Chapter 7

(Week 14)

The Application Layer (CONTINUATION

ANDREW S. TANENBAUM
COMPUTER NETWORKS
FOURTH EDITION
PP. 611-720

The Application Layer's topics

7.1. DNS – The Domain Name System

7.2. Electronic Mail

7.3. The World Wide Web

7.4. Multimedia

7.5. Summary

7.3. The World Wide Web

- Architectural Overview
- Static Web Documents
- Dynamic Web Documents
- HTTP – The HyperText Transfer Protocol
- Performance Ehnancements
- The Wireless Web

7.3. The World Wide Web

- The World Wide Web is an architectural framework for accessing linked documents spread out over millions of machines all over the Internet.

- Web or WWW is a system for linking hypertext documents

7.3. The World Wide Web

- Originally, each document was a page writtin in HTML with hyperlinks to other documents.

- Nowadays, XML is gradually starting to take over from HTML.

7.3. 1. Architectural Overview

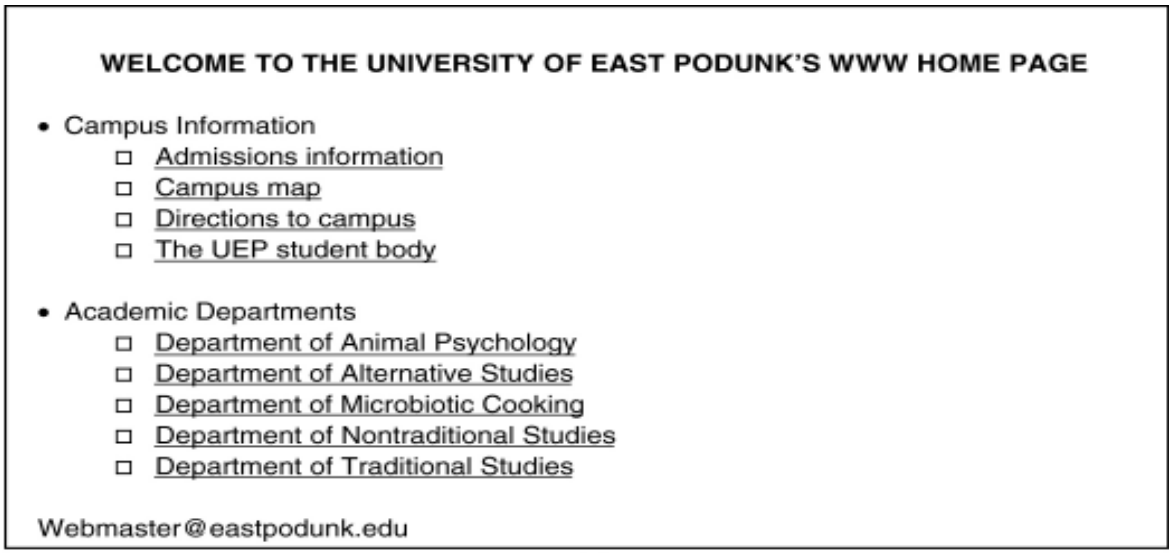
- From users' point of view, Web consists of a vast, worldwide collection of documents or Web pages, often just called pages for short.
- Each page may contain links to other pages anywhere in the world.

7.3. 1. Architectural Overview

- The idea of having one page point to another, now called hypertext.

- Pages are viewed with a program called a browser, of which Internet Explorer and Netscape Navigator are two popular ones.

Architectural Overview

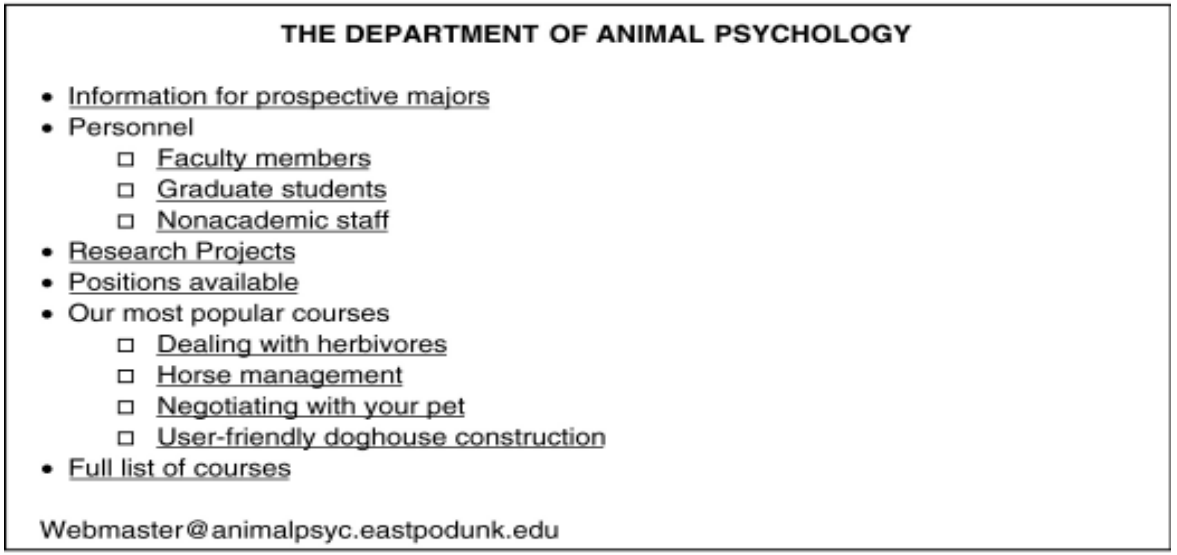


WELCOME TO THE UNIVERSITY OF EAST PODUNK'S WWW HOME PAGE

- Campus Information
  - Admissions information
  - Campus map
  - Directions to campus
  - The UEP student body
- Academic Departments
  - Department of Animal Psychology
  - Department of Alternative Studies
  - Department of Microbiotic Cooking
  - Department of Nontraditional Studies
  - Department of Traditional Studies

Webmaster@eastpodunk.edu

(a)



THE DEPARTMENT OF ANIMAL PSYCHOLOGY

- Information for prospective majors
- Personnel
  - Faculty members
  - Graduate students
  - Nonacademic staff
- Research Projects
- Positions available
- Our most popular courses
  - Dealing with herbivores
  - Horse management
  - Negotiating with your pet
  - User-friendly doghouse construction
- Full list of courses

Webmaster@animalpsyc.eastpodunk.edu

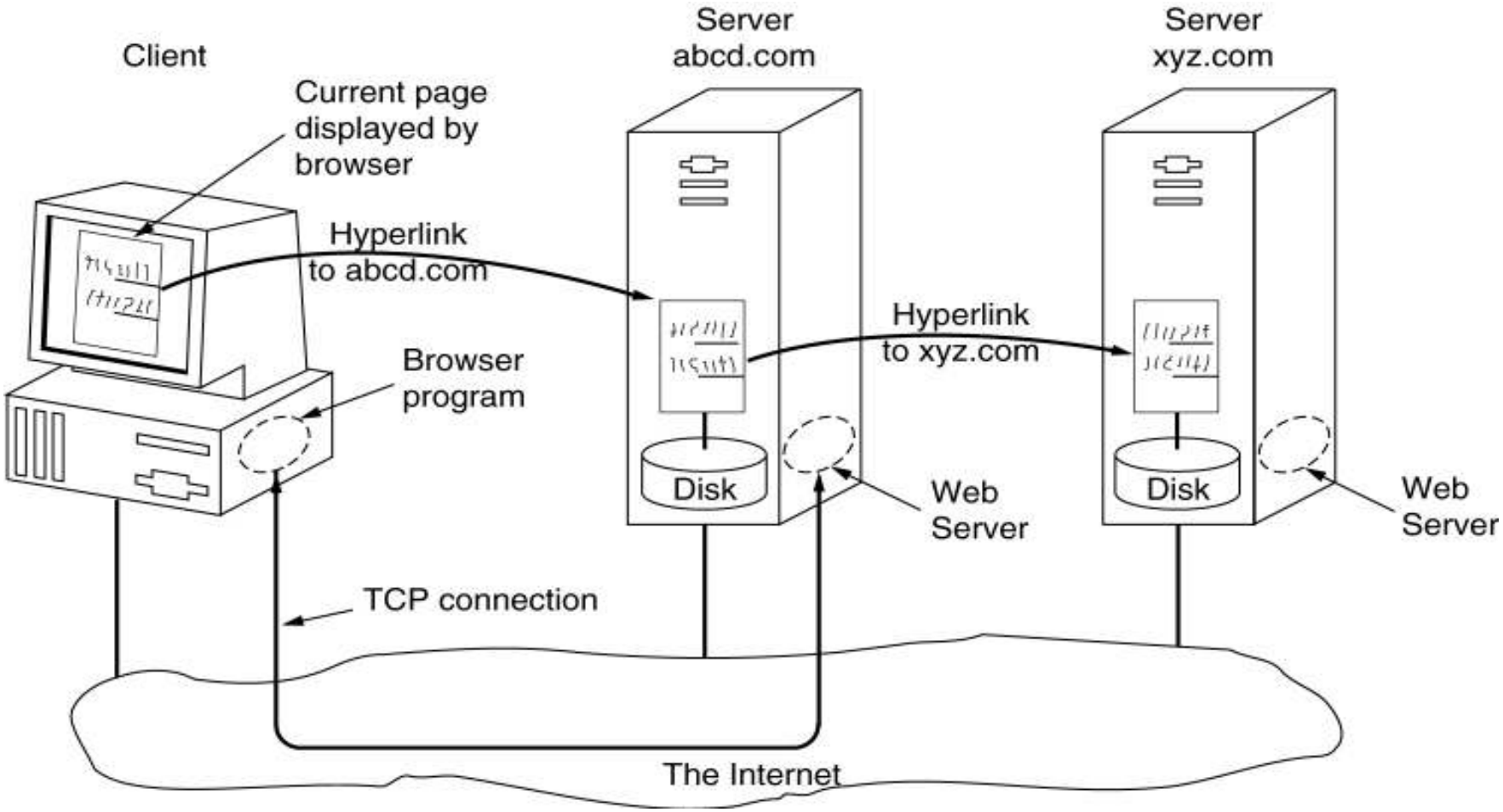
(b)

(a) A Web page (b) The page reached by clicking on Department of Animal Psychology.

7.3. 1. Architectural Overview

- Strings of text that are links to other pages, called hyperlinks, are often highlighted, by underlining, displaying them in a special color, or both.
- The basic model of how the Web works is shown in next slide.

Architectural Overview



The parts of the Web model.

7.3. 1. Architectural Overview
The Client Side

- When an item is selected, browser follows the hyperlink and fetches the page selected.
- Therefore, the embedded hyperlink needs a way to name any other page on the Web.
- Pages are named using URLs (Uniform Resource Locators)

7.3. 1. Architectural Overview
The Client Side

- A typical URL is
- http://www.abcd.com/products.html
- URL has three parts: the name of the protocol (http), the DNS name of the machine where the page is located (www.abcd.com), and (usually) the name of the file containing the page (products.html)

7.3. 1. Architectural Overview
The Client Side

- When a user clicks on a hyperlink, the browser carries out a series of steps in order to fetch the page pointed to.
- Suppose that a user is browsing the Web and finds a link on Internet telephony that points to ITU's home page, which is http://www.itu.org/home/index.html

7.3. 1. Architectural Overview
The Client Side

- Let us trace the steps that occur when this link is selected.

a) Browser determines the URL

b) Browser asks DNS for IP address of www.itu.org

c) DNS replies with 156.106.192.32

d) Browser makes a TCP connection to port 80 on 156.106.192.32

7.3. 1. Architectural Overview
The Client Side

e) It then sends over a request asking for file /home/index.html

f) www.itu.org server sends the file /home/index.html

h) TCP connection is released

g) Browser displays all the text in /home/index.html

i) Browser fetches and displays all images in this file

7.3. 1. Architectural Overview
The Client Side

- To allow all browsers to understand all Web pages, they are written in a standardized language called HTML.
- Not all pages contain HTML. A page may consist of a formatted document in PDF format, an icon in GIF format, a photograph in JPEG format, a song in MP3 format, a video in MPEG format, or any one of hundreds of other file types.

7.3. 1. Architectural Overview
The Client Side

- Since standard HTML pages may link to any of these, the browser has a problem when it encounters a page it cannot interpret.

- There are two possibilities : plug-ins and helper applications.

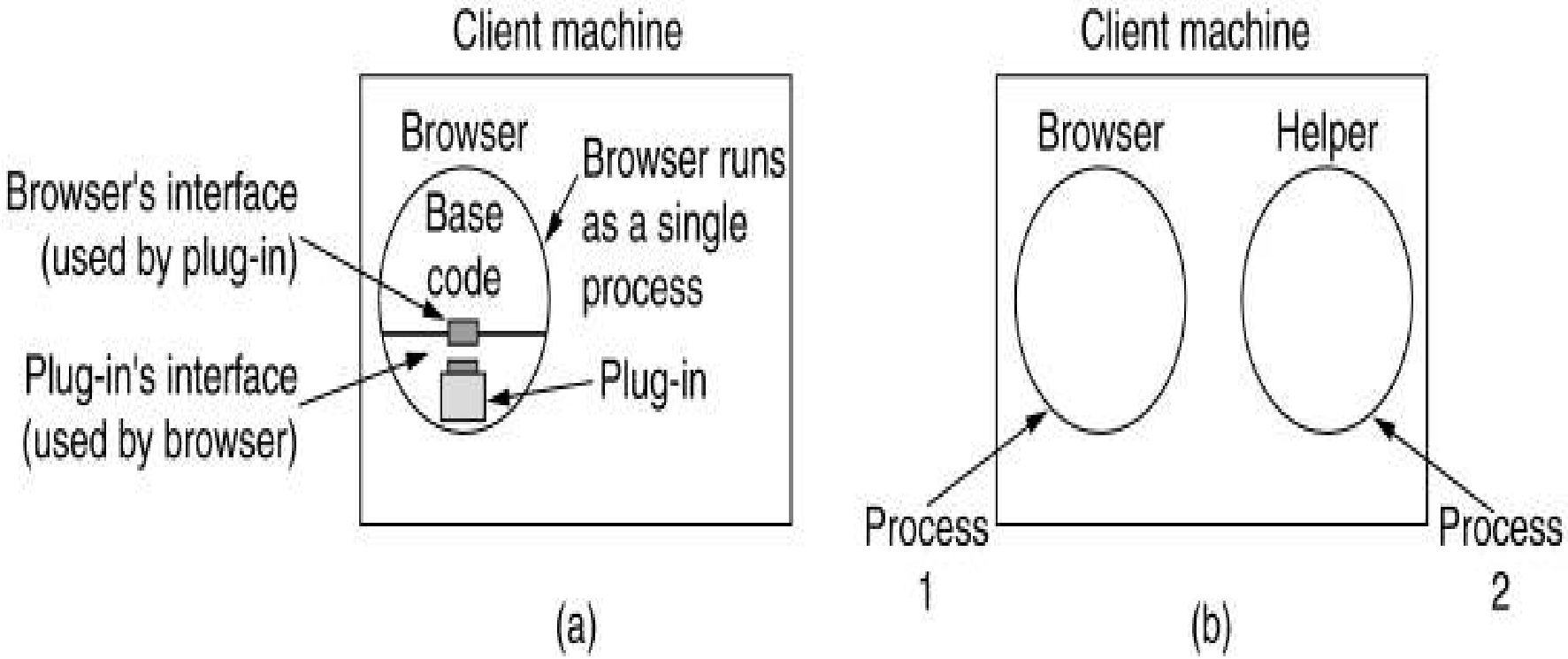
7.3. 1. Architectural Overview
The Client Side

- Plug-in is a code module that the browser fetches from a special directory on the disk and installs as an extension to itself.
- Because plug-ins run inside the browser, they have access to the current page and can modify its appearence.

7.3. 1. Architectural Overview
The Client Side

- After the plug-in has done its job (usually after the user has moved to a different Web page), the plug-in removed from the browser's memory.
- Before a plug-in can be used, it must be installed.

The Client Side



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

7.3. 1. Architectural Overview
The Client Side

- The other way to extend a browser is to use a helper application.
- This is a complete program, running as a separate process.
- Typically, helpers are large programs that exist independently of the browser, such as Adobe's Acrobat Reader for displaying PDF files or Microsoft Word.

7.3. 1. Architectural Overview
The Server Side

- The steps that the server performs are:

a) Accept a TCP connection from a client (a browser).

b) Get the name of the file requested.

c) Get the file (from disk).

d) Return the file to the client.

e) Release the TCP connection.

7.3. 1. Architectural Overview
The Server Side

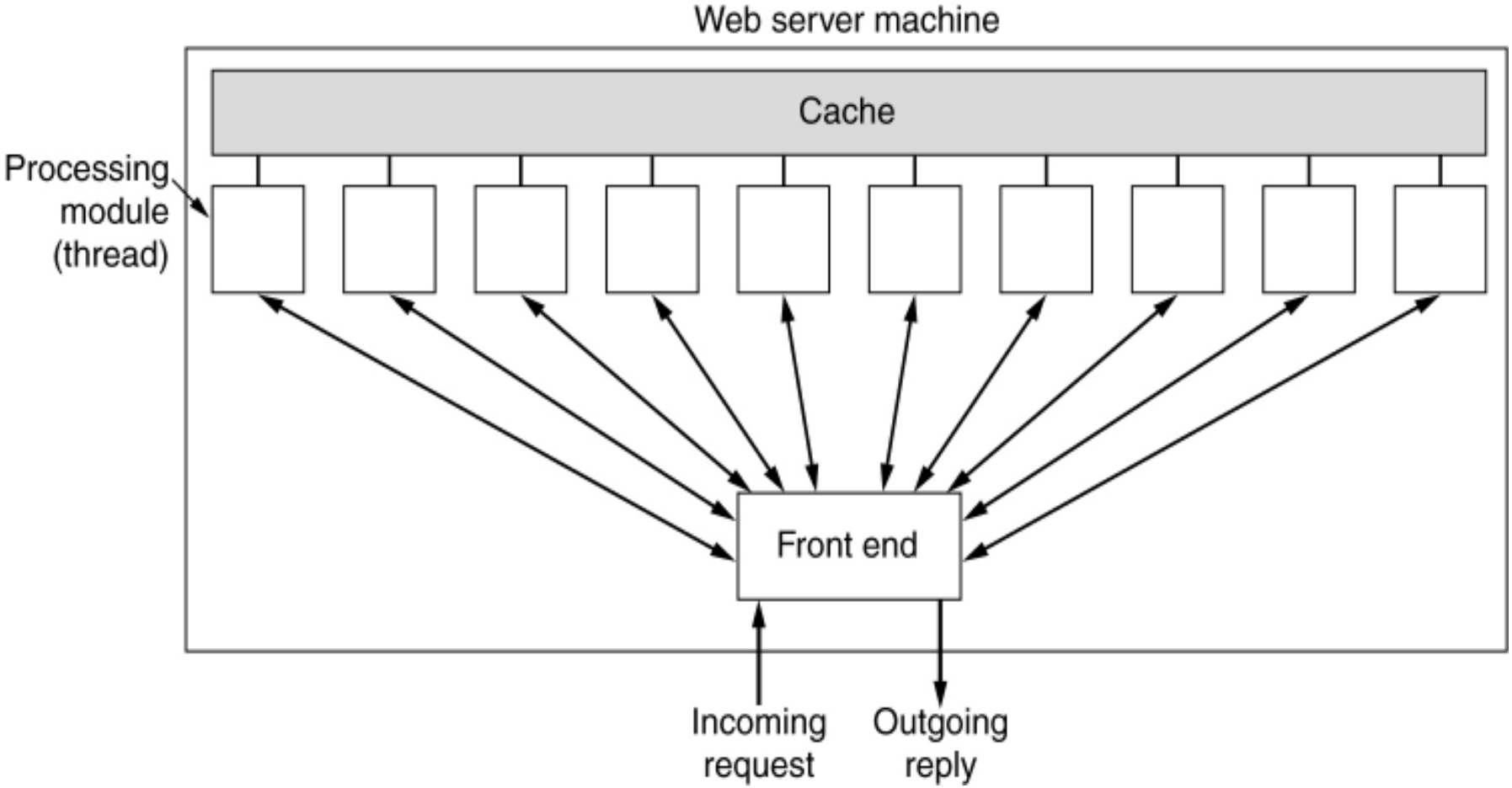
- A problem with this design is that every request requires making a disk access to get the file.

- The result is that the Web server cannot serve more requests per second than it can make disk accesses.

7.3. 1. Architectural Overview
The Server Side

- One obvious improvement (used by all Web servers) is to maintain a cache in memory of the ***n*** most recently used files.
- Other improvement for building a faster server is to make the server multithreaded.

The Server Side



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

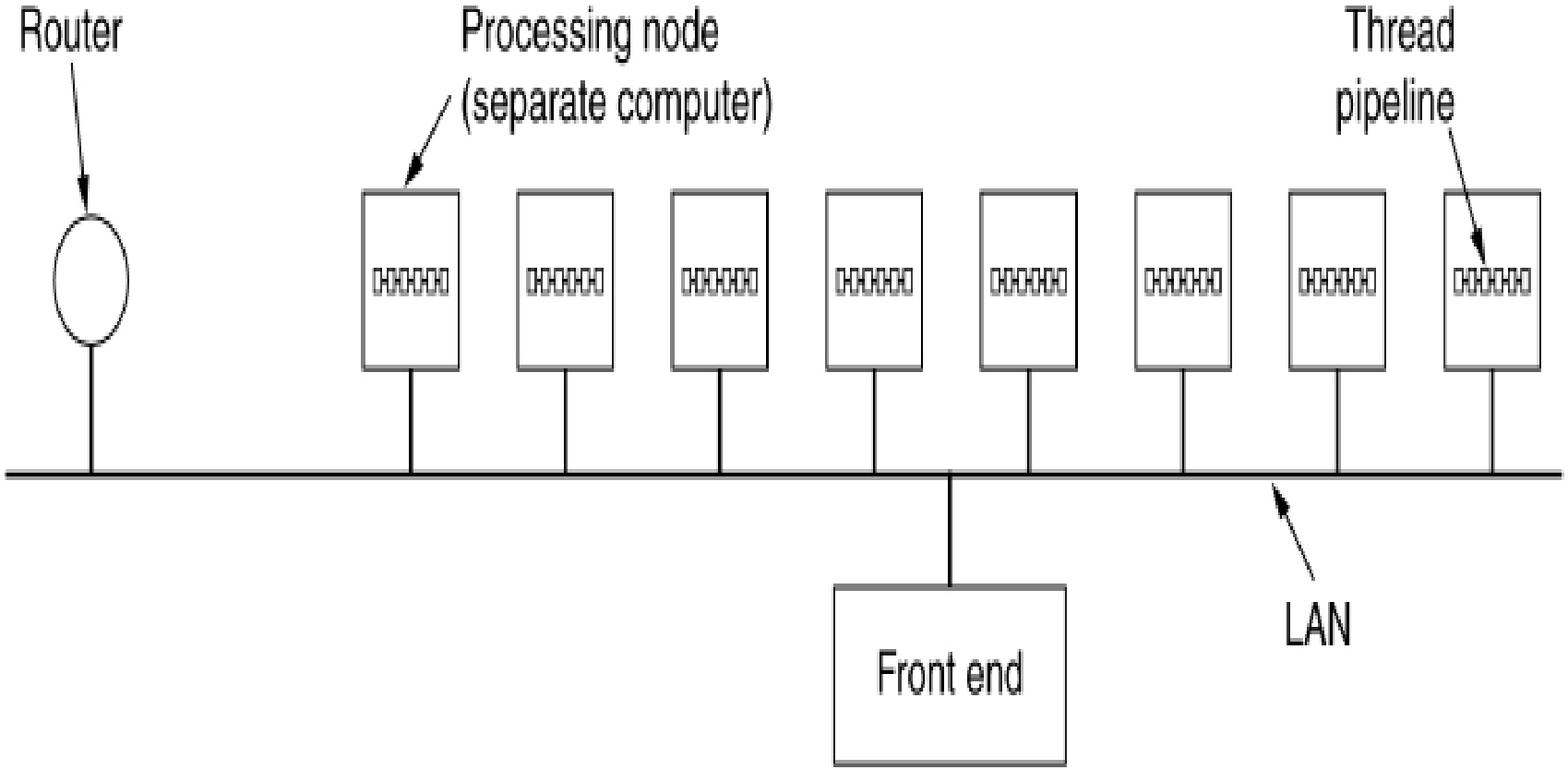
7.3. 1. Architectural Overview
The Server Side

- In one design, when a request comes in, the front end accept it and builds a short record describing it.
- It then hands the record to one of the processing moduls.
- In another design, the front end is eliminated and each processing module tries to acquire its own requests.

7.3. 1. Architectural Overview
The Server Side

- If too many requests come in each second, the CPU will not be able to handle the processing load, no matter how many disks are used in parallel.
- The solution is to add more nodes (computers), possibly with replicated disks to avoid having the disks become the next bottleneck.
- This leads to the server farm model.

The Server Side

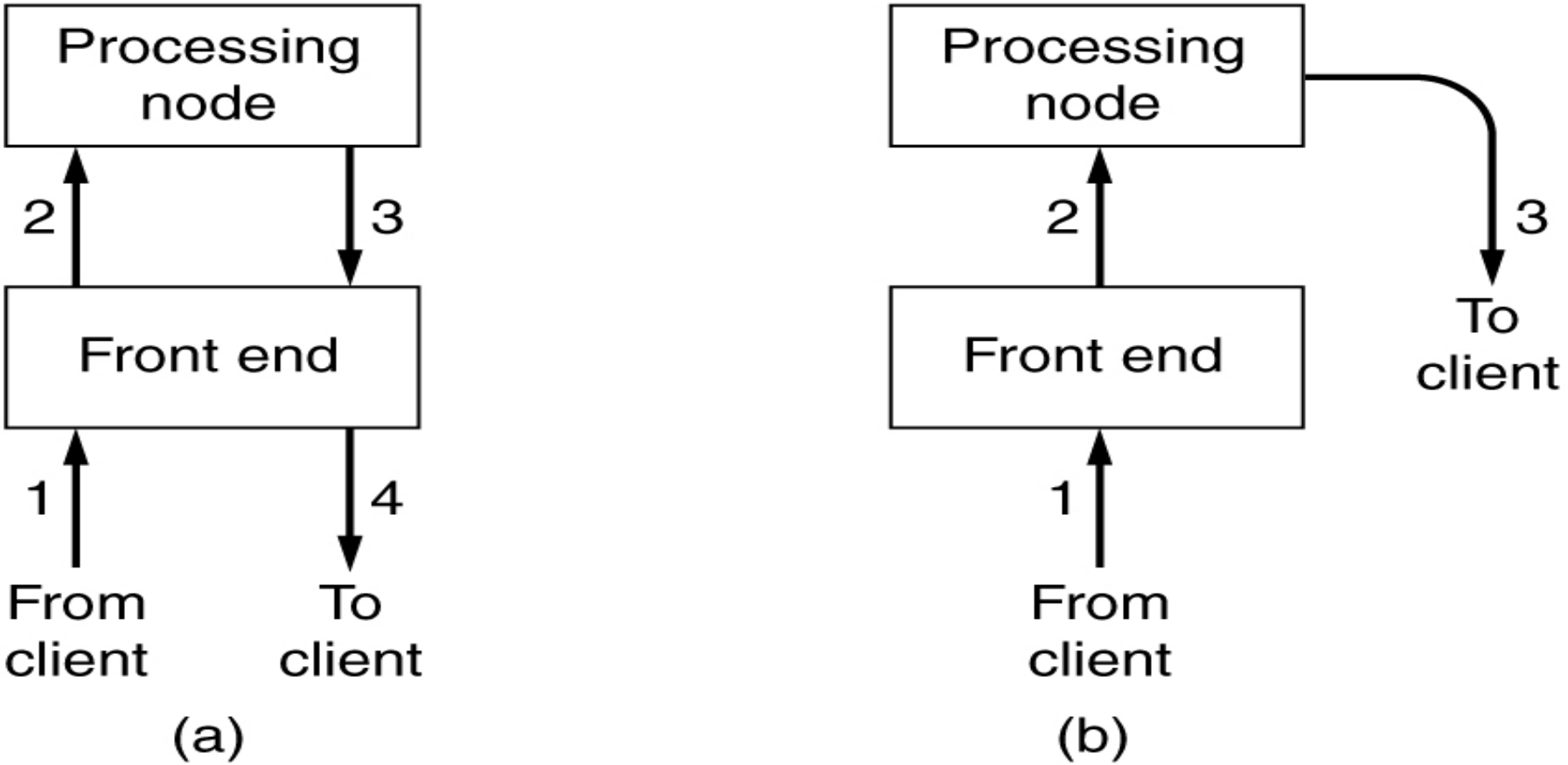


A server farm.

7.3. 1. Architectural Overview
The Server Side

- One problem with server farms is that there is no longer a shared cache because each processing node has its own memory – unless an expensive shared-memory multiprocessor is used.
- Another problem with server farms is that the client's TCP connection terminates at the front end, so the reply must go through the front end.

The Server Side



(a) Normal request-reply message sequence.

(b) Sequence when TCP handoff is used.

7.3. 1. Architectural Overview
URLs – Uniform Resource Locaters

- Web pages may contain pointers to other Web pages.
- How these pointers are implemented.
- When Web was first created, it was immediately apparent that having one page point to another Web page required mechanisms for naming and locating pages.

7.3. 1. Architectural Overview
URLs – Uniform Resource Locaters

- In particular, three questions had to be answered before a selected page could be displayed:

a) What is the page called?

b) Where is the page located?

c) How can the page be accessed?

7.3. 1. Architectural Overview
URLs – Uniform Resource Locaters

- Each page is assigned a URL (Uniform Resource Locator) that effectively serves as the page's worldwide name.
- URLs have three parts: the protocol (also known as the scheme), the DNS name of the machine on which the page is located, and a local name uniquely indicating the specific page (usually just a file name on the machine where it resides).

URLs – Uniform Resource Locaters

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

Some common URLs.

7.3. 1. Architectural Overview
Statelessness and Cookies

- Web is basically stateless. There is no concept of a login session.
- The browser sends a request to a server and gets back a file.
- When Web was just used for retrieving publicly available documents, this model was perfectly adequate.
- But as Web started to acquire other functions, it caused problems.

7.3. 1. Architectural Overview
Statelessness and Cookies

- For example, some Web sites require clients to register (and possibly pay money) to use them.
- This raises the question of how servers can distinguish between requests from registered users and everyone else.
- Other examples are e-commerce, customized Web portals such as Yahoo

7.3. 1. Architectural Overview
Statelessness and Cookies

- Cookies are solve this problem
- When a client requests a Web page, the server can supply additional information along with the requested page.
- This information may include a cookie, which is a small (at most 4 KB) file (or string).

7.3. 1. Architectural Overview
Statelessness and Cookies

- A cookie may contain up to five fields.

Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

Some examples of cookies.

7.3.2. Static Web Documents

- The basic of the Web is transferring Web pages from server to client.
- In the simplest form, Web pages are static, that is, are just files sitting on some server waiting to be retrieved.
- In this context, even a video is a static Web page because it is just a file.

7.3.2. Static Web Documents

- Web pages are currently written in a language called HTML (Hyper Text Markup Language).
- HTML allows users to produce Web pages that include text, graphics, and pointers to other Web pages.
- HTML is a markup language, a language for describing how documents are to be formatted.

7.3.2. Static Web Documents

- Markup languages thus contain explicit commands for formatting.
- For example, in HTML, means start boldface mode, and means leave boldface mode.
- The advantage: the browser simply has to understand the markup commands.

7.3.2. Static Web Documents

- By embedding all the markup commands within each HTML file and standardizing them, it becomes possible for any Web browser to read and reformat any Web page.
- A page may have been produced in a 1600x1200 window with 24 bit color or may have to be displayed in a 640x320 window configured for 8-bit color.

HTML – HyperText Markup Language

```
<html>
<head><title> AMALGAMATED  WIDGET, INC. </title> </head>
<body> <h1> Welcome to AWI's Home Page</h1>
<img src="http://www.widget.com/images/logo.gif" ALT="AWI Logo"> <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)

(b)

(a)The HTML for a sample Web page.

(b) The formatted page.

HTML

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

A selection of common HTML tags.
some can have additional parameters.

Forms

(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>   <td> x <td> x <td> x </tr>
<tr> <th> Forms <td>   <td> x <td> x <td> x </tr>
<tr> <th> Equations <td>   <td>   <td> x <td> x </tr>
<tr> <th> Toolbars <td>   <td>   <td> x <td> x </tr>
<tr> <th> Tables <td>   <td>   <td> x <td> x </tr>
<tr> <th> Accessibility features <td>   <td>   <td>   <td> x </tr>
<tr> <th> Object embedding <td>   <td>   <td>   <td> x </tr>
<tr> <th> Scripting <td>   <td>   <td>   <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)

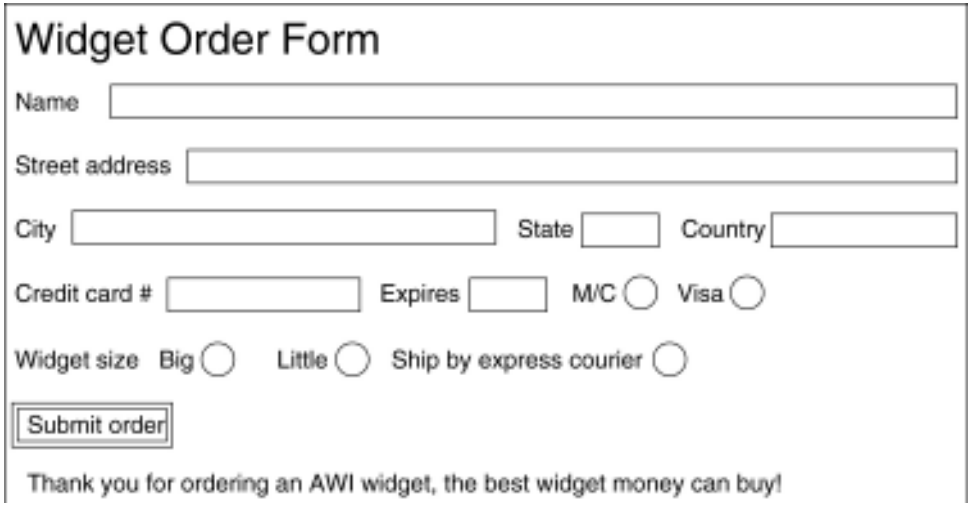
Forms (2)

(a) The HTML for an order form.

(b) The formatted page.

```
<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)

Forms (3)

```
customer=John+Doe&address=100+Main+St.&city=White+Plains&
state=NY&country=USA&cardno=1234567890&expires=6/98&cc=mastercard&
product=cheap&express=on
```

A possible response from the browser to the server with information filled in by the user.

7.3.2. Static Web Documents

- HTML, with or without forms, does not provide any structure to Web pages.
- It also mixes the content with the formatting.
- As e-commerce and other applications become more common, there is an increasing need for structuring Web pages and separating the content from the formatting.

7.3.2. Static Web Documents

- XML (eXtensible Markup Language)
- These structures are extremely simple
- It is permitted to have structures with repeated fields (e.g., multiple authors), optional fields (e.g., title of included CD-ROM), and alternative fields (e.g., URL of a bookstore if it is in print or URL of an auction site if it is out of print).

XML

A simple Web page in XML.

```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>

<book_list>

<book>
   <title> Computer Networks, 4/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 2003 </year>
</book>

<book>
   <title> Modern Operating Systems, 2/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 2001 </year>
</book>

<book>
   <title> Structured Computer Organization, 4/e </title>
   <author> Andrew S. Tanenbaum </author>
   <year> 1999 </year>
</book>

</book_list>
```

7.3.2. Static Web Documents

- XSL (eXtensible Style Language)
- Previous slide defines a book list containing three books.
- It says nothing about how to display the Web page on the screen.
- To provide the formatting information, we need a second file, *book_list.xsl,* containing the XSL definition.
- This file is a style sheet that tells how to display the page.

XML and XSL (2)

A style sheet in XSL.

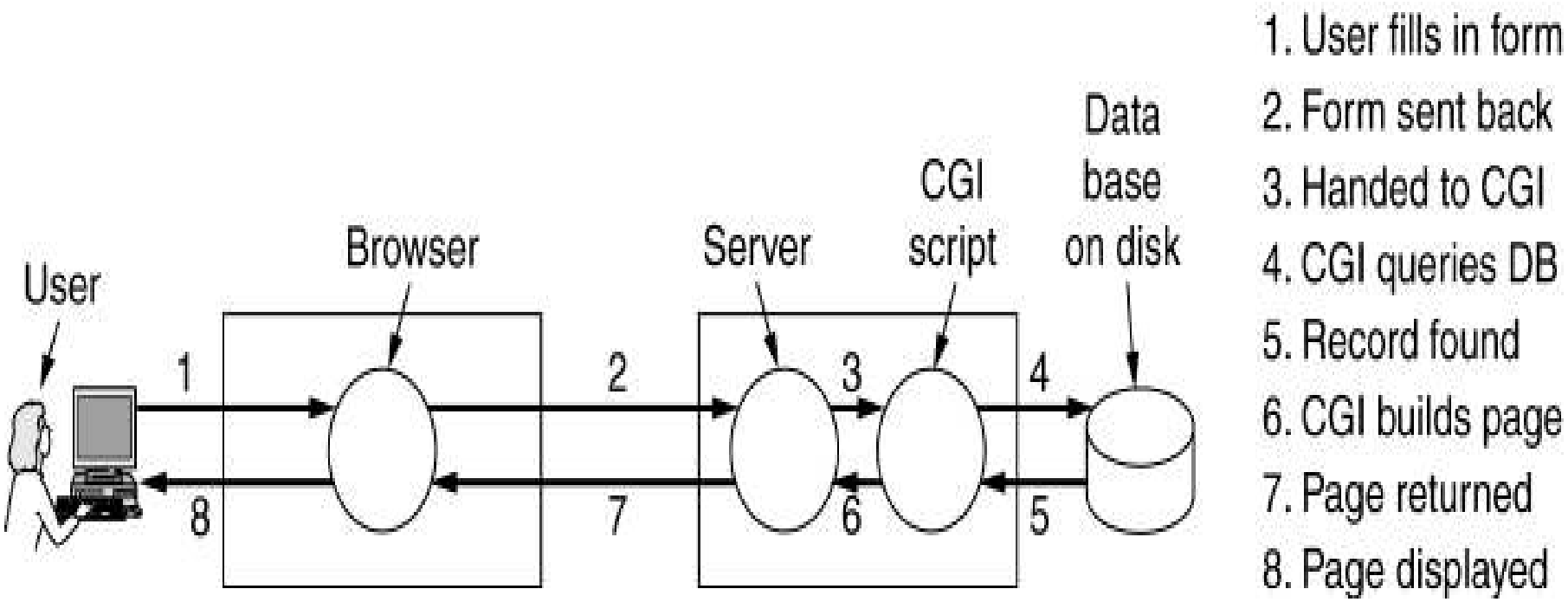
```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">

<html>
<body>

<table border="2">
  <tr>
     <th> Title</th>
     <th> Author</th>
     <th> Year </th>
  </tr>

  <xsl:for-each select="book_list/book">
  <tr>
     <td> <xsl:value-of select="title"/> </td>
     <td> <xsl:value-of select="author"/> </td>
     <td> <xsl:value-of select="year"/> </td>
 </tr>
 </xsl:for-each>
</table>

</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

7.3.3. Dynamic Web Documents

- According to client-server model, the client sends a file name to the server, which then returns the file.
- In the early days of the Web, all content was, in fact, static like this (just files).
- In recent years, more and more content has become dynamic, that is, generated on demand, rather than stored on disk.
- Content generation can take place either on the servere side or on the client side.

Dynamic Web Documents

Servere-side dynamic Web page generation



Steps in processing the information from an HTML form.

Dynamic Web Documents

Servere-side dynamic Web page generation

```
<html>
<body>

<h2> This is what I know about you </h2>
<?php echo $HTTP_USER_AGENT ?>

</body>
</html>
```

A sample HTML page with embedded PHP.

Dynamic Web Documents

Servere-side dynamic Web page generation

```
<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)

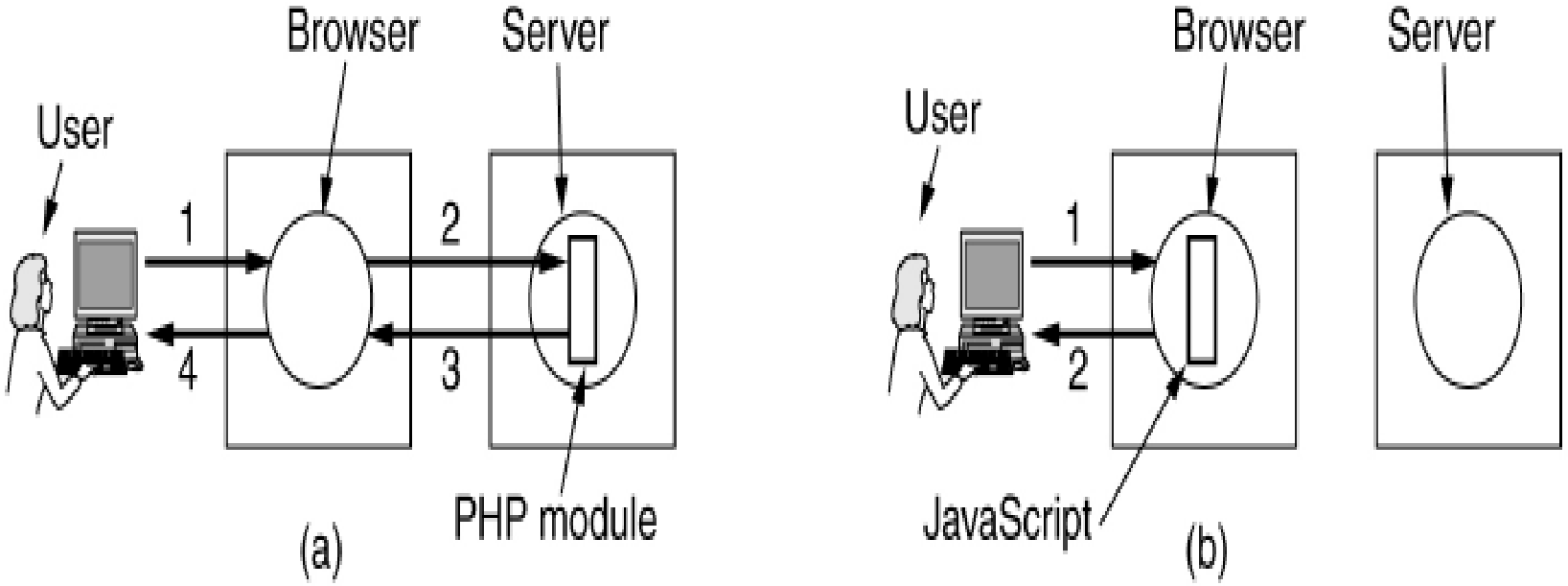
(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.

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)

```
<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>
```

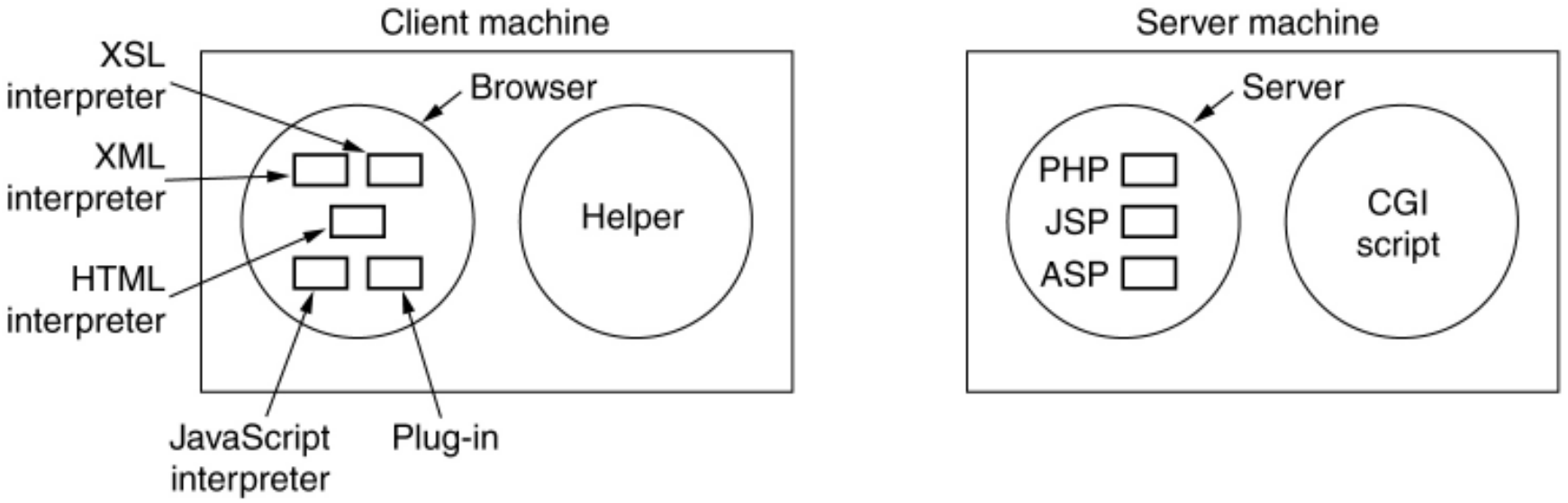
A JavaScript program for computing and printing factorials.

Client-Side Dynamic Web Page Generation (4)

```
<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>
```

An interactive Web page that responds to mouse movement.

Client-Side Dynamic Web Page Generation (5)



The various ways to generate and display content.

7.3.4. HTTP (1)

- The transfer protocol used throughout the World Wide Web is HTTP (HyperText Transfer Protocol)
- It specifies what message client may send to servers and what responses they get back in return.
- Each interaction consists of one ASCII request, followed by one RFC 822 MIME-like response

7.3.4. HTTP (2)
Connections

- The usual way for a browser to contact a server is to establish a TCP connection to port 80 on the server's machine.
- HTTP 1.0: after the connection was established, a single request was sent over and a single response was sent back. Then the TCP connection was released.
- HTTP 1.1: supports persistent connections. Establishing a TCP connection, sending a request and getting a response, and then sending additional requests and getting additional responses.

7.3.4. HTTP 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

The built-in HTTP request methods.

7.3.4. HTTP Methods

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

The status code response groups.

7.3.4. 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

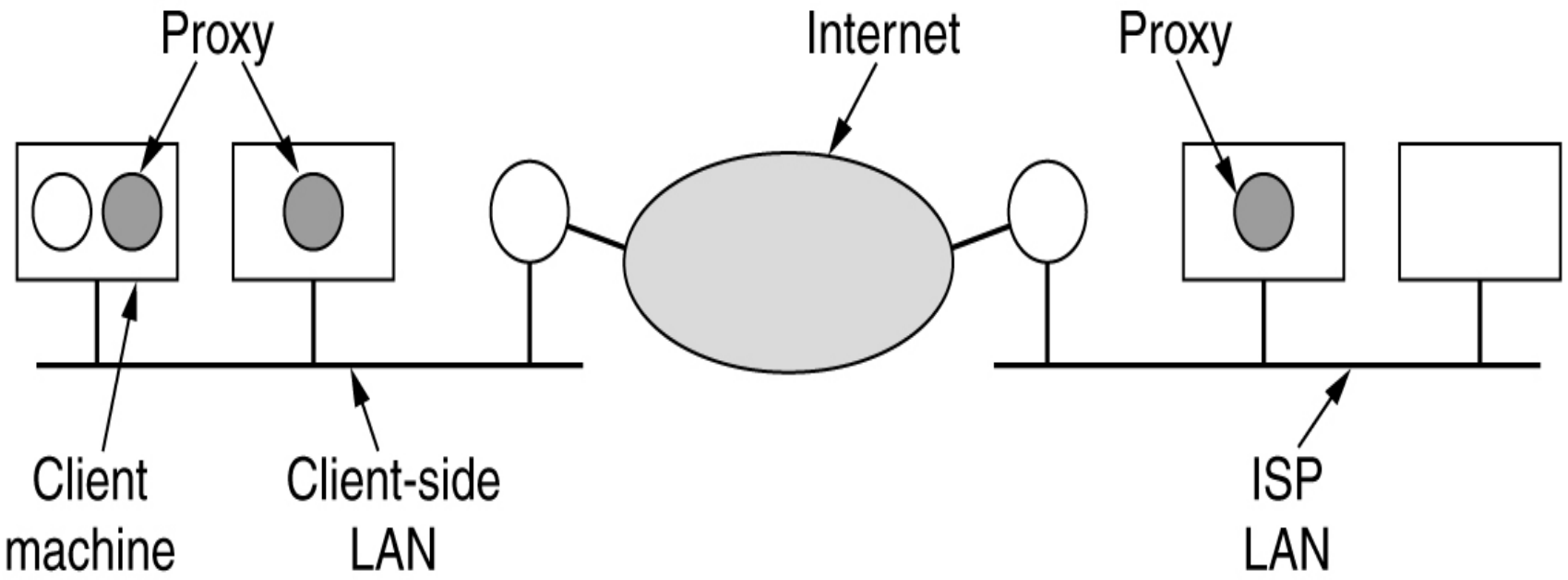
Some HTTP message headers.

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>
```

7.3.5. Performance Enhancements Caching



Hierarchical caching with three proxies.

Content Delivery Networks

```
<html>
<head> <title> Furry Video </title> </head>
<body>
<h1> Furry Video's Product List </h1>
<p> Click below for free samples. </p>
<a href="bears.mpg"> Bears Today </a> <br>
<a href="bunnies.mpg"> Funny Bunnies </a> <br>
<a href="mice.mpg"> Nice Mice </a> <br>
</body>
</html>
```

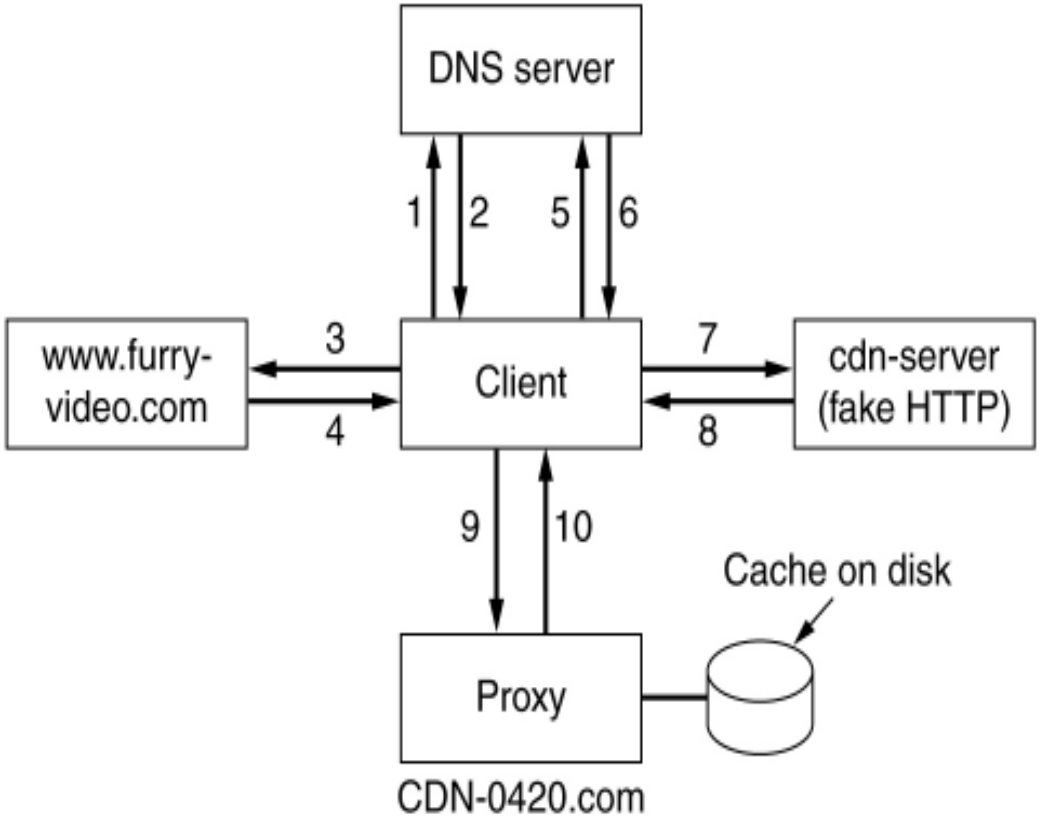
(a)

```
<html>
<head> <title> Furry Video </title> </head>
<body>
<h1> Furry Video's Product List </h1>
<p> Click below for free samples. </p>
<a href="http://cdn-server.com/furryvideo/bears.mpg"> Bears Today </a> <br>
<a href="http://cdn-server.com/furryvideo/bunnies.mpg"> Funny Bunnies </a> <br>
<a href="http://cdn-server.com/furryvideo/mice.mpg"> Nice Mice </a> <br>
</body>
</html>
```

(b)

(a) Original Web page. (b) Same page after transformation.

Content Delivery Networks



1. Look up www.furryvideo.com
2. Furry's IP address returned
3. Request HTML page from Furry
4. HTML page returned
5. After click, look up cdn-server.com
6. IP address of cdn-server returned
7. Ask cdn-server for bears.mpg
8. Client told to redirect to CDN-0420.com
9. Request bears.mpg
10. Cached file bears.mpg returned

Steps in looking up a URL when a CDN is used.

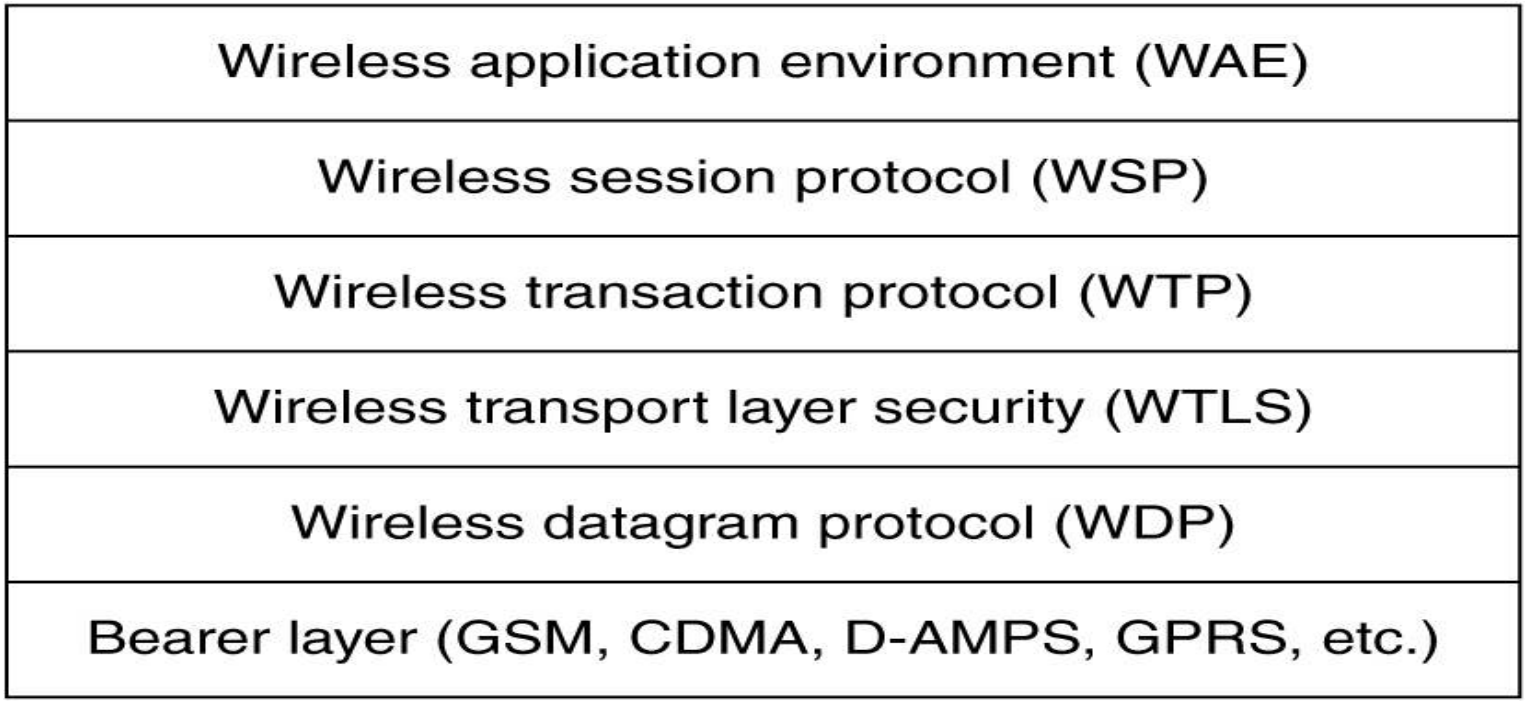
7.3.6. The Wireless Web

- There is considerable interest in small portable devices capable of accessing the Web via a wireless link.
- It is still worth examining some of the current ideas relating to the wireless Web to see where we are now and where we might be heading.
- There are many wide area wireless growing up, two of them are WAP and i-mode

7.3.6. The Wireless Web
WAP

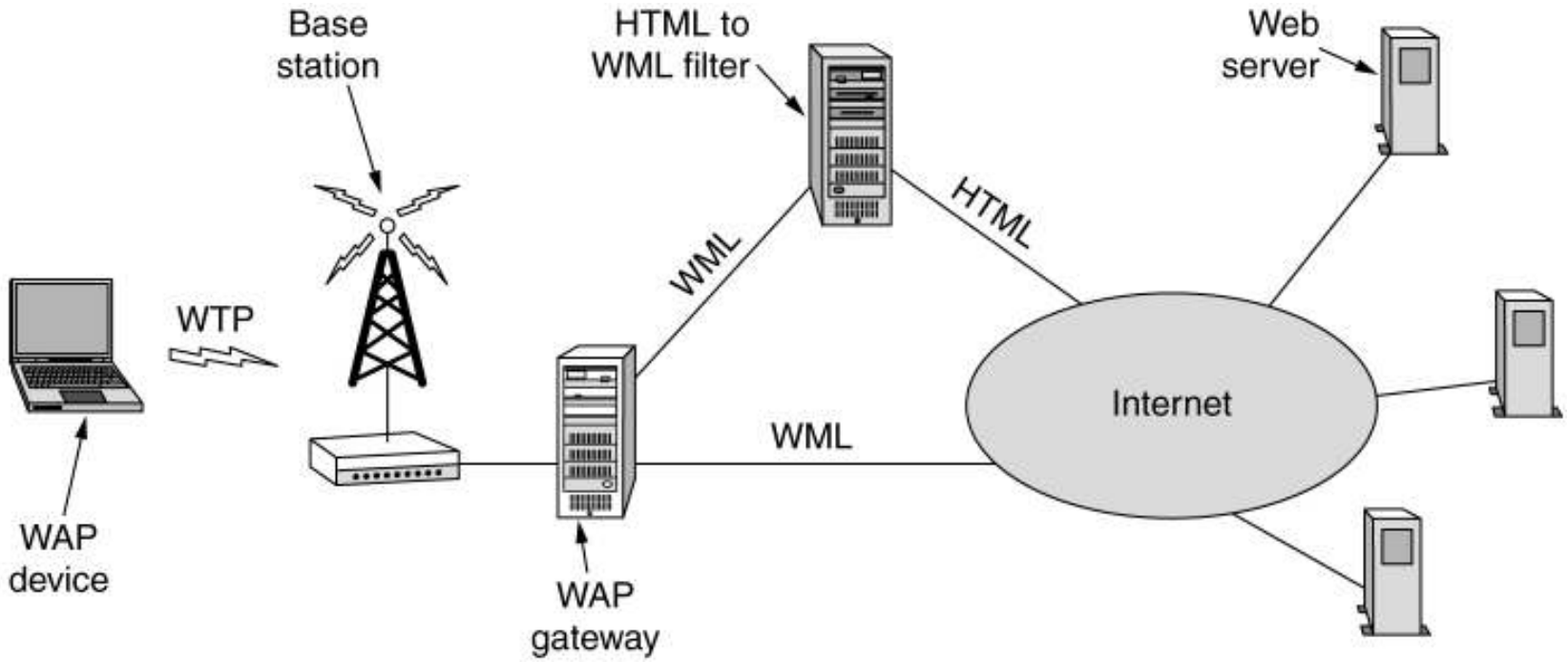
- Nokia, Ericsson, Motorola, etc. got the idea to combine the Internet and mobile phones into a mobile phone with a built-in screen for wireless access to e-mail and the Web.
- The system is called WAP (The Wireless Application Protocol)

7.3.6. The Wireless Web
WAP – The Wireless Application Protocol



The WAP protocol stack.

WAP

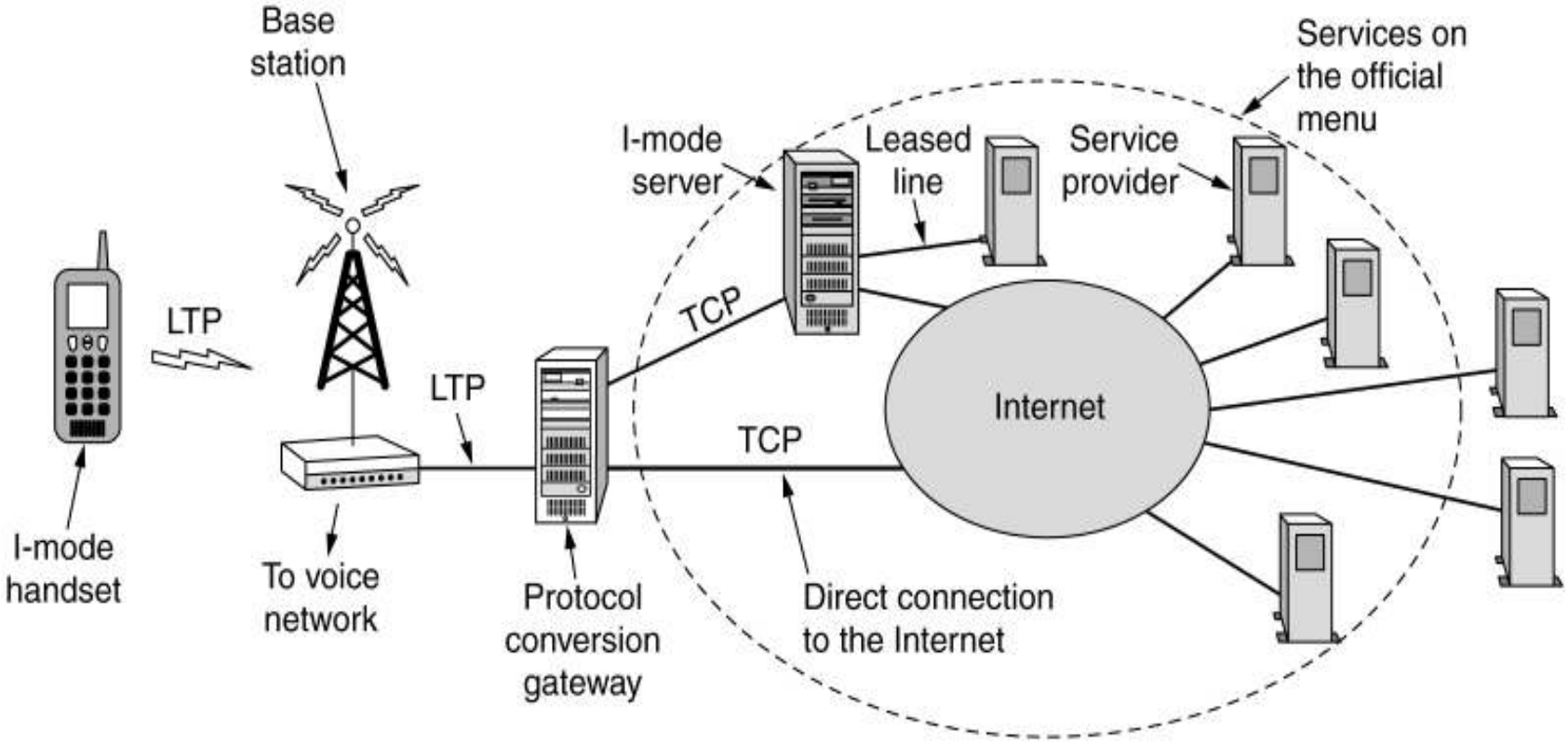


The WAP architecture.

7.3.6. The Wireless Web
I-Mode

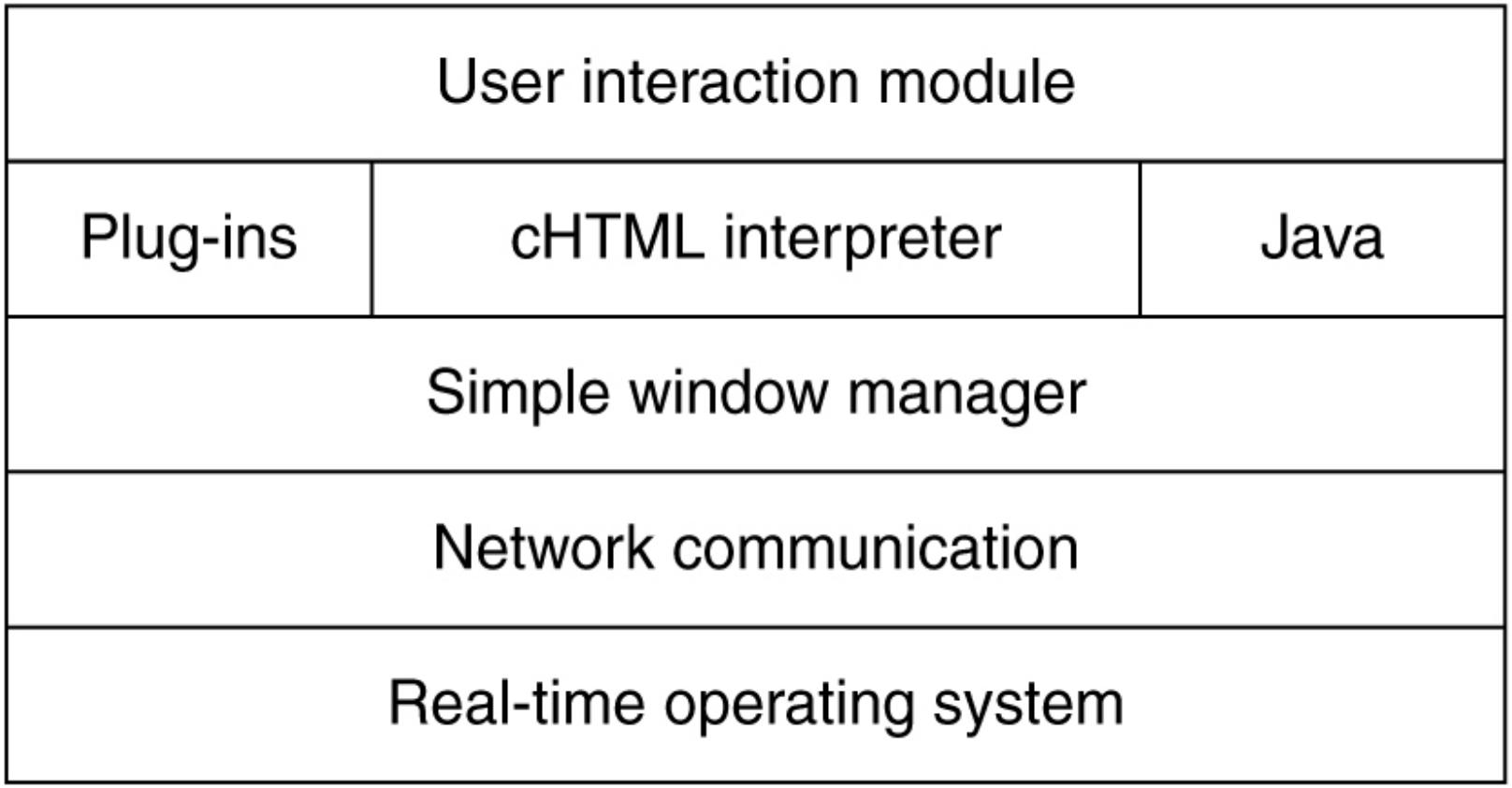
- A Japanesw woman invented a different approach to the wireles Web called i-mode (information-mode)
- The i-mode system has three major components: a new transmition system, a new handset, and a new language for Web page design

7.3.6. The Wireless Web
I-Mode



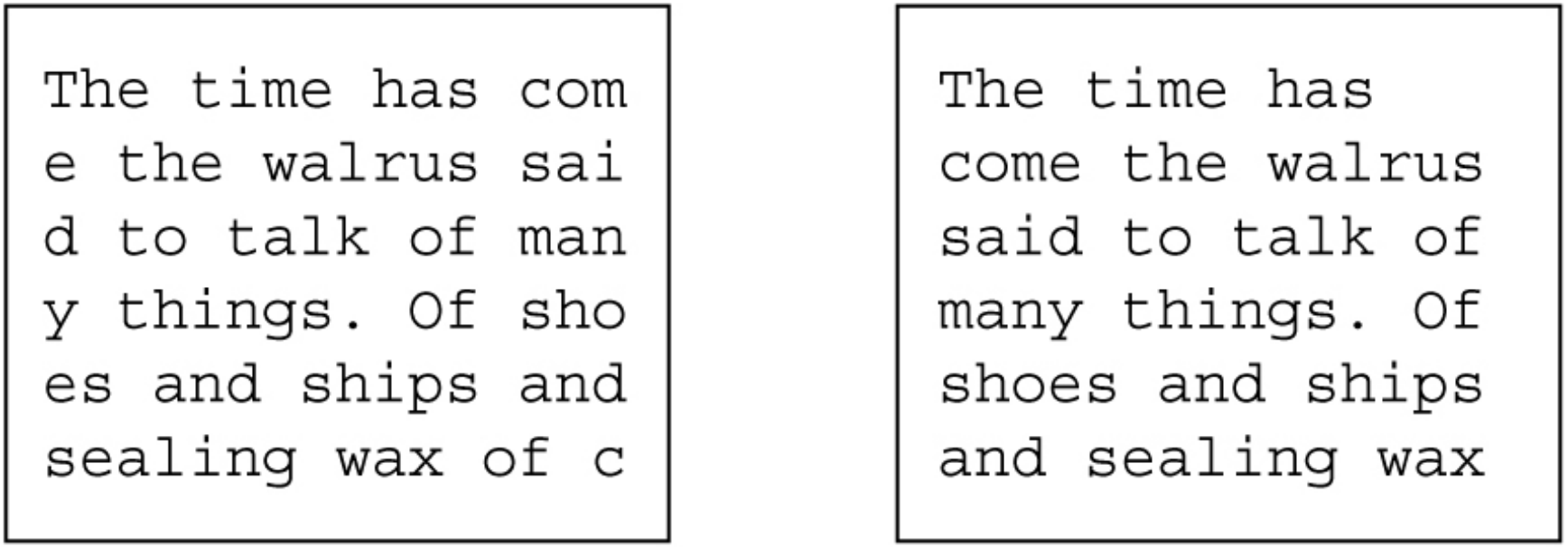
Structure of the i-mode data network showing the transport protocols.

I-Mode (2)



Structure of the i-mode software.

I-Mode (3)



Lewis Carroll meets a 16 x 16 screen.

I-Mode (4)

```
<html>
<body>
<h1> Select an option </h1>
<a href="messages.chtml" accesskey="1"> Check voicemail </a> <br>
<a href="mail.chtml" accesskey="2"> Check e-mail </a> <br>
<a href="games.chtml" accesskey="3"> Play a game </a>
</body>
</html>
```

An example of cHTML file.

Second-Generation Wireless Web

Feature	WAP	I-mode
What it is	Protocol stack	Service
Device	Handset, PDA, notebook	Handset
Access	Dial up	Always on
Underlying network	Circuit-switched	Two: circuit + packet
Data rate	9600 bps	9600 bps
Screen	Monochrome	Color
Markup language	WML (XML application)	cHTML
Scripting language	WMLscript	None
Usage charges	Per minute	Per packet
Pay for shopping	Credit card	Phone bill
Pictograms	No	Yes
Standardization	WAP forum open standard	NTT DoCoMo proprietary
Where used	Europe, Japan	Japan
Typical user	Businessman	Young woman

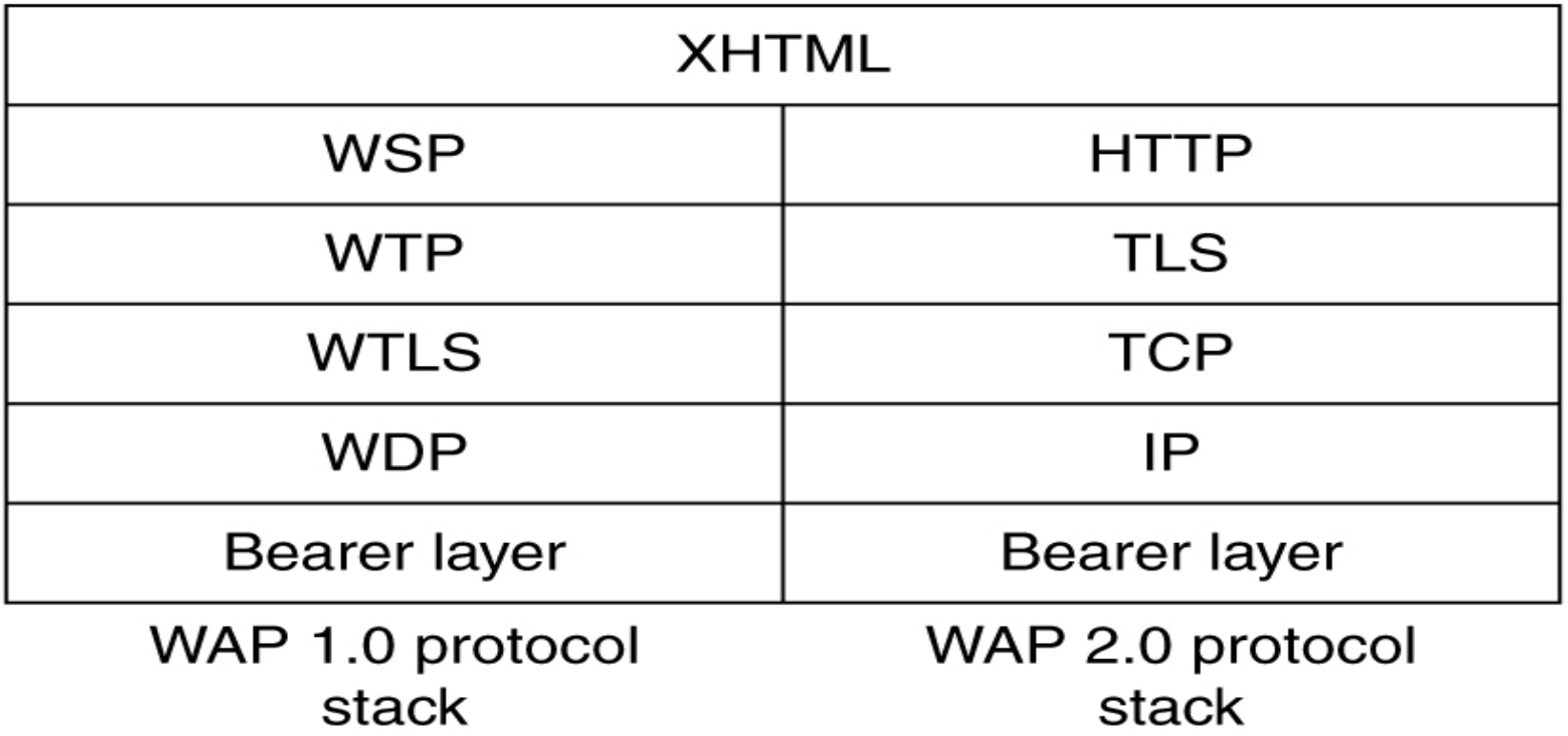
A comparison of first-generation WAP and i-mode.

Second-Generation Wireless Web (2)

New features of WAP 2.0.

- Push model as well as pull model.
- Support for integrating telephony into apps.
- Multimedia messaging.
- Inclusion of 264 pictograms.
- Interface to a storage device.
- Support for plug-ins in the browser.

Second-Generation Wireless Web (3)



WAP 2.0 supports two protocol stacks.

Second-Generation Wireless Web (4)

Module	Req.?	Function	Example tags
Structure	yes	Doc. structure	body, head, html, title
Text	yes	Information	br, code, dfn, em, h*n*, kbd, p, strong
Hypertext	yes	Hyperlinks	a
List	yes	Itemized lists	dl, dt, dd, ol, ul, li
Forms	No	Fill-in forms	form, input, label, option, textarea
Tables	No	Rectangular tables	caption, table, td, th, tr
Image	No	Pictures	img
Object	No	Applets, maps, etc.	object, param
Meta-information	No	Extra info	meta
Link	No	Similar to <a>	link
Base	No	URL starting point	base

The XHTML Basic modules and tags.

7.4. Multimedia

- Multimedia is also a rising star in the networking firmament.
- It allows audio and video to be digitized and transported electronically for display.
- Audio requires less bandwidth, so it is further along.
- Streaming audio, Internet radio, and voice over IP are a reality now.

7.4. Multimedia

- Video on demand is an up-and-coming area in which there is great interest.
- Finally, the MBone is an experimental, worldwide digital live television service sent over the Internet.
- For many people, multimedia is the holy grail of networking.

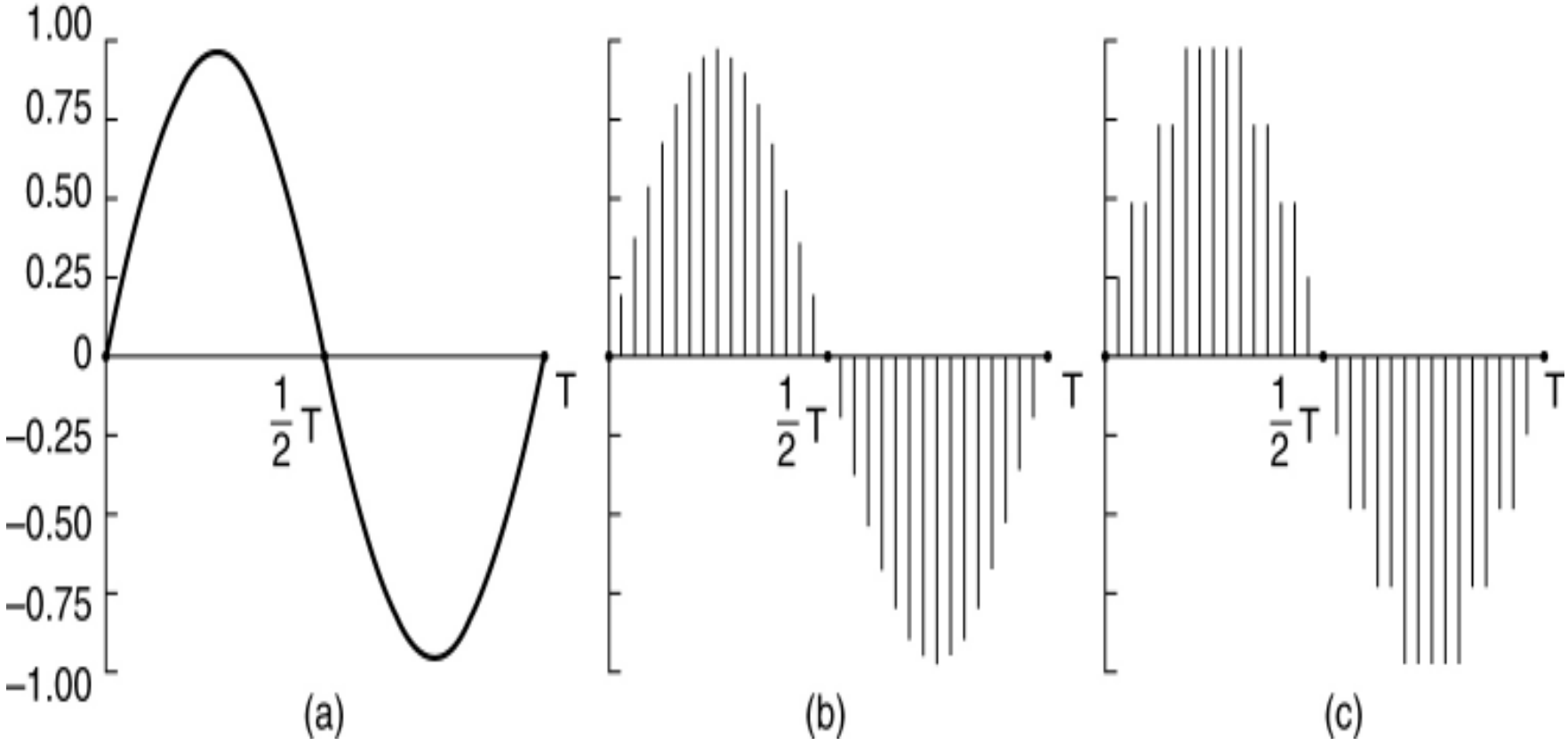
7.4. Multimedia

- Introduction to Audio
- Audio Compression
- Streaming Audio
- Internet Radio
- Voice over IP
- Introduction to Video
- Video Compression
- Video on Demand
- The MBone – The Multicast Backbone

7.4.1.Introduction to Audio

- An audio (sound) wave is a one-dimensional acoustic (pressure) wave.
- When an acoustic wave strikes a microphone, the microphone generates an electrical signal, representing the sound amplitude as a function of time.
- The representation, processing, storage, and transmission of such audio signals are a major part of the study of multimedia systems.

Introduction to Audio



(a) A sine wave. (b) Sampling the sine wave.
(c) Quantizing the samples to 4 bits.

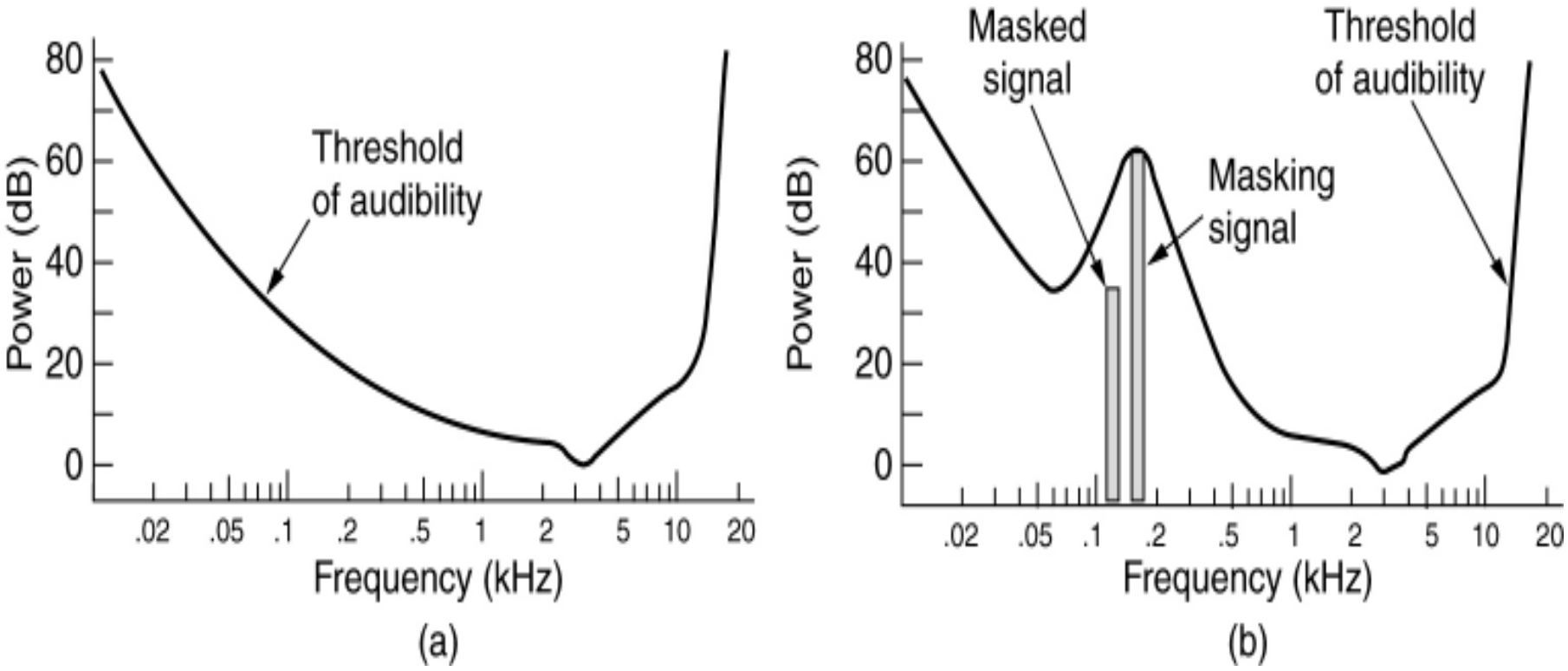
7.4.1.Introduction to Audio

- Audio waves can be converted to digital form by an ADC (Analog Digital Convertor)
- ADC takes an electrical voltage as input and generates a binary number as output.
- Telephone, audio compact discs are examples where sound is sampled.
- Music, speech are special cases of general audio.

7.4.2. Audio Compression

- CD-quality audio requires a transmission bandwidth of 1.411 Mbps.
- Clearly, substantial compression is needed to make transmission over the Internet practical.
- For this reason, various audio compression algorithms have been developed (MP3 is the most powerful and best known).

7.4.2. Audio Compression



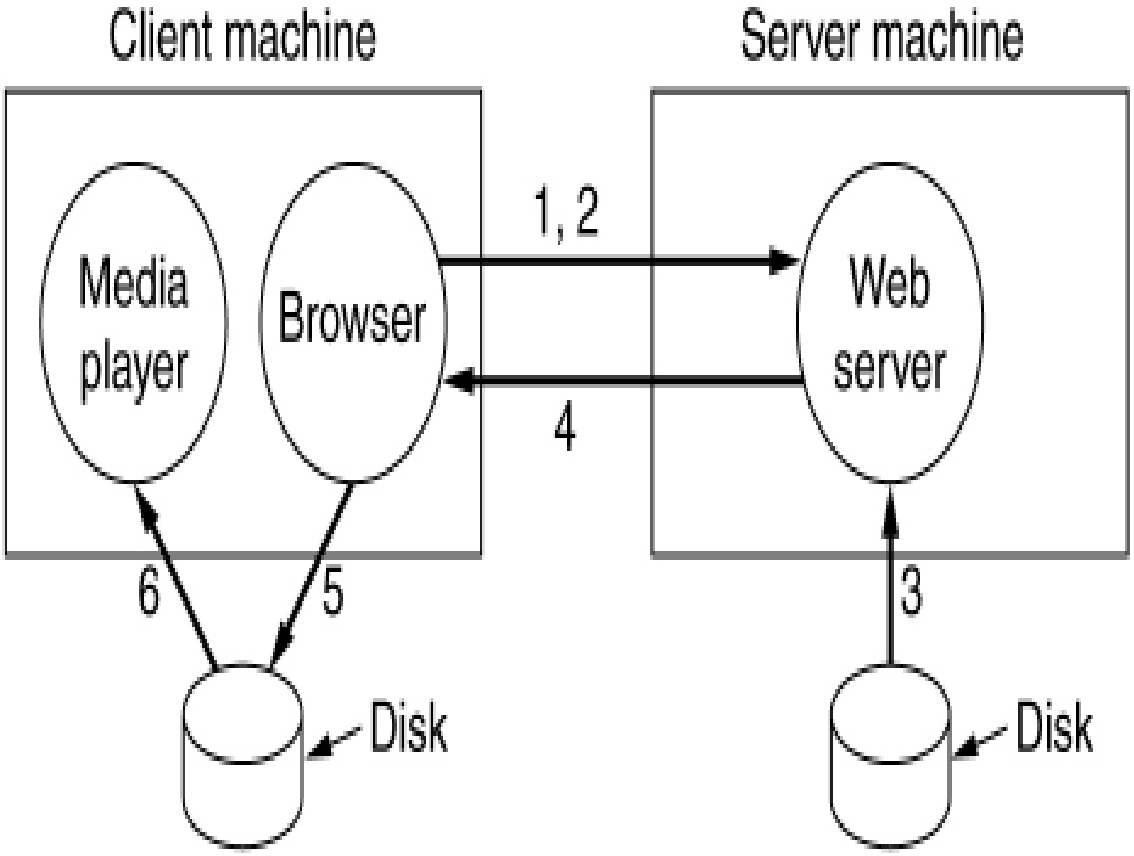
(a) The threshold of audibility as a function of frequency.

(b) The masking effect.

7.4.3. Streaming Audio

- Streaming Audio is listening to sound over the Internet.
- This is also called music on demand.
- The Internet is full of music Web sites, many of which list song title that users can click on to play the songs.

Streaming Audio



1. Establish TCP connection
2. Send HTTP GET request
3. Server gets file from disk
4. File sent back
5. Browser writes file to disk
6. Media player fetches file block by block and plays it

A straightforward way to implement clickable music on a Web page.

7.4.3. Streaming Audio

- The media player has four major jobs to do:

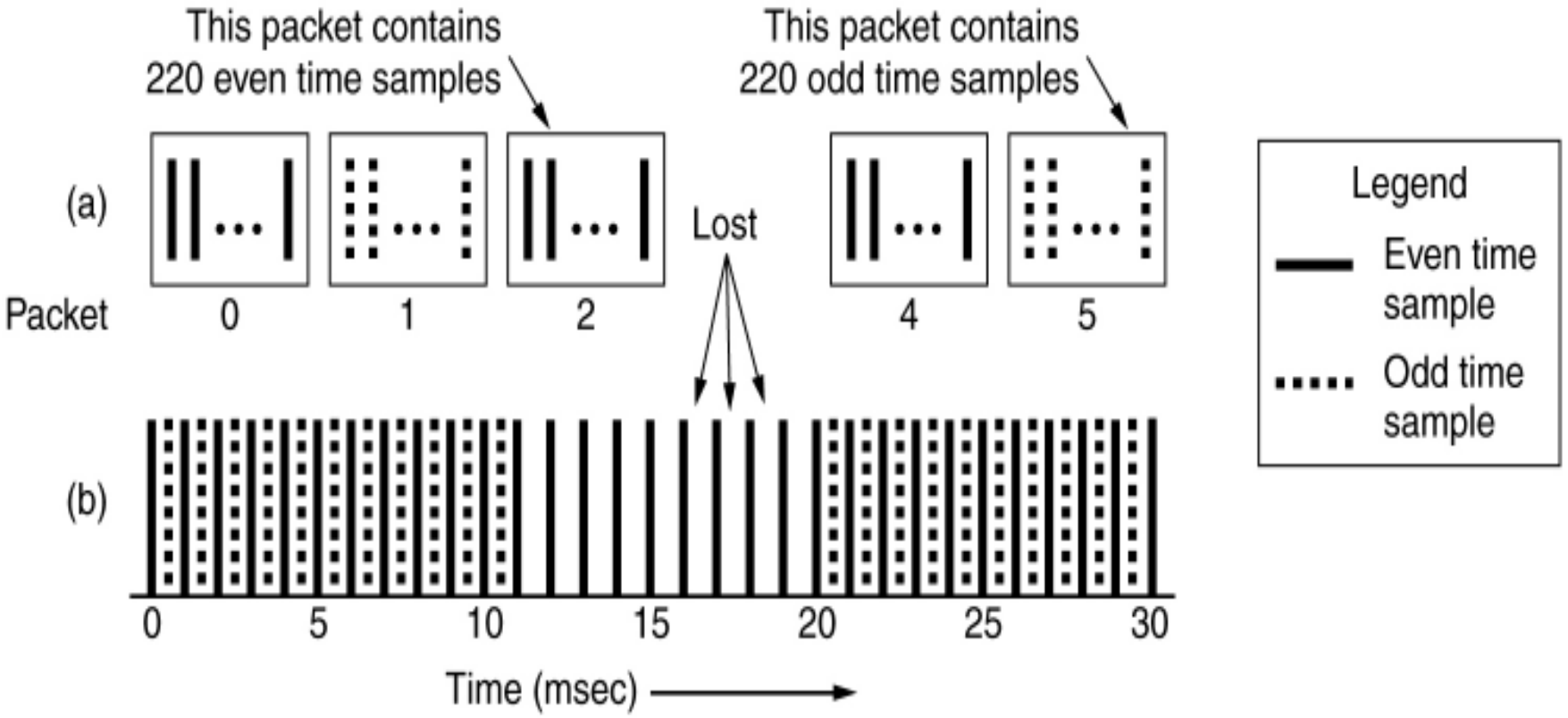
a) Manage the user interface

b) Handle transmission errors

c) Decompress the music

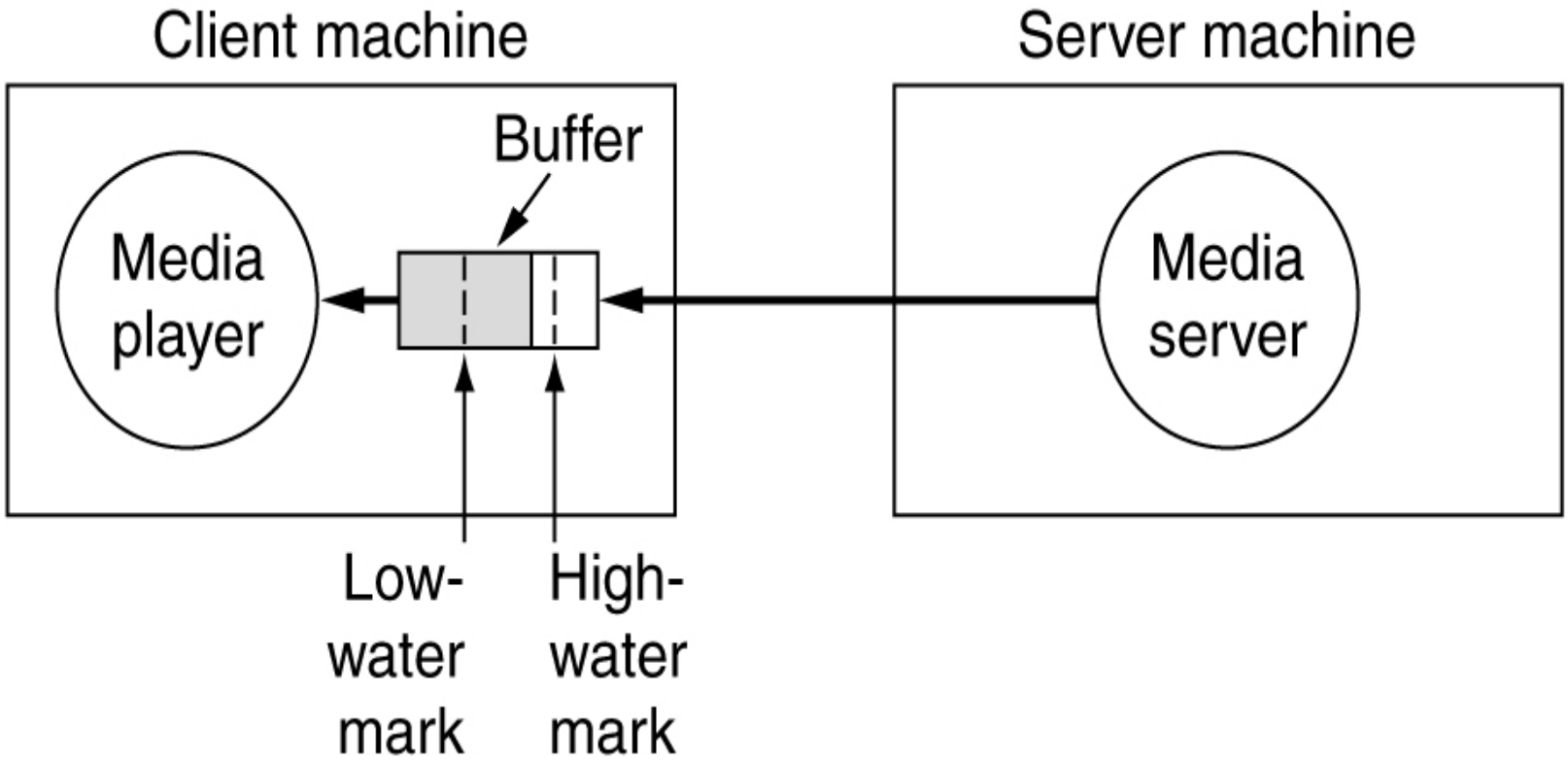
d) Eliminate jitter

7.4.3. Streaming Audio



When packets carry alternate samples, the loss of a packet reduces the temporal resolution rather than creating a gap in time.

7.4.3. Streaming Audio



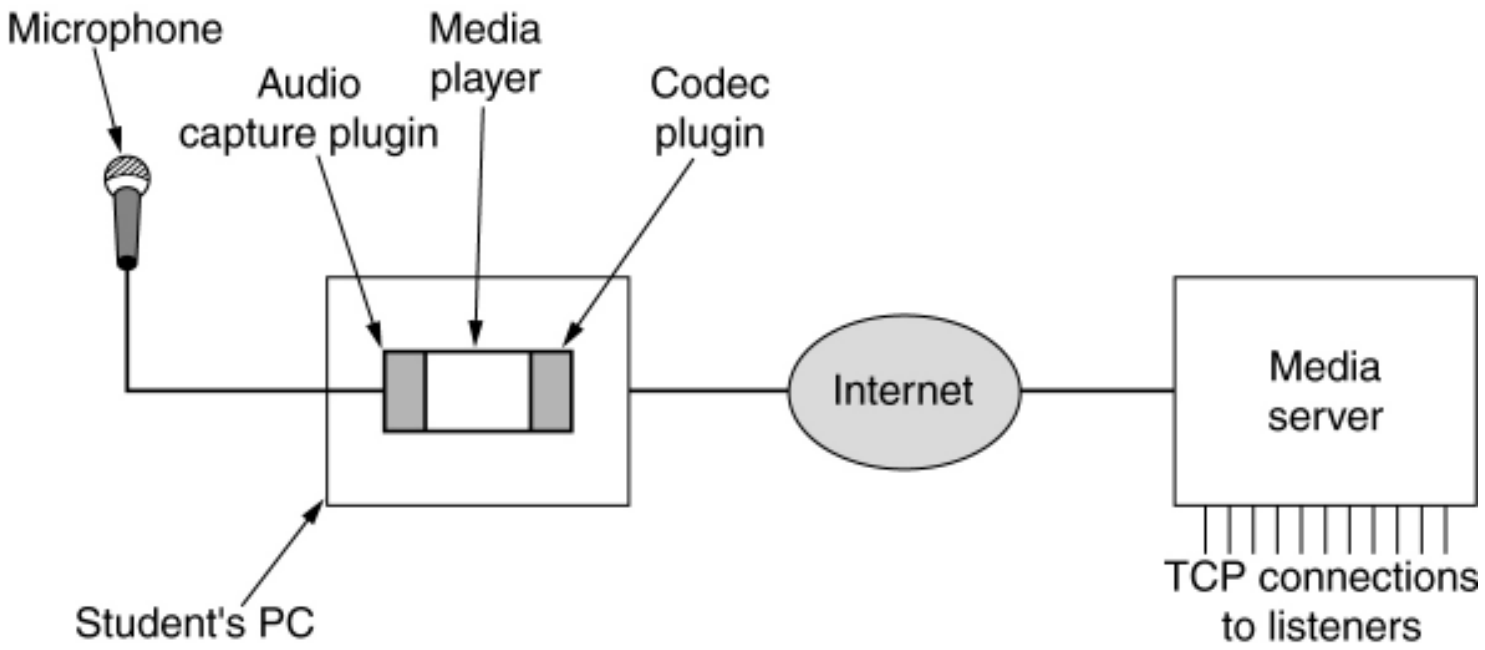
The media player buffers input from the media server and plays from the buffer rather than directly from the network.

7.4.3. Streaming Audio

Command	Server action
DESCRIBE	List media parameters
SETUP	Establish a logical channel between the player and the server
PLAY	Start sending data to the client
RECORD	Start accepting data from the client
PAUSE	Temporarily stop sending data
TEARDOWN	Release the logical channel

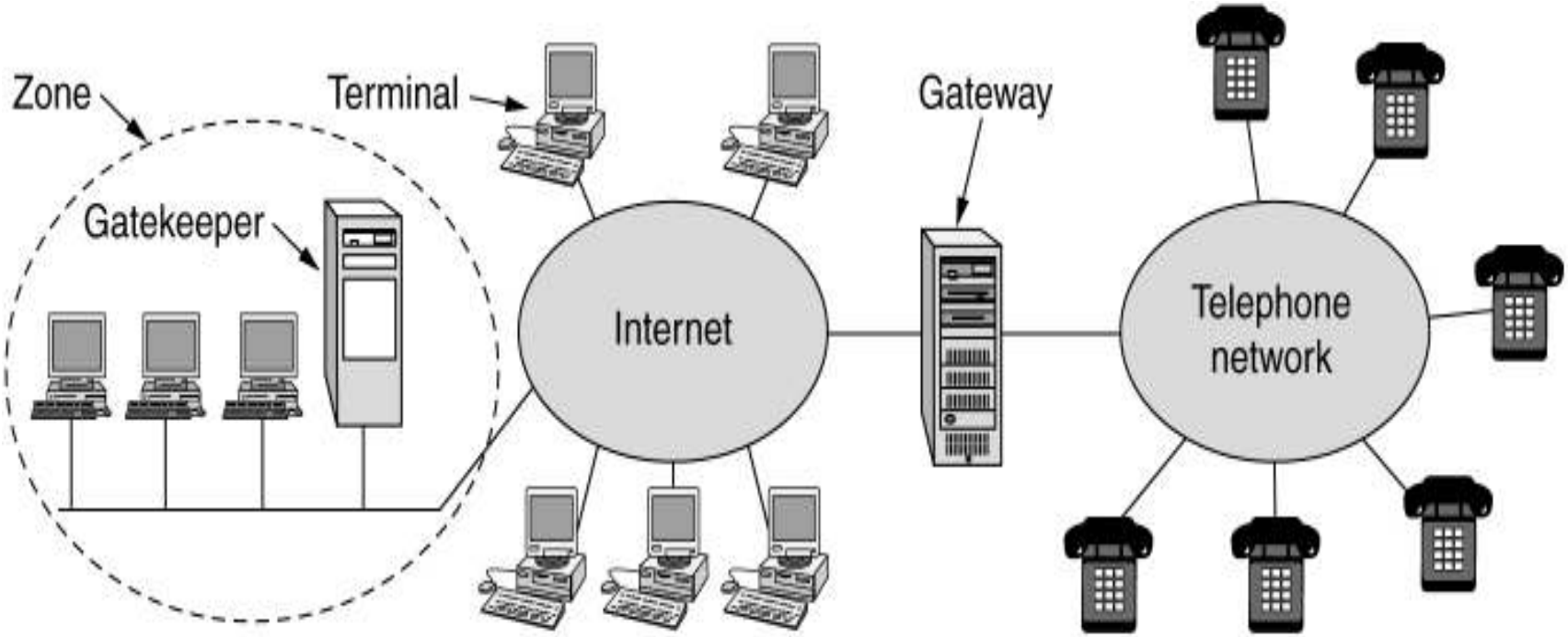
RTSP commands from the player to the server.

7.4.4 Internet Radio



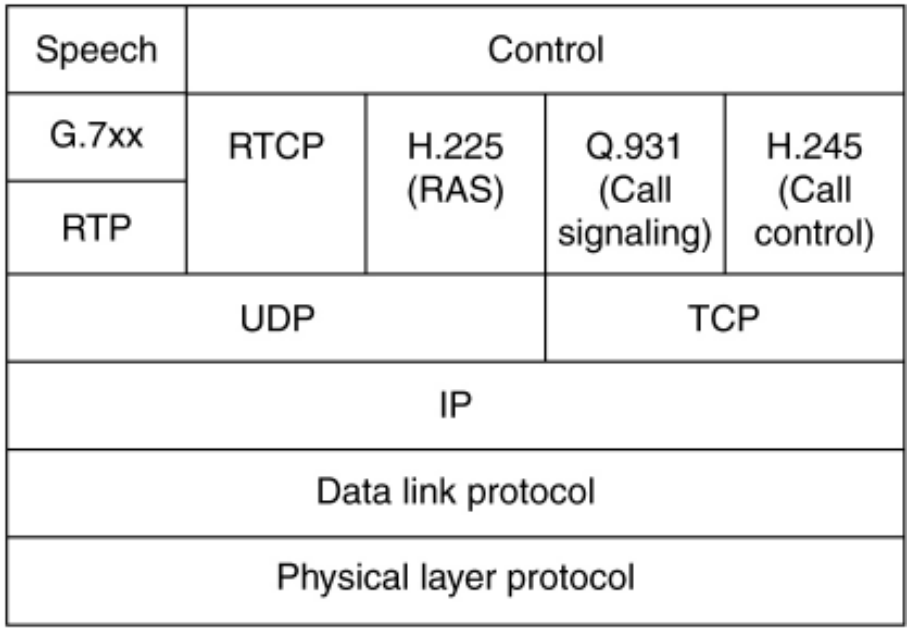
A student radio station.

7.4.5. Voice over IP



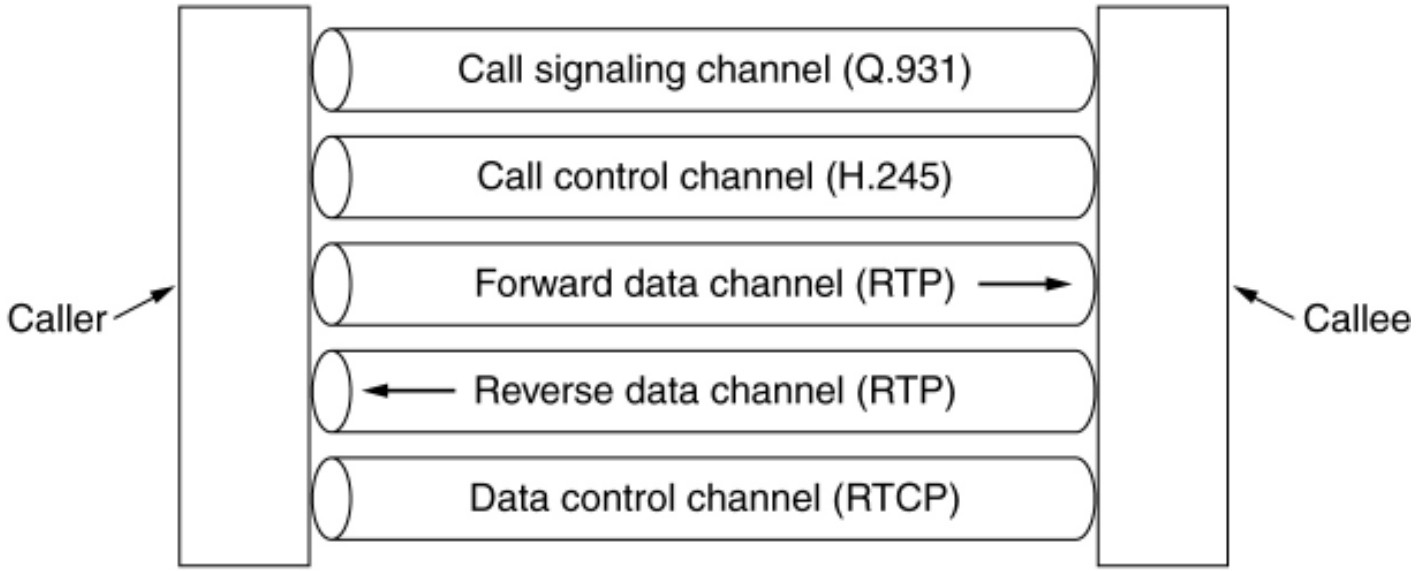
The H323 architectural model for Internet telephony.

7.4.5. Voice over IP (2)



The H323 protocol stack.

7.4.5. Voice over IP (3)



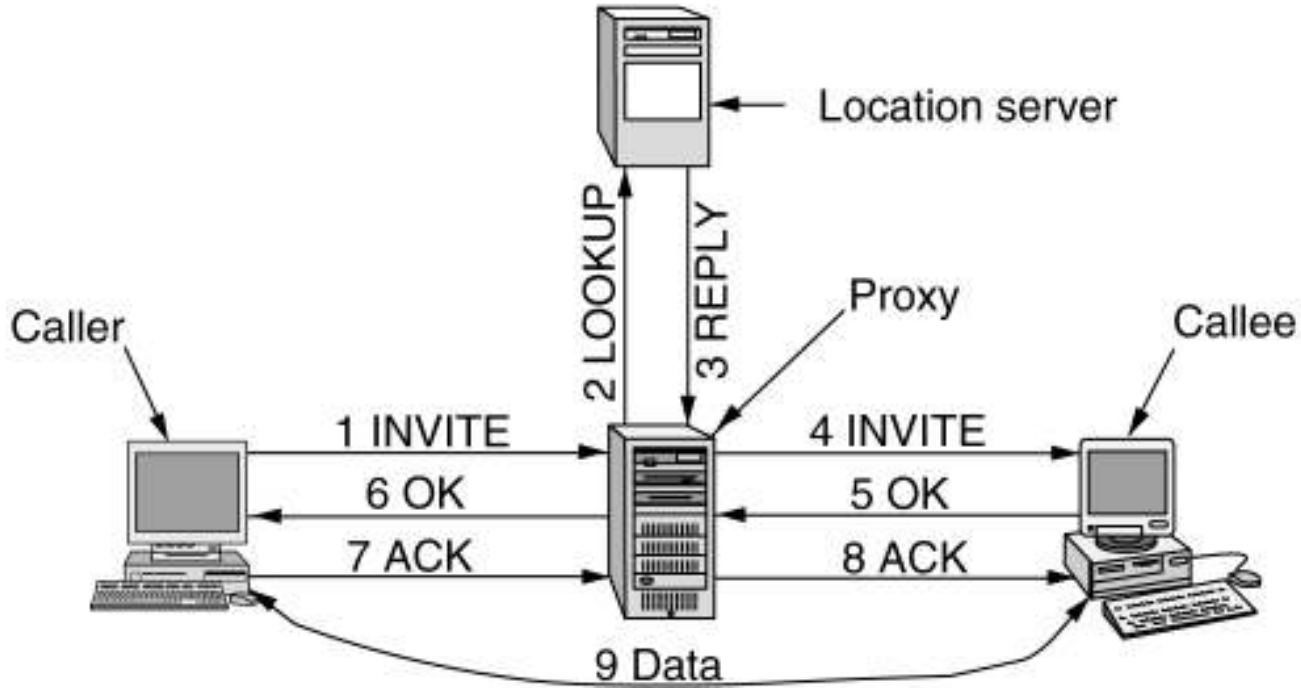
Logical channels between the caller and callee during a call.

7.4.5. SIP – The Session Initiation Protocol

Method	Description
INVITE	Request initiation of a session
ACK	Confirm that a session has been initiated
BYE	Request termination of a session
OPTIONS	Query a host about its capabilities
CANCEL	Cancel a pending request
REGISTER	Inform a redirection server about the user's current location

The SIP methods defined in the core specification.

7.4.5. SIP (2)



Use a proxy and redirection servers with SIP.

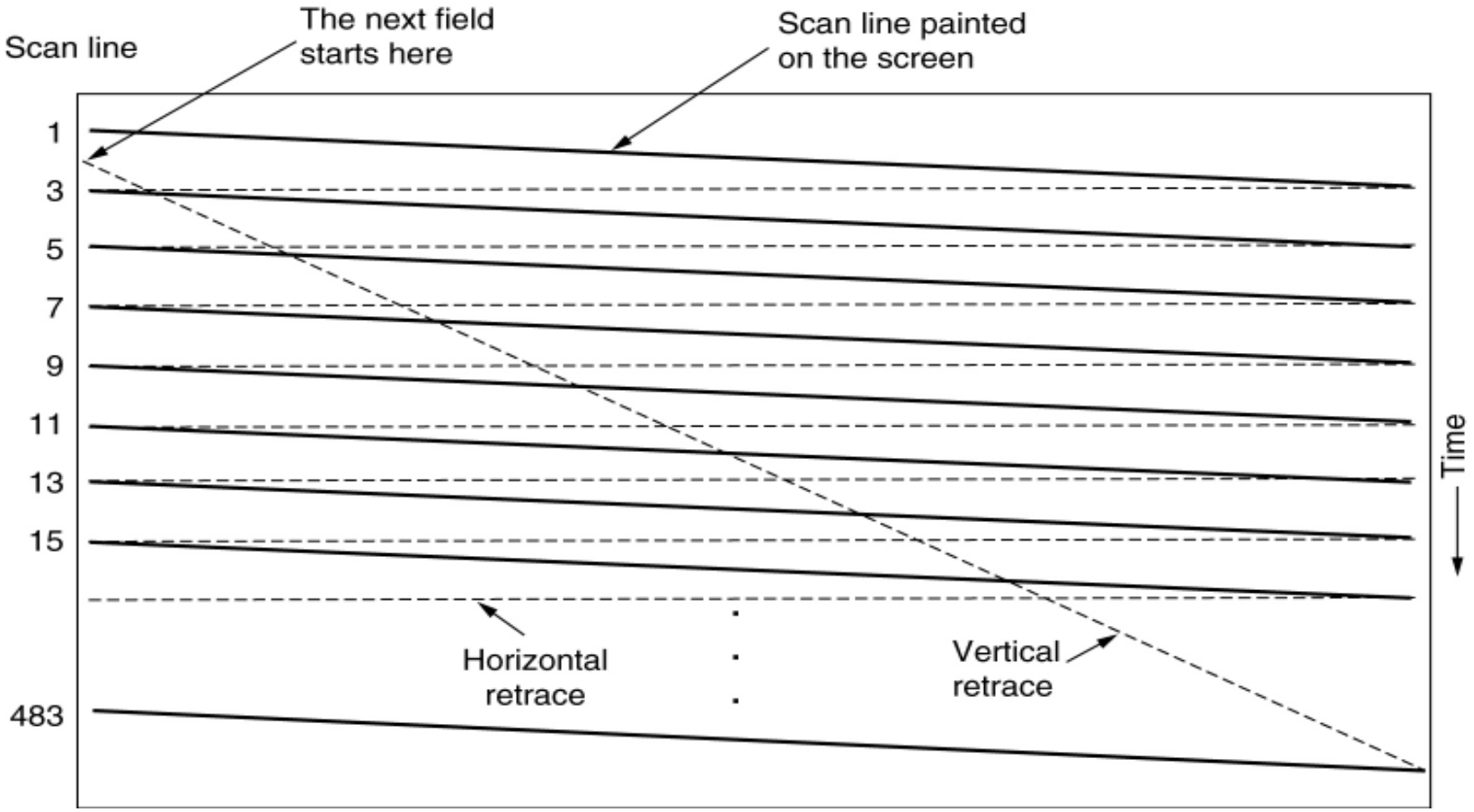
7.4.5. Comparison of H.323 and SIP

Item	H.323	SIP
Designed by	ITU	IETF
Compatibility with PSTN	Yes	Largely
Compatibility with Internet	No	Yes
Architecture	Monolithic	Modular
Completeness	Full protocol stack	SIP just handles setup
Parameter negotiation	Yes	Yes
Call signaling	Q.931 over TCP	SIP over TCP or UDP
Message format	Binary	ASCII
Media transport	RTP/RTCP	RTP/RTCP
Multiparty calls	Yes	Yes
Multimedia conferences	Yes	No
Addressing	Host or telephone number	URL
Call termination	Explicit or TCP release	Explicit or timeout
Instant messaging	No	Yes
Encryption	Yes	Yes
Size of standards	1400 pages	250 pages
Implementation	Large and complex	Moderate
Status	Widely deployed	Up and coming

7.4.6. Introduction to Video

- The human eye has the property that when an image appears on the retina, the image is retained for some number of milliseconds before decaying.
- If a sequence of images is drawn line by line at 50 images/sec, the eye does not notice that it is looking at discrete images.
- All video (i.e. television) systems exploit this principle to produce moving pictures.

Video Analog Systems



The scanning pattern used for NTSC video and television.

7.4.7. Video Compression

- All compression systems require two algorithms: one for compressing the data at the source, and another for decompressing it at the destination.
- In the literature, these algorithms are referred to as the encoding and decoding algorithms, respectively.

7.4.7. Video Compression
The JPEG Standard

- A video is just a sequence of images (plus sound).
- If we could find a good algorithm for encoding a single image, this algorithm could be applied to each image in succession to achieve video compression.
- The JPEG – Joint Photographic Experts Group

7.4.7. Video Compression
The JPEG Standard

- The JPEG Standard – for compressing continuous-tone still pictures (e.g., photographs).
- JPEG has four modes and many options.

The JPEG Standard

The operation of JPEG in lossy sequential mode.

The JPEG Standard (2)

(a) RGB input data.

(b) After block preparation.

The JPEG Standard (3)

(a) (b)

(a) One block of the *Y* matrix.
(b) The DTC coefficients.

The JPEG Standard (4)

Computation of the quantized DTC coefficients.

The JPEG Standard (5)

The order in which the quantized values are transmitted.

The MPEG Standard

- MPEG (Motion Picture Experts Group) standards.
- This are the main algorithms used to compress videos and have been international standards since 1993.
- Because movies contain both images and sound, MPEG can compress both audio and video.

The MPEG Standard

Synchronization of the audio and video streams in MPEG-1.

The MPEG Standard (2)

Three consecutive frames.

Video on Demand

Overview of a video-on-demand system.

Video Servers

A video server storage hierarchy.

Video Servers (2)

The hardware architecture of a typical video server.

The MBone – The Multicast Backbone

MBone consists of multicast islands connected by tunnels.