COMP2001J Computer Networks

Lecture 13 – Application Layer

Dr. Shen WANG (王燊)

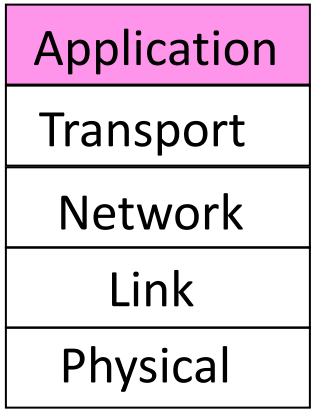
shen.wang@ucd.ie

Outline

- Architectural Overview
- HTTP
- DNS
- CDN
- EMAIL

Application Layer

- Uses transport services to build distributed applications

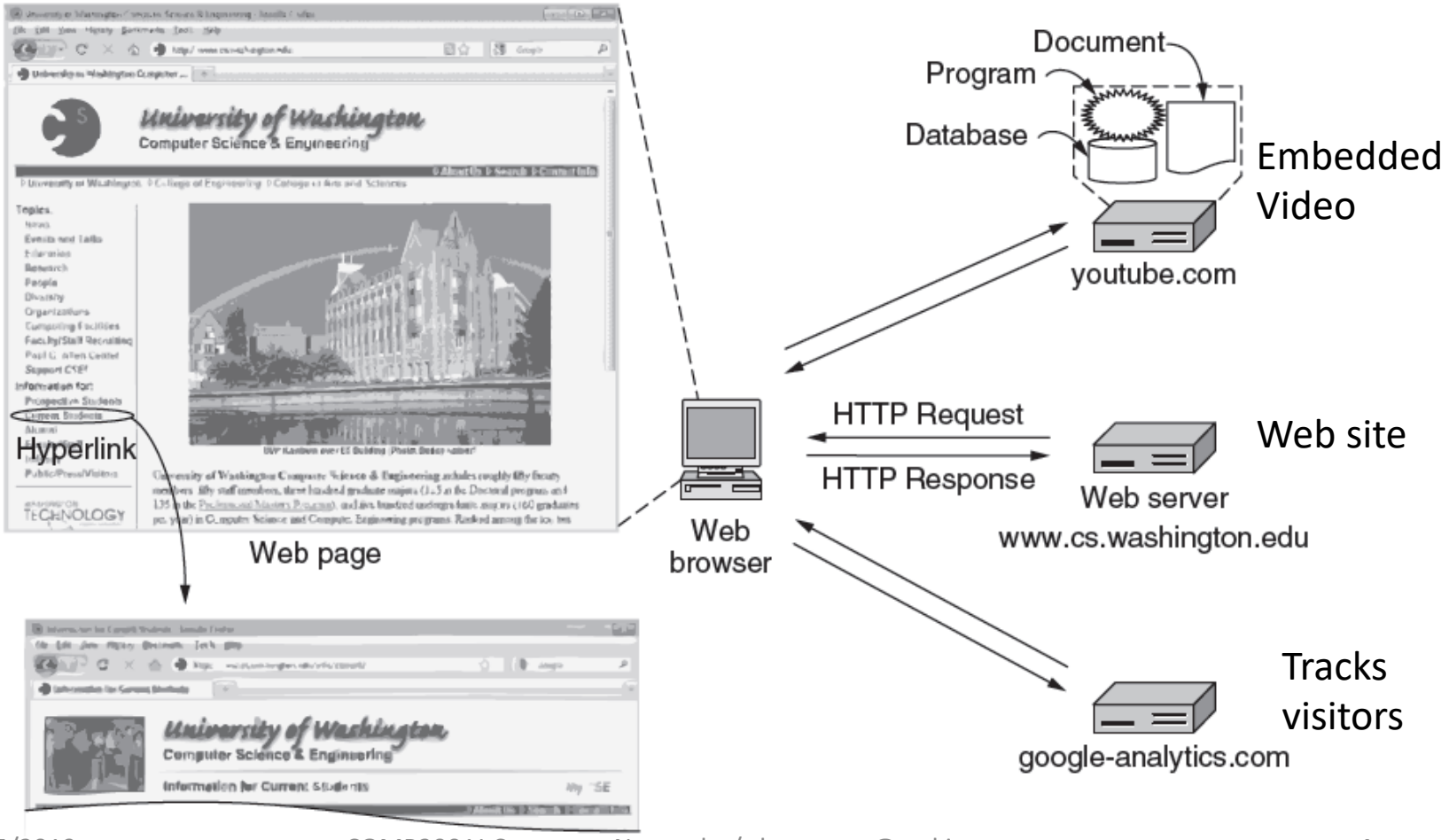


Architectural Overview

- the Web consists of a vast, worldwide collection of content in the form of **Web pages**, often just called **pages** for short
- The idea of having one page point to another, by clicking the *hyperlink*, now called **hypertext**
- Pages are generally viewed with a program called a **browser** (e.g. IE, Firefox, Chrome)
  - The browser fetches the page requested, interprets the content, and displays the page, properly formatted, on the screen.
- The page is a **static page** if it is a document that is the same every time it is displayed.
- In contrast, if it was generated on demand by a program or contains a program it is a **dynamic page**

Architectural Overview

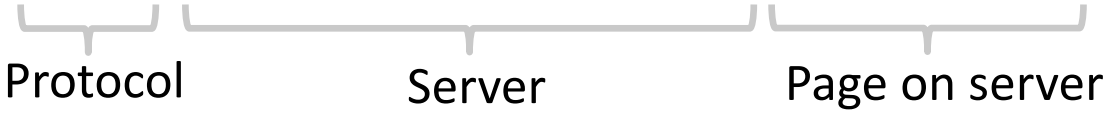
HTTP transfers pages from servers to browsers



URL

Pages are named with URLs (Uniform Resource Locators)

- Example: http://www.phdcomics.com/comics.php



Our focus →

Name	Used for	Example
http	Hypertext (HTML)	http://www.ee.uwa.edu/~rob/
https	Hypertext with security	https://www.bank.com/accounts/
ftp	FTP	ftp://ftp.cs.vu.nl/pub/minix/README
file	Local file	file:///usr/suzanne/prog.c
mailto	Sending email	mailto:JohnUser@acm.org
rtsp	Streaming media	rtsp://youtube.com/montypython.mpg
sip	Multimedia calls	sip:eve@adversary.com
about	Browser information	about:plugins

Common URL protocols

MIME Type

- MIME (Multipurpose Internet Mail Extensions).
  - It is widely used for mail messages that are sent across the Internet, as well as to describe content for other applications such as Web browsing.

Type	Example subtypes	Description
text	plain, html, xml, css	Text in various formats
image	gif, jpeg, tiff	Pictures
audio	basic, mpeg, mp4	Sounds
video	mpeg, mp4, quicktime	Movies
model	vrml	3D model
application	octet-stream, pdf, javascript, zip	Data produced by applications
message	http, rfc822	Encapsulated message
multipart	mixed, alternative, parallel, digest	Combination of multiple types

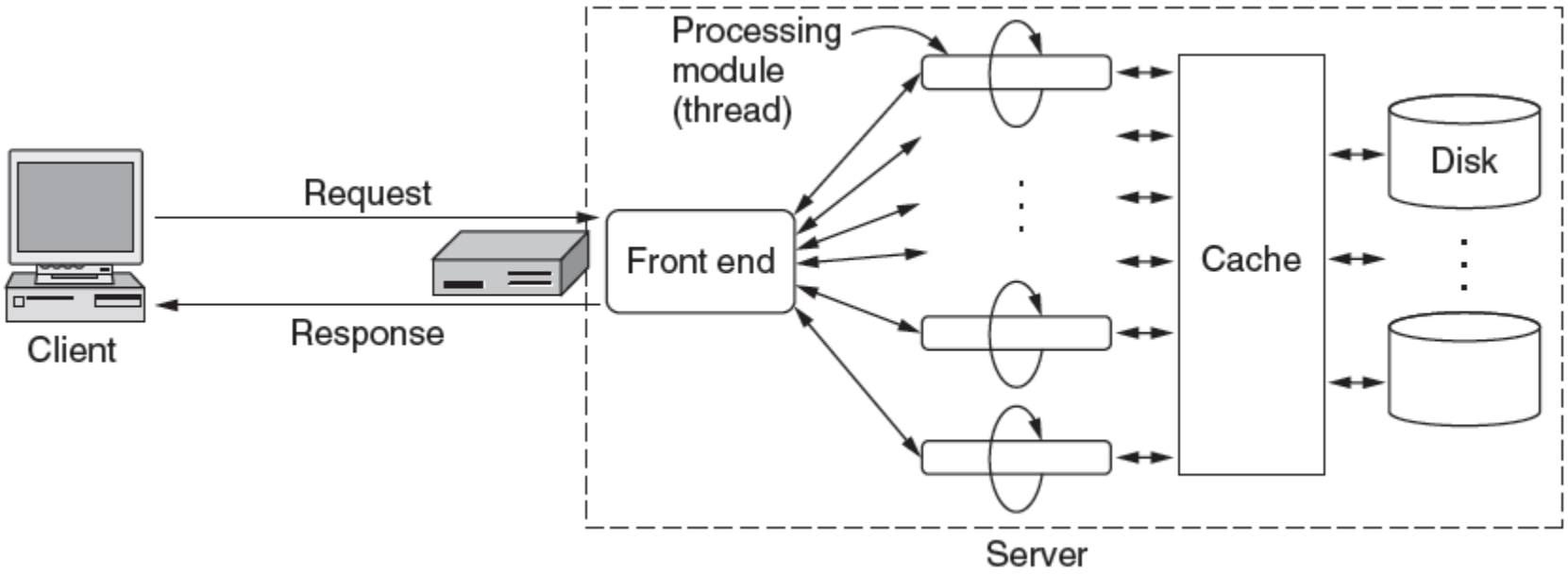
Client & Server

- Steps a **client** (browser) takes to follow a hyperlink:
  - Determine the protocol (HTTP)
  - Ask DNS for the IP address of server
  - Make a TCP connection to server
  - Send request for the page; server sends it back
  - Fetch other URLs as needed to display the page
  - Close idle TCP connections

- Steps a **server** takes to serve pages:
  - Accept a TCP connection from client
  - Get page request and map it to a resource (e.g., file name)
  - Get the resource (e.g., file from disk)
  - Send contents of the resource to the client.
  - Release idle TCP connections

Web Servers

- To scale performance, Web servers can use:
  - Caching, multiple threads, and a front end



Revisited server steps

- These steps occur after the TCP connection:

1. Resolve name of Web page requested
2. Perform access control on the Web page
3. Check the cache
4. Fetch requested page from disk or run program
5. Determine the rest of the response
6. Return the response to the client
7. Make an entry in the server log

Cookies

- Once TCP connection is released, all information related is gone
- When a client requests a Web page, the server can supply additional information in the form of a **cookie** along with the requested page.
- The cookie is a rather small, named string (of at most 4 KB) that the **server** can associate with a browser.
  - This association is not the same thing as a user, but it is much closer and more useful than an IP address.
- Browsers store the offered cookies for an interval, usually in a cookie directory on the client's disk so that the cookies persist across browser invocations, unless the user has disabled cookies.
- Cookies are just strings, not executable programs.

Cookies

- Cookies support **stateful** client/server interactions
  - Server sends cookies (state) with page response
  - Client stores cookies across page fetches
  - Client sends cookies back to server with requests

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=297793521	15-10-10 17:00	Yes
jills-store.com	/	Cart=1-00501;1-07031;2-13721	11-1-11 14:22	No
aportal.com	/	Prefs=Stk:CSCO+ORCL;Spt:Jets	31-12-20 23:59	No
sneaky.com	/	UserID=4627239101	31-12-19 23:59	No

Examples of cookies

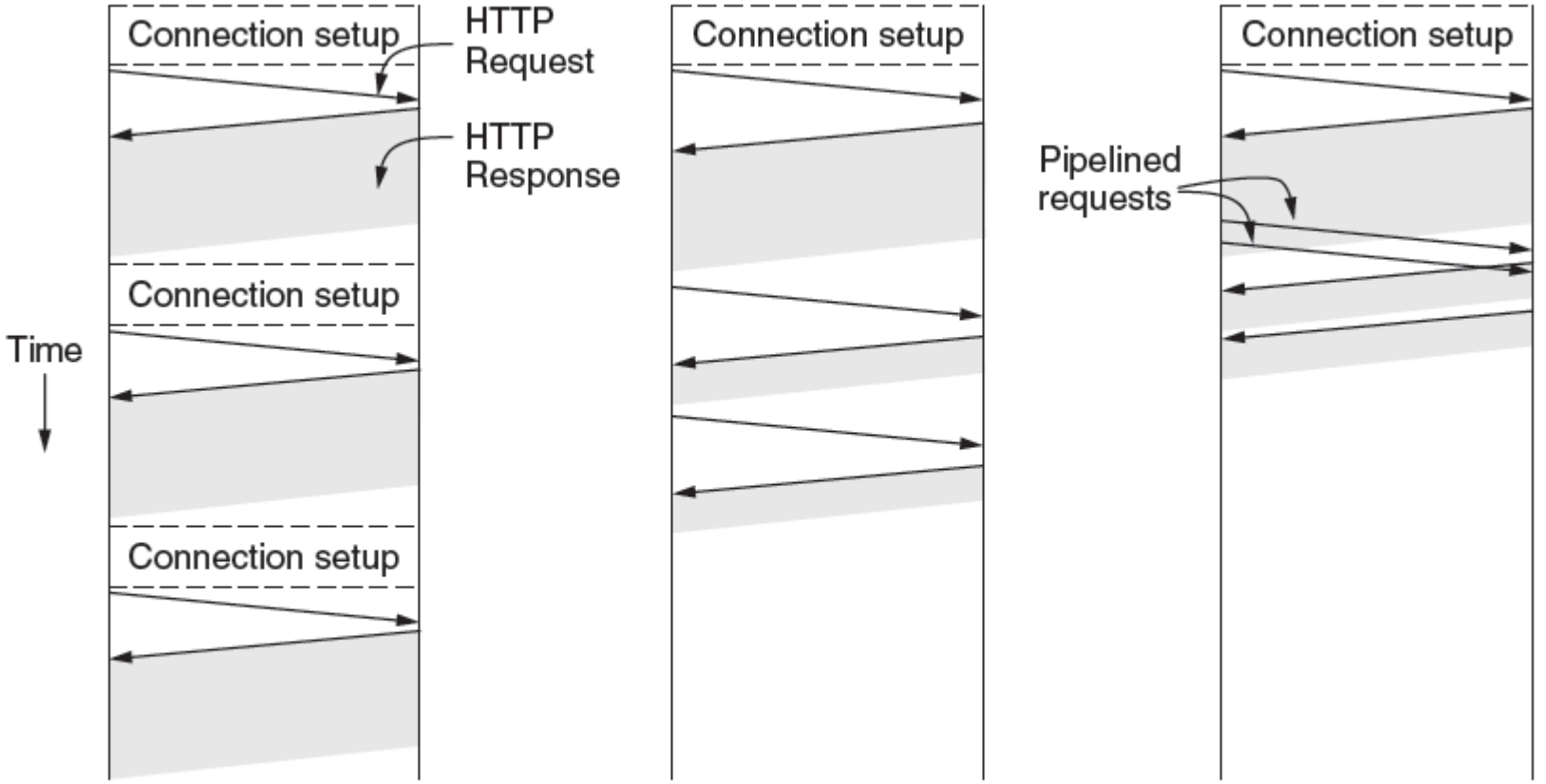
HTTP

- HTTP (HyperText Transfer Protocol) is a request-response protocol that runs on top of TCP.
- It specifies what messages clients may send to servers and what responses they get back in return.
  - Fetches pages from server to client
  - Server usually runs on port 80
  - Headers are given in readable ASCII
  - Content is described with MIME types
  - Protocol has support for pipelining requests
  - Protocol has support for caching

HTTP – Connection

- HTTP 1.0
  - the connection was established -> a single request was sent over and a single response was sent back -> the TCP connection was released.
  - Establishing a separate TCP connection to transport each single object (e.g.: pic, text) became a very expensive way to operate
- HTTP 1.1
  - **persistent connections**: establish a TCP connection -> send a request and get a response -> and then send additional requests and get additional responses.
  - **pipeline requests**: send request 2 before the response to request 1 has arrived.

HTTP – Connection



One connection for each request

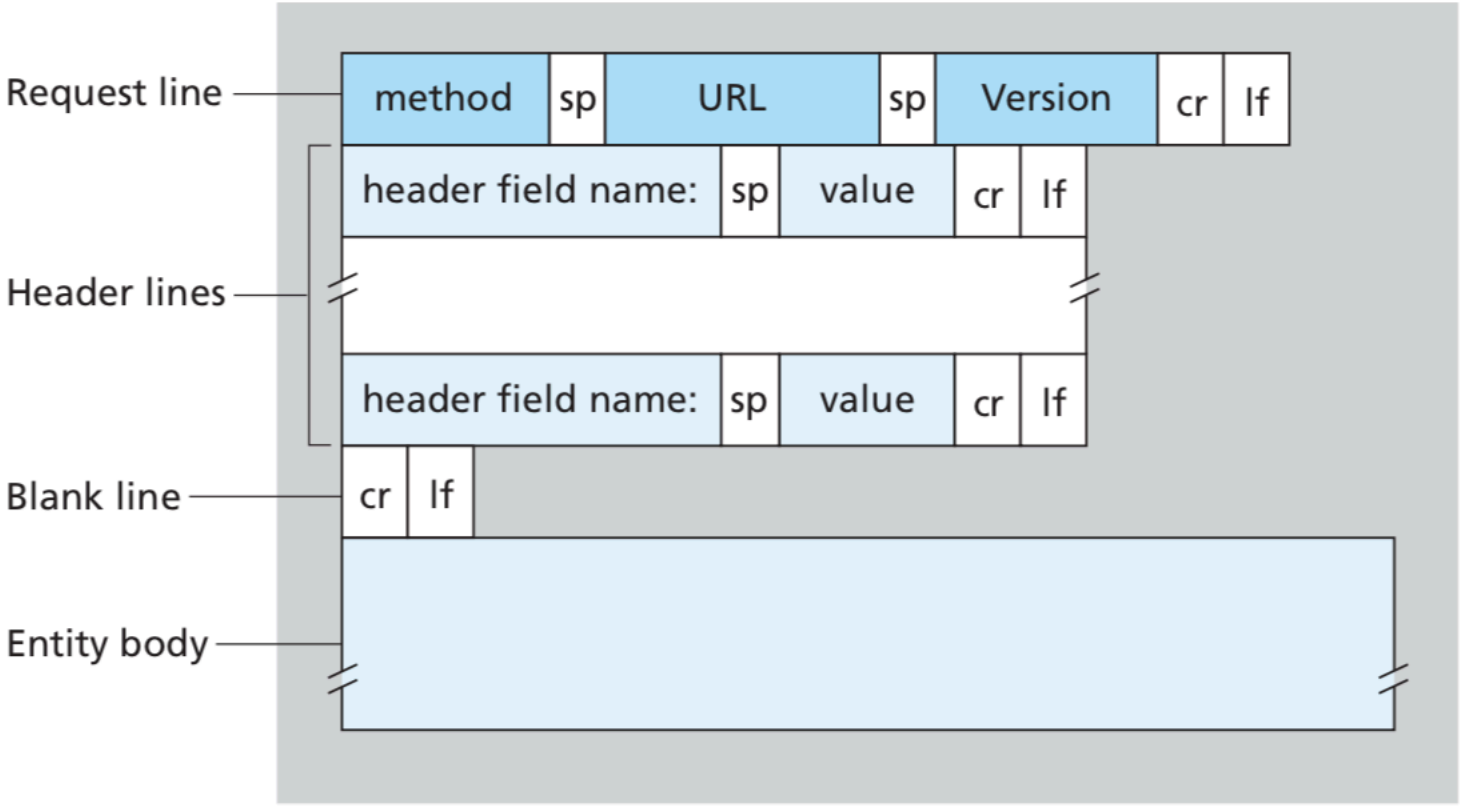
Sequential requests on one connection

Pipelined requests on one connection

HTTP Request

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

General Format



Example

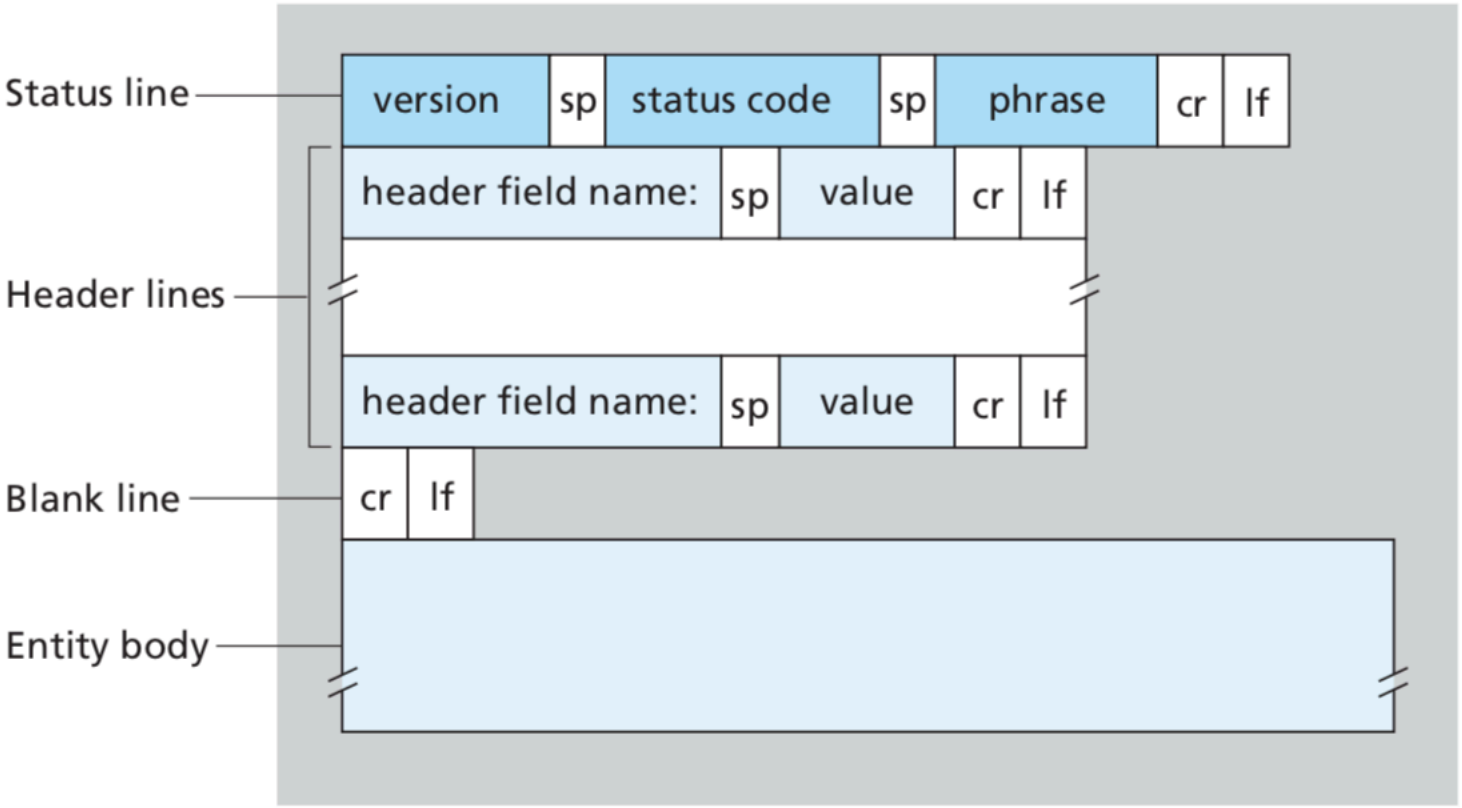
Request Methods

Fetch a page →

Used to send input data to a server program →

Method	Description
GET	Read a Web page
HEAD	Read a Web page's header
POST	Append to a Web page
PUT	Store a Web page
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Connect through a proxy
OPTIONS	Query options for a page

HTTP Response - General Format



HTTP Response - Example

```
HTTP/1.1 200 OK Connection: close
Date: Tue, 09 Aug 2011 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data ...)
```

Response Code

- The status line contains a three-digit status code telling whether the request was satisfied and, if not, why not.

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

Header Fields

- Many headers carry (key, value) information:

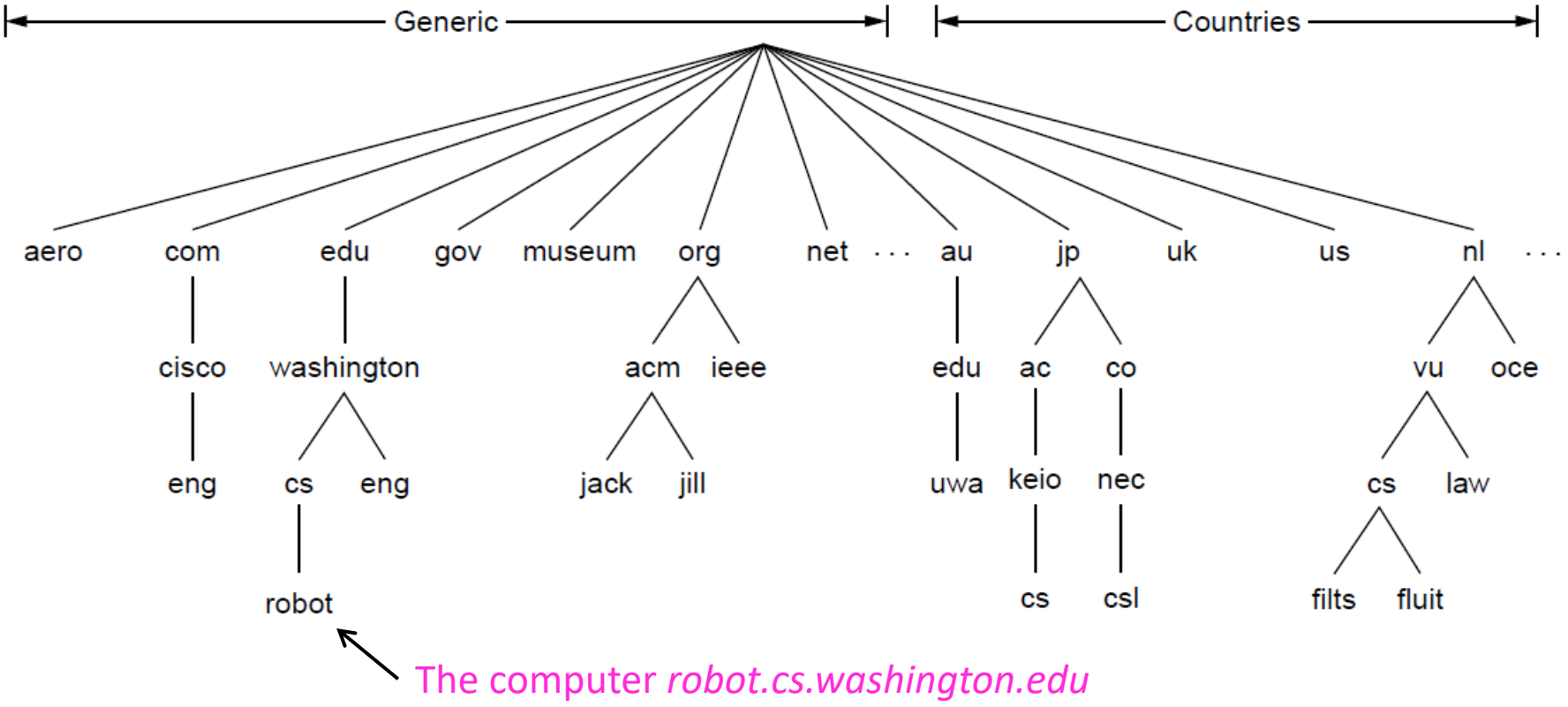
Function	**Example Headers**
Browser capabilities (client → server)	User-Agent, Accept, Accept-Charset, Accept-Encoding, Accept-Language
Caching related (mixed directions)	If-Modified-Since, If-None-Match, Date, Last-Modified, Expires, Cache-Control, ETag
Browser context (client → server)	Cookie, Referer, Authorization, Host
Content delivery (server → client)	Content-Encoding, Content-Length, Content-Type, Content-Language, Content-Range, Set-Cookie

DNS – Domain Name System

- The DNS resolves high-level human readable names for computers to low-level IP addresses
  - DNS name space
  - Domain Resource records
  - Name servers

The DNS Name Space

- DNS namespace is hierarchical from the root down
- Different parts delegated to different organizations



Generic top-level domains

- Generic top-level domains (**TLD**) are controlled by ICANN who appoints registrars to run them

Domain	Intended use	Start date	Restricted?
com	Commercial	1985	No
edu	Educational institutions	1985	Yes
gov	Government	1985	Yes
int	International organizations	1988	Yes
mil	Military	1985	Yes
net	Network providers	1985	No
org	Non-profit organizations	1985	No
aero	Air transport	2001	Yes
biz	Businesses	2001	No
coop	Cooperatives	2001	Yes
info	Informational	2002	No
museum	Museums	2002	Yes
name	People	2002	No
pro	Professionals	2002	Yes
cat	Catalan	2005	Yes
jobs	Employment	2005	Yes
mobi	Mobile devices	2005	Yes
tel	Contact details	2005	Yes
travel	Travel industry	2005	Yes
xxx	Sex industry	2010	No

Domain Resource Records

- The key resource records in the namespace are IP addresses (A/AAAA) and name servers (NS), but there are others too (e.g., MX)

Type	Meaning	Value
SOA	Start of authority	Parameters for this zone
A	IPv4 address of a host	32-Bit integer
AAAA	IPv6 address of a host	128-Bit integer
MX	Mail exchange	Priority, domain willing to accept email
NS	Name server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
SPF	Sender policy framework	Text encoding of mail sending policy
SRV	Service	Host that provides it
TXT	Text	Descriptive ASCII text

Domain Resource Records

A portion of a possible DNS database for *cs.vu.nl*.

```
; Authoritative data for cs.vu.nl
cs.vu.nl.       86400   IN   SOA     star boss (9527,7200,7200,241920,86400)
cs.vu.nl.       86400   IN   MX      1 zephyr
cs.vu.nl.       86400   IN   MX      2 top
cs.vu.nl.       86400   IN   NS      star

star            86400   IN   A       130.37.56.205
zephyr          86400   IN   A       130.37.20.10
top             86400   IN   A       130.37.20.11
www             86400   IN   CNAME   star.cs.vu.nl
ftp             86400   IN   CNAME   zephyr.cs.vu.nl

flits           86400   IN   A       130.37.16.112
flits           86400   IN   A       192.31.231.165
flits           86400   IN   MX      1 flits
flits           86400   IN   MX      2 zephyr
flits           86400   IN   MX      3 top

rowboat                 IN   A       130.37.56.201
                        IN   MX      1 rowboat
                        IN   MX      2 zephyr

little-sister           IN   A       130.37.62.23

laserjet                IN   A       192.31.231.216
```

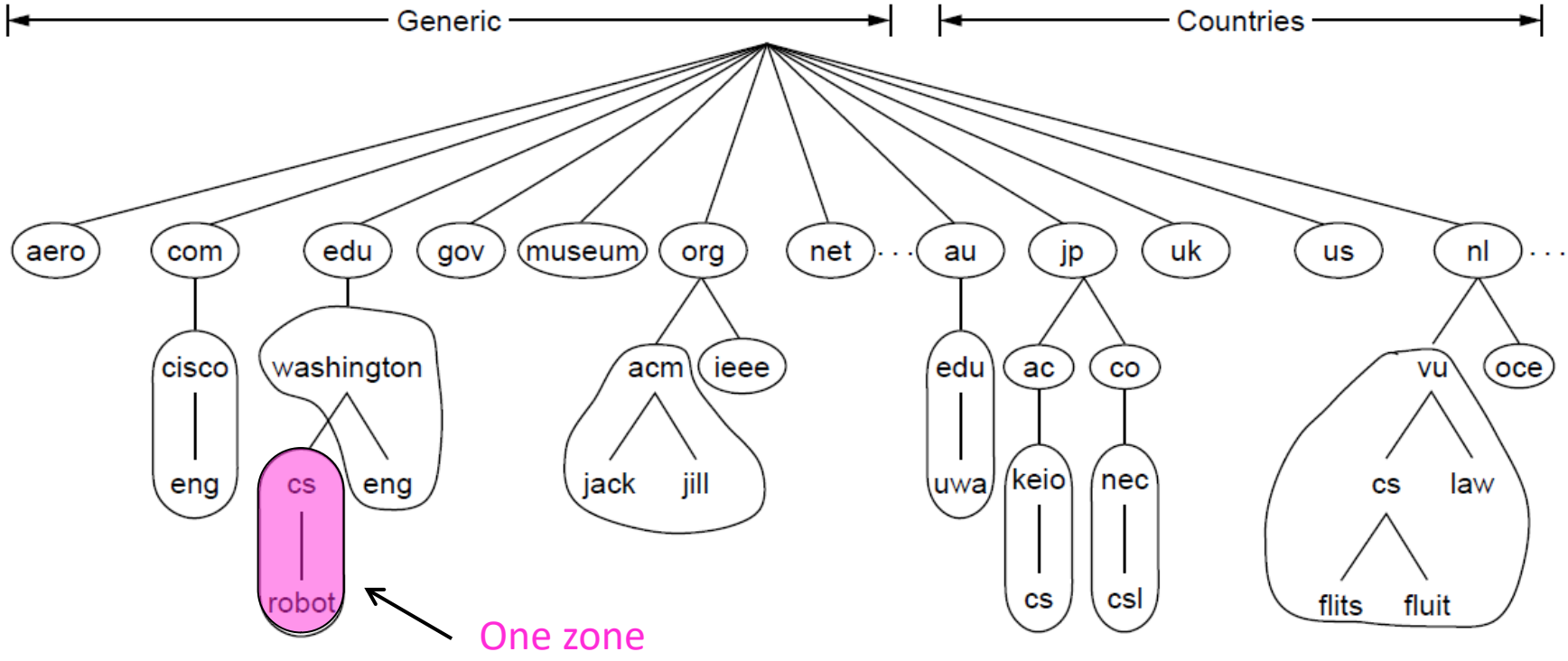
← Name server

← IP addresses of computers

← Mail gateways

Name Servers

- Name servers contain data for portions of the name space called zones (circled).

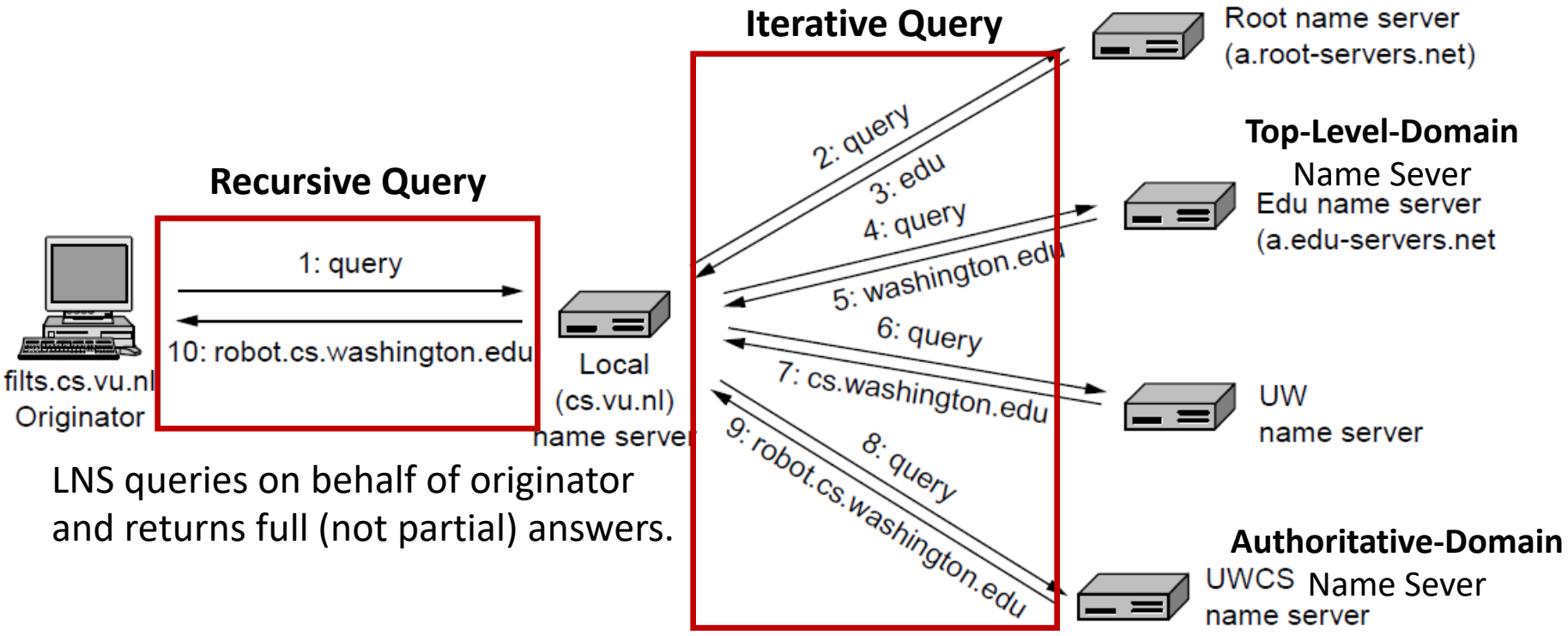


Name Servers

- Finding the IP address for a given hostname is called resolution and is done with the DNS protocol.

- Resolution:
  - Computer requests **local name server (LNS)** to resolve
    - LNS returns requested results if it exists in its cache, but if not:
  - Local name server asks the **root name server**
  - Root returns the name server for a lower zone
  - Continue down zones until name server can answer
- DNS protocol:
  - Runs on UDP port 53, retransmits lost messages
  - LNS **caches** name server answers for better performance

DNS Resolution



LNS queries on behalf of originator and returns full (not partial) answers.

Example of a computer looking up the IP for a name:
robot.cs.washington.edu

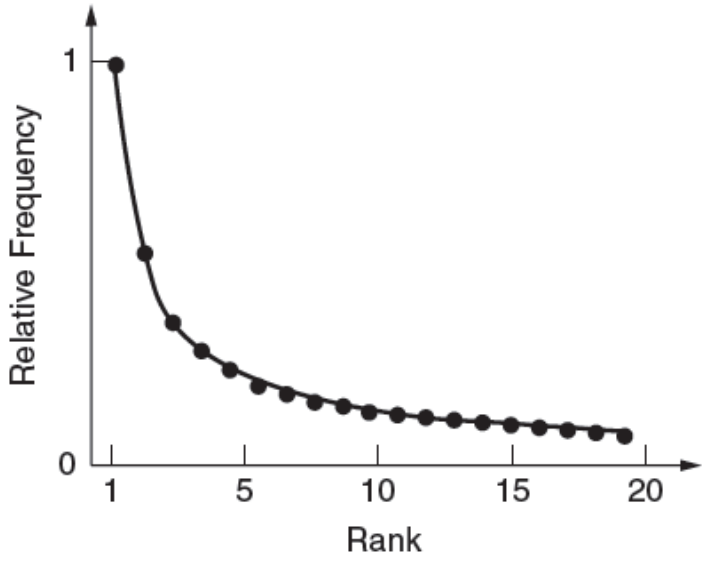
“Authoritative”

- An **authoritative record** is one that comes from the authority that manages the record and is thus always correct.

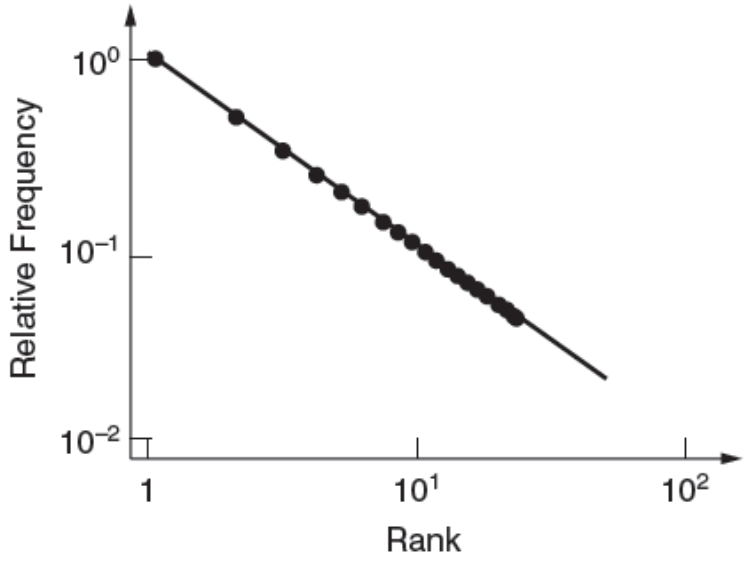
- Authoritative records are in contrast to **cached records**, which may be out of date.

Content and Internet Traffic

- Delivery of content, especially Web and video, to users is a major component of Internet traffic
- Internet traffic:
  - Shifts seismically (email→FTP→Web→P2P→video)
  - Has <u>many</u> small/unpopular and <u>few</u> large/popular flows



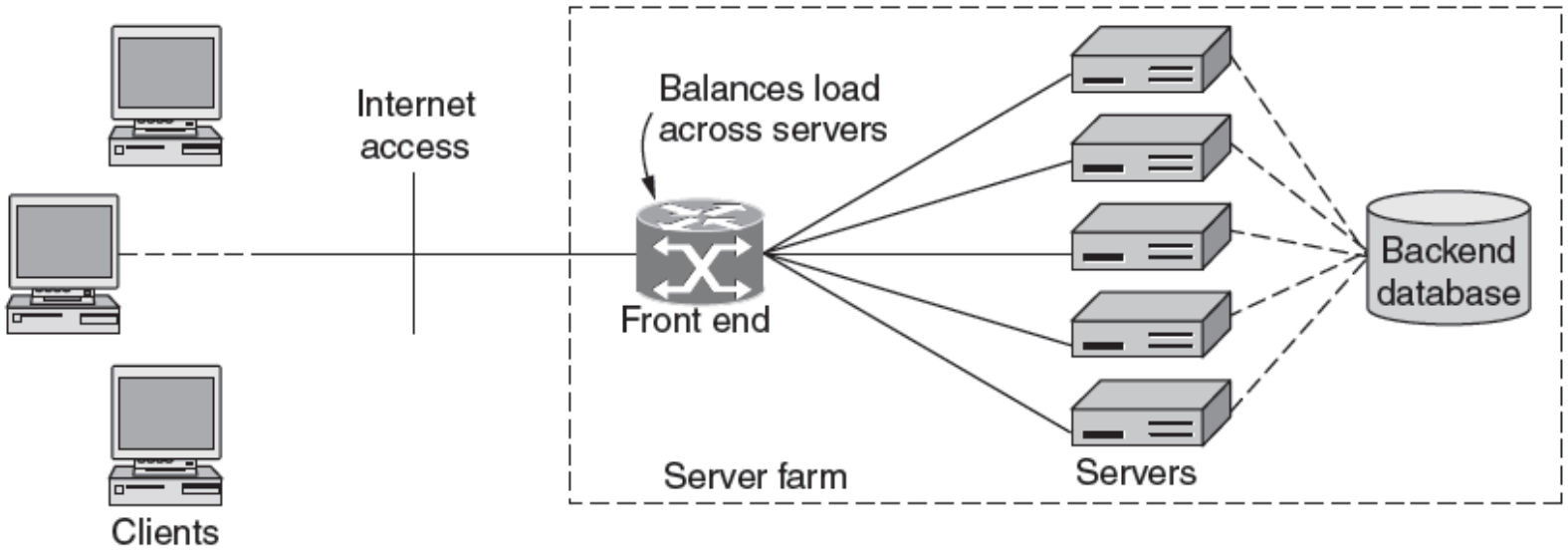
Zipf popularity distribution, 1/k



Shows up as a line on log-log plot

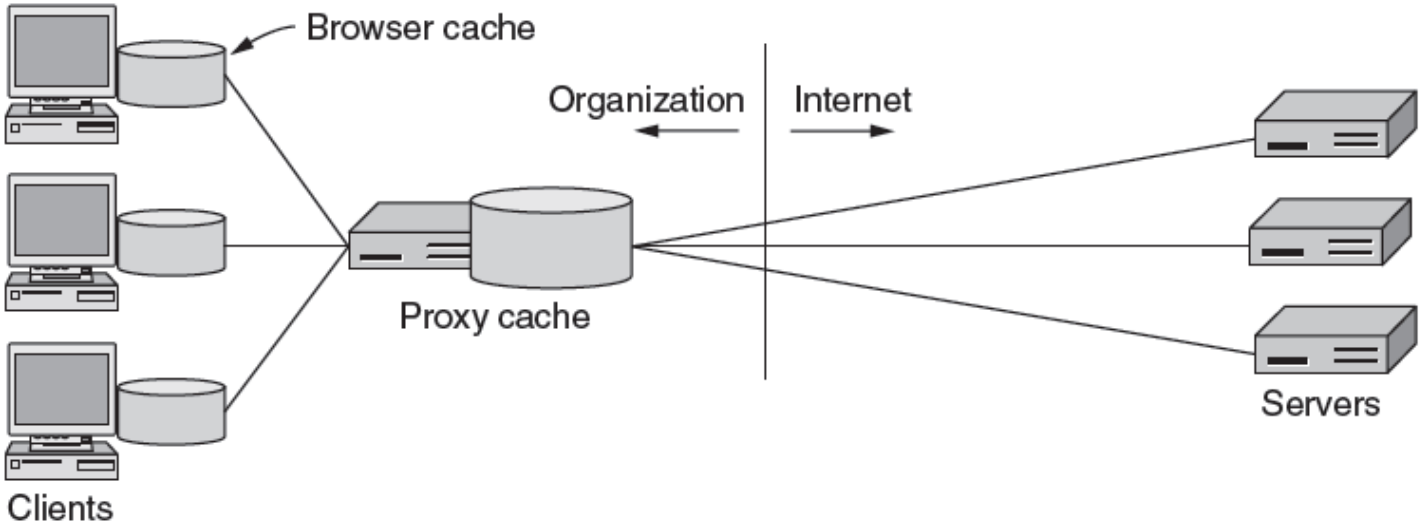
Server Farms

- Server farms enable large-scale Web servers:
  - Front-end load-balances requests over servers
  - Servers access the same backend database



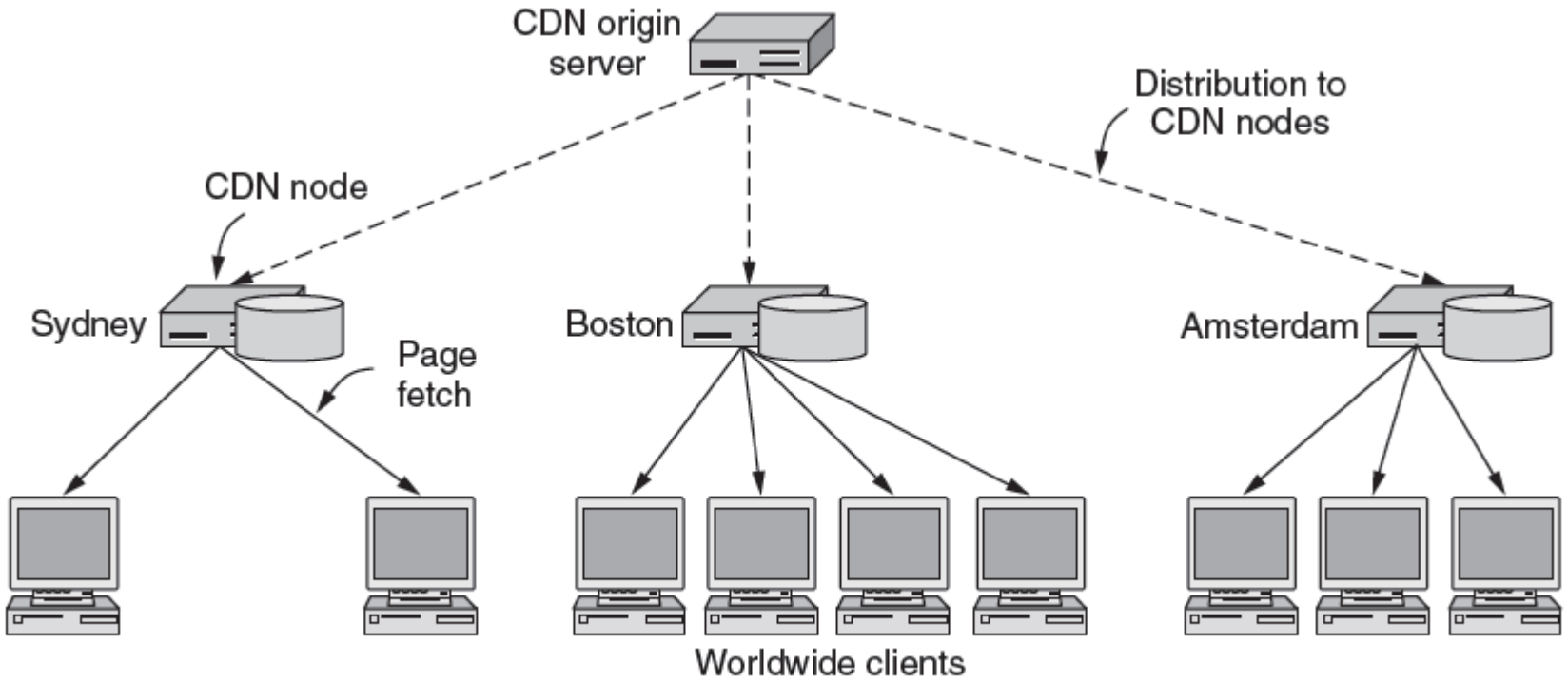
Web Proxies

- Proxy caches help organizations to scale the Web
  - Caches server content over clients for performance
  - Also implements organization policies (e.g., access)



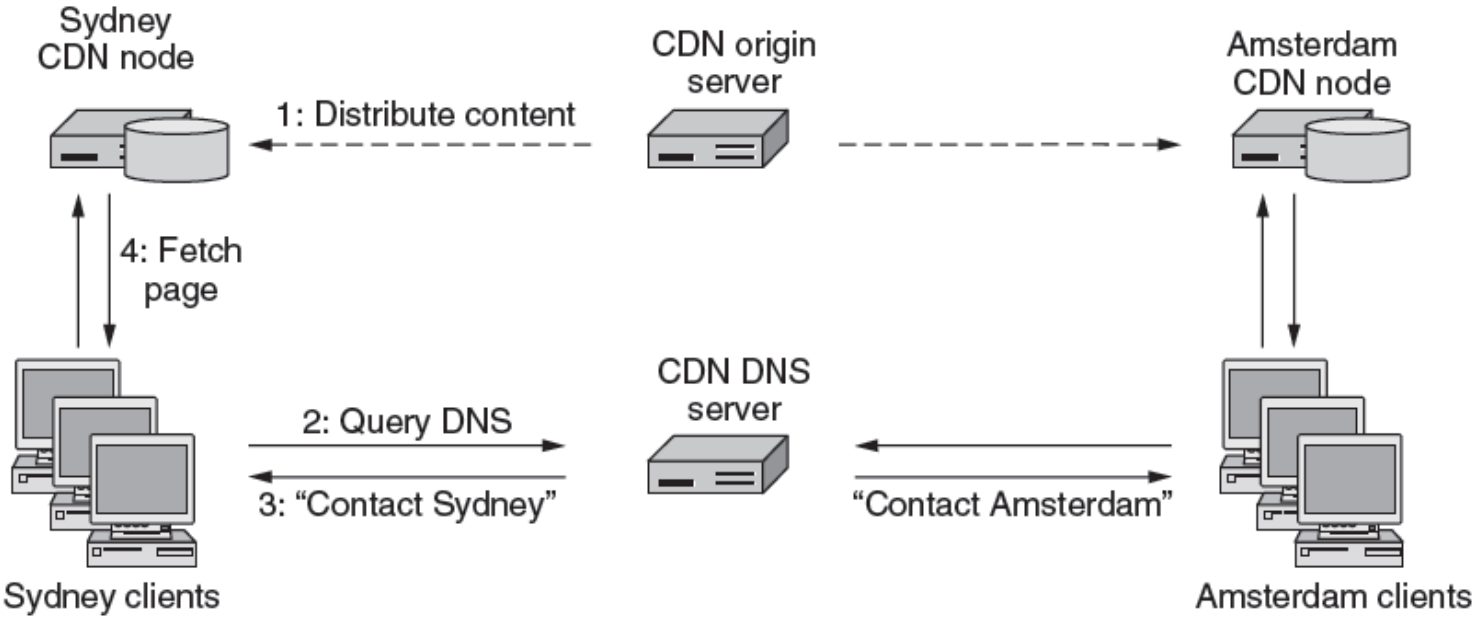
Content Delivery Networks (CDNs)

- CDNs scale Web servers by having clients get content from a nearby CDN node (cache)



CDNs

- Directing clients to nearby CDN nodes with DNS:
  - Client query returns local CDN node as response
  - Local CDN node caches content for nearby clients and reduces load on the origin server



CDNs

```
<html>
<head> <title> Fluffy Video </title> </head>
<body>
<h1> Fluffy Video's Product List </h1>
<p> Click below for free samples. </p>

<a href="koalas.mpg"> Koalas Today </a> <br>
<a href="kangaroos.mpg"> Funny Kangaroos </a> <br>
<a href="wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
```

Traditional Web page on server

Origin server rewrites pages to serve content via CDN

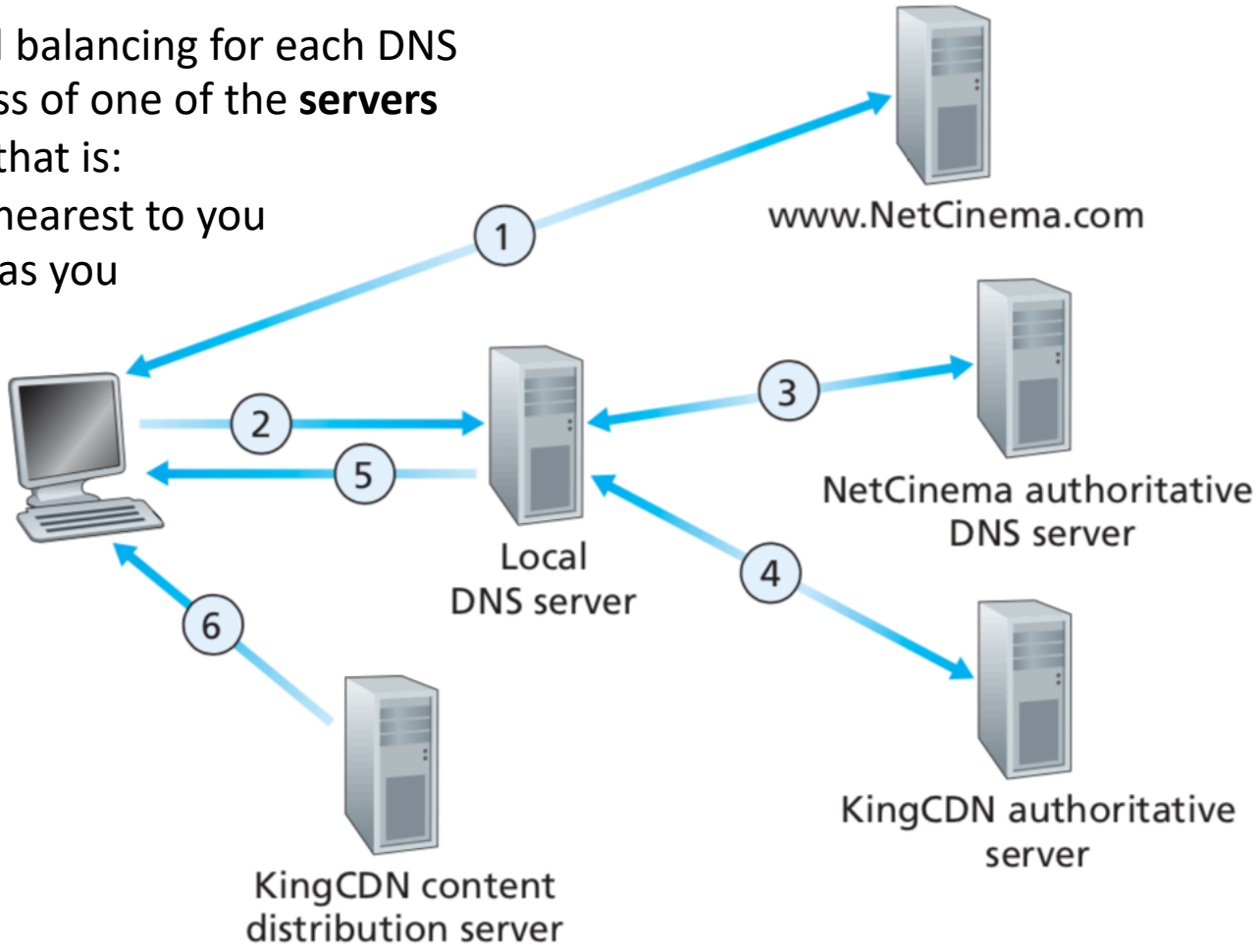
```
<html>
<head> <title> Fluffy Video </title> </head>
<body>
<h1> Fluffy Video's Product List </h1>
<p> Click below for free samples. </p>

<a href="http://www.cdn.com/fluffyvideo/koalas.mpg"> Koalas Today </a> <br>
<a href="http://www.cdn.com/fluffyvideo/kangaroos.mpg"> Funny Kangaroos </a> <br>
<a href="http://www.cdn.com/fluffyvideo/wombats.mpg"> Nice Wombats </a> <br>
</body>
</html>
```

Page that distributes content via CDN

DNS + CDN = Load Balancing

The results of this load balancing for each DNS would be the IP address of one of the **servers** by requested website that is:

- geographically the nearest to you
- using the same ISP as you



Sample Steps

1. The user visits the Web page at NetCinema

2. When the user clicks on the link `http://video.netcinema.com/6Y7B23V`, the user's host sends a DNS query for `video.netcinema.com`.

3. The user's Local DNS Server (LDNS) relays the DNS query to an authoritative DNS server for NetCinema, which observes the string "video" in the hostname `video.netcinema.com`.
   - To "hand over" the DNS query to KingCDN, instead of returning an IP address, the NetCinema authoritative DNS server returns to the LDNS a hostname (**CNAME**) in the KingCDN's domain, for example, `a1105.kingcdn.com`.

Sample Steps

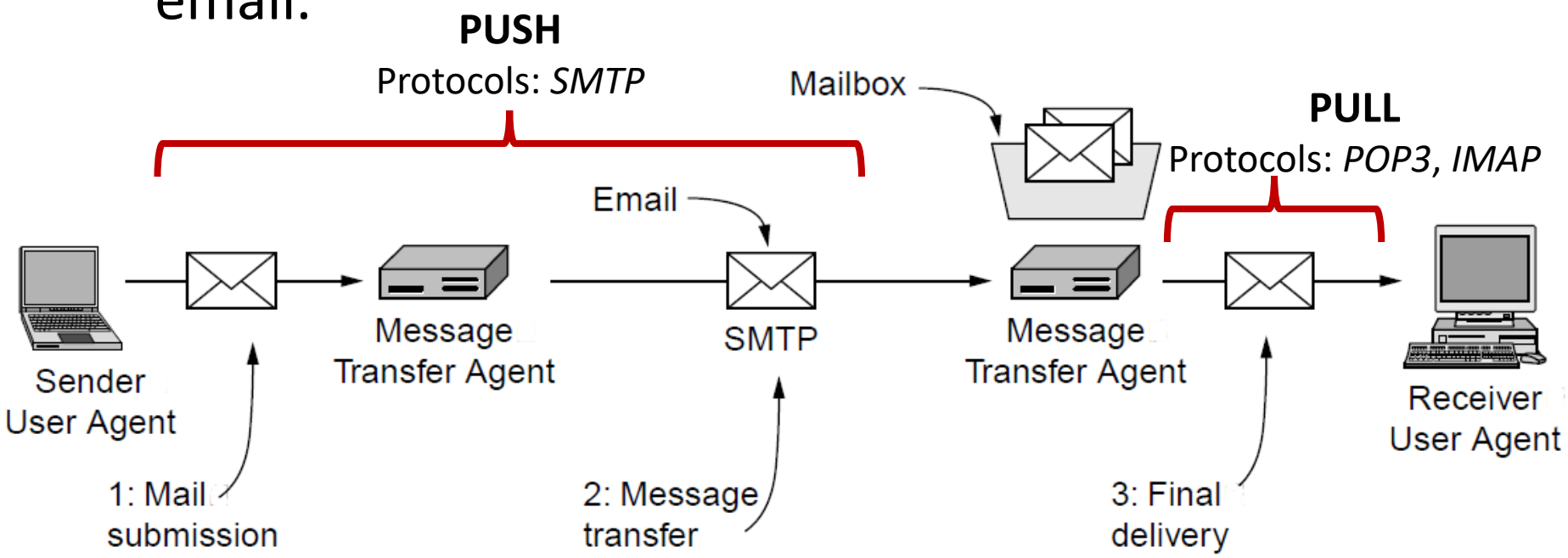
4. The user's LDNS then sends a second query, now for `a1105.kingcdn.com`, and KingCDN's DNS system eventually returns the IP addresses of a KingCDN **content server** to the LDNS.

5. The LDNS forwards the IP address of the content-serving CDN node to the user's host.

6. Once the client receives the IP address for a KingCDN content server, it establishes a direct TCP connection with the server at that IP address and issues an HTTP GET request for the video.

Electronic Mail

- The key components and steps (numbered) to send email.



Architecture of the email system