# User-server state: cookies

Many major Web sites use cookies

#### Four components:

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

## <u>Example:</u>

- Susan always access
   Internet always from PC
- visits specific ecommerce site for first time
- when initial HTTP requests arrives at site, site creates:
  - Inique ID
  - entry in backend database for ID









- cache acts as both client and server
- typically cache is installed by ISP (university, company, residential ISP)

## Why Web caching?

- reduce response time for client request
- reduce traffic on an institution's access link.
- Internet dense with caches: enables "poor" content providers to effectively deliver content (but so does P2P file sharing)

















### Sample commands:

- sent as ASCII text over control channel
- USER username
- PASS password
- LIST return list of file in current directory
- RETR filename retrieves (gets) file
- STOR filename stores (puts) file onto remote host

### Sample return codes

- status code and phrase (as in HTTP)
- 331 Username OK, password required
- 125 data connection already open; transfer starting
- 425 Can't open data connection
- 1 452 Error writing file

2: Application Layer 52

## FTP: let us try it out... and get /pub/rfcs/rfc-index.txt.pdf (submit results via elearning platform)

## Usage of commands:

Make a telnet connection to *ftp.di.uminho.pt* , *port 21* , *and use:* 

- □ USER anonymous
- □ PASS any-password
- PASV enter the passive mode
- look and record PASV <response>
- Other commands: RETR filename, LIST filename, QUIT

### Data connection:

- A data connection must be opened. Where to?
- <response> =
  (X,Y,Z,W,PH,PL) where
- IP address = X.Y.Z.W
   (or X\*256^3 + Y\*256^2
   + Z\*256 + W)
- Port # = PH\*256 + PL
- ... ... ... (how can you
  get the file?)



- 2.1 Principles of network applications
- 2.2 Web and HTTP
- 2.3 FTP
- 2.4 Electronic Mail
   SMTP, POP3, IMAP
- **2.5 DNS**

- □ 2.6 P2P applications
- 2.7 Socket programming with TCP
- 2.8 Socket programming with UDP



















- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7bit ASCII
- SMTP server uses
   CRLF.CRLF to determine
   end of message

## Comparison with HTTP:

- HTTP: pull
- □ SMTP: push
- both have ASCII command/response interaction, status codes
- HTTP: each object encapsulated in its own response msg
- SMTP: multiple objects sent in multipart msg













- 2.1 Principles of network applications
- 2.2 Web and HTTP
- **2.3 FTP**
- □ 2.4 Electronic Mail
- 2.6 P2P applications
- 2.7 Socket programming with TCP
- 2.8 Socket programming with UDP
- 2.9 Building a Web server



























## Let's try it out... DNS: (submit answers via elearning platform) Submit answers via http://elearning.uminho.pt 1. Which are the Name Servers and IP addresses for di.uminho.pt., uminho.pt., google.com. 1. 2. Get Authoritative answer for the IP address of: SOA server for sapo.pt., yahoo.com., publico.pt. 1. MX record for di.uminho.pt., up.pt. 2. 3. Identify completely (including domain name, email address, surface address, telephone #) an hypothetical attacker: 193.136.19.190, 193.137.89.146 1. 193.137.90.45 2. 4. identify temporal parameters for domain "gcom.di.uminho.pt"