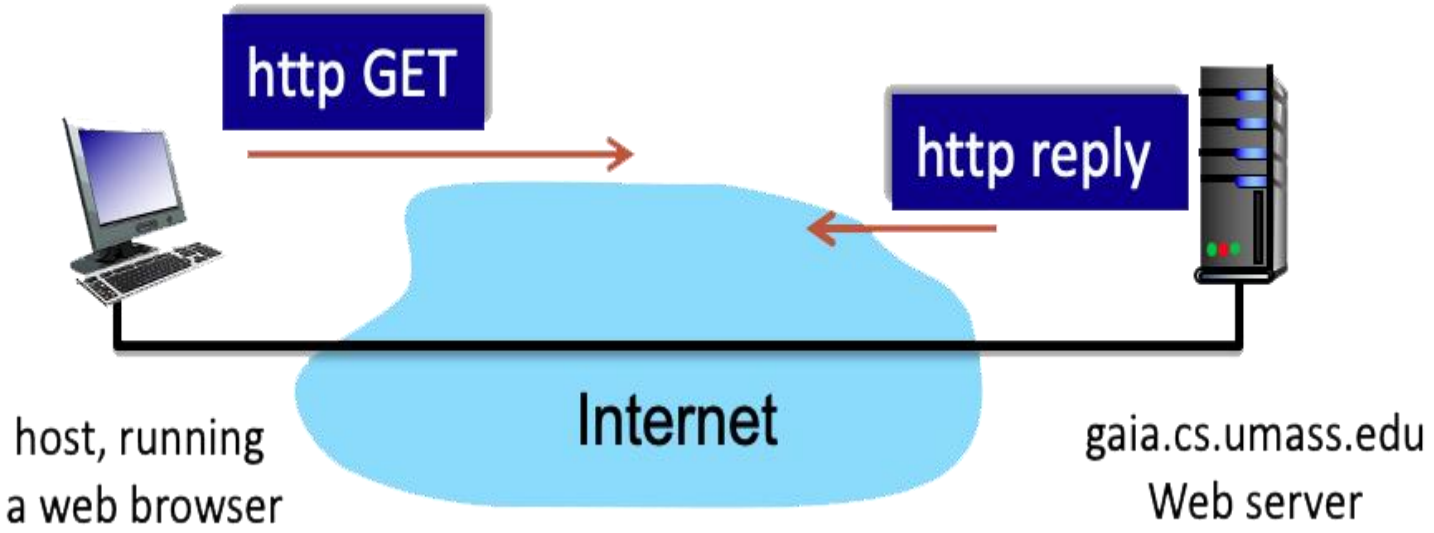
The HTTP GET message: Solution

1. The name of the file is quotation9.htm.
2. The client is running on HTTP/1.1
3. True. In the 'Accept' field the client includes 'text/html' files.
4. False. The client does NOT include 'image/jpeg' in its 'Accept' field.

5. The client's preferred version of English is American English. Any language without a defined q value has a default value of 1
6. The client's least preferred version of English is English because it has the lowest q value.
7. False. The client does NOT include German in its 'Accepted-Language' field.
8. True. The client has a cached copy of the file that was updated on: Mon, 25 Aug 2025 03:08:41 -0700

The HTTP RESPONSE message

Consider the figure below, where the server is sending a HTTP RESPONSE message back the client



Suppose the server-to-client HTTP RESPONSE message is the following:

HTTP/1.0 404 Not Found
Date: Mon, 25 Aug 2025 10:05:41 +0000
Server: Apache/2.2.3 (CentOS)
Content-Length: 384
Connection: Close
Content-type: image/html

1. Is the response message using HTTP 1.0 or HTTP 1.1?
2. Was the server able to send the document successfully? Yes or No
3. How big is the document in bytes?
4. Is the connection persistent or nonpersistent?
5. What is the type of file being sent by the server in response?
6. What is the name of the server and its version? Write your answer as server/x.y.z

The HTTP RESPONSE message: Solution

1. The response is using HTTP/1.0

2. Since the response code is 404 Not Found, the document was NOT received successfully.

3. The document is 384 bytes.

4. The connection is nonpersistent.

5. The file type the server is sending is image/html.

6. The name and version of the server is Apache/2.2.3

Trying out HTTP (client side) for yourself

1. Telnet to your favorite Web server:

```
telnet gaia.cs.umass.edu 80
```

- opens TCP connection to port 80 (default HTTP server port) at gaia.cs.umass. edu.
- anything typed in will be sent to port 80 at gaia.cs.umass.edu

2. type in a GET HTTP request:

```
GET /kurose_ross/interactive/index.php HTTP/1.1
Host: gaia.cs.umass.edu
```

- by typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

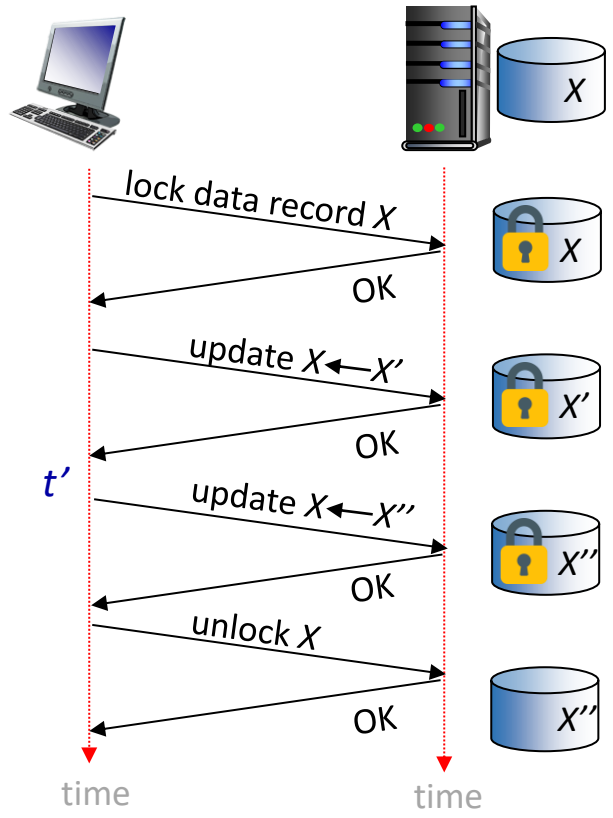
3. look at response message sent by HTTP server!

Maintaining user/server state: cookies

Recall: HTTP GET/response interaction is *stateless*

- no notion of multi-step exchanges of HTTP messages to complete a Web "transaction"
  - no need for client/server to track "state" of multi-step exchange
  - all HTTP requests are independent of each other
  - no need for client/server to "recover" from a partially-completed-but-never-completely-completed transaction

a stateful protocol: client makes two changes to X, or none at all

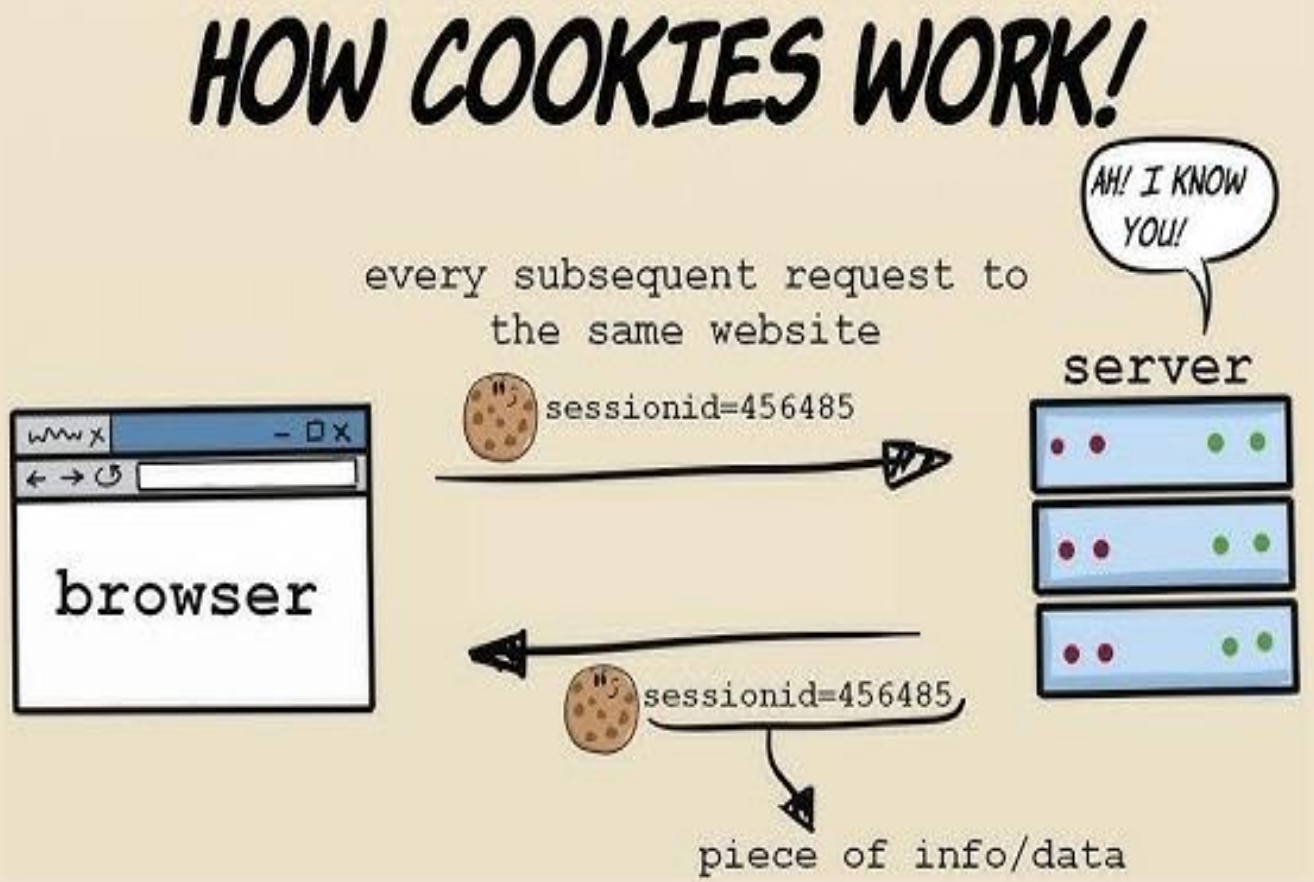


Q: what happens if network connection or client crashes at *t'* ?

What Are Cookies?

- Cookies are like bookmarks for websites, helping them remember things such as:
  - Your login session
  - Site preferences
  - Shopping cart contents
  - Analytics tracking info
  - Every time you revisit a site, your browser sends these cookies back to the server, allowing the site to "remember" who you are.

How cookies work!



Maintaining user/server state: cookies

Web sites and client browser use *cookies* to maintain some state between transactions

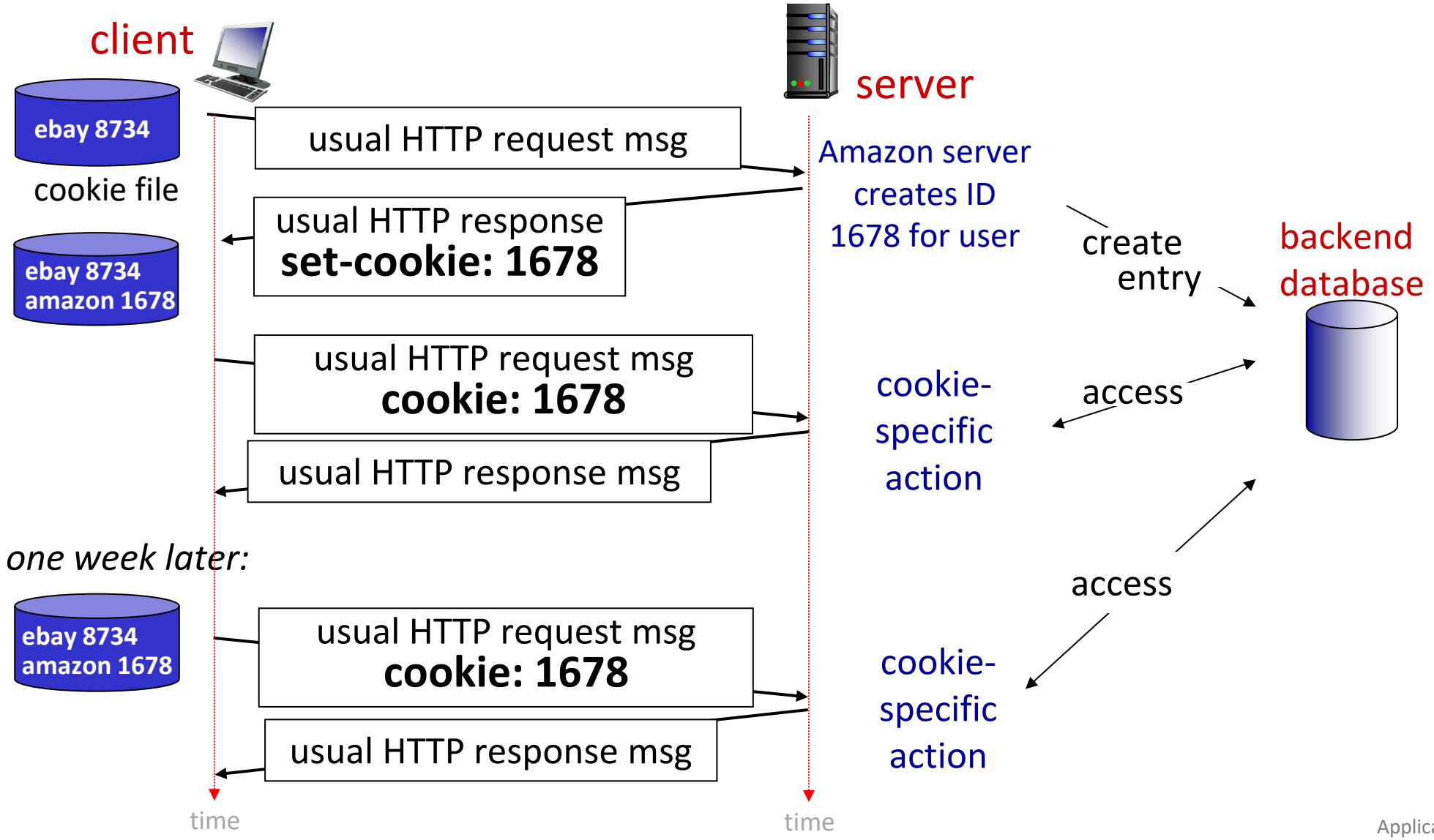
four components:

1) cookie header line of HTTP *response* message
2) cookie header line in next HTTP *request* message
3) cookie file kept on user's host, managed by user's browser
4) back-end database at Web site

Example:

- Prince uses browser on laptop, visits specific e-commerce site for first time
- when initial HTTP requests arrives at site, site creates:
  - unique ID (aka "cookie")
  - entry in backend database for ID
- subsequent HTTP requests from Prince to this site will contain cookie ID value, allowing site to "identify" Prince

Maintaining user/server state: cookies



HTTP cookies: comments

What cookies can be used for:

- authorization
- shopping carts
- recommendations
- user session state (Web e-mail)

Challenge: How to keep state:

- protocol endpoints: maintain state at sender/receiver over multiple transactions
- cookies: HTTP messages carry state

aside

cookies and privacy:

- cookies permit sites to *learn* a lot about you on their site.
- third party persistent cookies (tracking cookies) allow common identity (cookie value) to be tracked across multiple web sites

Cookie Central includes extensive information on the cookie controversy.