Lecture 4
Sessions and Personalization

The Web Remembers You

- How does a shopping cart stay full?
- How do I log in?
- How does Google remember my past queries?

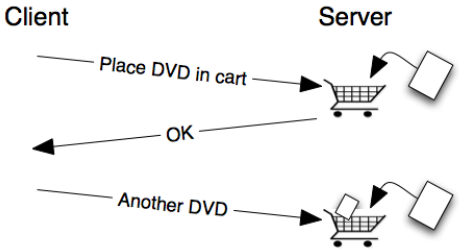
2

Overview

- Sessions
- Login
- Cookies
- Personalization
- Single Sign-on

3

Session Requirements



•Must maintain state with stateless HTTP

4

Sessions

- A session is a single "interaction" between the site and user
  - Precise def'n depends on application
  - HTTP is session-less
- TCP has connections, similar to sessions
  - Relationship between TCP and "sessions"?

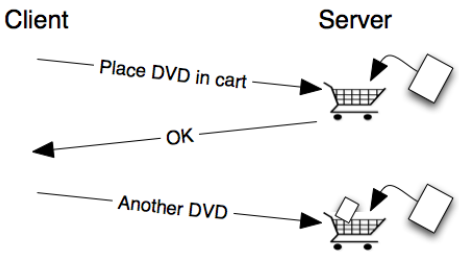
5

Stateful HTTP?

- HTTP is stateless, so we want to use a "session protocol" on top of it
  - Like TCP does on IP
- But, uh, there's no such thing as a "session protocol"
- Implemented at app-layer instead
  - State maintained in session variables
  - Data stored in one request can be accessed by later request, as long as within same session

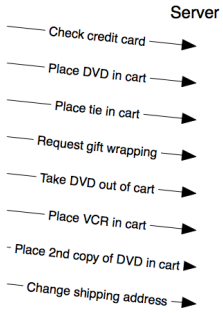
6

Session Perspectives - Client



7

Session Perspectives - Server



- Server must track many sessions at once

8

Server Session Model

- Sessions like files, or call-stack frames
  - Collection of attr/value pairs
  - E.g., `name=mike, numEmails=20`
- Sessions explicitly opened, closed
  - Old variables must be explicitly deleted
  - PHP does a lot for you by default
- Server issues:
  - When to create + close sessions?
  - How to link HTTP requests to a session?
  - How to link sessions to users?
  - How to store session data?

9

Server session processing

1. HTTP request arrives
   - If new session, init session structure
   - Else, load session structure from storage
2. Service HTTP Request; update session
   - If end of session, dealloc session struct
   - Else, save session updates to storage
3. HTTP response to client

10

Begin and end

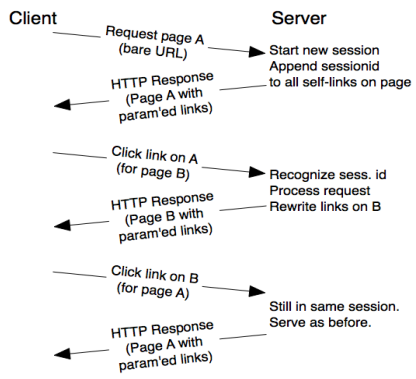
- Starting a session is easy
  - Is request associated with a session?
- Ending is harder
  - We can't rely on logout
  - Timeouts needed for almost all apps
    - When should the online game be reset?
    - When should Google forget your search?
    - When has your cart been abandoned?
    - When have you started searching for a different flight?
  - Timeout from first request or most recent?

11

Request-to-Session

- URL Encoding
  - http://google.com/search?q=hello
  - URL parameters come after ?
  - Attr/val pairs
  - Rewrite the URLs on a page to reflect the session id or other embedded data

12



13

Request-to-Session

http://www.google.com/webhp?hl=en#hl=en&source=hp&q=eecs485&aq=f&aql=&aqi=g-sx1&oq=&fp=292ac4760832f3c4

- http://www.google.com/webhp
- q=eecs485
- hl=en#hl=en
- source=hp
- aq=f
- aql=aqi, aqi=g-sx1
- oq=
- fp=292ac4760832f3c4

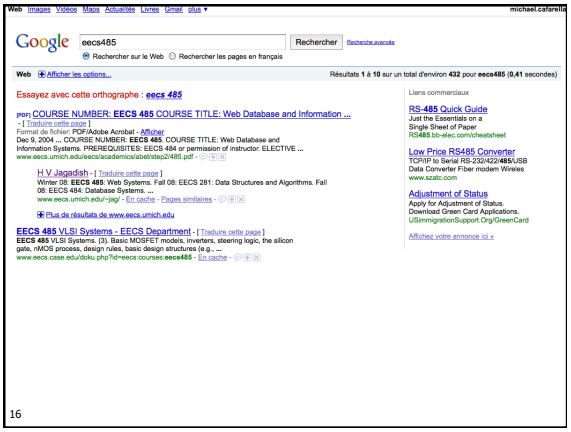
14

Request-to-Session

http://www.google.com/webhp?hl=en#hl=en&source=hp&q=eecs485&aq=f&aql=&aqi=g-sx1&oq=&fp=292ac4760832f3c4

- http://www.google.com/webhp
- q=eecs485
- hl=en#hl=en
- source=hp
- aq=f
- aql=aqi, aqi=g-sx1
- oq=
- fp=292ac4760832f3c4

15



16

Request-to-Session

http://www.google.com/webhp?hl=en#hl=en&source=hp&q=eecs485&aq=f&aql=&aqi=g-sx1&oq=&fp=292ac4760832f3c4

- http://www.google.com/webhp
- q=eecs485
- hl=en#hl=en
- source=hp
- aq=f
- aql=aqi, aqi=g-sx1
- oq=
- fp=292ac4760832f3c4

17

Request-to-Session

http://www.google.com/webhp?hl=en#hl=en&source=hp&q=eecs485&aq=f&aql=&aqi=g-sx1&oq=&fp=292ac4760832f3c4

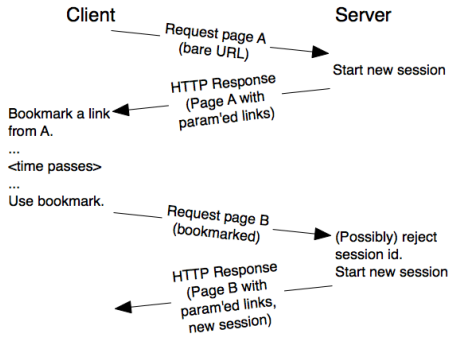
- http://www.google.com/webhp
- q=eecs485
- hl=en
- source=hp
- aq=f
- aql=aqi, aqi=g-sx1
- oq=
- fp=292ac4760832f3c4

18

Bookmarking

- What if the session is no longer valid?
- What if URLs are bookmarked?
  - Shared?
  - Intercepted?

19



20

Bookmarking

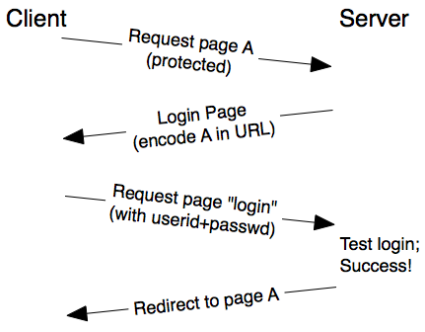
- What if the session is no longer valid?
- What if URLs are bookmarked?
  - Shared?
  - Intercepted?
- Session ids should:
  - Have some resistance to guess-attacks
  - Be unlikely to be accidentally replicated
    - Incrementing a counter is bad
  - Contain validating data (like timestamp)

21

Logins

- Sessions can be anonymous
- Sometimes nicer to authenticate
  - Store state per-user basis, not per-session
  - User can move to different machines
  - Don't have to timeout sessions; some state should be long-lasting
- Encode authenticated user id
  - Often helpful to encode userid *and* session
  - Authentication next week
- Combine login w/webpage access ctrl

22



23

Sessions and Usability

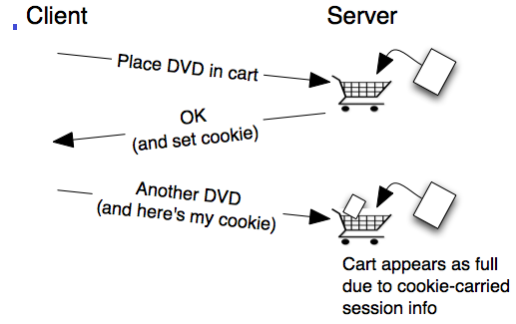
- Login failure has default 401 response
  - Nicer to redirect to login page
- If request requires login first, don't just dead-end at login page
  - Store target as session var; visit after login
- Store userid for faster future login
- Customization: explicit user mods
- Individualization: auto-tailored experience

24

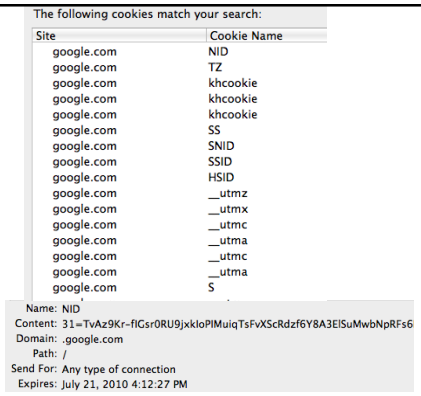
Cookies

- URL-encoding is not perfect
  - What if you type in the URL yourself?
  - Some software refuse too-long URLs
  - Ugly
- Cookies are small files on client machine
  - Carry state between HTTP requests
  - attr/val pairs, just like URL params
- Set by server, but not requested
- Sent by client, but never edited
- Either side can delete/ignore cookies

25



26

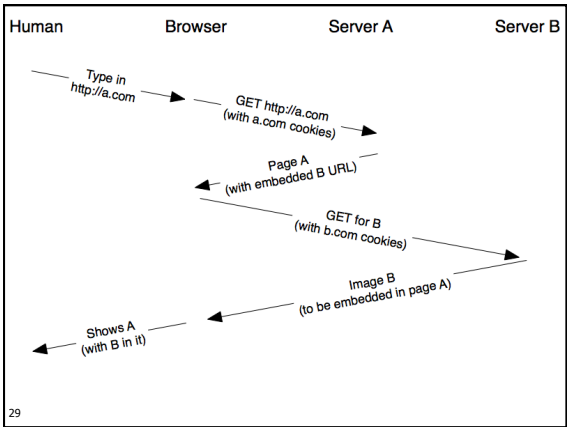


27

Cookie Contents

- **Path** specifies scope of cookie
  - catalog vs catalog/page1.html
- **Expiration** tells client when to delete
- **Secure** is how cookie may be xmitted
- Contents up to server: encrypted? OK!
- Supported by HTTP
- Cookies only overwritable by src domain and path-prefix
- Browser per-domain, total limits

28



29

Uses and Abuses

- Cookies make sessions quiet, ubiquitous
  - Even at IHopeNooneSeesMeHere.com
- Third-party cookies
  - Page may contain objects from many srcs
    - Images, ads, other plugins
  - These "3rd-party" objects set/get cookies
  - AdSense can track across sites this way
  - "Web Bugs" are invisible images or frames, used for tracking
- Cookies also add to HTTP overhead
- Best practice is to use them sparingly

30