



---

# Computer Networks 1

## (Mạng Máy Tính 1)

Lectured by: Dr. Phạm Trần Vũ

MEng. Nguyễn Cao Đạt

[cuu duong than cong . com](http://cuuduongthancong.com)



---

# Lecture 10: Application Layer

cuu duong than cong . com



# Application Layer

---

- ❑ Where our applications are running
- ❑ Using services provided by layers below
- ❑ We will look at:
  - Domain Name System
  - Email
  - File Transfer Protocol
  - World Wide Web



# Domain Name System - DNS

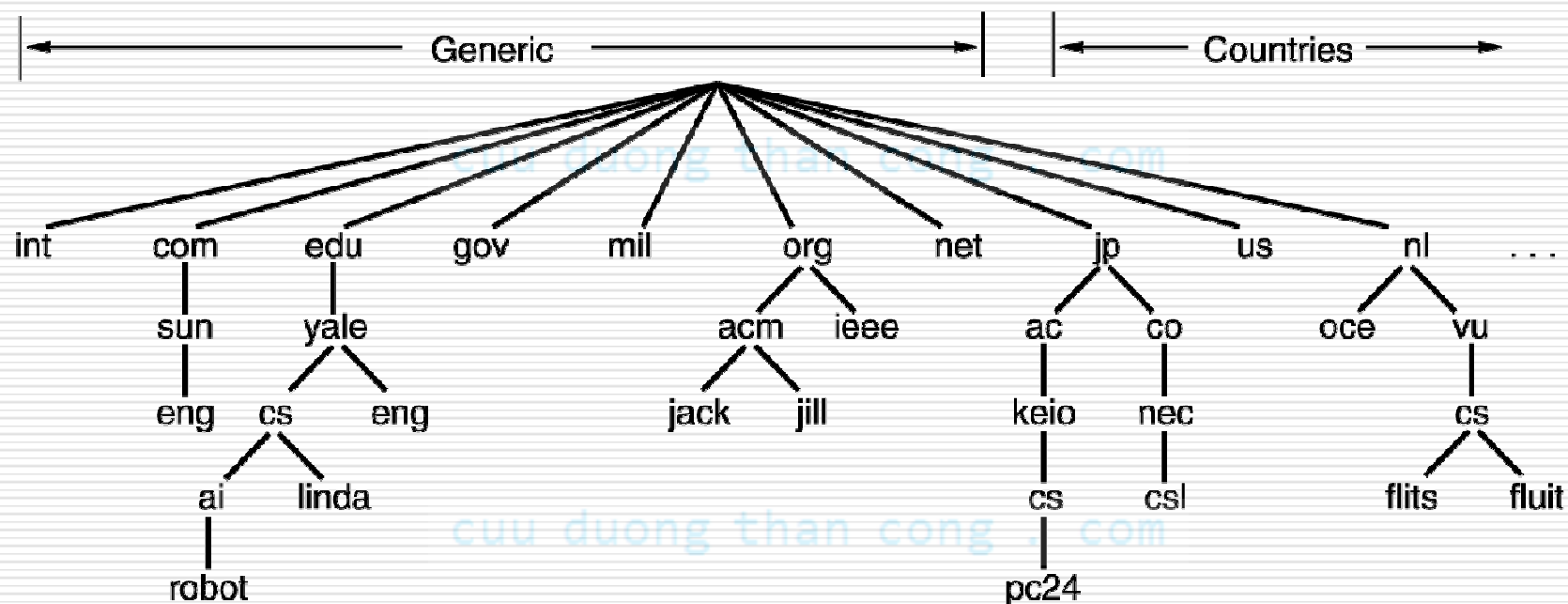
---

- ❑ IP addresses can be used to identify a host machine on the Internet
  - As people moves around, the addresses need to be changed accordingly as well
- ❑ ASCII names have been invented to decouple host names and their IPs to provide more flexibility
- ❑ The DNS was invented to manage and resolve host names into IP addresses



# The DNS Name Space

A portion of the Internet domain name space.





# Resource Records

- Every domain has a set of records associated with
- The principal DNS resource records types.

| Type  | Meaning              | Value                                     |
|-------|----------------------|-------------------------------------------|
| SOA   | Start of Authority   | Parameters for this zone                  |
| A     | IP address of a host | 32-Bit integer                            |
| MX    | Mail exchange        | Priority, domain willing to accept e-mail |
| NS    | Name Server          | Name of a server for this domain          |
| CNAME | Canonical name       | Domain name                               |
| PTR   | Pointer              | Alias for an IP address                   |
| HINFO | Host description     | CPU and OS in ASCII                       |
| TXT   | Text                 | Uninterpreted ASCII text                  |



# Resource Records (2)

|                                   |       |    |       |                                            |
|-----------------------------------|-------|----|-------|--------------------------------------------|
| ; Authoritative data for cs.vu.nl |       |    |       |                                            |
| cs.vu.nl.                         | 86400 | IN | SOA   | star boss (952771,7200,7200,2419200,86400) |
| cs.vu.nl.                         | 86400 | IN | TXT   | "Divisie Wiskunde en Informatica."         |
| cs.vu.nl.                         | 86400 | IN | TXT   | "Vrije Universiteit Amsterdam."            |
| cs.vu.nl.                         | 86400 | IN | MX    | 1 zephyr.cs.vu.nl.                         |
| cs.vu.nl.                         | 86400 | IN | MX    | 2 top.cs.vu.nl.                            |
|                                   |       |    |       |                                            |
| flits.cs.vu.nl.                   | 86400 | IN | HINFO | Sun Unix                                   |
| flits.cs.vu.nl.                   | 86400 | IN | A     | 130.37.16.112                              |
| flits.cs.vu.nl.                   | 86400 | IN | A     | 192.31.231.165                             |
| flits.cs.vu.nl.                   | 86400 | IN | MX    | 1 flits.cs.vu.nl.                          |
| flits.cs.vu.nl.                   | 86400 | IN | MX    | 2 zephyr.cs.vu.nl.                         |
| flits.cs.vu.nl.                   | 86400 | IN | MX    | 3 top.cs.vu.nl.                            |
| www.cs.vu.nl.                     | 86400 | IN | CNAME | star.cs.vu.nl                              |
| ftp.cs.vu.nl.                     | 86400 | IN | CNAME | zephyr.cs.vu.nl                            |
|                                   |       |    |       |                                            |
| rowboat                           |       | IN | A     | 130.37.56.201                              |
|                                   |       | IN | MX    | 1 rowboat                                  |
|                                   |       | IN | MX    | 2 zephyr                                   |
|                                   |       | IN | HINFO | Sun Unix                                   |
|                                   |       |    |       |                                            |
| little-sister                     |       | IN | A     | 130.37.62.23                               |
|                                   |       | IN | HINFO | Mac MacOS                                  |
|                                   |       |    |       |                                            |
| laserjet                          |       | IN | A     | 192.31.231.216                             |
|                                   |       | IN | HINFO | "HP Laserjet IIISi" Proprietary            |

A portion of a possible DNS database for *cs.vu.nl.* 7



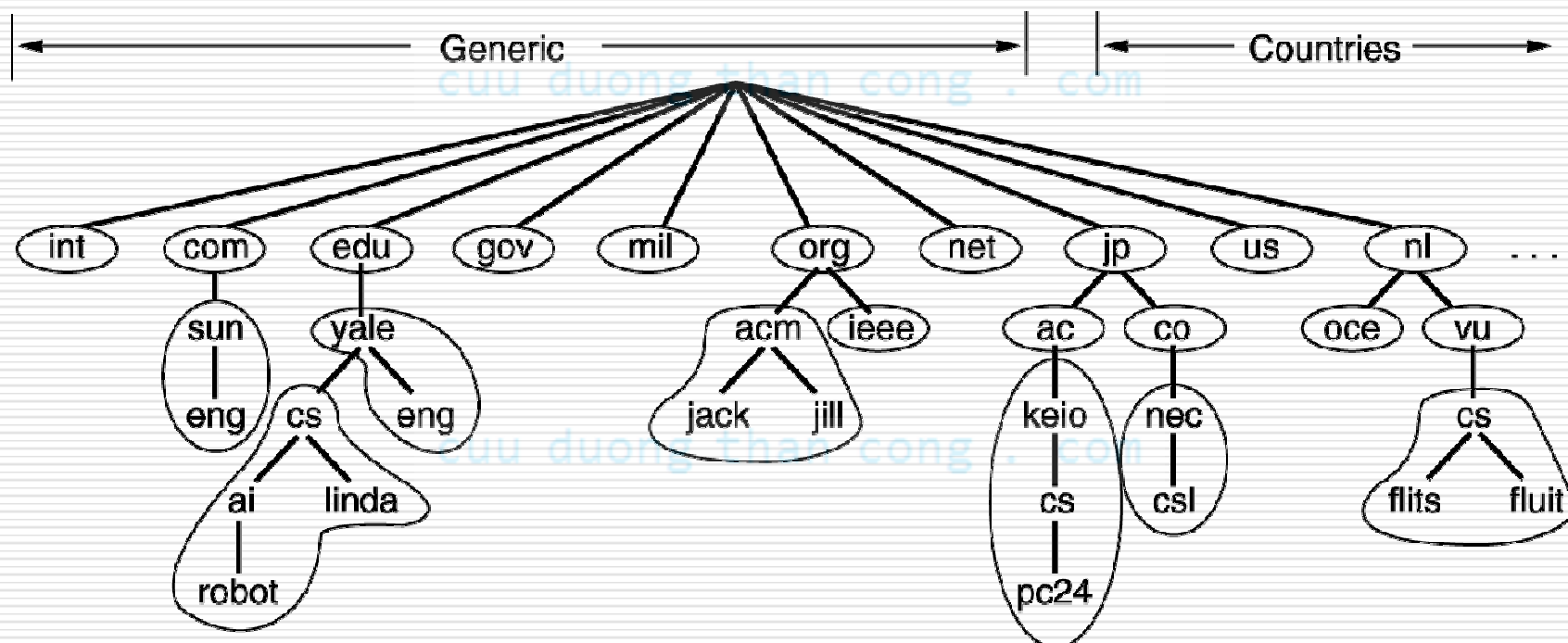
## Resource Records (3)

```
hcmut.edu.vn. IN SOA hcmut-server.hcmut.edu.vn. webmaster.hcmut.edu.vn. (  
    2004110800; serial  
    7200; refresh  
    3600; retry  
    604800; expire  
    86400 ); minimum  
hcmut.edu.vn. 86400 IN NS vnuserv.vnuhcm.edu.vn.  
hcmut.edu.vn. 86400 IN NS server.vnuhcm.edu.vn.  
hcmut.edu.vn. 86400 IN MX 0 webmailserv.hcmut.edu.vn.  
hcmut.edu.vn. 86400 IN MX 5 vnuserv.vnuhcm.edu.vn.  
hcmut-server.hcmut.edu.vn. 86400 IN A 172.28.2.2  
stu-mailserv.hcmut.edu.vn. 86400 IN A 172.28.2.3  
webmailserv.hcmut.edu.vn. 86400 IN A 172.28.2.4  
pop3.student.hcmut.edu.vn. 86400 IN CNAME stu-mailserv.hcmut.edu.vn.  
www.student.hcmut.edu.vn 86400 IN CNAME stu-mailserv.hcmut.edu.vn.
```



# Name Servers

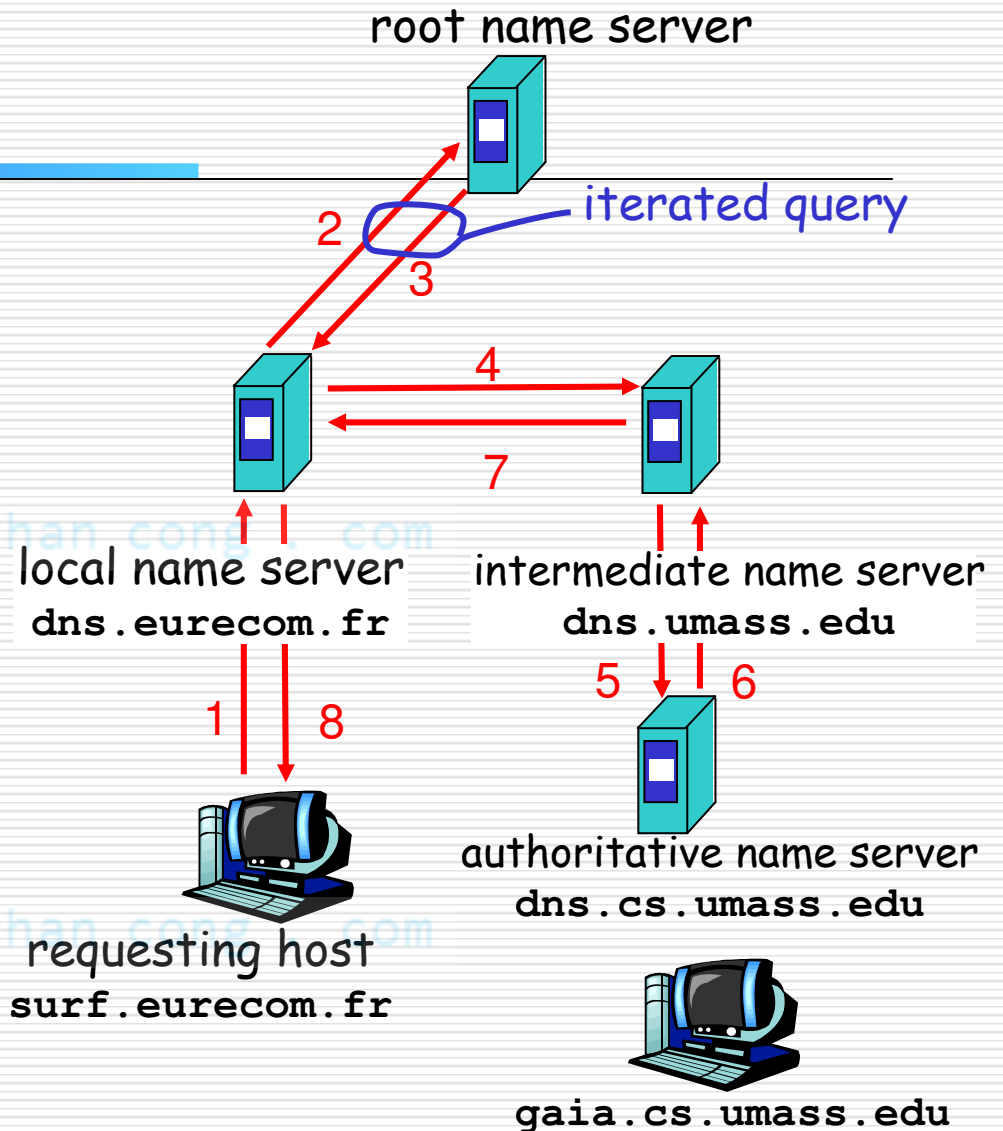
- DNS Name Space is organised into non-overlapping zones
- Each zone has Name Servers holding information about it





# DNS - Query

- **recursive query**
  - puts burden of name resolution on contacted name server.
  - heavy load ?
- **iterated query**
  - contacted server replies with name of server to contact.
  - “I don’t know this name, surf.eurecom.fr but ask this server”





# Electronic Mail – Email (or E-mail)

---

- ❑ Has been around since the early days of Internet
- ❑ Is widely used today
- ❑ Informal form of communication
- ❑ Simple and easy to use



## Electronic Mail (2)

Some smileys :-).

| Smiley | Meaning       | Smiley | Meaning         | Smiley | Meaning       |
|--------|---------------|--------|-----------------|--------|---------------|
| : -)   | I'm happy     | = :-)  | Abe Lincoln     | :+)    | Big nose      |
| :-(    | I'm sad/angry | =):-)  | Uncle Sam       | : -))  | Double chin   |
| : -    | I'm apathetic | *<:-)  | Santa Claus     | : -{)  | Mustache      |
| ;-)    | I'm winking   | <:- (  | Dunce           | #:-)   | Matted hair   |
| :-(O)  | I'm yelling   | (-:    | Australian      | 8-)    | Wears glasses |
| :-(*)  | I'm vomiting  | : -)X  | Man with bowtie | C:-)   | Large brain   |



# Architecture and Services

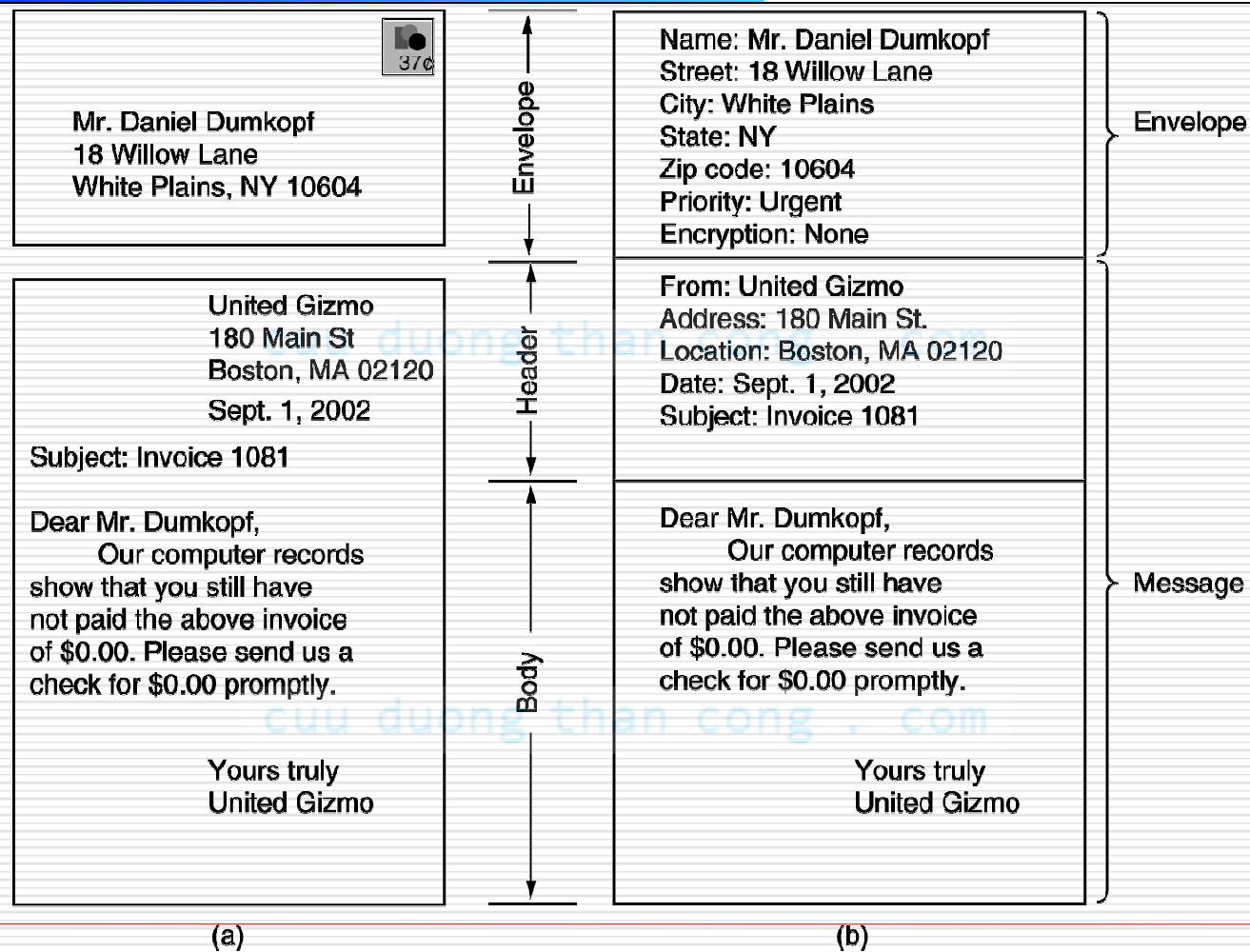
---

## Basic email functions

- Composition
- Transfer [cuu duong than cong . com](http://cuuduongthancong.com)
- Reporting
- Displaying
- Disposition [cuu duong than cong . com](http://cuuduongthancong.com)



# Email Message Structure



Envelopes and messages. (a) Paper mail. (b) Electronic mail.



# Email Systems

---

- Has two basic parts:
  - User agent: a program that accepts a variety of commands for composing, receiving, and replying to messages, as well as for manipulating mailboxes
  - Message transfer agents: relaying messages from the originator to the recipient



# Reading E-mail

---

An example display of the contents of a mailbox.

| # | Flags | Bytes  | Sender      | Subject                              |
|---|-------|--------|-------------|--------------------------------------|
| 1 | K     | 1030   | asw         | Changes to MINIX                     |
| 2 | KA    | 6348   | trudy       | Not all Trudys are nasty             |
| 3 | K F   | 4519   | Amy N. Wong | Request for information              |
| 4 |       | 1236   | bal         | Bioinformatics                       |
| 5 |       | 104110 | kaashoek    | Material on peer-to-peer             |
| 6 |       | 1223   | Frank       | Re: Will you review a grant proposal |
| 7 |       | 3110   | guido       | Our paper has been accepted          |
| 8 |       | 1204   | dmr         | Re: My student's visit               |



# Message Formats

---

RFC 822 header fields related to message transport.

| Header       | Meaning                                           |
|--------------|---------------------------------------------------|
| To:          | E-mail address(es) of primary recipient(s)        |
| Cc:          | E-mail address(es) of secondary recipient(s)      |
| Bcc:         | E-mail address(es) for blind carbon copies        |
| From:        | Person or people who created the message          |
| Sender:      | E-mail address of the actual sender               |
| Received:    | Line added by each transfer agent along the route |
| Return-Path: | Can be used to identify a path back to the sender |



# Message Formats (2)

---

Some fields used in the RFC 822 message header.

| Header       | Meaning                                               |
|--------------|-------------------------------------------------------|
| Date:        | The date and time the message was sent                |
| Reply-To:    | E-mail address to which replies should be sent        |
| Message-Id:  | Unique number for referencing this message later      |
| In-Reply-To: | Message-Id of the message to which this is a reply    |
| References:  | Other relevant Message-Ids                            |
| Keywords:    | User-chosen keywords                                  |
| Subject:     | Short summary of the message for the one-line display |



# MIME – Multipurpose Internet Mail Extensions

---

- Some problems when using ASCII formatted messages:
  - Languages with accents (French, German).
  - Languages in non-Latin alphabets (Hebrew, Russian).
  - Languages without alphabets (Chinese, Japanese).
  - Messages not containing text at all (audio or images).
- MIME adds structure to the message body and defines encoding rules for non-ASCII messages



# MIME (2)

---

## RFC 822 headers added by MIME.

| Header                     | Meaning                                              |
|----------------------------|------------------------------------------------------|
| MIME-Version:              | Identifies the MIME version                          |
| Content-Description:       | Human-readable string telling what is in the message |
| Content-Id:                | Unique identifier                                    |
| Content-Transfer-Encoding: | How the body is wrapped for transmission             |
| Content-Type:              | Type and format of the content                       |



# MIME (3)

The MIME types and subtypes defined in RFC 2045.

| Type        | Subtype       | Description                                 |
|-------------|---------------|---------------------------------------------|
| Text        | Plain         | Unformatted text                            |
|             | Enriched      | Text including simple formatting commands   |
| Image       | Gif           | Still picture in GIF format                 |
|             | Jpeg          | Still picture in JPEG format                |
| Audio       | Basic         | Audible sound                               |
| Video       | Mpeg          | Movie in MPEG format                        |
| Application | Octet-stream  | An uninterpreted byte sequence              |
|             | Postscript    | A printable document in PostScript          |
| Message     | Rfc822        | A MIME RFC 822 message                      |
|             | Partial       | Message has been split for transmission     |
|             | External-body | Message itself must be fetched over the net |
| Multipart   | Mixed         | Independent parts in the specified order    |
|             | Alternative   | Same message in different formats           |
|             | Parallel      | Parts must be viewed simultaneously         |
|             | Digest        | Each part is a complete RFC 822 message     |



# Message Transfer

---

- ❑ Message transfer agents are daemons running on mail servers
- ❑ Use Simple Mail Transfer Protocol
- ❑ Use TCP on port 25

cuu duong than cong . com



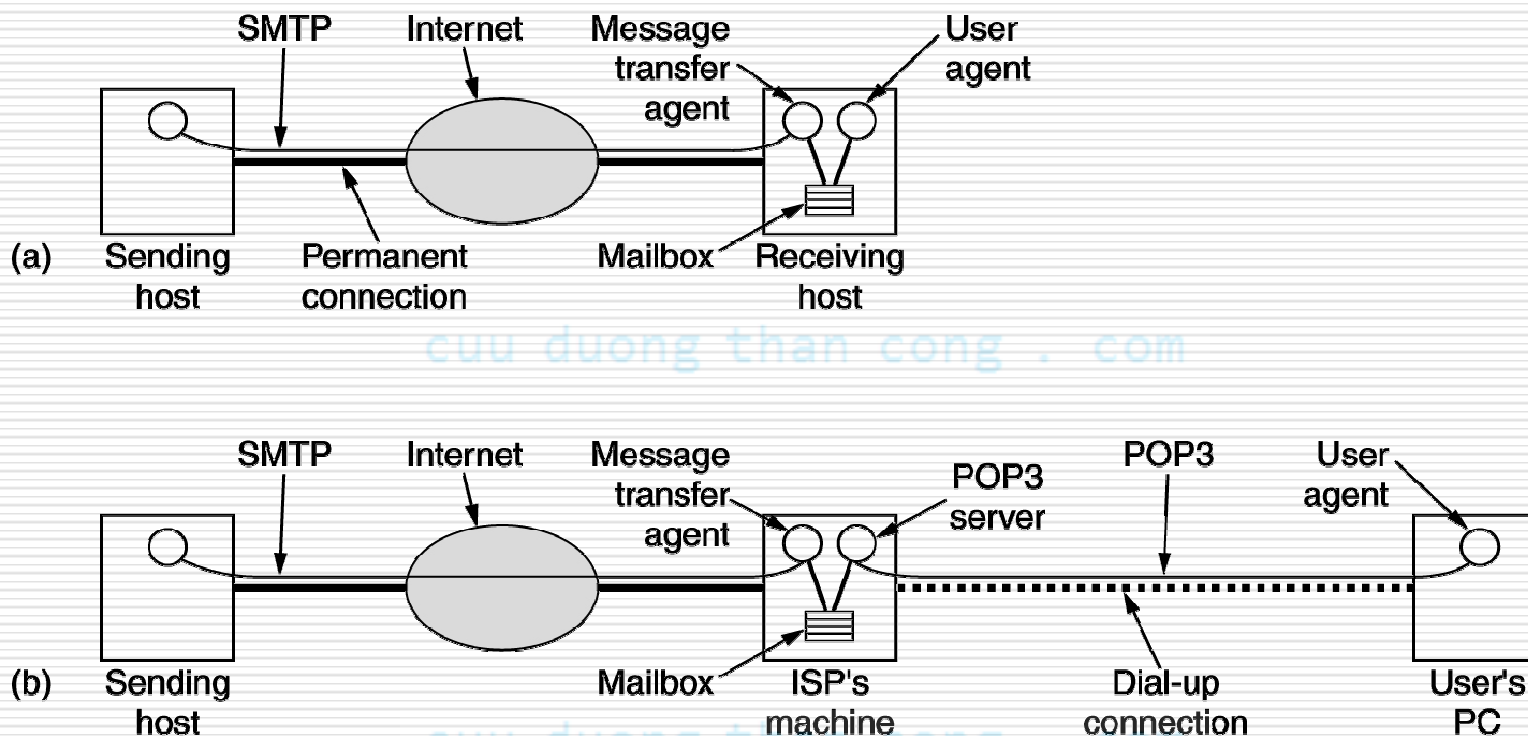
# Message Transfer (2)

Transferring a message  
from *elinore@abc.com*  
to *carolyn@xyz.com*  
Using SMTP.

```
S: 220 xyz.com SMTP service ready
C: HELO abcd.com
S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
S: 250 recipient ok
C: DATA
S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C:   access-type="anon-ftp";
C:   site="bicycle.abcd.com";
C:   directory="pub";
C:   name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
S: 250 message accepted
C: QUIT
S: 221 xyz.com closing connection
```



# Final Delivery



(a) Sending and reading mail when the receiver has a permanent Internet connection and the user agent runs on the same machine as the message transfer agent. (b) Reading e-mail when the receiver has a dial-up connection to an ISP.



# POP3

- Post Office Protocol Version 3
- Use TCP on port 110
- Is used to download messages from a mail server to client computers
- Example: Using POP3 to fetch three messages.

```
S: +OK POP3 server ready
C: USER carolyn
S: +OK
C: PASS vegetables
S: +OK login successful
C: LIST
S: 1 2505
S: 2 14302
S: 3 8122
S: .
C: RETR 1
S: (sends message 1)
C: DELE 1
C: RETR 2
S: (sends message 2)
C: DELE 2
C: RETR 3
S: (sends message 3)
C: DELE 3
C: QUIT
S: +OK POP3 server disconnecting
```



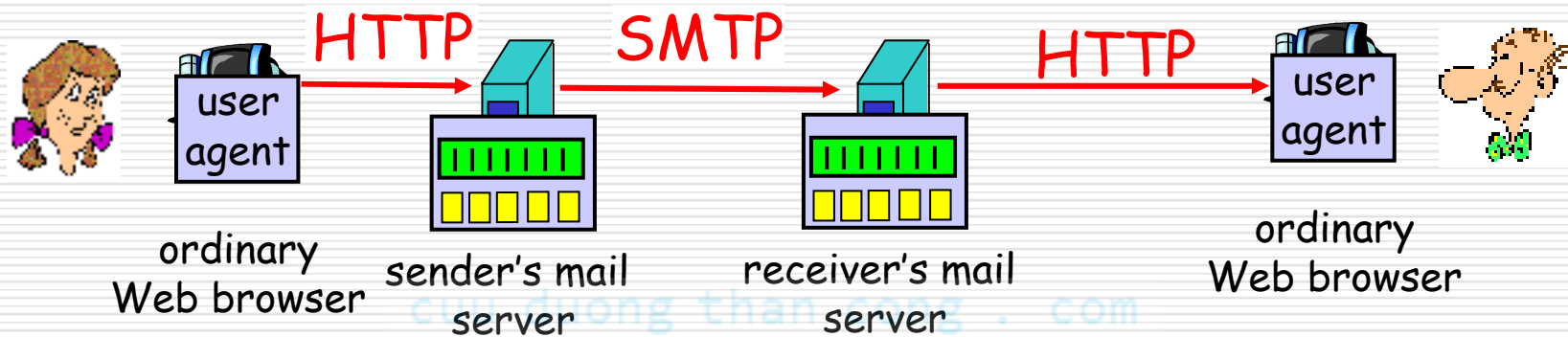
# IMAP (Internet Message Access Protocol)

- POP3 is not convenient when users frequently use different machines to read email from servers, as emails have to be downloaded to different computers more or less random
- IMAP can resolve this issues as emails will be always on the servers
- A comparison of POP3 and IMAP.

| Feature                        | POP3      | IMAP             |
|--------------------------------|-----------|------------------|
| Where is protocol defined?     | RFC 1939  | RFC 2060         |
| Which TCP port is used?        | 110       | 143              |
| Where is e-mail stored?        | User's PC | Server           |
| Where is e-mail read?          | Off-line  | On-line          |
| Connect time required?         | Little    | Much             |
| Use of server resources?       | Minimal   | Extensive        |
| Multiple mailboxes?            | No        | Yes              |
| Who backs up mailboxes?        | User      | ISP              |
| Good for mobile users?         | No        | Yes              |
| User control over downloading? | Little    | Great            |
| Partial message downloads?     | No        | Yes              |
| Are disk quotas a problem?     | No        | Could be in time |
| Simple to implement?           | Yes       | No               |
| Widespread support?            | Yes       | Growing          |



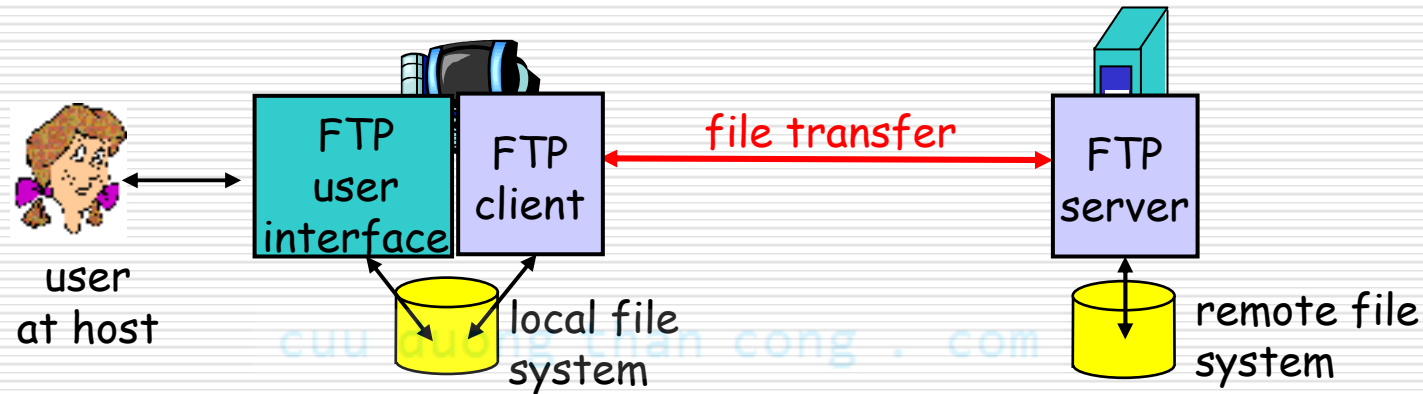
# Web Mail



- Convenient for the user on the go (Internet Café, WebTV, ...)
- User can organize their hierarchy of folders on servers
- May be slow:
  - server typically far from client
  - interaction with server through CGI scripts



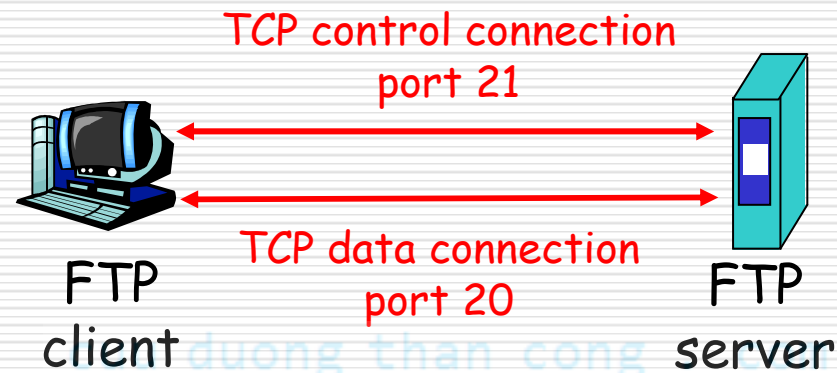
# FTP - File Transfer Protocol



- transfer file to/from remote host
- client/server model
  - *client*: side that initiates transfer (either to/from remote)
  - *server*: remote host
- ftp: RFC 959
- ftp server: port 21



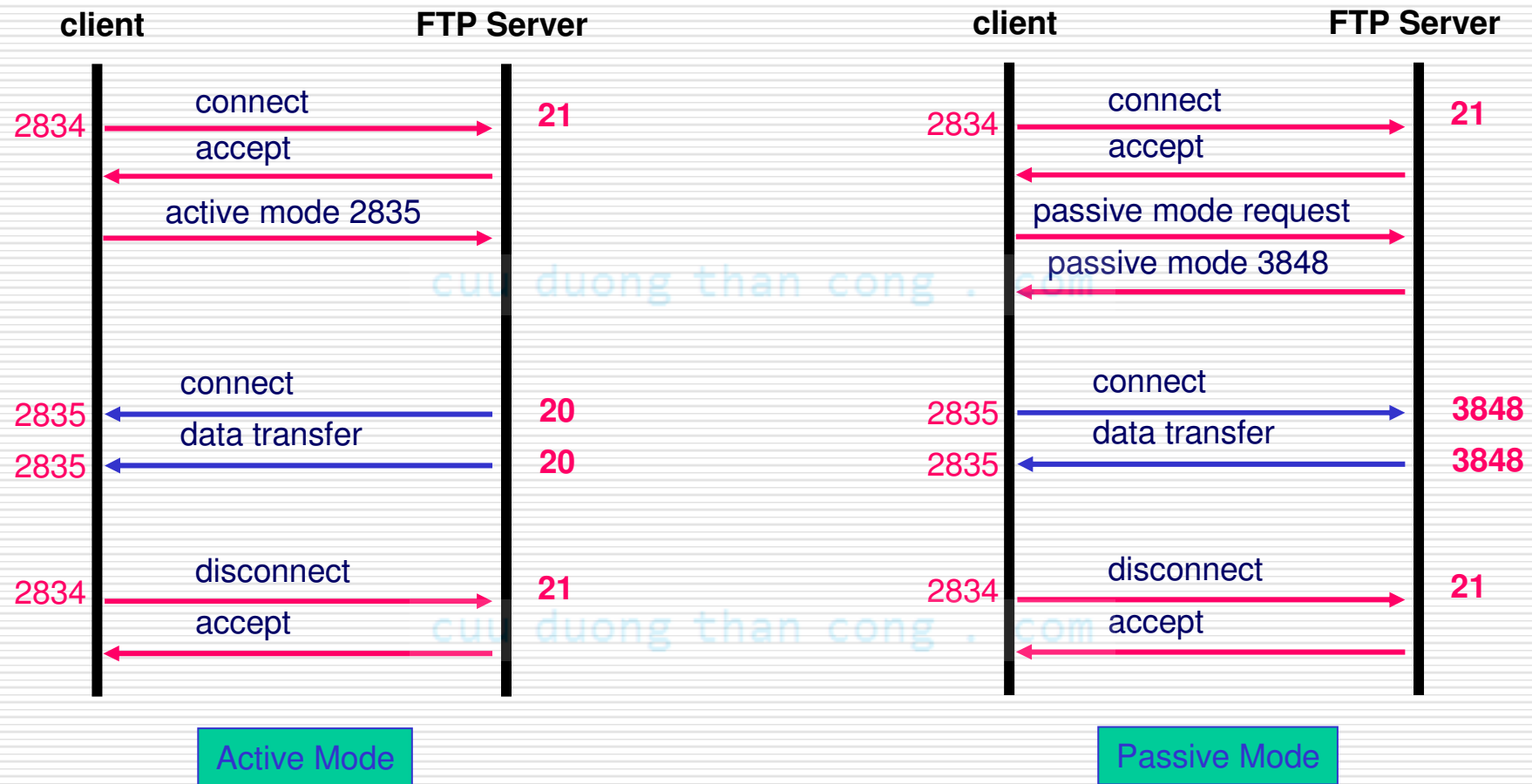
# FTP - Out of band control



- ❑ ftp client contacts ftp server at **port 21**, specifying TCP as transport protocol.
- ❑ two parallel TCP connections opened:
  - **control**: exchange commands, responses between client, server.
  - **data**: file data to/from server.
- ❑ ftp server maintains "state": current directory, earlier authentication.



# FTP - Transfer Mode





# The World Wide Web

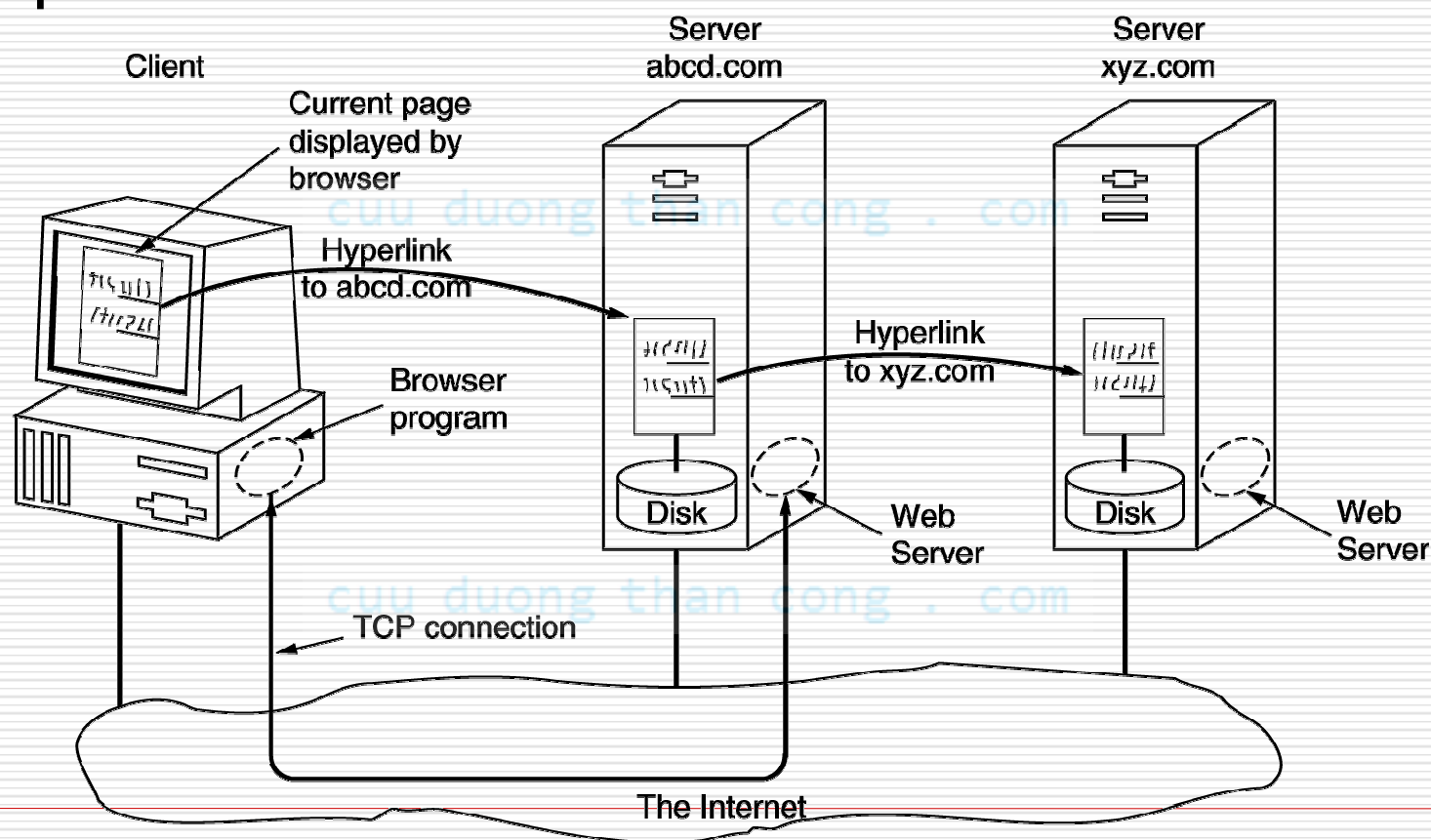
---

- Began in 1989 at CERN (Switzerland) by Tim Berners-Lee
- To be discussed
  - Architectural Overview
  - Static Web Documents
  - Dynamic Web Documents
  - HTTP – The HyperText Transfer Protocol



# Architectural Overview

## The parts of the Web model.





# The Client Side

---

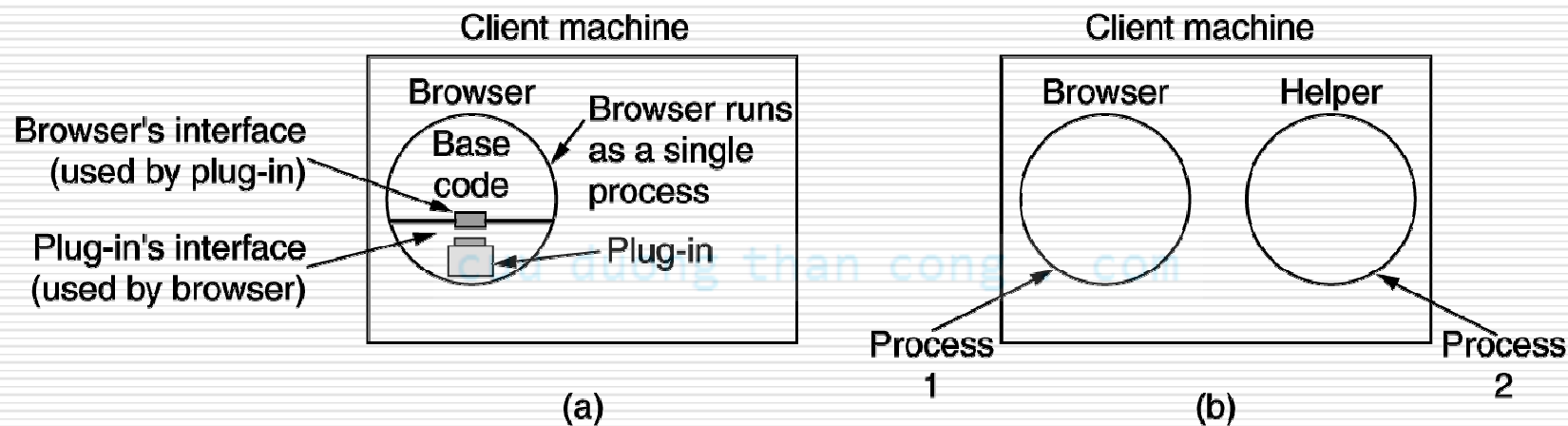
- Steps carried out by the browsers
  1. The browser determines the URL (by seeing what was selected).
  2. The browser asks DNS for the IP address of www.itu.org.
  3. DNS replies with 156.106.192.32.
  4. The browser makes a TCP connection to port 80 on 156.106.192.32.
  5. It then sends over a request asking for file /home/index.html.
  6. The www.itu.org server sends the file /home/index.html.
  7. The TCP connection is released.
  8. The browser displays all the text in /home/index.html.
  9. The browser fetches and displays all images in this file.



## The Client Side (2)

- To display contents other HTML, e.g. PDF file or a movie clip, browsers use plug-ins or helper applications

(a) A browser plug-in. (b) A helper application.





# The Client Side

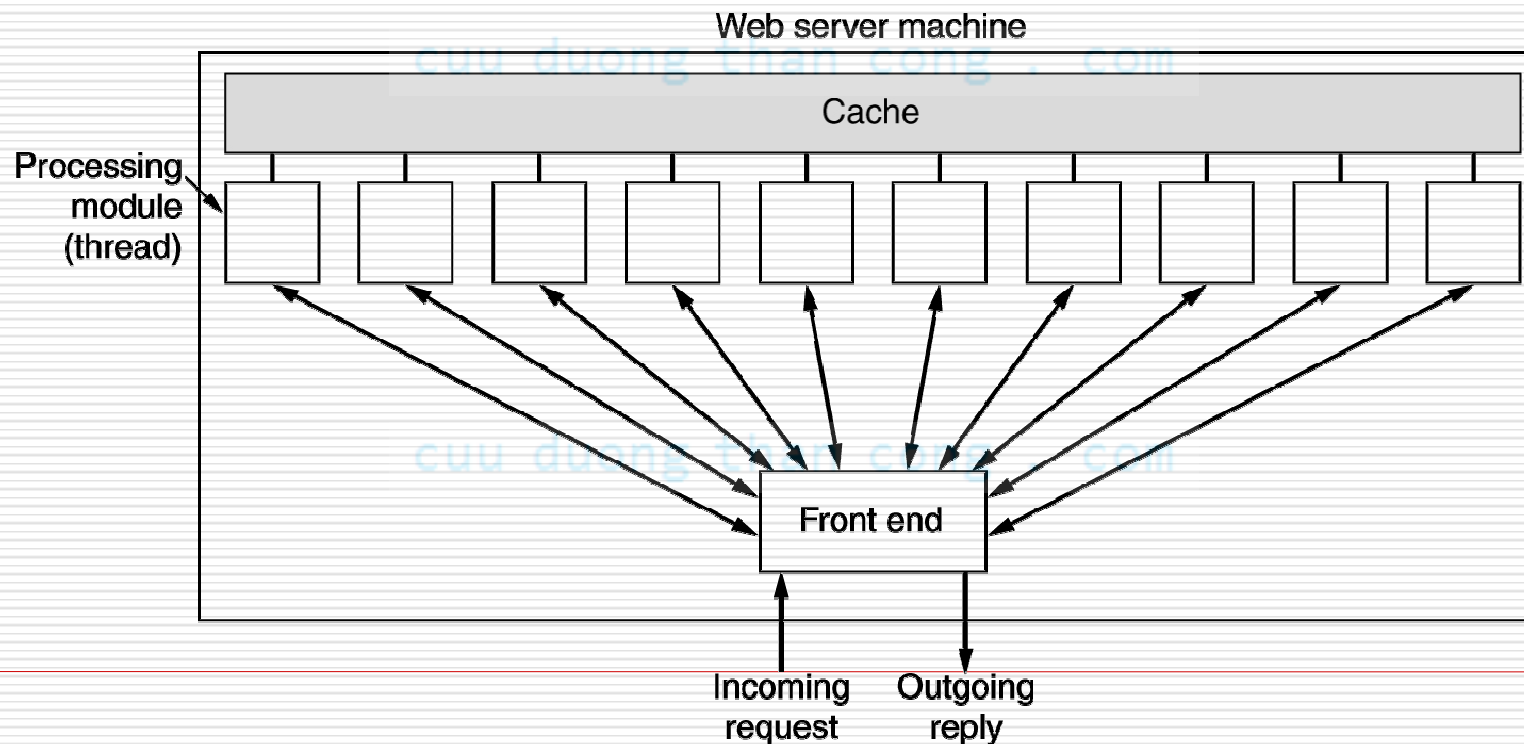
---

- Generally, steps that a server performs
  1. Accept a TCP connection from a client (a browser).
  2. Get the name of the file requested.
  3. Get the file (from disk).
  4. Return the file to the client.
  5. Release the TCP connection



## The Server Side (2)

A multithreaded Web server with a front end and processing modules.





# The Server Side (3)

---

- Steps performed by modern web servers
  1. Resolve the name of the Web page requested.
  2. Authenticate the client.
  3. Perform access control on the client.
  4. Perform access control on the Web page.
  5. Check the cache.
  6. Fetch the requested page from disk.
  7. Determine the MIME type to include in the response.
  8. Take care of miscellaneous odds and ends.
  9. Return the reply to the client.
  10. Make an entry in the server log.



# URLs – Uniform Resource Locators

- URLs are used to refer to resources in the Internet, such as a web page

Some common URLs.

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



# HTML – HyperText Markup Language

(a) The HTML for a sample Web page. (b) The formatted page.

```
<html>
<head><title> AMALGAMATED WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
 <br>
We are so happy that you have chosen to visit <b> Amalgamated Widget's </b>
home page. We hope <i> you </i> will find all the information you need here.
<p>Below we have links to information about our many fine products.
You can order electronically (by WWW), by telephone, or by fax. </p>
<hr>
<h2> Product information </h2>
<ul>
<li> <a href="http://widget.com/products/big"> Big widgets</a>
<li> <a href="http://widget.com/products/little"> Little widgets </a>
</ul>
<h2> Telephone numbers</h2>
<ul>
<li> By telephone: 1-800-WIDGETS
<li> By fax: 1-415-765-4321
</ul>
</body>
</html>
```

(a)

## Welcome to AWI's Home Page



We are so happy that you have chosen to visit **Amalgamated Widget's** home page. We hope you will find all the information you need here.

Below we have links to information about our many fine products. You can order electronically (by WWW), by telephone, or by FAX.

---

### Product Information

- [Big widgets](#)
- [Little widgets](#)

### Telephone numbers

- 1-800-WIDGETS
- 1-415-765-4321

(b)



# HTML Tags

| 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 <i>n</i> > ... </h <i>n</i> > | Delimits a level <i>n</i> heading             |
| <b> ... </b>                     | Set ... in boldface                           |
| <i> ... </i>                     | Set ... in italics                            |
| <center> ... </center>           | Center ... on the page horizontally           |
| <ul> ... </ul>                   | Brackets an unordered (bulleted) list         |
| <ol> ... </ol>                   | Brackets a numbered list                      |
| <li>                             | Starts a list item (there is no </li>)        |
| <br>                             | Forces a line break here                      |
| <p>                              | Starts a paragraph                            |
| <hr>                             | Inserts a Horizontal rule                     |
|                   | Displays an image here                        |
| <a href="..."> ... </a>          | Defines a hyperlink                           |



# Tables

(a) An HTML table.

(b) A possible rendition of this table.

```
<html>
<head> <title> A sample page with a table </title> </head>
<body>
<table border=1 rules=all>
<caption> Some Differences between HTML Versions </caption>
<col align=left>
<col align=center>
<col align=center>
<col align=center>
<col align=center>
<tr> <th>Item <th>HTML 1.0 <th>HTML 2.0 <th>HTML 3.0 <th>HTML 4.0 </tr>
<tr> <th> Hyperlinks <td> x <td> x <td> x <td> x </tr>
<tr> <th> Images <td> x <td> x <td> x <td> x </tr>
<tr> <th> Lists <td> x <td> x <td> x <td> x </tr>
<tr> <th> Active Maps and Images <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Forms <td> &nbsp; <td> x <td> x <td> x </tr>
<tr> <th> Equations <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Toolbars <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Tables <td> &nbsp; <td> &nbsp; <td> x <td> x </tr>
<tr> <th> Accessibility features <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Object embedding <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
<tr> <th> Scripting <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x </tr>
</table>
</body>
</html>
```

(a)

Some Differences between HTML Versions

| Item                   | HTML 1.0 | HTML 2.0 | HTML 3.0 | HTML 4.0 |
|------------------------|----------|----------|----------|----------|
| Hyperlinks             | x        | x        | x        | x        |
| Images                 | x        | x        | x        | x        |
| Lists                  | x        | x        | x        | x        |
| Active Maps and Images |          | x        | x        | x        |
| Forms                  |          | x        | x        | x        |
| Equations              |          |          | x        | x        |
| Toolbars               |          |          | x        | x        |
| Tables                 |          |          | x        | x        |
| Accessibility features |          |          |          | x        |
| Object embedding       |          |          |          | x        |
| Scripting              |          |          |          | x        |

(b)



# Forms

(a) The HTML for an order form.

```
<html>
<head> <title> AWI CUSTOMER ORDERING FORM </title> </head>
<body>
<h1> Widget Order Form </h1>
<form ACTION="http://widget.com/cgi-bin/widgetorder" method=POST>
<p> Name <input name="customer" size=46> </p>
<p> Street Address <input name="address" size=40> </p>
<p> City <input name="city" size=20> State <input name="state" size =4>
Country <input name="country" size=10> </p>
<p> Credit card # <input name="cardno" size=10>
Expires <input name="expires" size=4>
M/C <input name="cc" type=radio value="mastercard">
VISA <input name="cc" type=radio value="visacard"> </p>
<p> Widget size Big <input name="product" type=radio value="expensive">
Little <input name="product" type=radio value="cheap">
Ship by express courier <input name="express" type=checkbox> </p>
<p><input type=submit value="submit order"> </p>
Thank you for ordering an AWI widget, the best widget money can buy!
</form>
</body>
</html>
```

(a)

(b) The formatted page.

### Widget Order Form

Name

Street address

City  State  Country

Credit card #  Expires  M/C ☐ Visa ☐

Widget size Big ☐ Little ☐ Ship by express courier ☐

Thank you for ordering an AWI widget, the best widget money can buy!

(b)



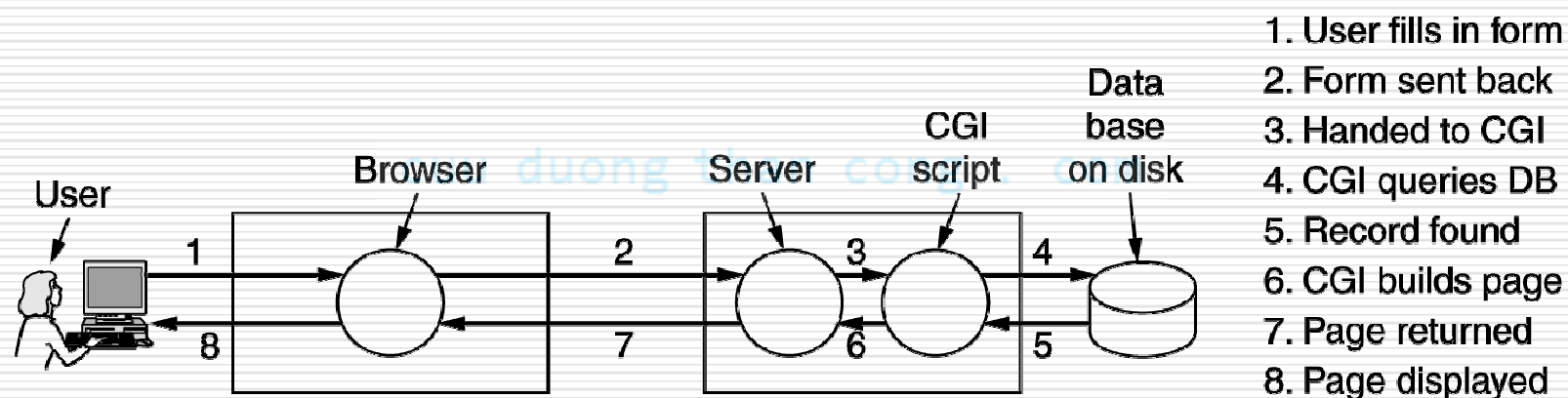
# Dynamic Web Documents

---

- ❑ Web contents are generated dynamically on demand
- ❑ Dynamic Web documents are now popular in the Internet
- ❑ Dynamic contents can be generated on client side or/and server side



# Server Side Dynamic Web Documents



Steps in processing the information from an HTML form.



## Server Side Dynamic Web Documents (2)

---

A sample HTML page with embedded PHP.

```
<html>
<body>cuu duong than cong . com
<h2> This is what I know about you </h2>
<?php echo $HTTP_USER_AGENT ?>

</body>
</html>cuu duong than cong . com
```



# Server Side Dynamic Web Documents (3)

(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.

```
<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>
```

(a)

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

(b)

```
<html>
<body>
<h1> Reply: </h1>
Hello Barbara.
Prediction: next year you will be 25
</body>
</html>
```

(c)



# Client-Side Dynamic Web Page Generation

Use of JavaScript  
for processing  
a form.

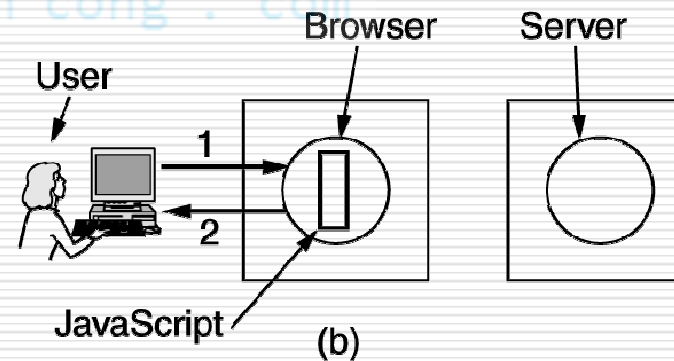
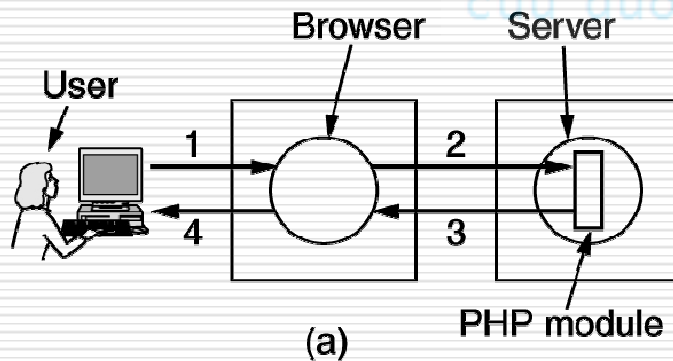
```
<head>
<script language="javascript" type="text/javascript">
function response(test form) {
    var person = test form.name.value;
    var years = eval(test form.age.value) + 1;
    document.open();
    document.writeln("<html> <body>");
    document.writeln("Hello " + person + ".<br>");
    document.writeln("Prediction: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

<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>
```



# Client-Side Dynamic Web Page Generation (2)

- (a) Server-side scripting with PHP.
- (b) Client-side scripting with JavaScript.





# Client-Side Dynamic Web Page Generation

## (3)

---

A JavaScript program for computing and printing factorials.

```
<html>
<head>
<script language="javascript" type="text/javascript">
function response(test_form) {
    function factorial(n) {if (n == 0) return 1; else return n * factorial(n - 1);}
    var r = eval(test_form.number.value); // r = typed in argument
    document.myform.mytext.value = "Here are the results.\n";
    for (var i = 1; i <= r; i++) // print one line from 1 to r
        document.myform.mytext.value += (i + "! = " + factorial(i) + "\n");
}
</script>
</head>
<body>
<form name="myform">
Please enter a number: <input type="text" name="number">
<input type="button" value="compute table of factorials" onclick="response(this.form)">
<p>
<textarea name="mytext" rows=25 cols=50> </textarea>
</form>
</body>
</html>
```



# Client-Side Dynamic Web Page Generation (4)

An interactive Web page that responds to mouse movement.

```
<html>
<head>
<script language="javascript" type="text/javascript">
if (!document.myurl) document.myurl = new Array();
document.myurl[0] = "http://www.cs.vu.nl/~ast/im/kitten.jpg";
document.myurl[1] = "http://www.cs.vu.nl/~ast/im/puppy.jpg";
document.myurl[2] = "http://www.cs.vu.nl/~ast/im/bunny.jpg";
function pop(m) {
    var urx = "http://www.cs.vu.nl/~ast/im/cat.jpg";
    popupwin = window.open(document.myurl[m], "mywind", "width=250,height=250");
}
</script>
</head>

<body>
<p> <a href="#" onmouseover="pop(0); return false;" > Kitten </a> </p>
<p> <a href="#" onmouseover="pop(1); return false;" > Puppy </a> </p>
<p> <a href="#" onmouseover="pop(2); return false;" > Bunny </a> </p>
</body>
</html>
```



# HTTP Methods

---

The built-in HTTP 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                         |



# HTTP Methods (2)

---

The status code response groups.

| 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 Message Headers

| Header           | Type     | Contents                                              |
|------------------|----------|-------------------------------------------------------|
| User-Agent       | Request  | Information about the browser and its platform        |
| Accept           | Request  | The type of pages the client can handle               |
| Accept-Charset   | Request  | The character sets that are acceptable to the client  |
| Accept-Encoding  | Request  | The page encodings the client can handle              |
| Accept-Language  | Request  | The natural languages the client can handle           |
| Host             | Request  | The server's DNS name                                 |
| Authorization    | Request  | A list of the client's credentials                    |
| Cookie           | Request  | Sends a previously set cookie back to the server      |
| Date             | Both     | Date and time the message was sent                    |
| Upgrade          | Both     | The protocol the sender wants to switch to            |
| Server           | Response | Information about the server                          |
| Content-Encoding | Response | How the content is encoded (e.g., gzip)               |
| Content-Language | Response | The natural language used in the page                 |
| Content-Length   | Response | The page's length in bytes                            |
| Content-Type     | Response | The page's MIME type                                  |
| Last-Modified    | Response | Time and date the page was last changed               |
| Location         | Response | A command to the client to send its request elsewhere |
| Accept-Ranges    | Response | The server will accept byte range requests            |
| Set-Cookie       | Response | The server wants the client to save a cookie          |

Some HTTP message headers.



# Example HTTP Usage

The start of the output of  
*www.ietf.org/rfc.html*.

```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is '^'.
HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: "2a79d-c8b-39bce48d"
Accept-Ranges: bytes
Content-Length: 3211
Content-Type: text/html
X-Pad: avoid browser bug
```

```
<html>
<head>
<title>IETF RFC Page</title>

<script language="javascript">
function url() {
  var x = document.form1.number.value
  if (x.length == 1) {x = "000" + x }
  if (x.length == 2) {x = "00" + x }
  if (x.length == 3) {x = "0" + x }
  document.form1.action = "/rfc/rfc" + x + ".txt"
  document.form1.submit
}
</script>

</head>
```



# Caching

Hierarchical caching with three proxies.

