CS4220
Computer Networks

Lecture 8 The Application Layer

Dr. Xiaobo "Charles" Zhou
Department of Computer Science

1

Application Layer
Chapter 7

- DNS – Domain Name System
- Electronic Mail
- The Web
- Streaming Audio and Video
- Content Delivery

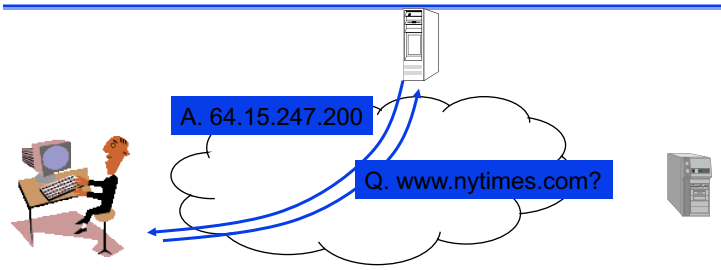
2

The Application Layer

- Uses transport services to build distributed applications

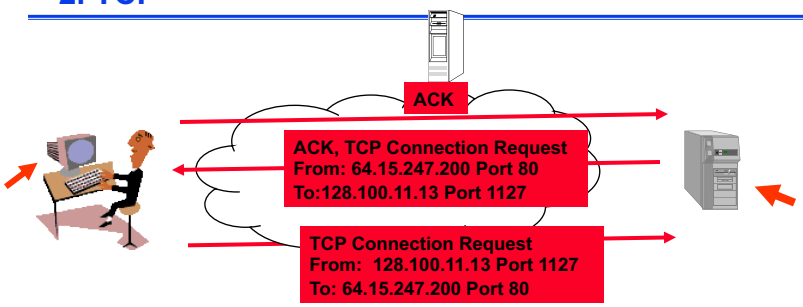
Application
Transport
Network
Link
Physical

1. DNS



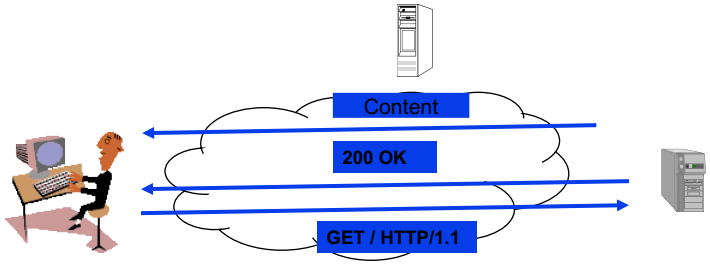
- User clicks on http://www.nytimes.com/
- URL contains Internet name of machine (www.nytimes.com), but not Internet address
- Internet needs Internet address to send information to a machine
- Browser software uses Domain Name System (DNS) protocol to send query for Internet address
- DNS system responds with Internet address

2. TCP



- Browser software uses HyperText Transfer Protocol (HTTP) to send request for document
- HTTP server waits for requests by listening to a well-known port number (80 for HTTP)
- HTTP client sends request messages through an "ephemeral port number," e.g. 1127
- HTTP needs a Transmission Control Protocol (TCP) connection between the HTTP client and the HTTP server to transfer messages reliably

3. HTTP



- HTTP client sends its request message: "GET …"
- HTTP server sends a status response: "200 OK"
- HTTP server sends requested file
- Browser displays document
- Clicking a link sets off a chain of events across the Internet!
- Let's see how protocols & layers come into play…

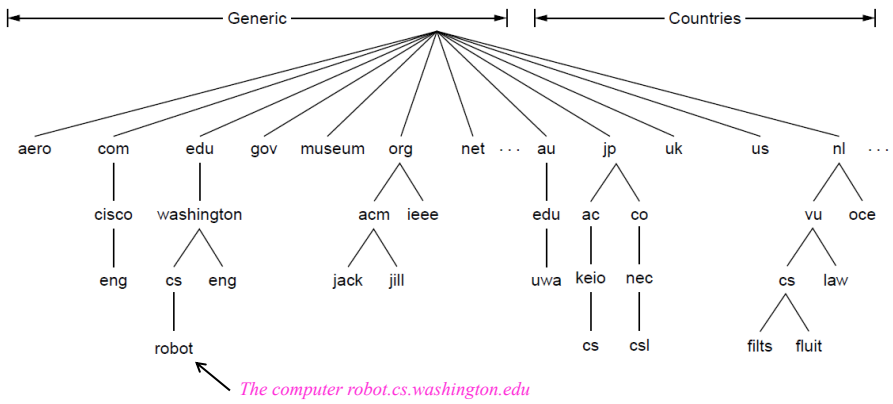
DNS – Domain Name System

- 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 (1)

DNS namespace is hierarchical from the root down

- Different parts delegated to different organizations



The DNS Name Space (2)

- **Generic top-level domains 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

This one was controversial

Domain Resource Records (1)

- **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

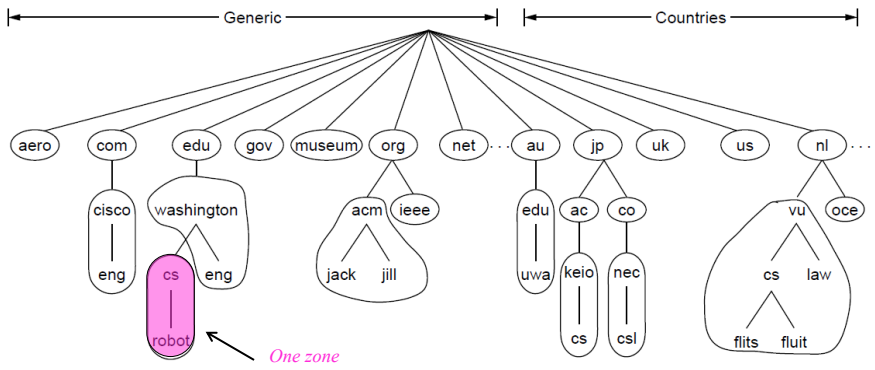
Domain Resource Records (2)

```
; 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                    <-- Name server

star            86400   IN   A       130.37.56.205
zephyr          86400   IN   A       130.37.20.10
top             86400   IN   A       130.37.20.11            <-- IP addresses of computers
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               <-- Mail gateways
                        IN   MX      2 zephyr

little-sister           IN   A       130.37.62.23

laserjet                IN   A       192.31.231.216
```

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

Name Servers (1)

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

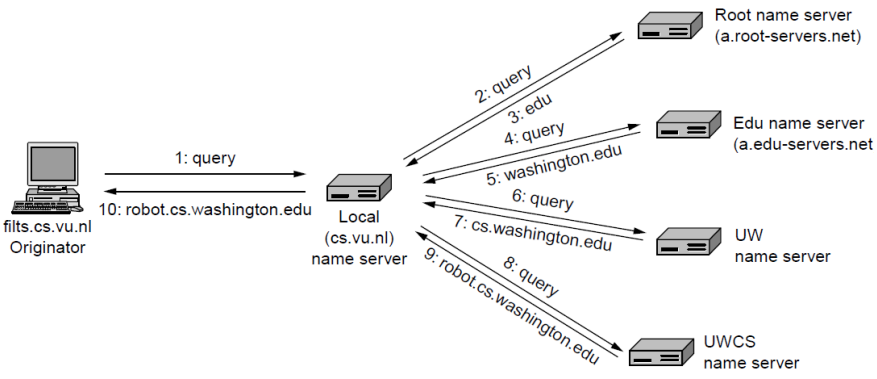


Name Servers (2)

- Finding the IP address for a given hostname is called resolution and is done with the DNS protocol.
- Resolution:
  - Computer requests local name server to resolve
  - 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
  - Caches name server answers for better performance

Name Servers (3)

- Example of a computer looking up the IP for a name

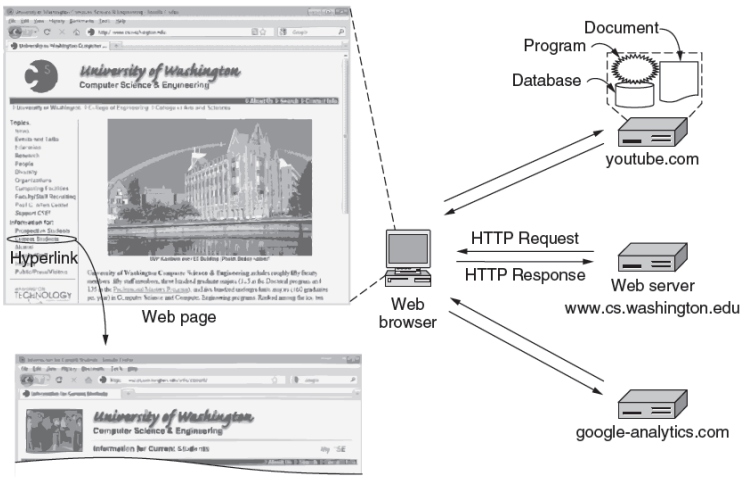


The World Wide Web

- Architectural overview »
- Static Web pages »
- Dynamic pages and Web applications »
- HTTP – HyperText Transfer Protocol »
- The mobile Web »
- Web search »

Architectural Overview (1)

° HTTP transfers pages from servers to browsers



Architectural Overview (2)

° **Pages are named with URLs (Uniform Resource Locators)**

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

Protocol *Server* *Page on server*

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

Architectural Overview (3)

Steps a client (browser) takes to follow a hyperlink:

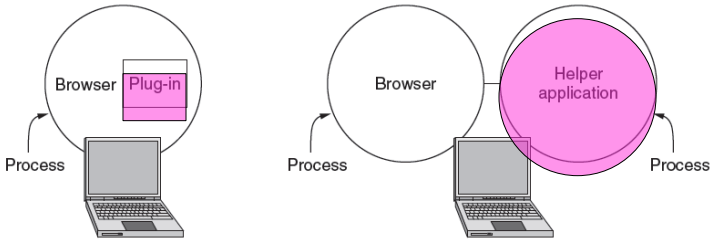
- Determine the protocol (HTTP)
- Ask DNS for the IP address of server
- Make a TCP connection to server
- Send request for the page; server sends it back
- Fetch other URLs as needed to display the page
- Close idle TCP connections

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

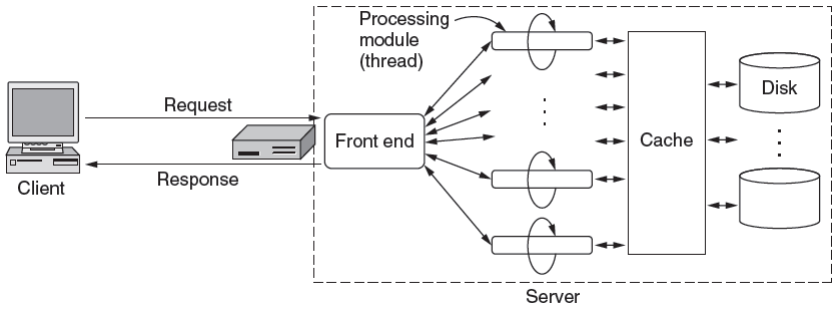
Architectural Overview (4)

- Content type is identified by MIME types
  - Browser takes the appropriate action to display
  - Plug-ins / helper apps extend browser for new types



Architectural Overview (5)

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



Architectural Overview (6)

Server steps, revisited:

- Resolve name of Web page requested
- Perform access control on the Web page
- Check the cache
- Fetch requested page from disk or run program
- Determine the rest of the response
- Return the response to the client
- Make an entry in the server log

Architectural Overview (7)

- **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

Static Web Pages (1)

- Static Web pages are simply files
  - Have the same contents for each viewing
- Can be visually rich and interactive nonetheless:
  - HTML that mixes text and images
  - Forms that gather user input
  - Style sheets that tailor presentation
  - Vector graphics, videos, and more (over) . . .

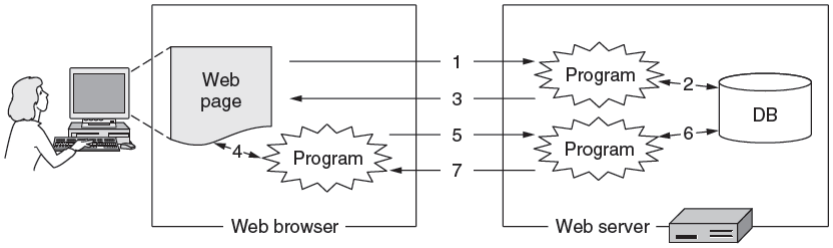
Static Web Pages (2)

Progression of features through HTML 5.0

Item	HTML 1.0	HTML 2.0	HTML 3.0	HTML 4.0	HTML 5.0
Hyperlinks	x	x	x	x	x
Images	x	x	x	x	x
Lists	x	x	x	x	x
Active maps & images		x	x	x	x
Forms		x	x	x	x
Equations			x	x	x
Toolbars			x	x	x
Tables			x	x	x
Accessibility features				x	x
Object embedding				x	x
Style sheets				x	x
Scripting				x	x
Video and audio					x
Inline vector graphics					x
XML representation					x
Background threads					x
Browser storage					x
Drawing canvas					x

Dynamic Pages & Web Applications (1)

- **Dynamic pages are generated by programs running at the server (with a database) and the client**
  - **E.g., PHP at server, JavaScript at client**
  - **Pages vary each time like using an application**



Dynamic Pages & Web Applications (2)

Web page that gets form input and calls a server program

```
<html>
<body>
<form action="action.php" method="post">
<p> Please enter your name: <input type="text" name="name"> </p>
<p> Please enter your age: <input type="text" name="age"> </p>
<input type="submit">
</form>
</body>
</html>
```

PHP server program that creates a custom Web page

PHP calls

```
<html>
<body>
<h1> Reply: </h1>
Hello <?php echo $name; ?>.
Prediction: next year you will be <?php echo $age + 1; ?>
</body>
</html>
```

Resulting Web page (for inputs "Barbara" and "32")

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 33
</body>
</html>
```

Dynamic Pages & Web Applications (3)

JavaScript program produces result page in the browser

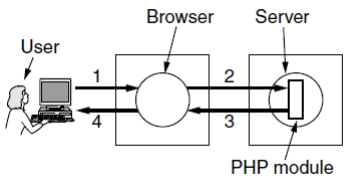
```
<html>
<head>
<script language="javascript" type="text/javascript">
function response(test_form) {
   var person = test_form.name.value;
   var years = eval(test_form.age.value) + 1;
   document.open();
   document.writeln("<html> <body>");
   document.writeln("Hello " + person + ".<br>");
   document.writeln("Prediction: next year you will be " + years + ".");
   document.writeln("</body> </html>");
   document.close();
}
</script>
</head>
```

First page with form, gets input and calls program above

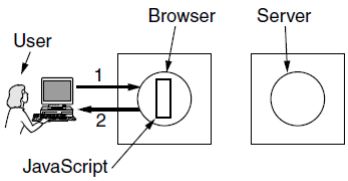
```
<body>
<form>
Please enter your name: <input type="text" name="name">
<p>
Please enter your age: <input type="text" name="age">
<p>
<input type="button" value="submit" onclick="response(this.form)">
</form>
</body>
</html>
```


Dynamic Pages & Web Applications (4)

- **The difference between server and client programs**



Server-side scripting with PHP



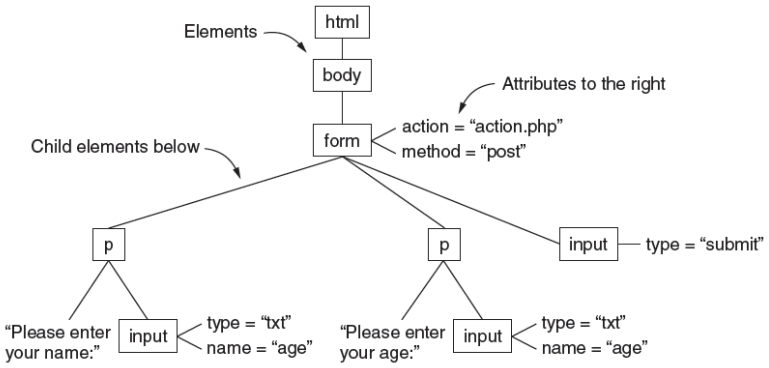
Client-side scripting with JavaScript

Dynamic Pages & Web Applications (5)

- **Web applications use a set of technologies that work together, e.g. AJAX:**
  - **HTML: present information as pages.**
  - **DOM: change parts of pages while they are viewed.**
  - **XML: let programs exchange data with the server.**
  - **Asynchronous way to send and retrieve XML data.**
  - **JavaScript as a language to bind all this together.**

Dynamic Pages & Web Applications (6)

- **The DOM (Document Object Model) tree represents Web pages as a structure that programs can alter**



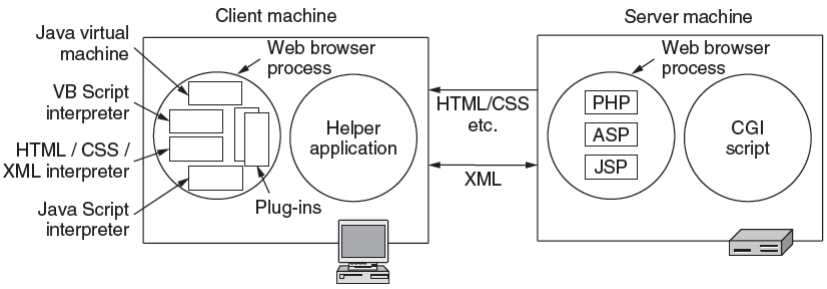
Dynamic Pages & Web Applications (7)

XML captures document structure, not presentation like HTML. Ex:

```
<?xml version="1.0" ?>

<book_list>

<book>
   <title> Human Behavior and the Principle of Least Effort </title>
   <author> George Zipf </author>
   <year> 1949 </year>
</book>

<book>
   <title> The Mathematical Theory of Communication </title>
   <author> Claude E. Shannon </author>
   <author> Warren Weaver </author>
   <year> 1949 </year>
</book>

<book>
   <title> Nineteen Eighty-Four </title>
   <author> George Orwell </author>
   <year> 1949 </year>
</book>

</book_list>
```


Dynamic Pages & Web Applications (8)

Web applications use a set of technologies, revisited:

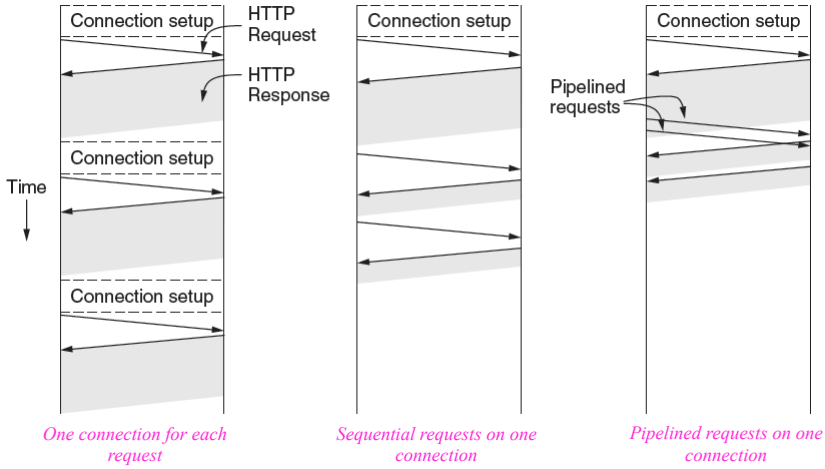


HTTP (1)

- HTTP (HyperText Transfer Protocol) is a request-response protocol that runs on top of TCP
  - 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 (2)

- HTTP uses persistent connections to improve performance



HTTP (3)

HTTP has several 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 (4)

Response codes tell the client how the request fared:

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

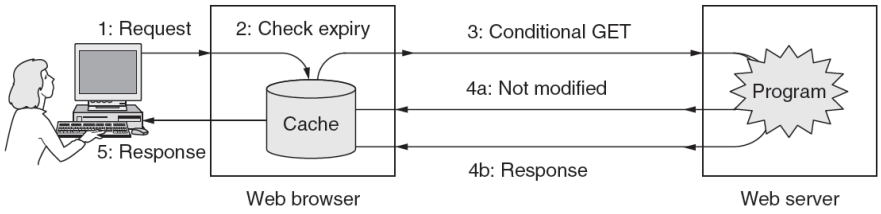
HTTP (5)

Many headers carry key 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

HTTP (6)

- **HTTP caching checks to see if the browser has a known fresh copy, and if not if the server has updated the page**
  - **Uses a collection of headers for the checks**
  - **Can include further levels of caching (e.g., proxy)**



The Mobile Web

- Mobiles (phones, tablets) are challenging as clients:
  - Relatively small screens
  - Limited input capabilities, lengthy input.
  - Network bandwidth is limited
  - Connectivity may be intermittent.
  - Computing power is limited
- Strategies to handle them:
  - Content: servers provide mobile-friendly versions; transcoding can also be used
  - Protocols: no real need for specialized protocols; HTTP with header compression sufficient

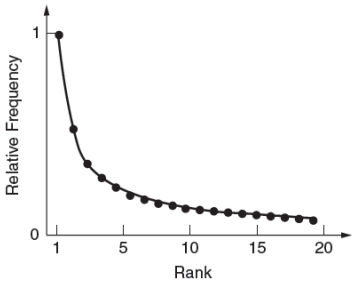
Content Delivery

- Delivery of content, especially Web and video, to users is a major component of Internet traffic
  - Content and Internet traffic »
  - Server farms and Web proxies »
  - Content delivery networks »
  - Peer-to-peer networks »

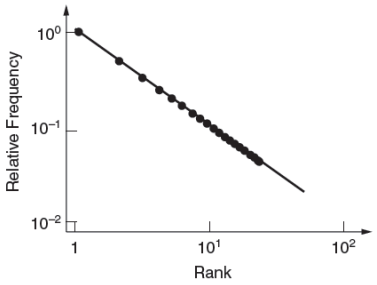
Content and Internet Traffic

Internet traffic:

1. **Shifts seismically (email→FTP→Web→P2P→video)**
2. **Has many small/unpopular and few large/popular flows – mice and elephants**



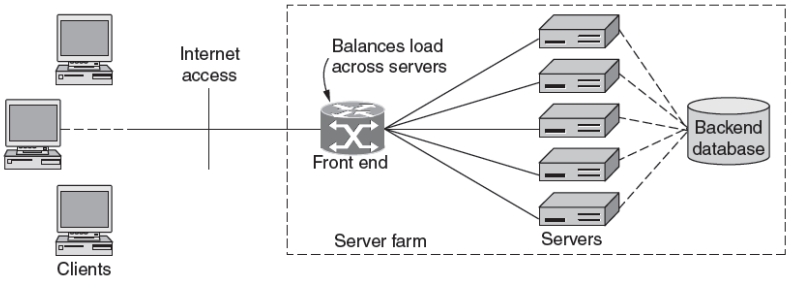
Zipf popularity distribution, 1/k



Shows up as a line on log-log plot

Server Farms and Web Proxies (1)

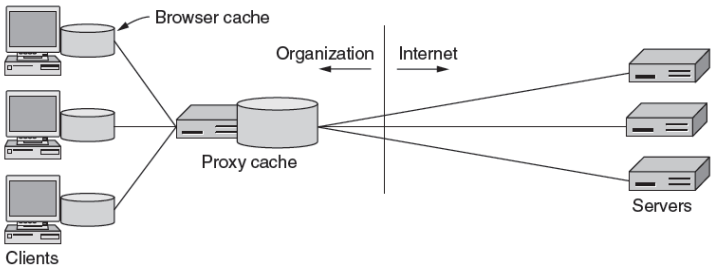
- **Server farms enable large-scale Web servers:**
  - **Front-end load-balances requests over servers**
  - **Servers access the same backend database**
  - **Cloud Computing**



Server Farms and Web Proxies (2)

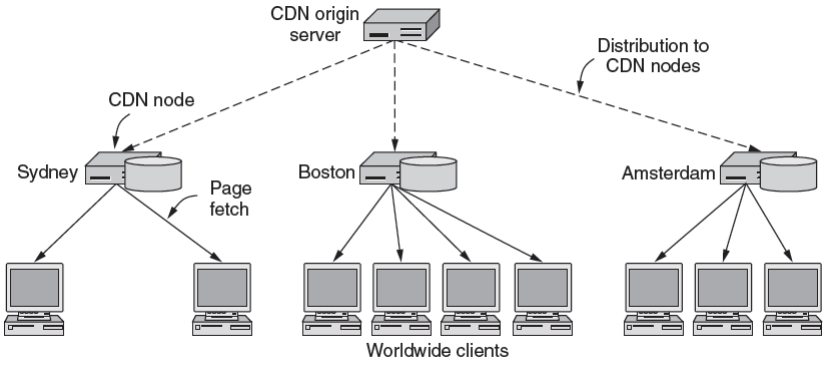
Proxy caches help organizations to scale the Web

- **Caches server content over clients for performance**
- **Also implements organization policies (e.g., access)**



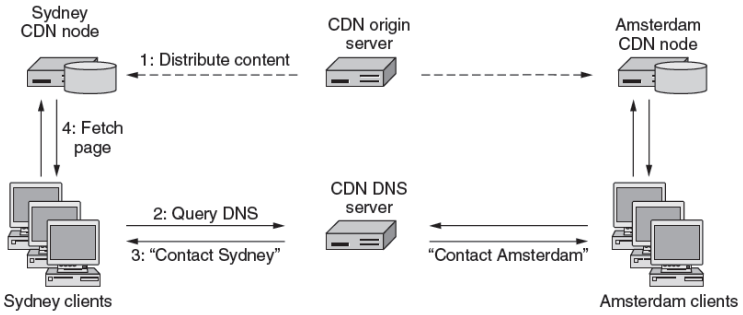
CDNs – Content Delivery Networks (1)

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



Content Delivery Networks (2)

- 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

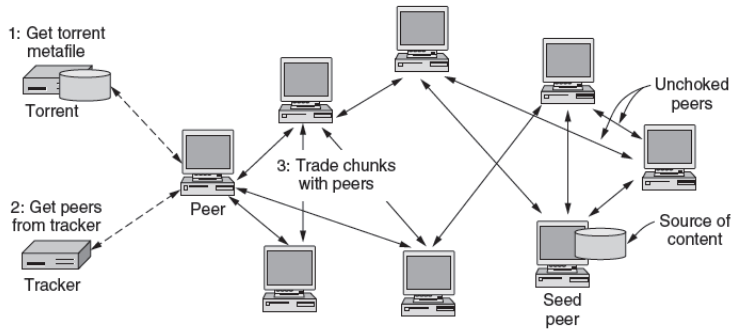


Peer-to-Peer Networks (1)

- P2P (Peer-to-Peer) is an alternative CDN architecture with no dedicated infrastructure (i.e., servers)
  - Clients serve content to each other as peers
- Challenges when servers are removed:
  1. How do peers find each other?
  2. How do peers support rapid content downloads?
  3. How do peers encourage each other to upload?

Peer-to-Peer Networks (2)

- BitTorrent lets peers download torrents
  - Peers find each other via Tracker in torrent file
  - Peers swap chunks (parts of content) with partners, preferring those who send most quickly [2]
  - Many peers speed download; preference helps uploads [3]



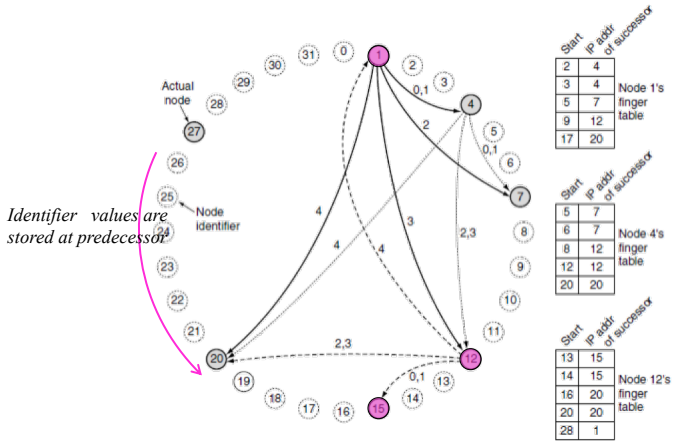
Peer-to-Peer Networks (3)

- Distributed Hash Tables (DHTs) are a fully distributed index that scales to very many clients/entries
  - Need to follow O(log N) path for N entries
  - Can use as Tracker to find peers with no servers [1]
  - Look up torrent (identifier) in DHT to find IP of peers
  - Kademlia is used in BitTorrent

Peer-to-Peer Networks (3)

° A Chord ring of 32 identifiers. Finger tables [at right, and as arcs] are used to navigate the ring.

- Example: path to look up 16 from 1 → 12 → 15



Reading

° Chapter 7 of the text