UNIT V

The Application Layer

PREVIOUS LAYERS

- THE PURPOSE OF THE PHYSICAL LAYER IS TO TRANSPORT A RAW BIT STREAM FROM ONE MACHINE TO ANOTHER.
- THE MAIN TASK OF THE DATA LINK LAYER IS TO TRANSFORM A RAW TRANSMISSION FACILITY INTO A LINE THAT APPEARS FREE OF UNDETECTED TRANSMISSION ERRORS TO THE NETWORK LAYER.

- THE NETWORK LAYER IS CONCERNED WITH GETTING PACKETS FROM THE SOURCE ALL THE WAY TO THE DESTINATION.
- THE TRANSPORT LAYER'S TASK IS TO PROVIDE RELIABLE, COST-EFFECTIVE DATA TRANSPORT FROM SOURCE MACHINE TO DESTINATION MACHINE, INDEPENDENTLY OF THE PHYSICAL Layer

The Application Layer

- Layers below the application layer are there to provide reliable transport, but they do not do real work for users.

- In this chapter we will study some real network applications.

The Application Layer

- In the application layer there is also need for support protocols, to allow the applications to function.
- One of them is DNS which handles naming within the Internet.
- Three real applications:

1. Electronic Mail
2. The World Wide Web
3. Multimedia

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

7.1. DNS – The Domain Name System

- 7.1.1. The DNS Name Space
- 7.1.2. Resource Records
- 7.1.3. Name Servers

7.1. DNS – The Domain Name System

- Network addresses (e.g., IP) are hard for people to remember.
- Also, sending e-mail to tana@128.111.24.41 means that if Tana's ISP moves the mail server to a different machine with different IP address, her e-mail address has to change.

7.1. DNS – The Domain Name System

- Consequently, ASCII names were introduced to decouple machine names from machine addresses.
- In this way, Tana's address might be something like tana@art.ucsb.edu.
- Nevertheless, network itself understand only numerical addresses, so some mechanism is required to convert the ASCII srtings to network addresses.

7.1. DNS – The Domain Name System

- DNS (Domain Name System) solves this problem.
- The essence of DNS is the invention of a hierarchical, domain-based naming scheme and a distributed database system for implementing this naming scheme.
- DNS is primarily used for mapping host names and e-mail destinations to IP addresses but can also be used for other purposes.

7.1. DNS – The Domain Name System

- DNS is used as follows.
- To map a name onto an IP address, an application program calls a library procedure called the resolver, passing it the name as a parameter.

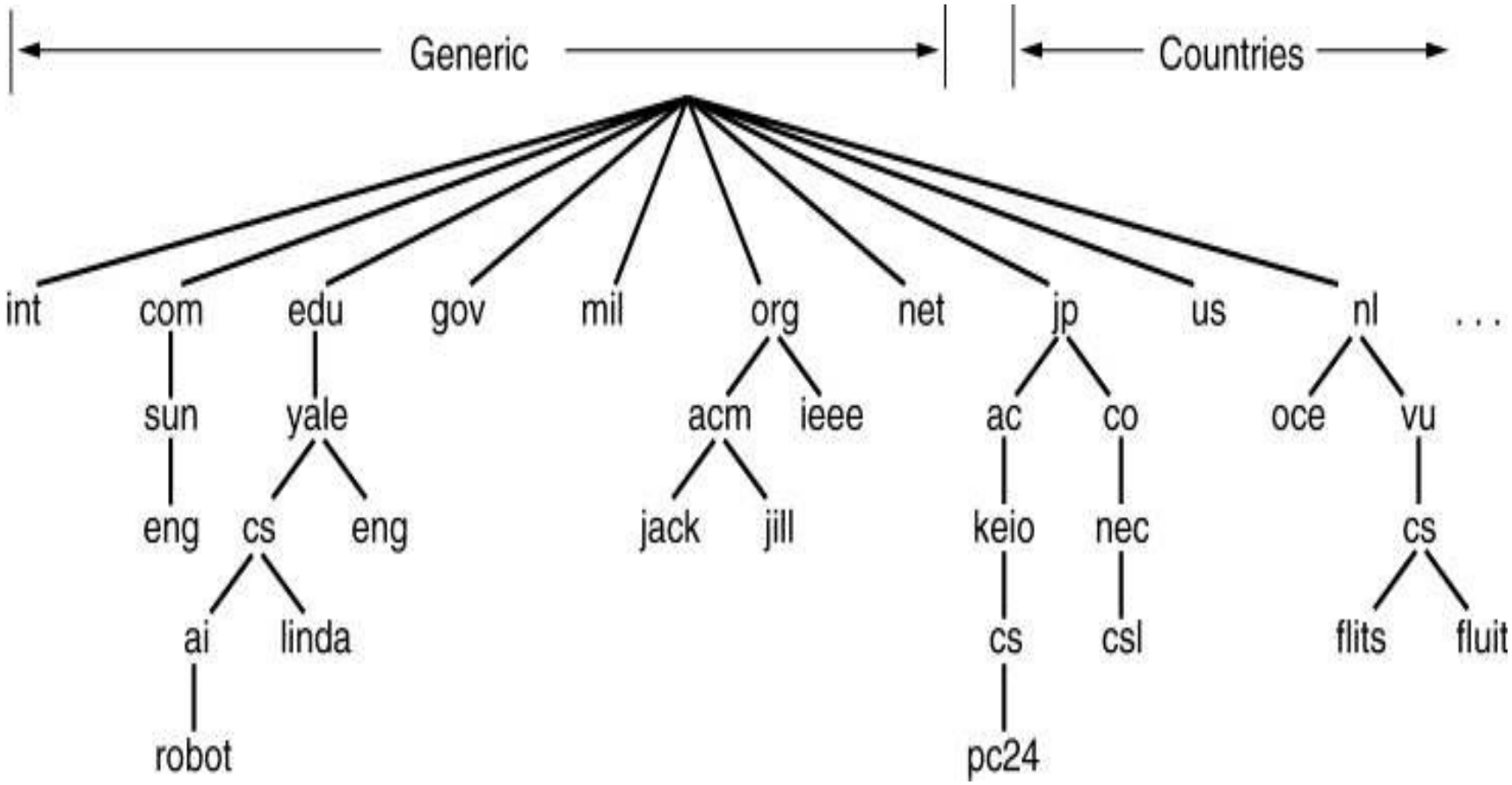
7.1. DNS – The Domain Name System

- Resolver sends a UDP packet to a local DNS server, which then looks up the name and returns the IP address to the resolver, which then returns it to the caller.
- Armed with the IP address, the program can then establish a TCP connection with the destination or send it UDP packets.

7.1.1. The DNS Name Space

- Internet is divided into over 200 top-level domains, where each domain covers many hosts.
- Each domain is partitioned into subdomains, and these are further partitioned, and so on.
- All these domains can be represented by a tree.

7.1.1. The DNS Name Space



A portion of the Internet domain name space.

7.1.1. The DNS Name Space

- The leaves of the tree represent domains that have no subdomains (but do contain machines, of course).
- A leaf domain may contain a single host, or it may represent a company and contain thousands of hosts.
- The top-level domains come in two flavors: generic and countries.

7.1.1. The DNS Name Space

- Top-level general-purpose domains:
- Commercial (com)
- Educational institutions (edu)
- Governments (gov)
- Certain international organizations (int)
- Armed forces (mil)
- Network providers (net)

7.1.1. The DNS Name Space

- Nonprofit organizations (org)

7.1.1. The DNS Name Space

- Business (biz) – 2000, ICANN
- Information (inf) – 2000, ICANN
- Peaple's names (name) – 2000, ICANN
- Professions (pro) – 2000, ICANN
- Aerospace industry (aero) - by request
- Co-operatives (coop) - by request

7.1.1. The DNS Name Space

- Museums (museum) - by request, etc

7.1.1. The DNS Name Space

- In general, getting a second-level domain, such as name-of-company.com, is easy.
- It requires going to a registrar for the corresponding top-level domain (com in this case) to check if the desired name is available and not somebody's trademark.
- If there are no problems, the register pays a small annual fee and gets the name.

7.1.1. The DNS Name Space

- Each domain is named by the path upward from it to the root.
- The components are separated by periods (pronounced “dot”).
- *eng.sun.com* is the engineering department at Sun Microsystems.
- *eng/sun/com* is UNIX-style

7.1.1. The DNS Name Space

- Domain names are case insensitive, so edu, Edu, and EDU mean the same thing.
- Component names can be up to 63 characters long, and full path names must not exceed 255 characters.
-

7.1.1. The DNS Name Space

- Domains can be inserted into the tree in two different ways.
- *cs.yale.edu*
- *cs.yale.ct.us*
- Most organization in the United States are under a generic domain, and most outside the United States are under the domain of

7.1.1. The DNS Name Space

their country.

7.1.1. The DNS Name Space

- Each domain controls how it allocates the domains under it.
- For example, Japon has domains ac.jp and co.jp that mirror edu and com
- The Netherlands does not make this distinction and puts all organization directly under nl

7.1.1. The DNS Name Space

- Thus, all three of the following are university computer science departments:
- cs.yale.edu (Yale University, in the US)
- cs.vu.nl (Vrije Universiteit, in the Netherl)
- cs.keio.ac.jp (Keio University, in Japan)
- Naming follows organizational boundaries, not physical networks.

7.1.2. Resource Records

- Every domain, whether it is a single host or a top-level domain, can have a set of resource records associated with it.
- For a single host, the most common resource record is just its IP address, but many other kinds of resource records also exist.

7.1.2. Resource Records

- When a resolver gives a domain name to DNS, what it gets back are the resource records associated with that name.
- Thus, the primary function of DNS is to map domain names onto resource records.

7.1.2. Resource Records

- A resource record is a five-tuple.
- Resource records are presented as ASCII text, one line per resource record.
- The format we will use is as follows:
- Domain_name Time_to_live Class Type Value

7.1.2. Resource Records

- Domain_name
- It tells the domain to which this record applies.
- Normally, many records exist for each domain and each copy of the database holds information about multiple domains.
- This field is thus the primary search key

7.1.2. Resource Records

used to satisfy queries.

7.1.2. Resource Records

- Time_to_live
- It gives an indication of how stable the record is.
- Information that is highly stable is assigned a large value, such as 86400 (the number of seconds in 1 day).
- Information that is highly volatile is assigned a small value, such as 60 (1

7.1.2. Resource Records

minute).

7.1.2. Resource Records

- Class
- For Internet information, it is always *IN*.
- Type
- It tells what kind of record this is
- The most important types are listed in following figure.

7.1.2. Resource Records

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

The principal DNS resource records types.

7.1.2. Resource Records

- Value
- This field can be a number, a domain name, or an ASCII string.
- The semantics depend on the record type.
- Following figure depicts part of a database for the cs.vu.nl domain shown on slide number 12.

7.1.2. Resource Records

```
; 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*.

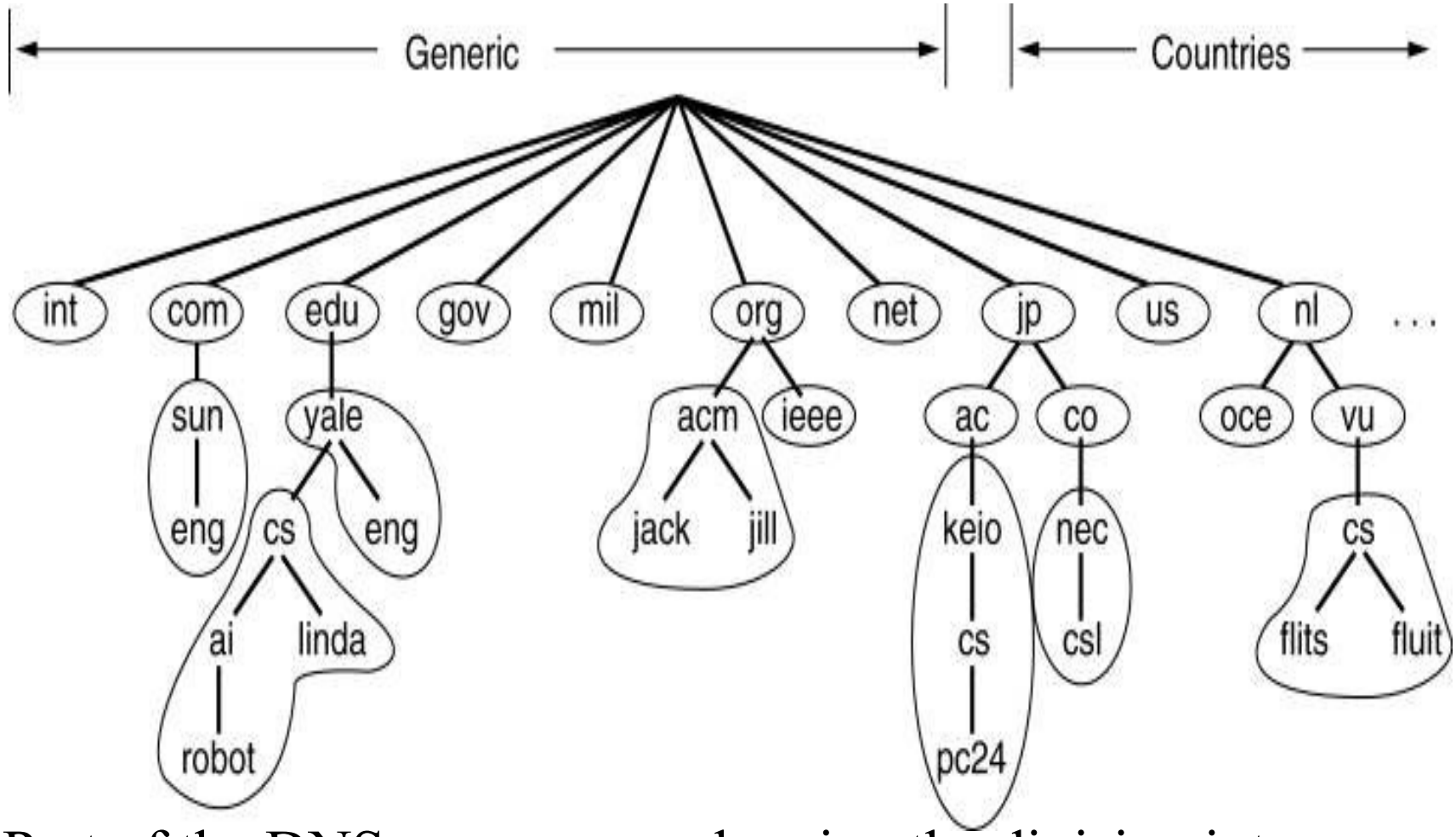
7.1.3. Name Servers

- In theoty at least, a single name server could contain the entire DNS database and respond to all queries about it.
- In practice, this server would be so overloaded as to be useless.
- Futhermore, if it ever went down, the entire Internet would be crippled.

7.1.3. Name Servers

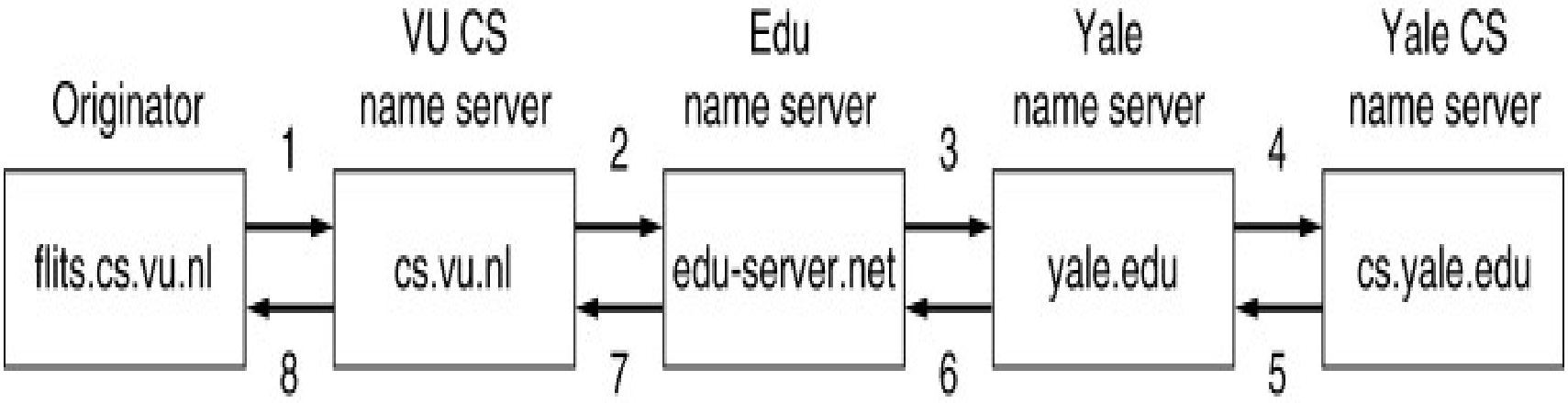
- To avoid the problems associated with having only a single source of information, the DNS name space is divided into nonoverlapping zones.
- One possible way to divide the name space on slide 12 is shown in following figure.

7.1.3. Name Servers



Part of the DNS name space showing the division into zones.

7.1.3. Name Servers



How a resolver looks up a remote name in eight steps.

7.2. Electronic Mail

- Electronic mail, or e-mail, like other forms of communication, has its own conventions and style.
- e-mail is very informal and has a low threshold of use.
- e-mail is full of jargon
- Many people also use litle ASCII symbols called smileys or emoticons in their e-mail…

7.2. Electronic Mail

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

Some smileys. They will not be on the final exam :-).

7.2. Electronic Mail

- The first e-mail systems simply consisted of file transfer protocols, with the convention that the first line of each message (i.e., file) contained the recipient's address.

7.2. Electronic Mail

- Some of the complaints were as follows:

a) Sending a message to a group of people was inconvenient.

b) Messages had no internal structure, making computer processing difficult.

c) The sender never knew if a message arrived or not.

d) It was not possible to create and send messages containing a mixture of text, drawings, facsimile, and voice, etc.

7.2. Electronic Mail

- 1982 – RFC 821 (transmission protocol) and RFC 822 (message format)
- 1984 – X.400
- After two decades of competition, e-mail systems based on RFC 822 are widely used.

7.2. Electronic Mail

- Architecture and Services
- The User Agent
- Message Formats
- Message Transfer
- Final Delivery

7.2.1. Architecture and Services

- What e-mail systems can do and how they are organized?
- There are two subsystems in e-mail systems

a) User agents allow people to read and send e-mail.

b) Message transfer agents move messages from the source to the destination.

7.2.1. Architecture and Services

e-mail systems support five basic functions

- Composition
- Transfer
- Reporting
- Displaying
- Disposition

7.2.1. Architecture and Services

A few of advanced features:

- Mailboxes
- Mailing list
- Carbon copies
- Blind carbon copies
- Etc.

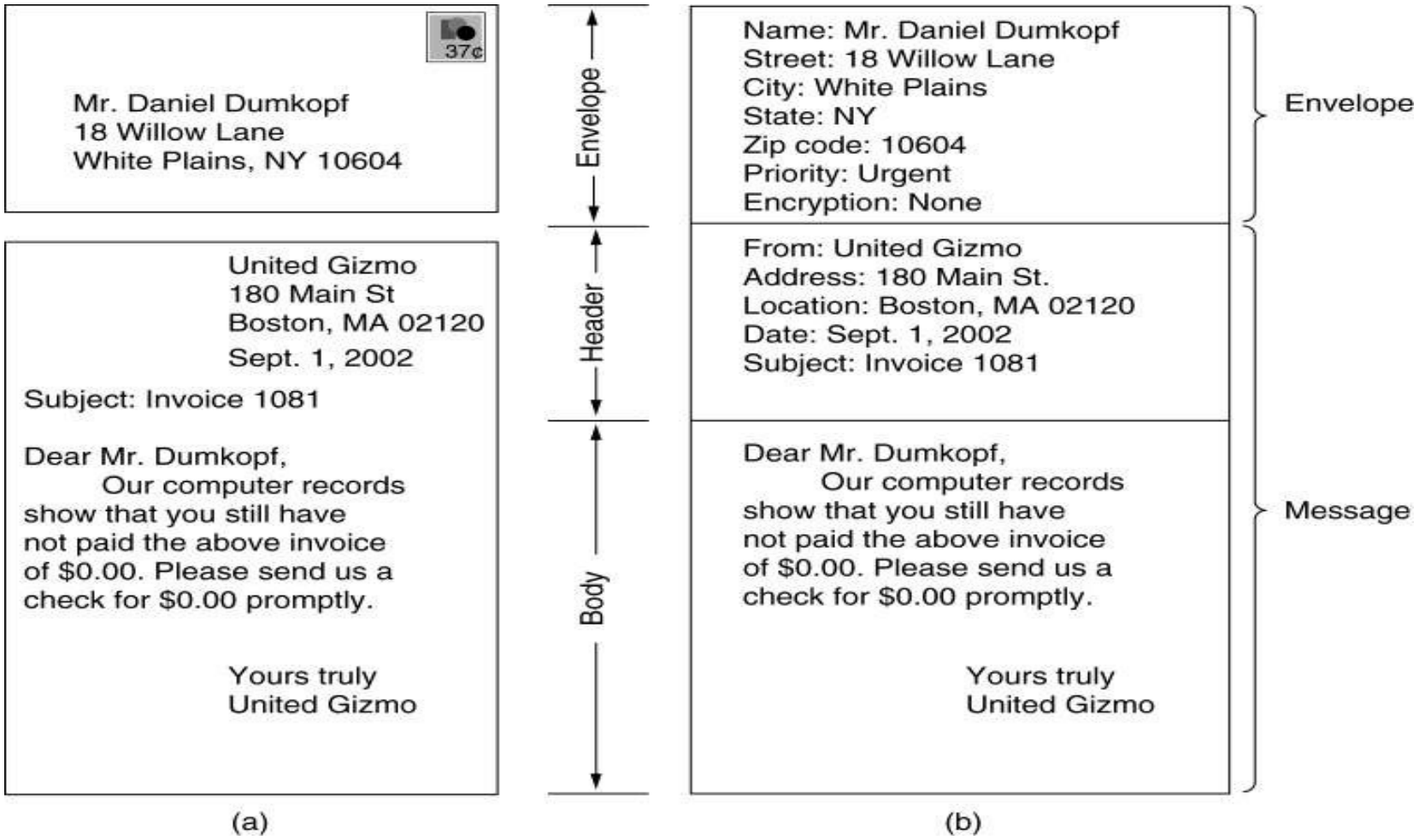
7.2.1. Architecture and Services

- A key idea in e-mail systems is the distinction between the envelope and its contents.
- The envelope encapsulates the message
- It contains all the information needed for transporting the message, such as the destination address, priority, and security level, all of which are distinct from the message itself.

7.2.1. Architecture and Services

- Message inside the envelope consists of two parts:
- The header and the body
- The header contains control information for the user agents.
- The body is entirely for the human recipient.

7.2.1. Architecture and Services



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

7.2.2. The User Agent

- A user agent is normally a program (sometimes called a mail reader) that accepts a variety of commands for composing, receiving, and replying to message, as well as for manipulating mailboxes.

7.2.2. The User Agent

- SENDING E-MAIL
- To send an e-mail message, a user must provide the message, the destination address, and possiply some other parameters,
- The message can be produced with a free-standing text editor, a word processing program, etc
- Most e-mail systems support mailing lists, so that a user can send same message to a list of people with a single command.

7.2.2. The User Agent

- READING E-MAIL
- Typically, when a user agent is started up, it looks at the user's mailbox for incoming e-mail before displaying anything on the screen.
- Then it may announce the number of messages in the mailbox or display a one-line summary of each one and wait for a command.

7.2.2. Reading E-mail

#	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

An example display of the contents of a mailbox.

7.2.3 Message Formats – RFC 822

- Messages consist of :
- a primitive envelope,
- some number of header fields,
- a blank line,
- and then message body

7.2.3 Message Formats – RFC 822

- Each header field consists of a single line of ASCII text containing the field name, a colon, and, for most fields, a value.

- In normal usage, the user agent builds a message and passes it to the message transfer agent, which then uses some of the header fields to construct the actual envelope.

7.2.3 Message Formats – RFC 822

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

RFC 822 header fields related to message transport.

Message Formats – RFC 822 (2)

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

Some fields used in the RFC 822 message header.

MIME – Multipurpose Internet Mail Extensions

- In the early days of the ARPANET, e-mail consisted exclusivly of text messages written in English and expressed in ASCII.
- For this environment, RFC 822 did thejob completely: it specified the headersbut left the content entirely up to the users.

MIME – Multipurpose Internet Mail Extensions

Problems with international languages:

- 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 – Multipurpose Internet Mail Extensions

- A solution was proposed in RFC 1341 andupdated in RFCs 2045-2049.
- This solution, called MIME (MultipurposeInternet Mail Extensions) is now widely used.
- The basic idea of MIME is to continue touse the RFC 822 format, but to add srtucture to the message body and define encoding rules for non-ASCII messages.

MIME (65)

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

RFC 822 headers added by MIME.

MIME ( )

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

The MIME types and subtypes defined in RFC 2045.

MIME (4)

```
From: elinor@abcd.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

This is the preamble. The user agent ignores it. Have a nice day.

--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/enriched

Happy birthday to you
Happy birthday to you
Happy birthday dear <bold> Carolyn </bold>
Happy birthday to you

--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
        access-type="anon-ftp";
        site="bicycle.abcd.com";
        directory="pub";
        name="birthday.snd"

content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--
```

7.2.4. Message Transfer

- The message transfer system is concernedwith relaying messages from the originatorto the recipient.

- The simplest way to do this is to establisha transport connection from the source machine to the destination machine and then just transfer the message.

7.2.4. Message Transfer

- SMTP – The Simple Mail TransferProtocol
- Within the Internet, e-mail is delivered byhaving the source machine establish a TCPconnection to port 25 of the destination machine.
- Listening to this port is an e-mail daemonthat speaks SMTP.

7.2.4. Message Transfer

- This daemon accepts incoming connections and copies messages from them into the appropriate mailboxes.

- If a message cannot be delivered, an errorreport containing the first part of the undeliverable message is returned to the sender.

7.2.4. Message Transfer

- SMTP is a simple ASCII protocol.
- After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first.

7.2.4. Message Transfer

- The server starts by sending a line of textgiving its identity and telling whether it isprepared to receive mail.

- If it is not, the client releases theconnection and tries again later.

7.2.4. Message Transfer

- If the server is willing to accept e-mail, the client announces whom the e-mail is coming from and whom it is going to.

- If such a recipient exists at the destination, the server gives the client the go-ahead to send the message.

7.2.4. Message Transfer

7.2.4. Message Transfer

- Then the client sends the message and theserver acknowledges it.
- No checksums are needed because TCPprovides a reliable byte stream.
- If there is more e-mail, that is now sent.
- When all the e-mail has been exchangedin both directions, the connection is

7.2.4. Message Transfer

released.

Message Transfer

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

```
                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
```

7.2.5. Final Delivery

- What happens when Elinor wants to sendCarolyn e-mail and Carolyn is not currently on-line?
- Elinor cannot establish a TCP connectionto Carolyn and thus cannot run the SMTP protocol.

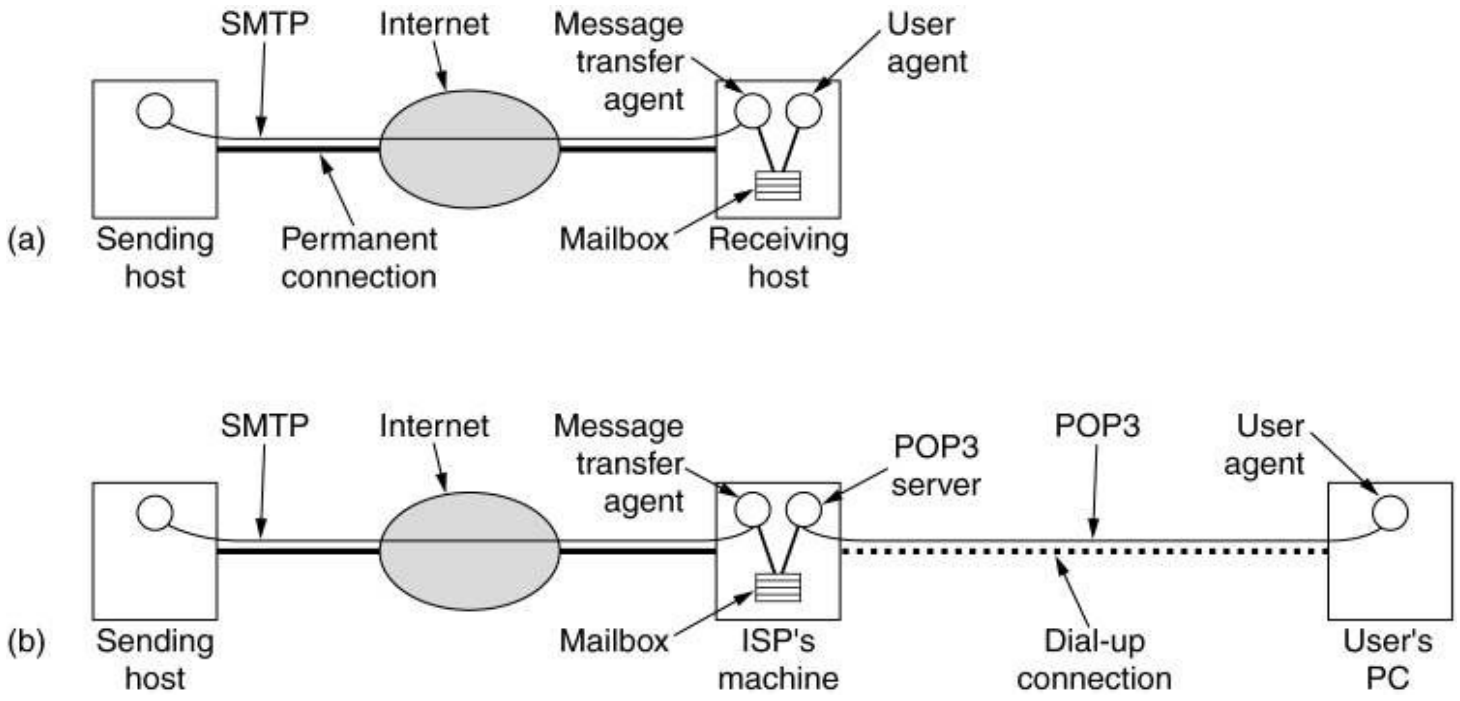
7.2.5. Final Delivery

- One solution is to have a message transferagent on an ISP machine accept e-mail for its customers and store it in their mailboxes on an ISP machine.
- Since this agent can be on-line all the time, e-mail can be sent to it 24 hours aday.

7.2.5. Final Delivery - POP3

- How does the user get the e-mail from theISP's message transfer agent?
- The solution to this problem is to create another protocol that allows user transfer agents (on client PCs) to contact the message transfer agent (on ISP's machine) and allow e-mail to be copied from the ISP to the user.

7.2.5. 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.

7.2.5. Final Delivery - POP3

```
                    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
```

7.2.5. Final Delivery - IMAP

- IMAP – Internet Message Access Protocol
- IMAP assumes that all the e-mail will remainon the server indefinitely in multiple mailboxes.
- But POP3 basically assumes that the user willclear out the mailbox on every contact and work off-line after that.
- IMAP – provides extensive mechanisms forreading message or even parts of

7.2.5. Final Delivery - IMAP

messages

UNIT V

The Application Layer

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

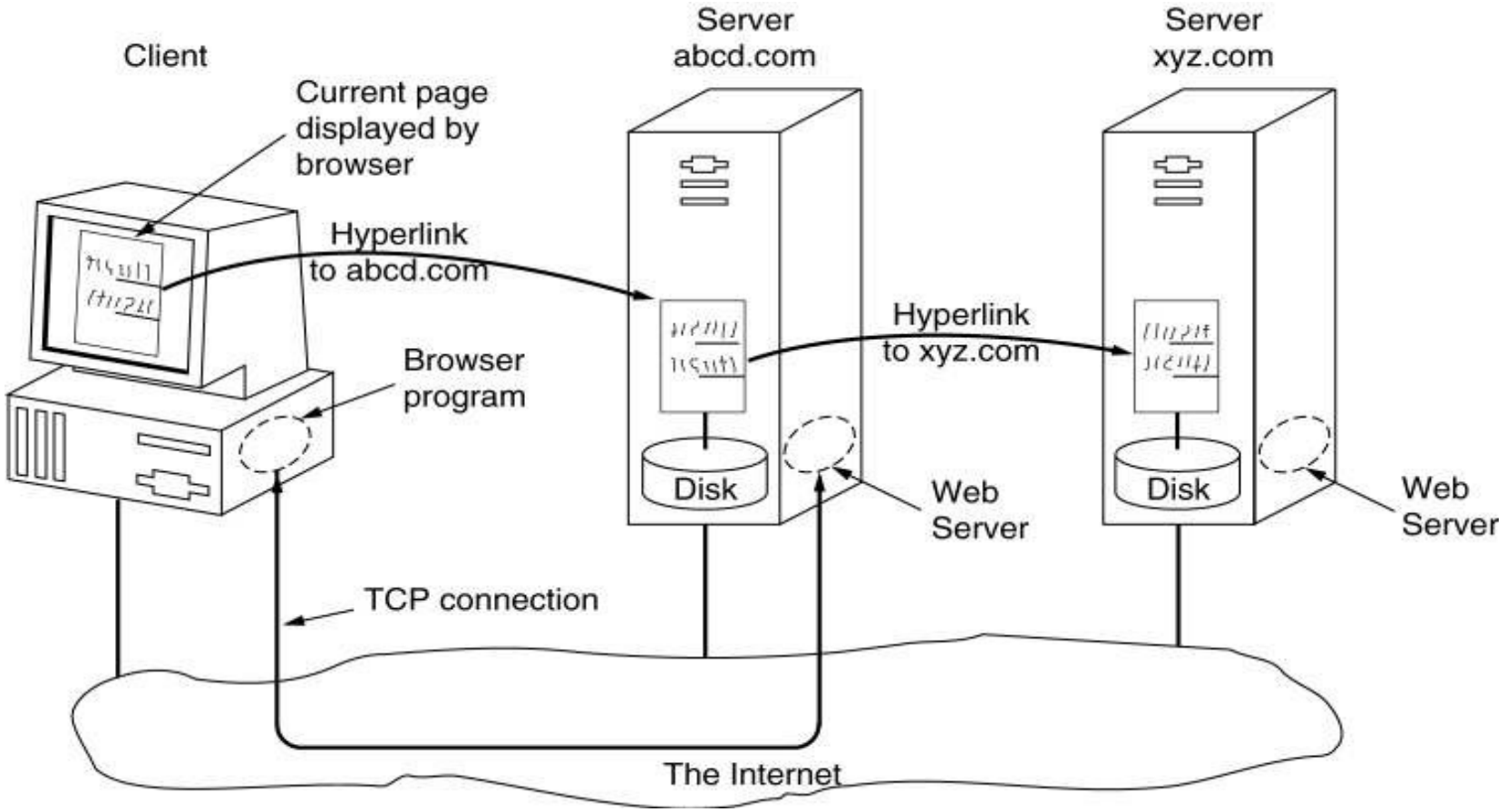
Webmaster@animalpsyc.eastpodunk.edu

(b)

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

7.3. 1. Architectural Overview
The Client Side
(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

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 allWeb 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

7.3. 1. Architectural Overview
The Client Side

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

- There are two possibilities : plug-ins andhelper applications.

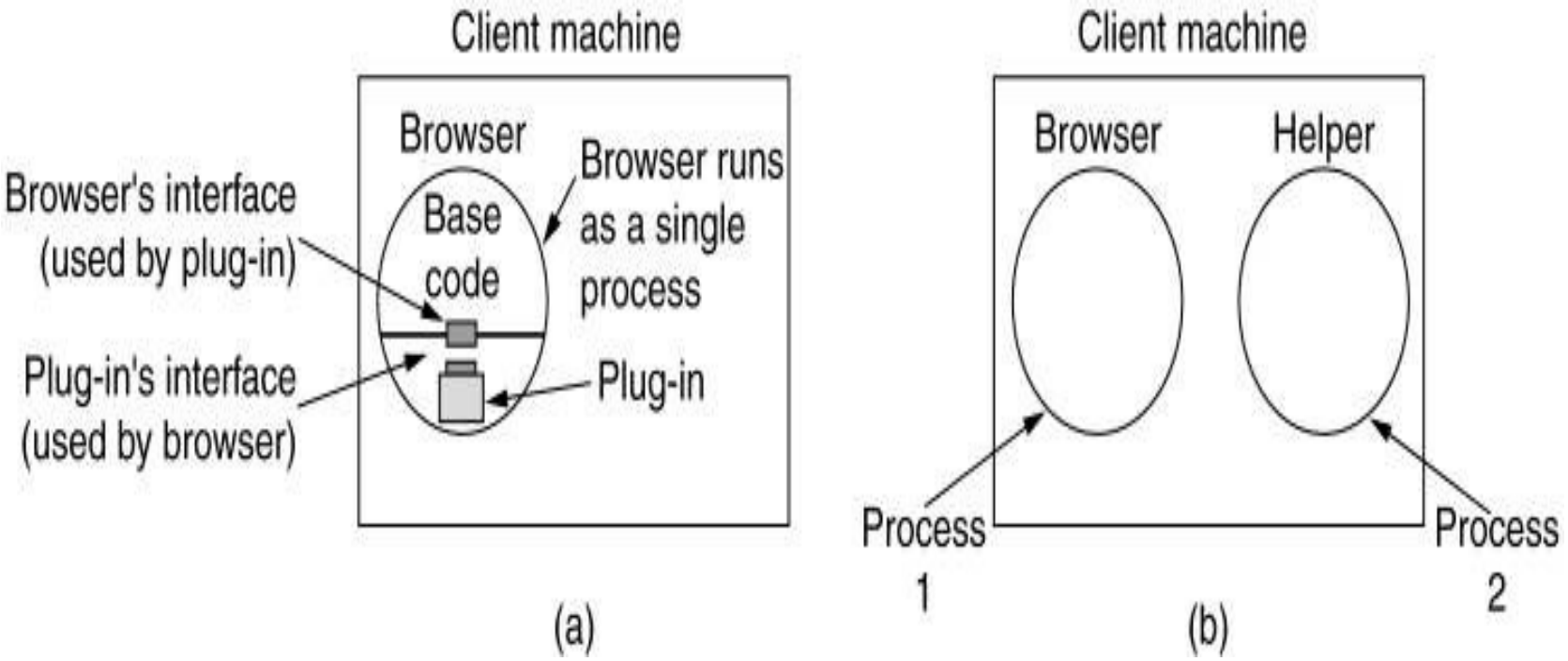
7.3. 1. Architectural Overview
The Client Side

- 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 beinstalled.

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

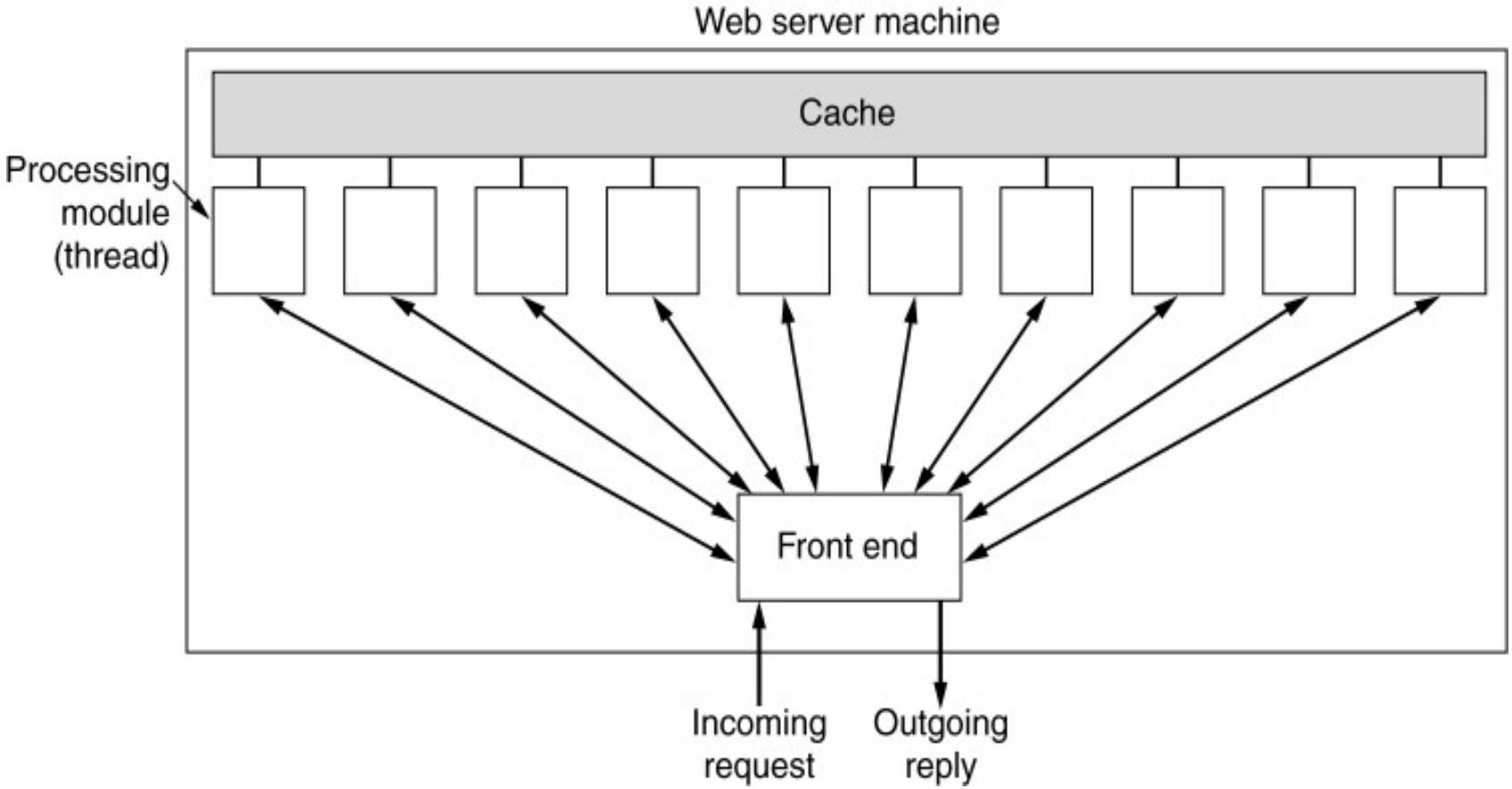
- A problem with this design is that every request requires making a disk access to get the file.

- 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

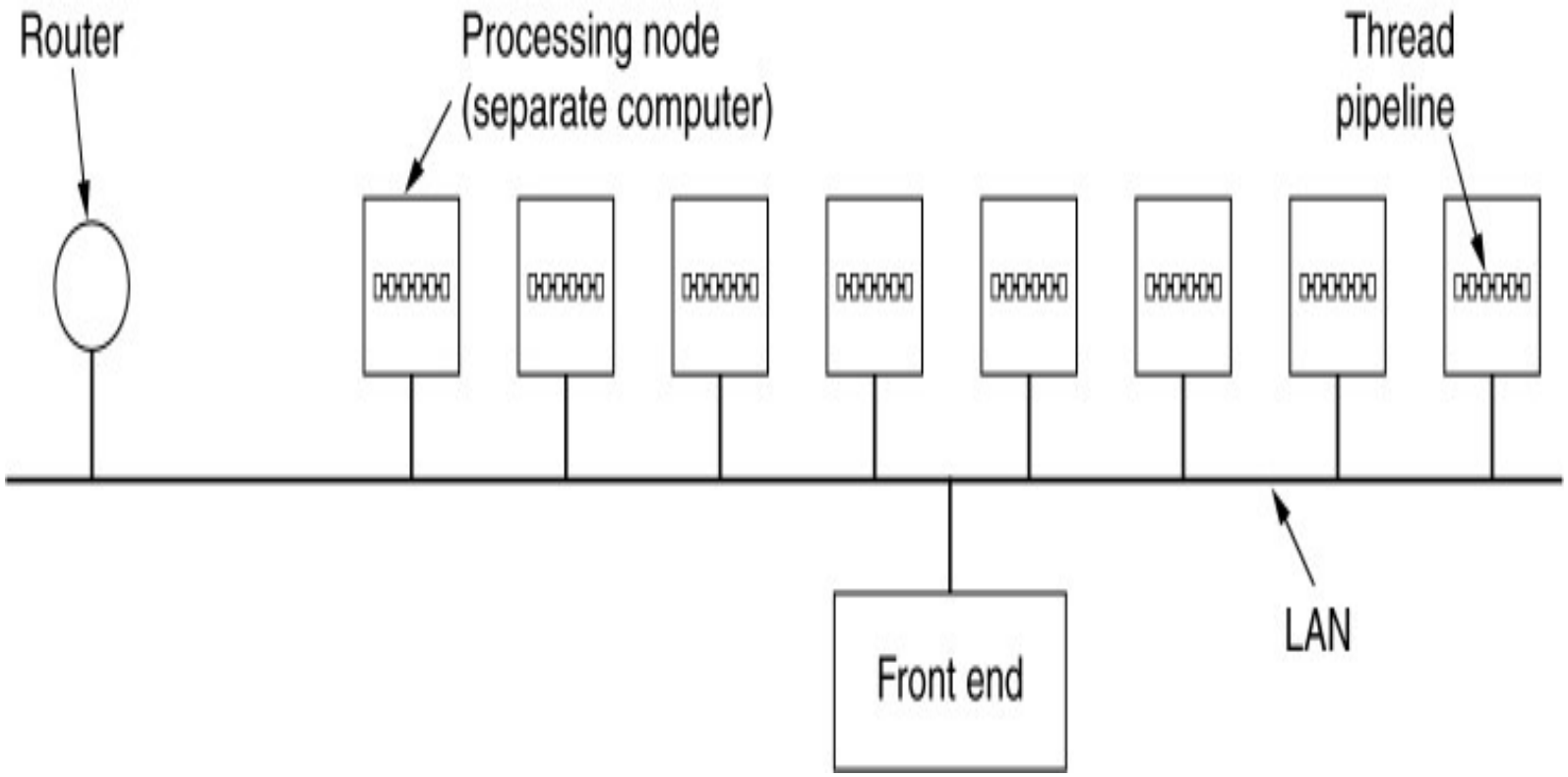
7.3. 1. Architectural Overview
The Server Side

- 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

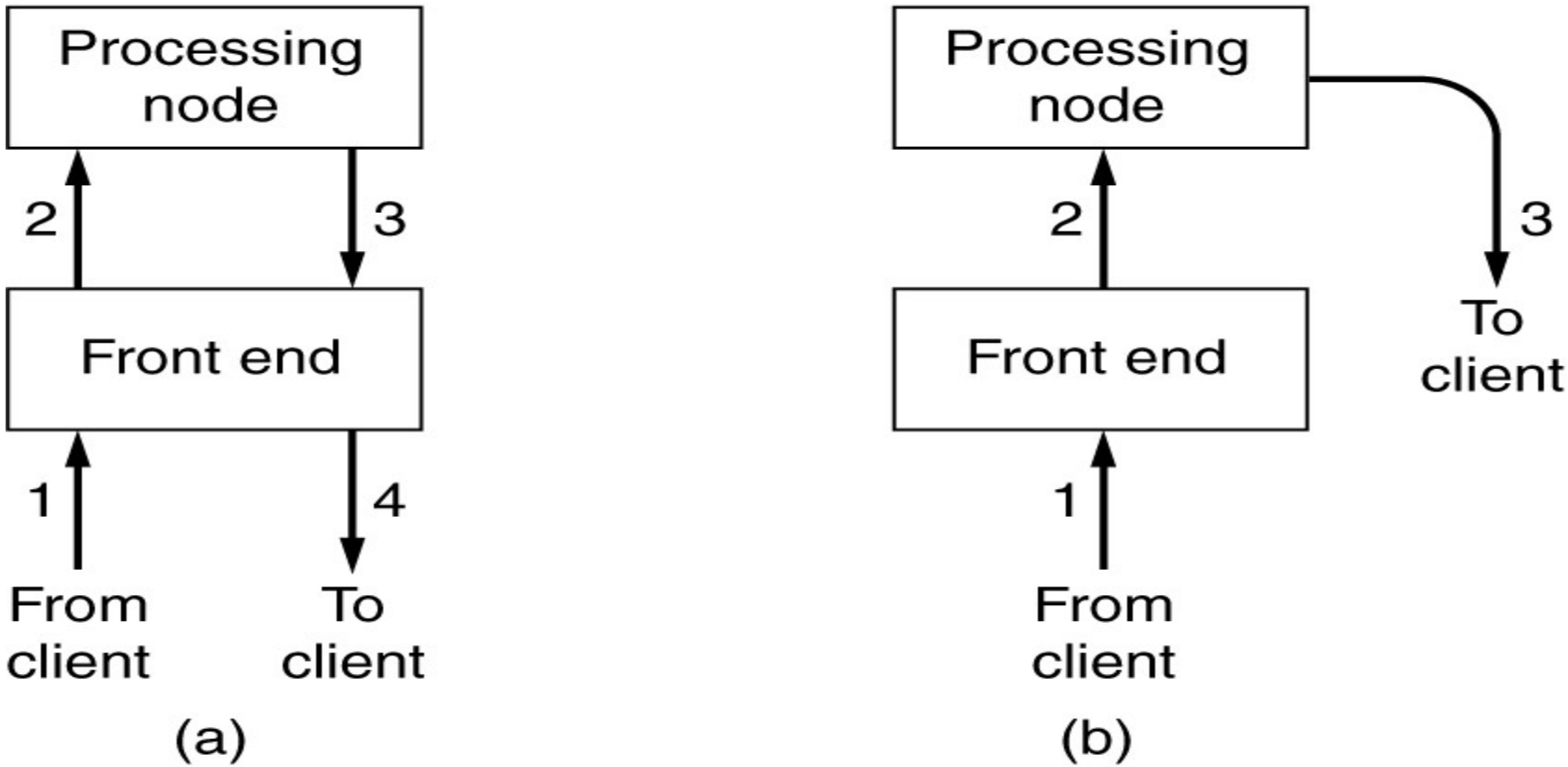


A server farm.

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



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

7.3. 1. Architectural Overview
URLs – Uniform Resource Locaters

locating pages.

- 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.

- 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

7.3. 1. Architectural Overview
Statelessness and Cookies

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

Forms

```
<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)

(a) An HTML table.

(b) A possible rendition of this table.

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)

- 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.

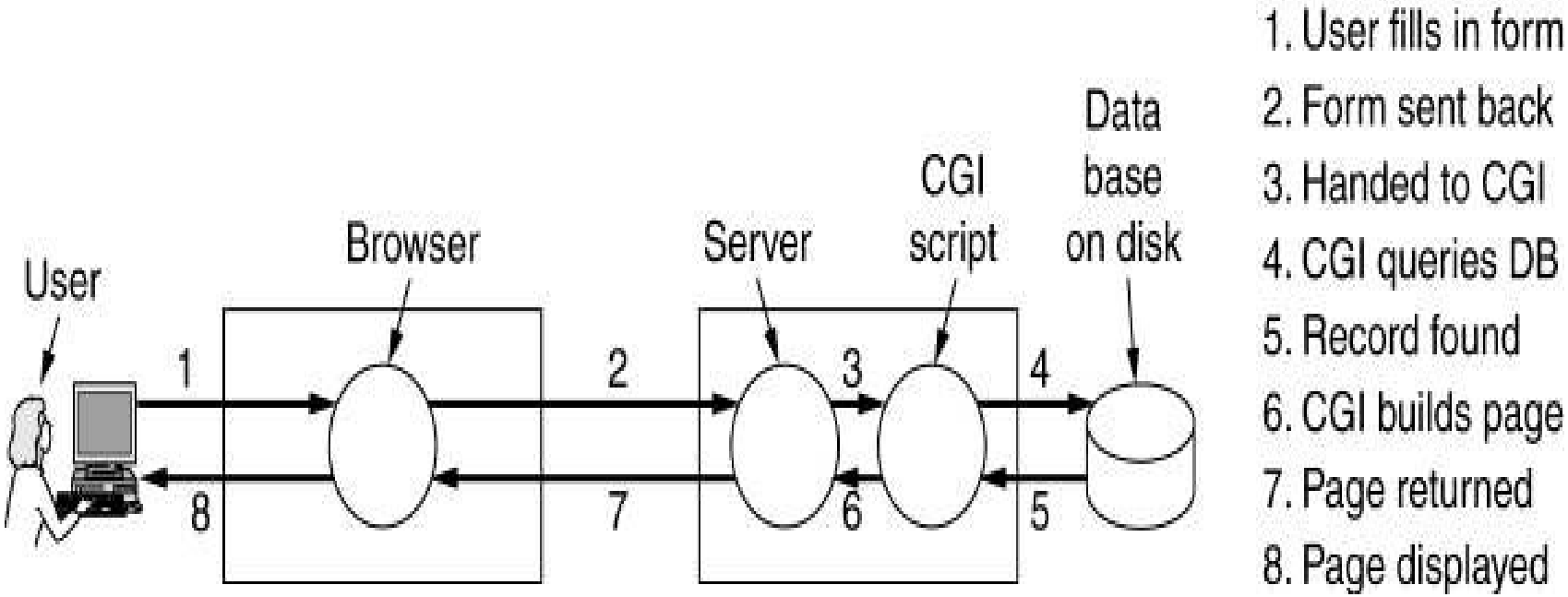
```
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:template match="/">

<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)

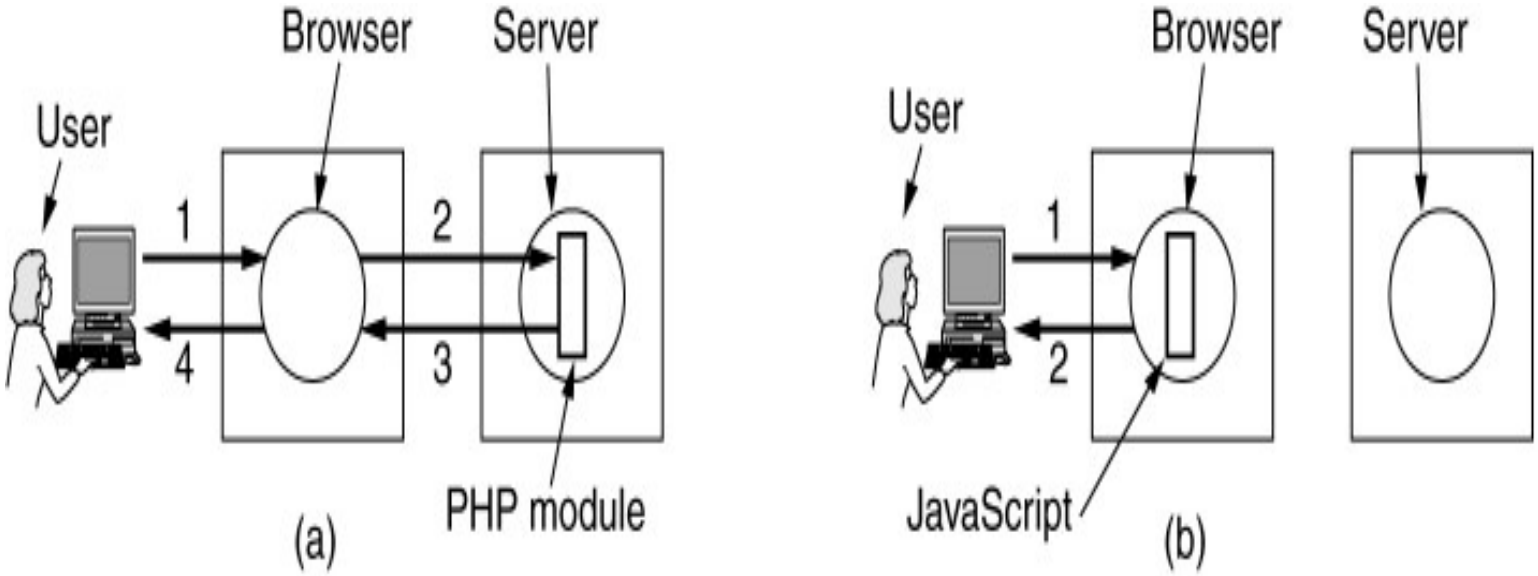
(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.

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 (



(a) Server-side scripting with PHP.

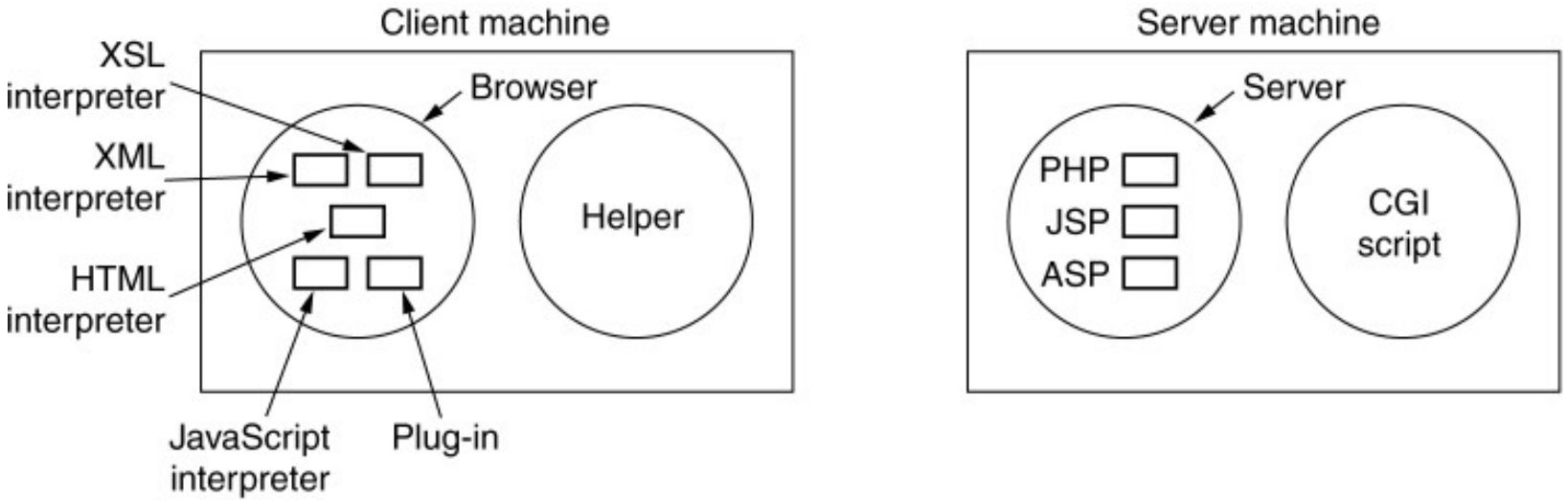
(b) Client-side scripting with JavaScript.

Client-Side Dynamic Web Page Generation (

```
<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>
```

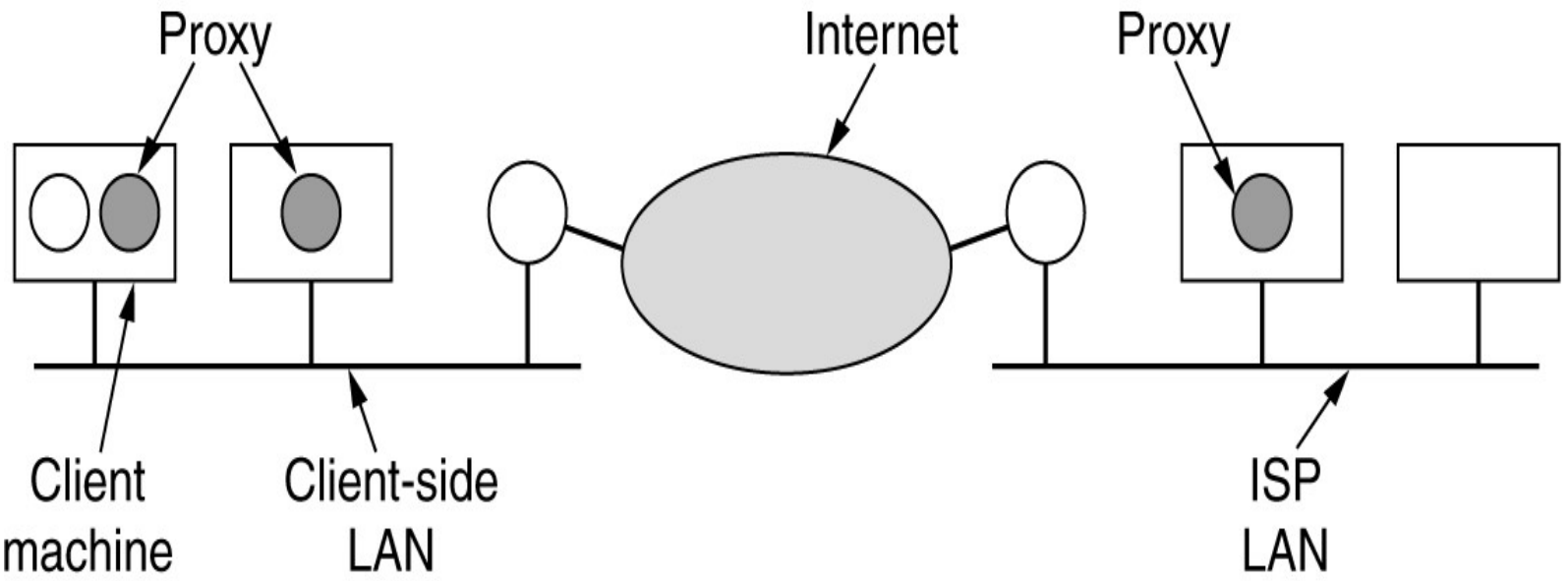
A JavaScript program for computing and printing factorials

Client-Side Dynamic Web Page Generation (



The various ways to generate and display content.

7.3.4. 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>
```

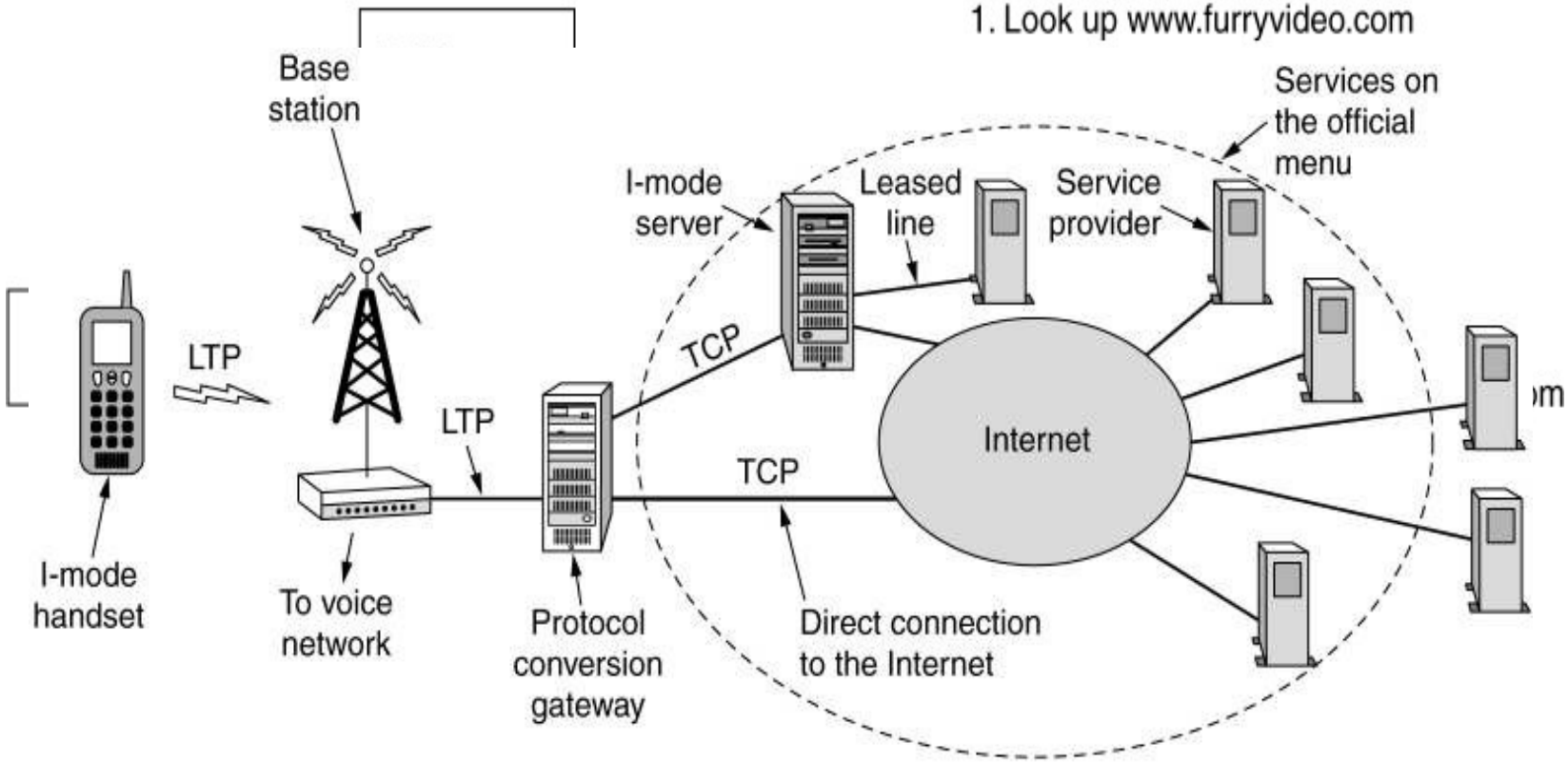
(a)

```
<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

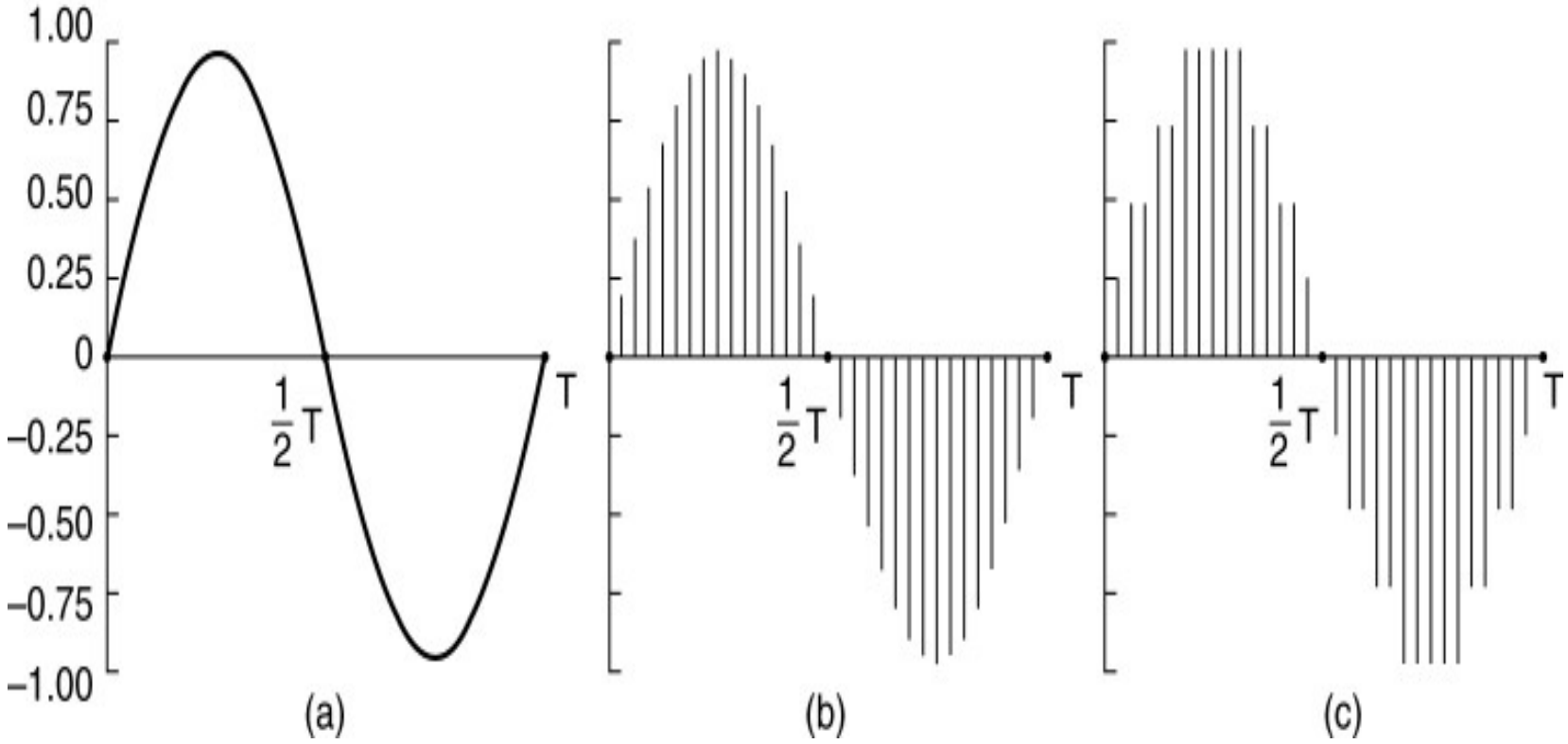


Steps in looking up a URL when a CDN is used

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

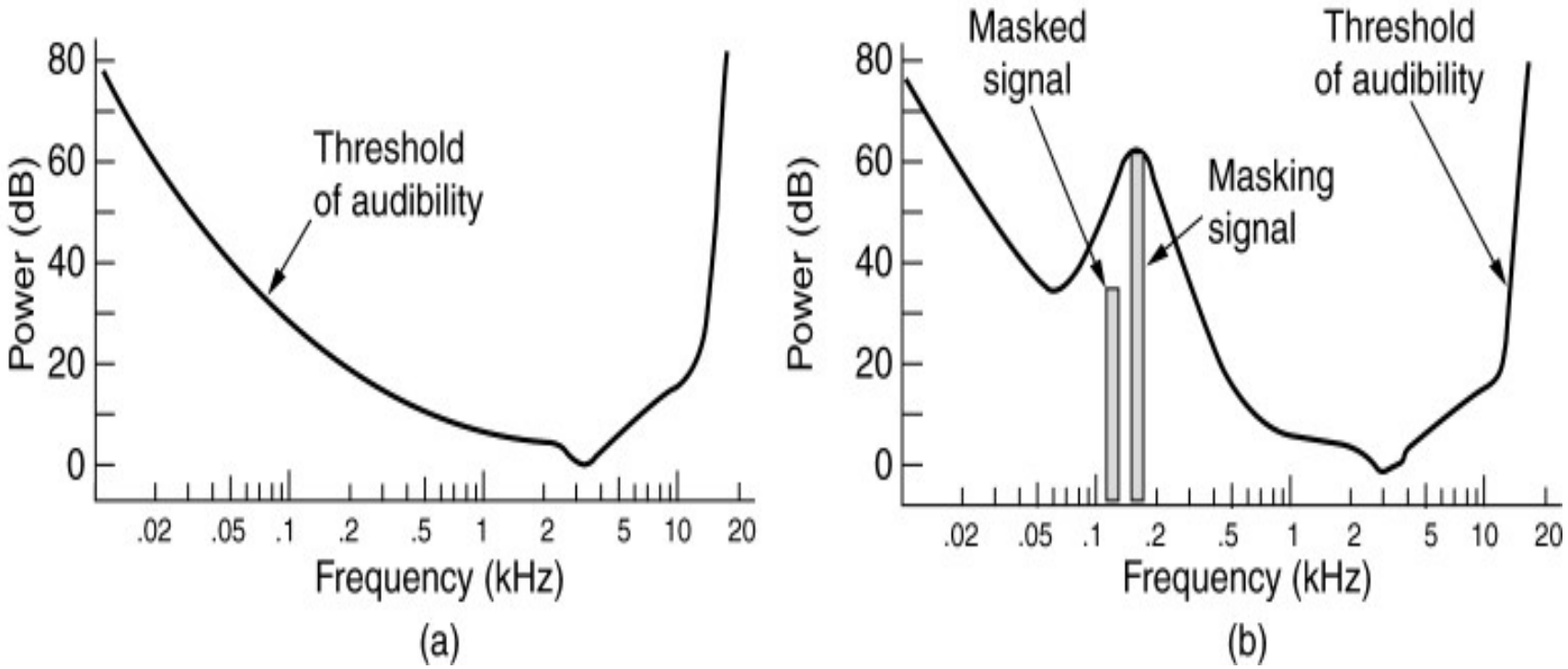
- Audio waves can be converted to digital form by an ADC (Analog Digital Convertor)
- ADC takes an electrical voltage as input and generates a binary number as output.
- Telephone, audio compact discs are examples where sound is sampled.
- Music, speech are special cases of general

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



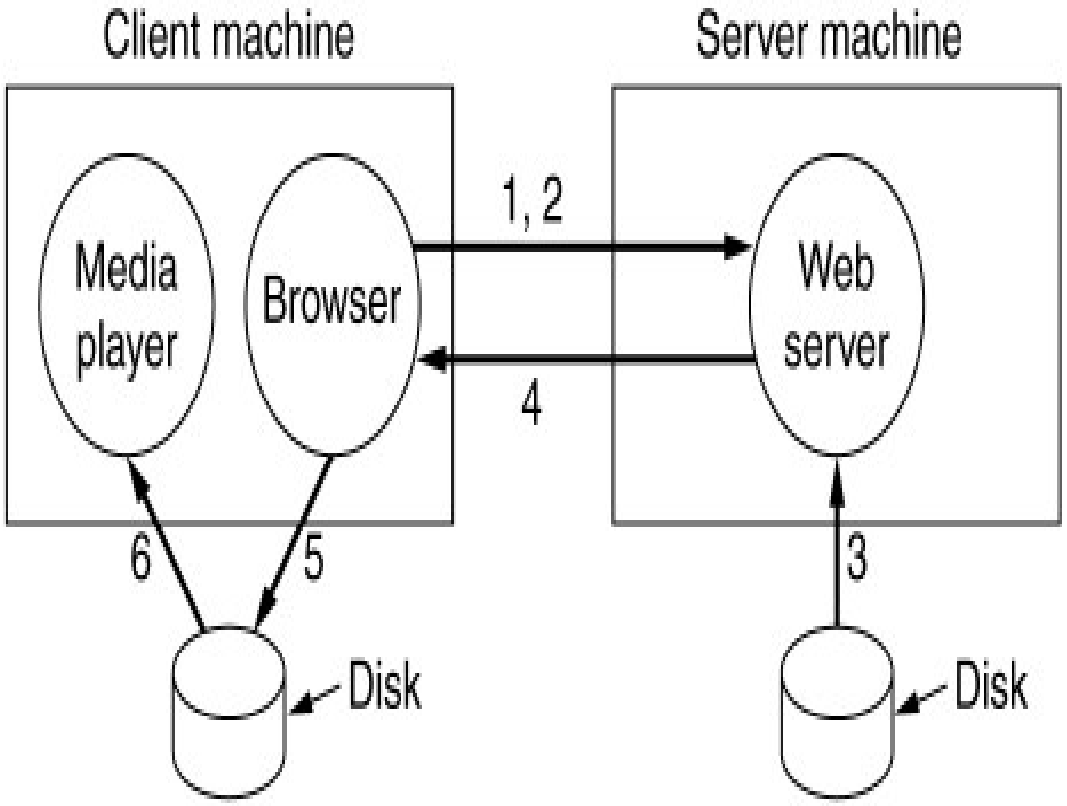
(a) The threshold of audibility as a function of frequency.

(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:

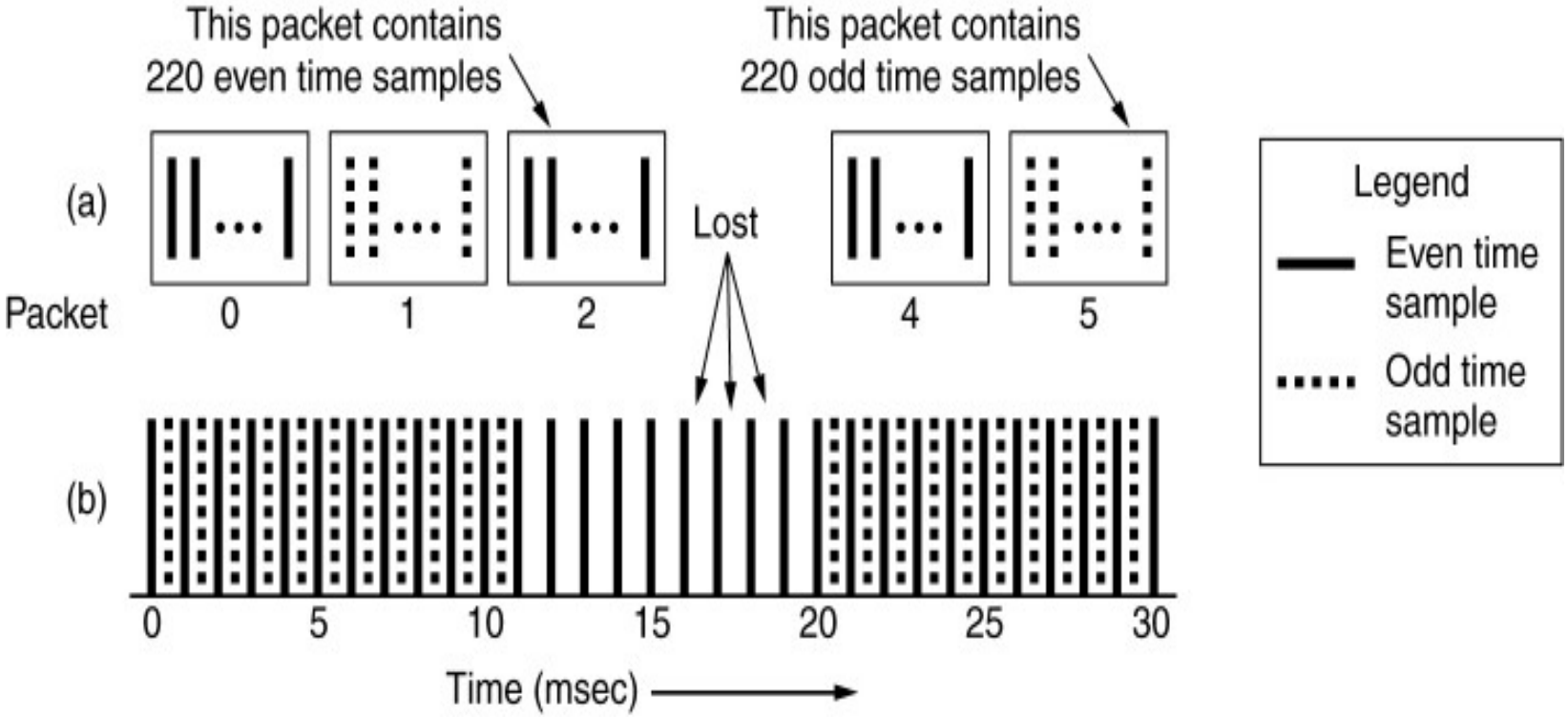
a) Manage the user interface

b) Handle transmission errors

c) Decompress the music

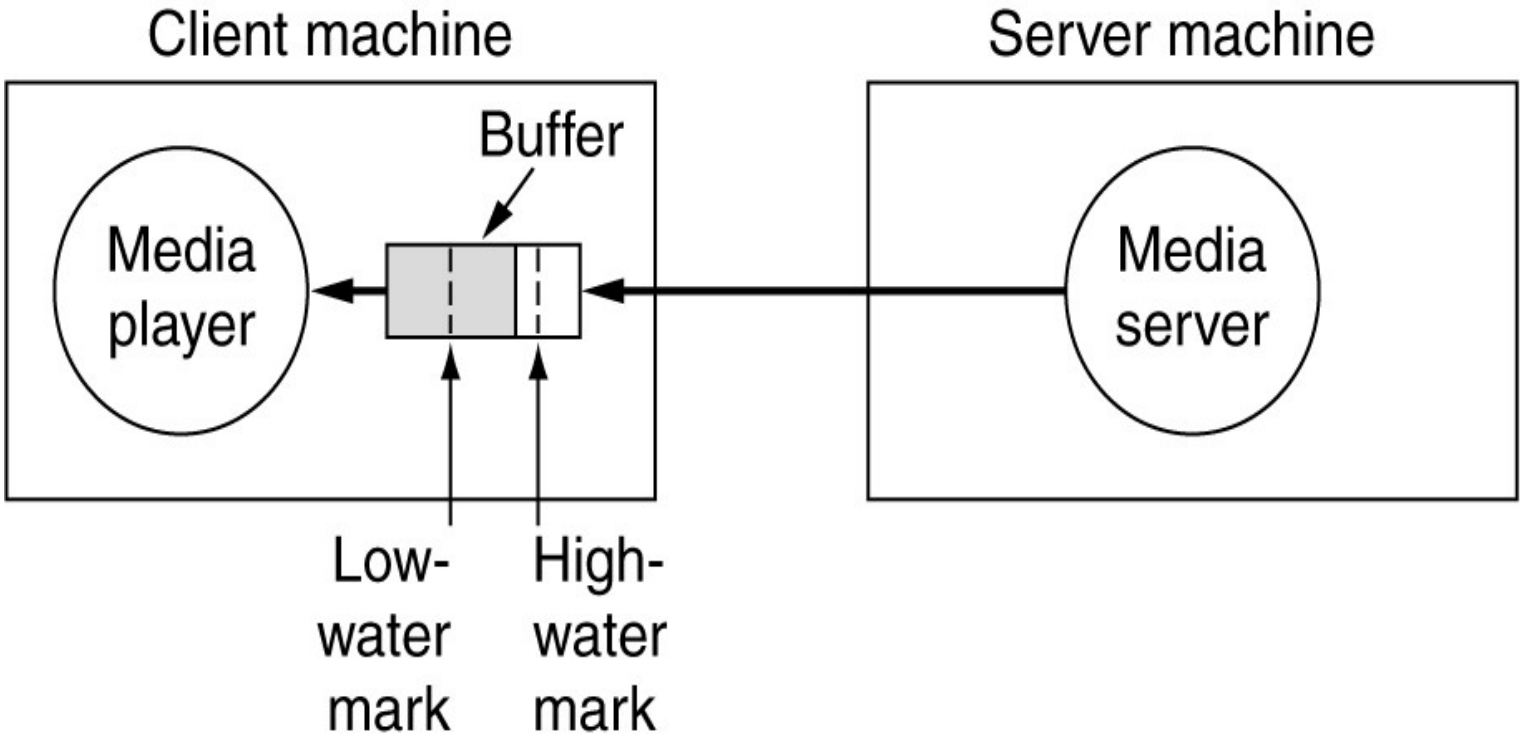
d) Eliminate jitter

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

7.4.5. Introduction to Video

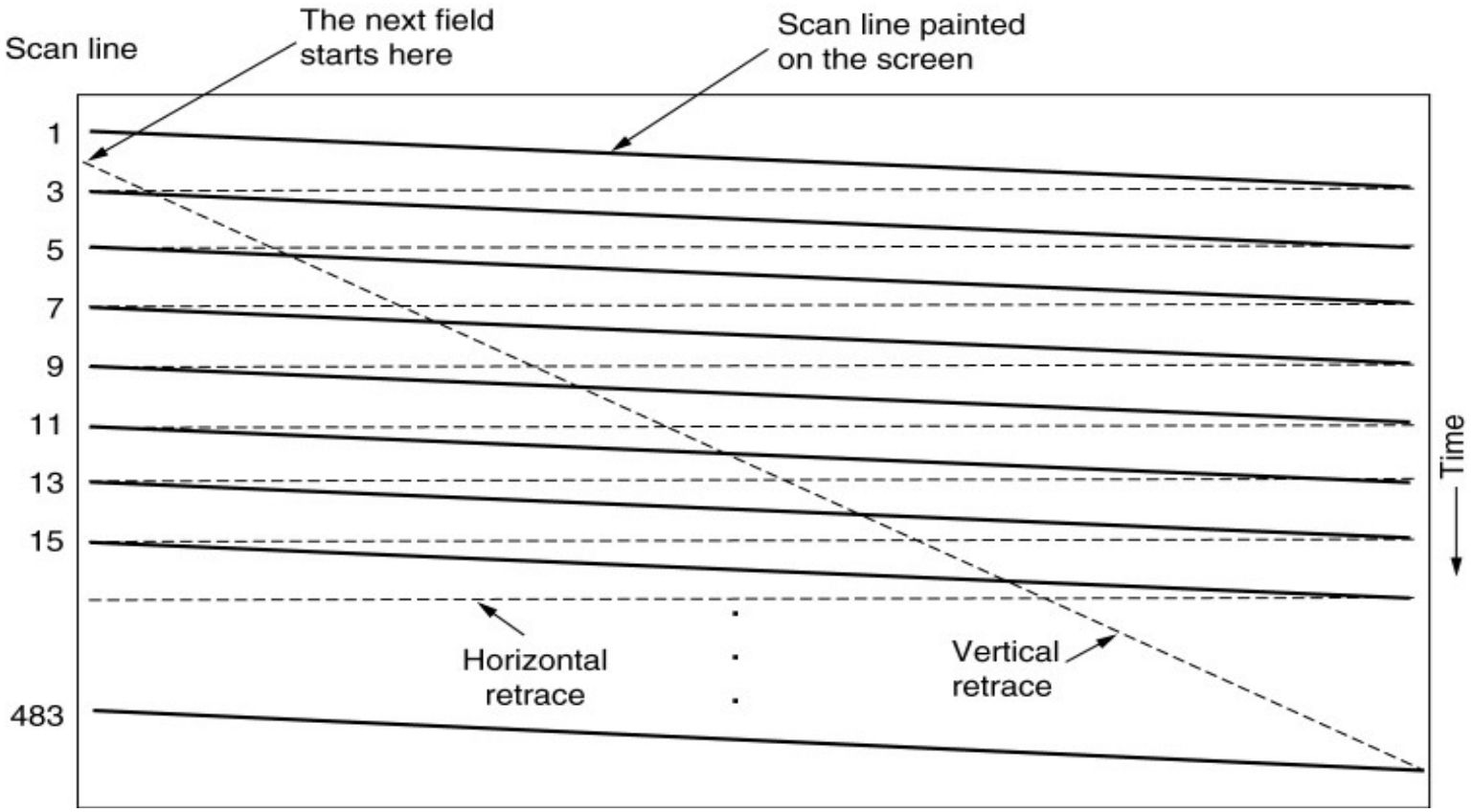
The human eye has the property that whenan image appears on the retina, the image is retained for some number of milliseconds before decaying.

- 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.

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.