Computer Networks

World Wide Web

Jianping Pan
Fall 2015

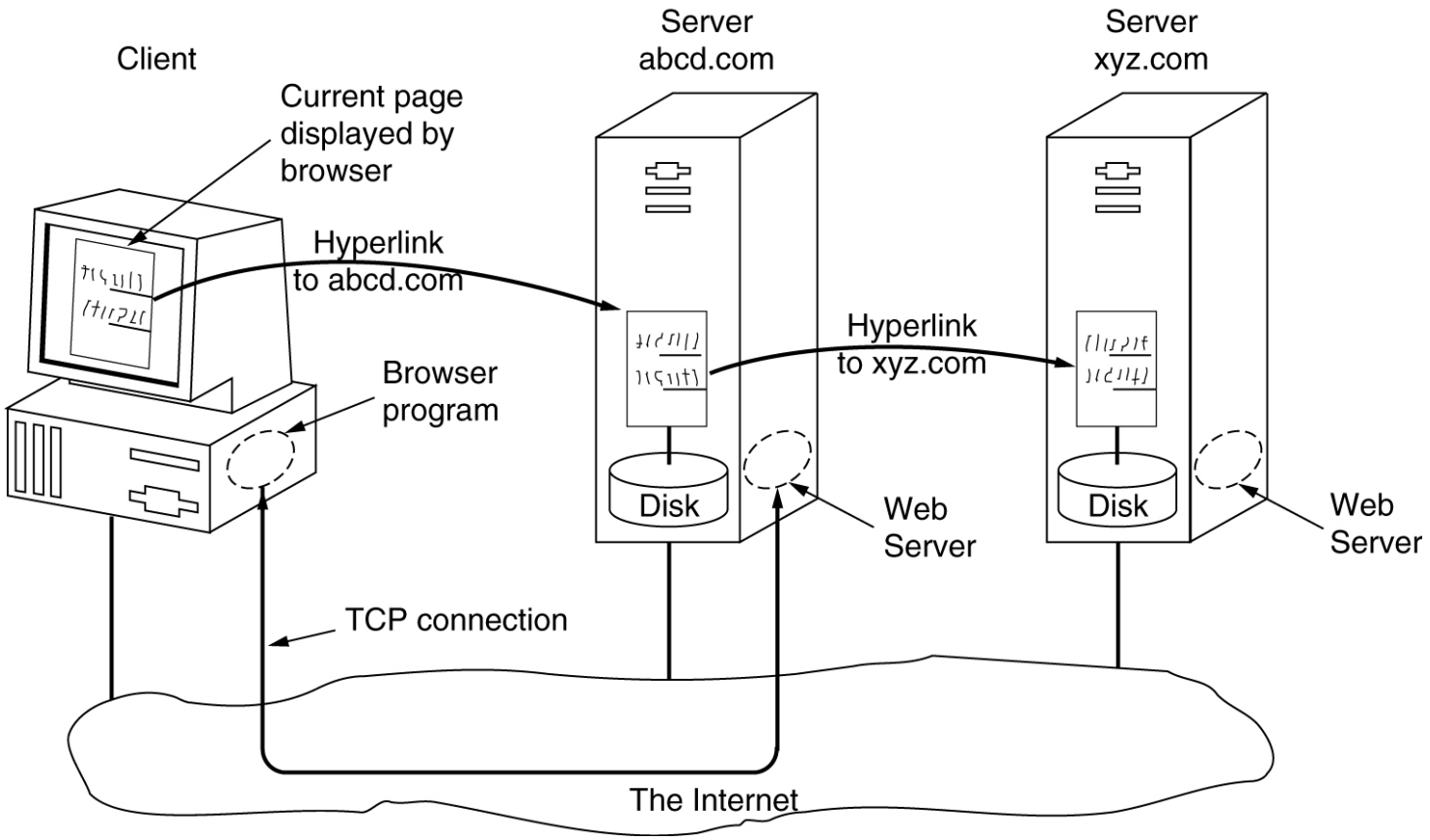
Worldwide web

- Sir Tim Berners-Lee
  - 1989, CERN; ... W3C; ... W3F!
- Hypertext and hypermedia
    - linked documents

- Marc Andreessen
  - 1993, Mosaic, NCSA@UIUC
- Netscape Comm
  - Netscape navigator vs MS Internet explorer

Q: Mozilla, IE, Firefox, Safari, Opera, Chrome?

Web overview



Uniform Resource Locator

- http://user:pass@host:port/path/file?input

Name	Used for	Example
http	Hypertext (HTML)	http://www.cs.vu.nl/~ast/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
news	Newsgroup	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending e-mail	mailto:JohnUser@acm.org
telnet	Remote login	telnet://www.w3.org:80

Q: DocumentRoot and mapping?

HTML tags

- Anchors
  - ...
- Objects
  - <img src="...">

Tag	Description
<html> ... </html>	Declares the Web page to be written in HTML
<head> ... </head>	Delimits the page's head
<title> ... </title>	Defines the title (not displayed on the page)
<body> ... </body>	Delimits the page's body
<h *n*> ... </h*n*>	Delimits a level *n* heading
...	Set ... in boldface
...	Set ... in italics
...	Center ... on the page horizontally
...	Brackets an unordered (bulleted) list
...	Brackets a numbered list
	Starts a list item (there is no)
 	Forces a line break here
<p>	Starts a paragraph
<hr>	Inserts a Horizontal rule
<img src="...">	Displays an image here
 ... 	Defines a hyperlink

Q: linked vs embedded objects?

HTTP

- Hyper text transfer protocol
  - application layer protocol, ASCII format
    - HTTP/1.0: RFC1945 (1996); 1.1: RFC2068 (1997)
  - typical client-server model: request-reply
    - client (browser): Mozilla, Opera, IE, Chrome, etc
    - server (web server)
      - Apache, Microsoft Internet information server (IIS)
  - normally uses service offered by TCP
    - http: 80; https: 443 (HTTP over SSL over TCP)

* HTTP/2.0?

HTTP requests

- Request methods

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

- Request parameters (control headers)

HTTP responses

- Response codes

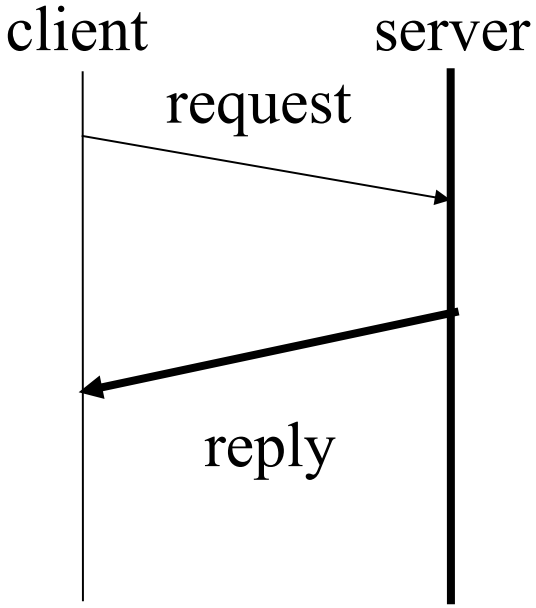
Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

  - 400: bad request
- Response parameters
- Response data

HTTP examples

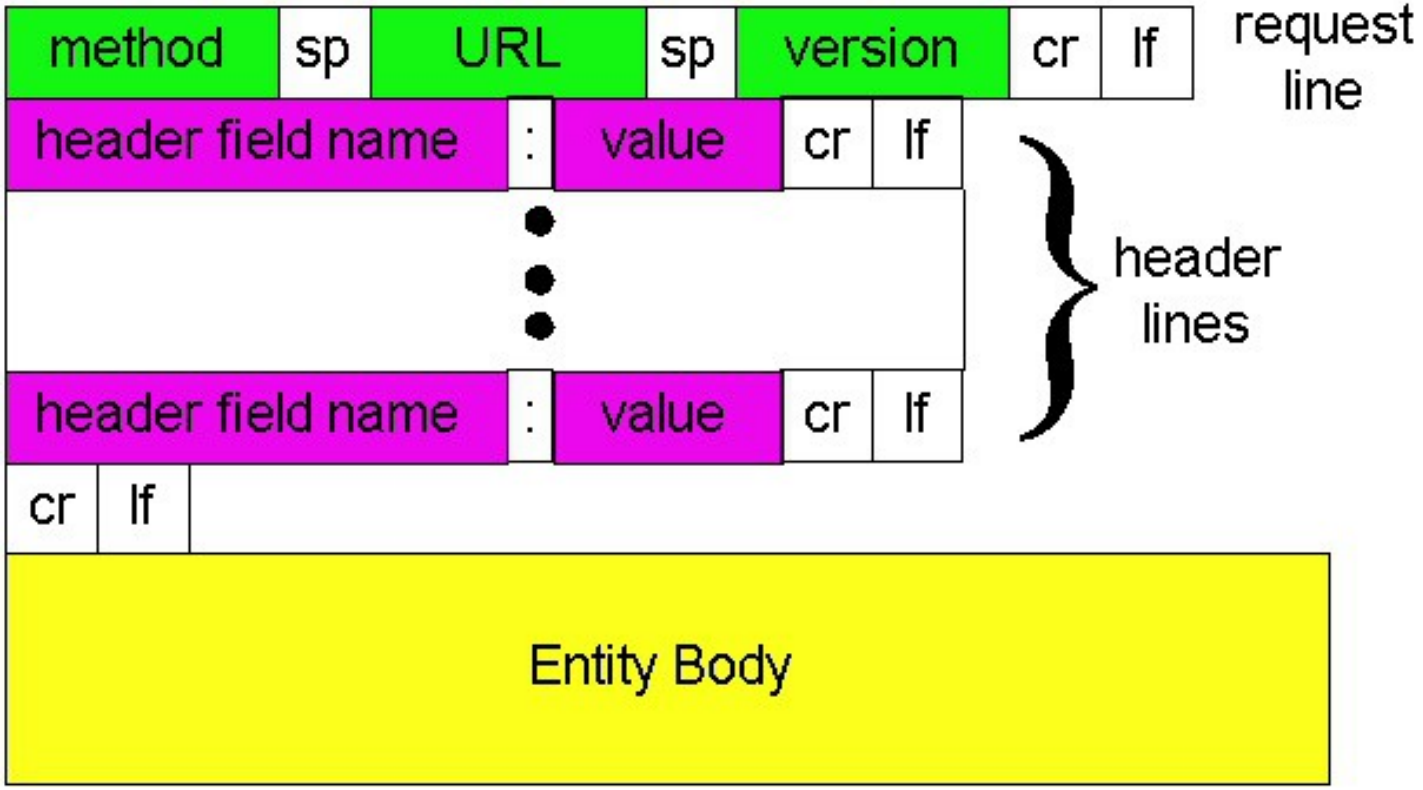
- # wget -d www.google.com

  Connecting to www.google.com:80... Caching www.google.com <-> 66.102.7.104
  Created fd 3.
  connected!
  ---request begin---
  GET / HTTP/1.0
  User-Agent: Wget/1.7
  Host: www.google.com
  Accept: */*
  Connection: Keep-Alive
  ---request end---
  HTTP request sent, awaiting response...
  HTTP/1.0 302 Found
  Location: http://www.google.ca/
  Cache-Control: private
  Content-Type: text/html
  Server: GWS/2.1
  Content-Length: 218



Q: syntax, semantics, synchronization?

Message-level view



Web browsing examples

- http://www.a.com/index.html
  <html>
  <img src="http://www.a.com/x.gif">

  <img src="http://www.b.com/z.gif">
  </html>
- In your favorite web browser
  - URL: http://www.a.com
  - Q: how many HTTP requests?

discussion on blackboard

This lecture

- Internet architecture
  - services, protocols, client/server
- HTTP basics
  - why “web”?
    - linked objects: links
    - embedded objects: e.g., images
  - URL
    - “http://user:pass@host:port/path/file?input”?!!
    - how to interpret?
  - HTTP service model
    - HTTP over TCP
      - connection-oriented, reliable data transfer service
    - request/response
      - classic client-server model

Next lecture

- HTTP intermediate
  - HTTP persistence
    - how many TCP connections do you need?
    - or, how fast can you get your web page?
      - a “metric” for browser wars
  - beyond the classic client-server model
    - web proxy, cache server, ... (you name it!)

- Read
  - KR4, Chapter 2