CSC8004

The Application Layer

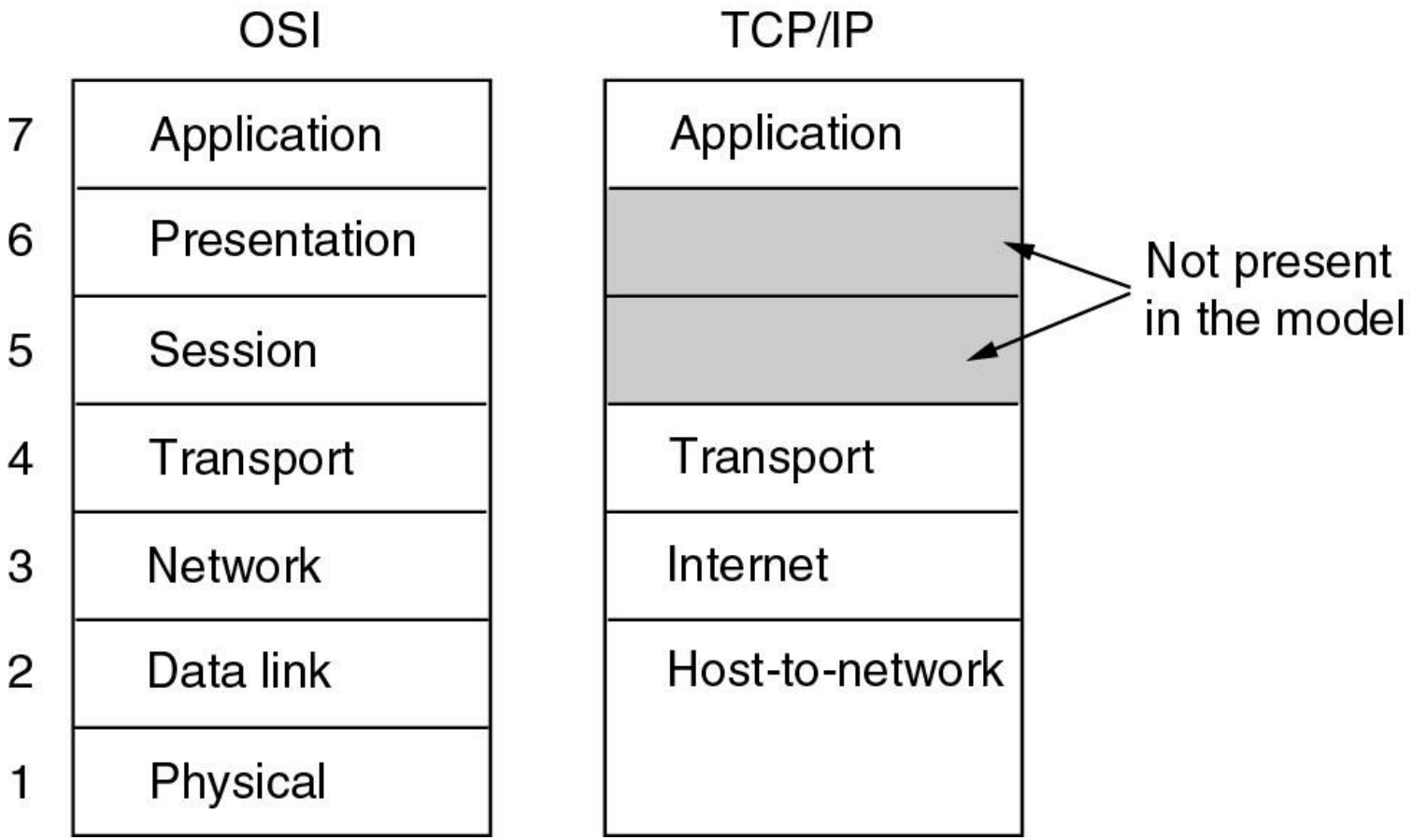
Ellis Solaiman

- Overview
- Network Services
- User Applications

Next

- Overview
- Network Services
- User Applications

Overview


OSI
TCP/IP
7
Application
6
Presentation
5
Session
4
Transport
3
Network
2
Data link
1
Physical
Application
Transport
Internet
Host-to-network
Not present
in the model

Overview (2)

The Application layer

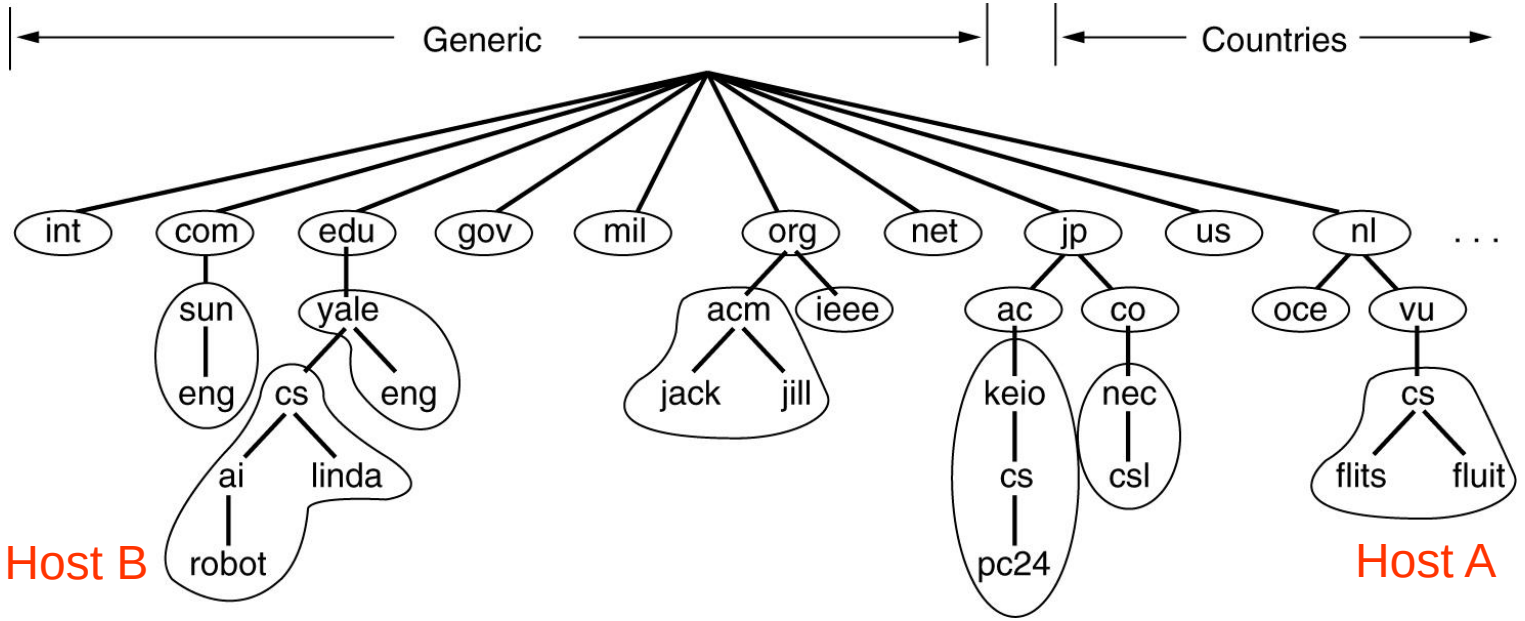
- is the interface between the network and its users
- deals with presentation/session layer issues
- contains network services (eg DNS)
- contains user applications (eg email, web browsing)

Next

- Overview
- Network Services
  - Domain Name Service - DNS
- User Applications

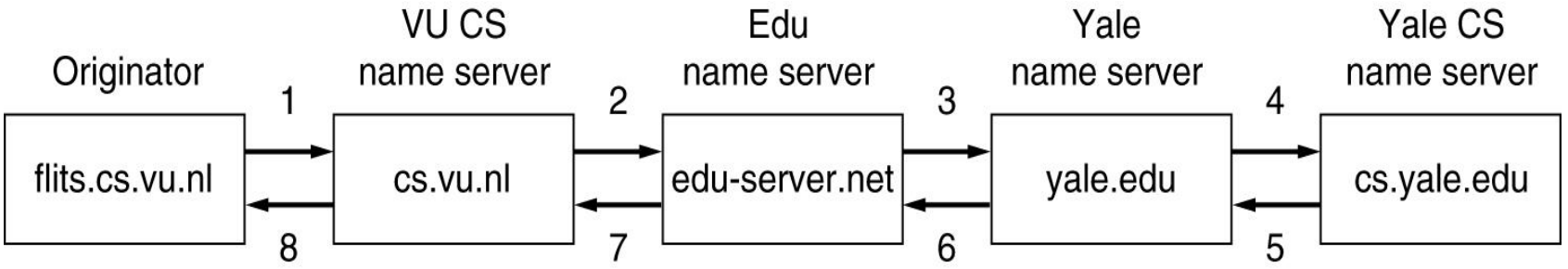
DNS name space

- DNS name space is hierarchical
- Name space is divided into zones
- At least one DNS server exists in each zone
- A DNS server knows about all hosts in its zone (its children)
- A DNS server knows about top level DNS servers



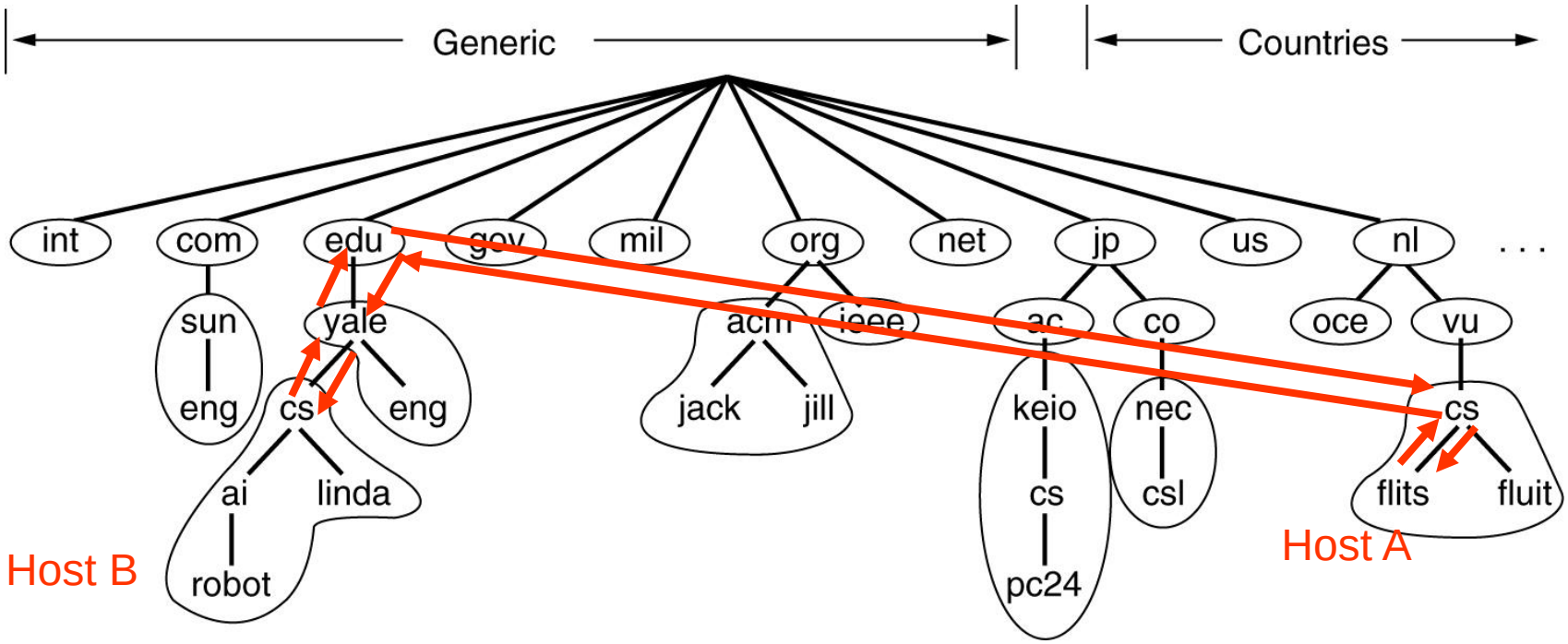
DNS name resolution

- DNS servers communicate to convert a symbolic Internet address into an IP address
- eg Host A (`flits.cs.vu.nl`) wants IP address of Host B (`robot.ai.cs.yale.edu`)



- Originating host contacts its local DNS server
- DNS servers communicate to find target
- DNS servers communicate to return IP address

DNS name resolution (2)



Host A (flits.cs.vu.nl) wants IP address of Host B (robot.ai.cs.yale.edu)

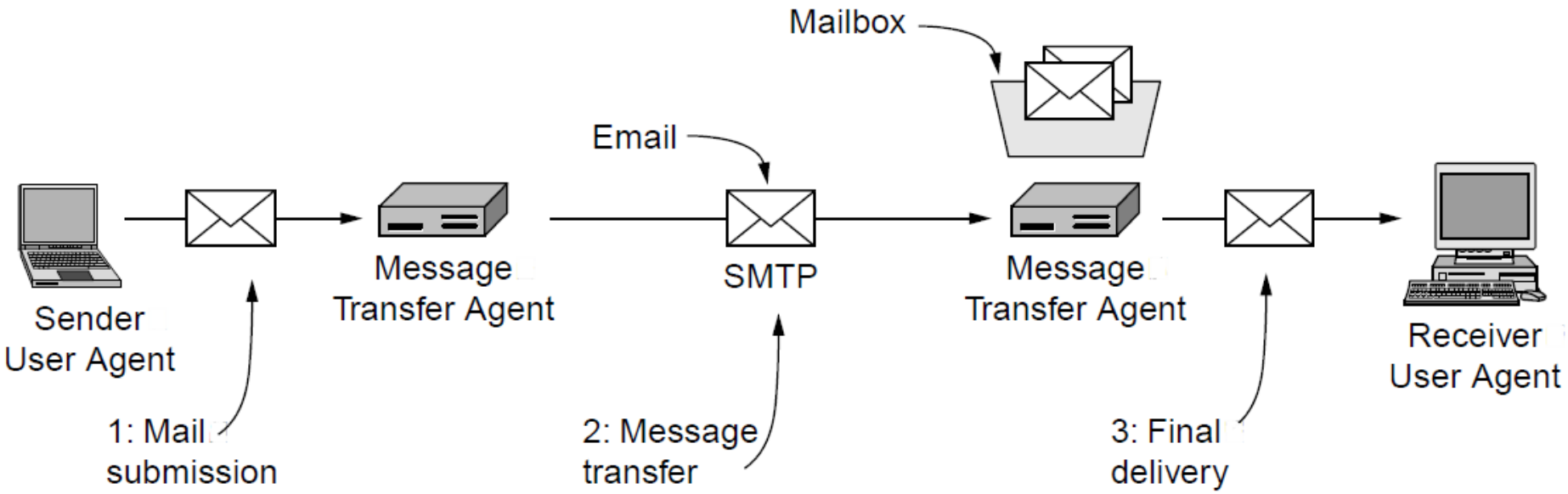
Next

- Overview
- Network Services
- User Applications
  - Email
  - World Wide Web

Email

- Mail is transferred from source host to ISP mailbox by a message transfer agent (eg MS Exchange) using SMTP
- Mail is transferred from ISP to destination host by a user agent (eg MS Outlook) using POP3

  (Alternatively, user can read mail online at ISP using browser)

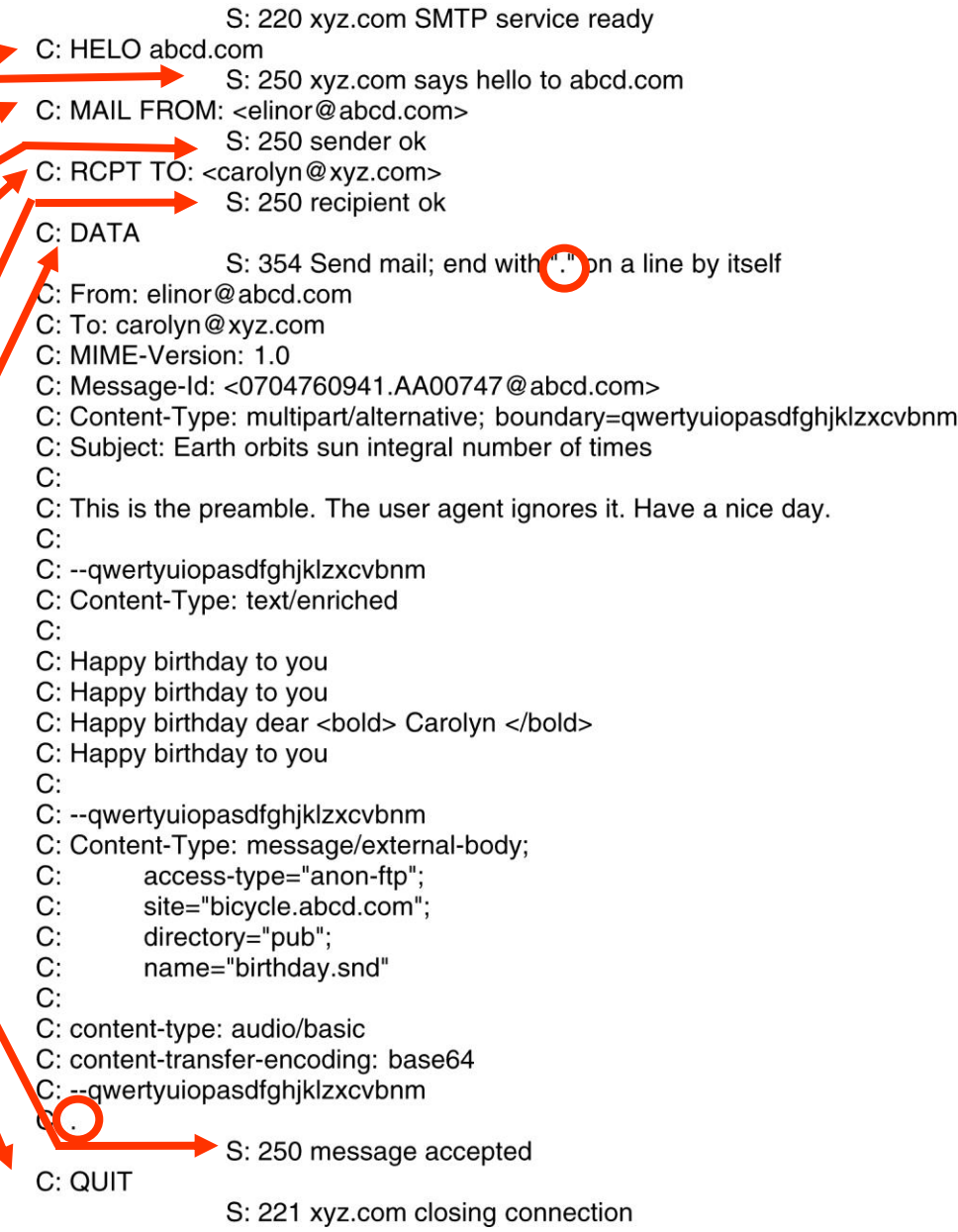


SMTP (a "push" protocol) (RFC 821)

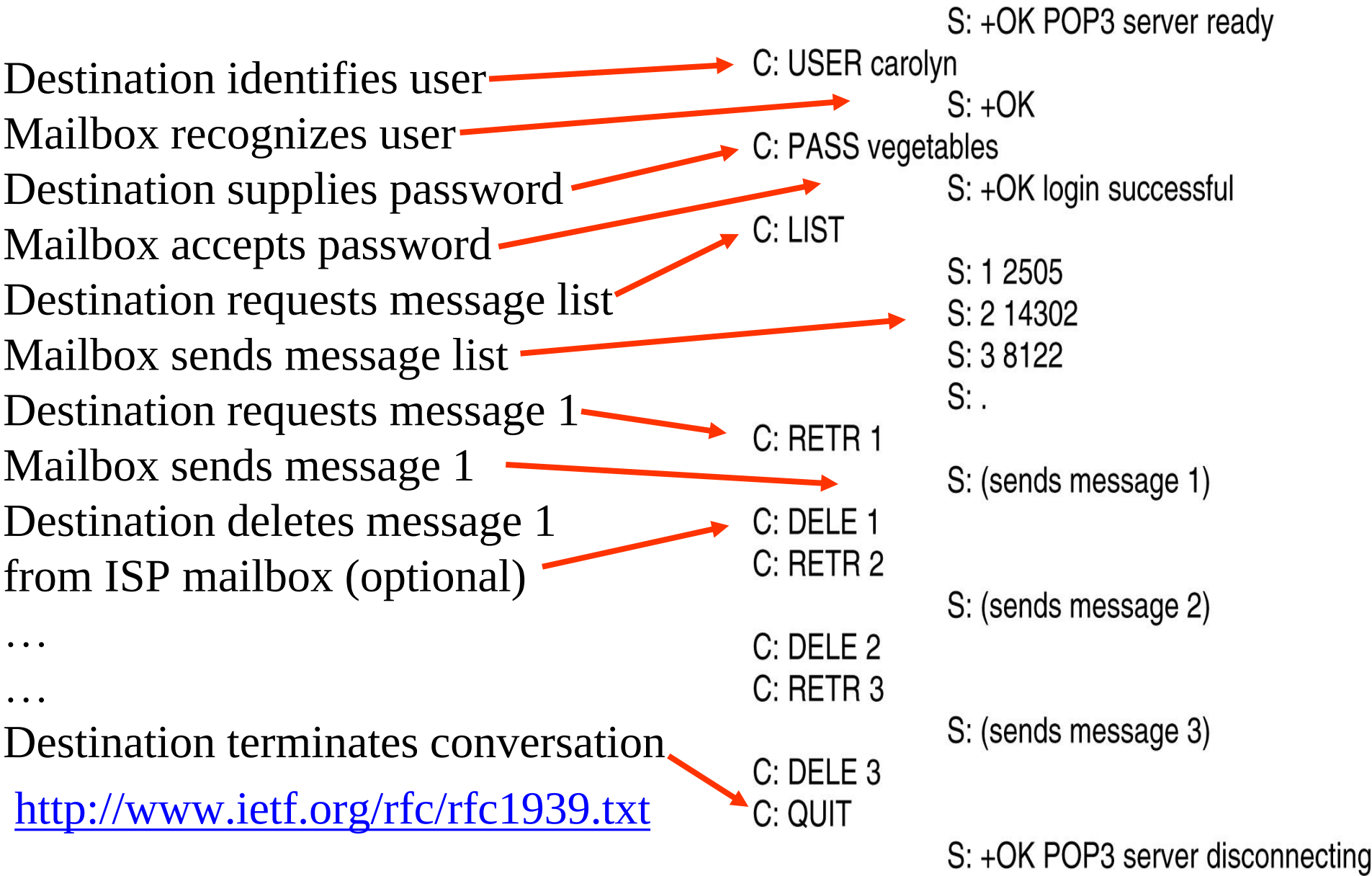
Source contacts destination
Mailbox says it is ready
Source identifies itself
Mailbox accepts
Source identifies recipient
Mailbox recognizes recipient
Negotiate any parameters
Source sends email
Mailbox accepts email
Source terminates conversation

eg transferring a message from
elinor@abcd.com
to *carolyn@xyz.com*

http://www.ietf.org/rfc/rfc0821.txt



POP3 (a “pull” protocol) (RFC 1939)



http://www.ietf.org/rfc/rfc1939.txt

Message format (text) (RFC 822)

- http://www.ietf.org/rfc/rfc0822.txt

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

MIME extensions

- MIME - Multipurpose Internet Mail Extensions
- Necessary to handle multiple languages
- Necessary to handle multimedia

MIME message types (RFC 2045)

http://www.ietf.org/rfc/rfc1939.txt

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

MIME example

Alternative 1

Text with formatting

(Will be displayed by client)

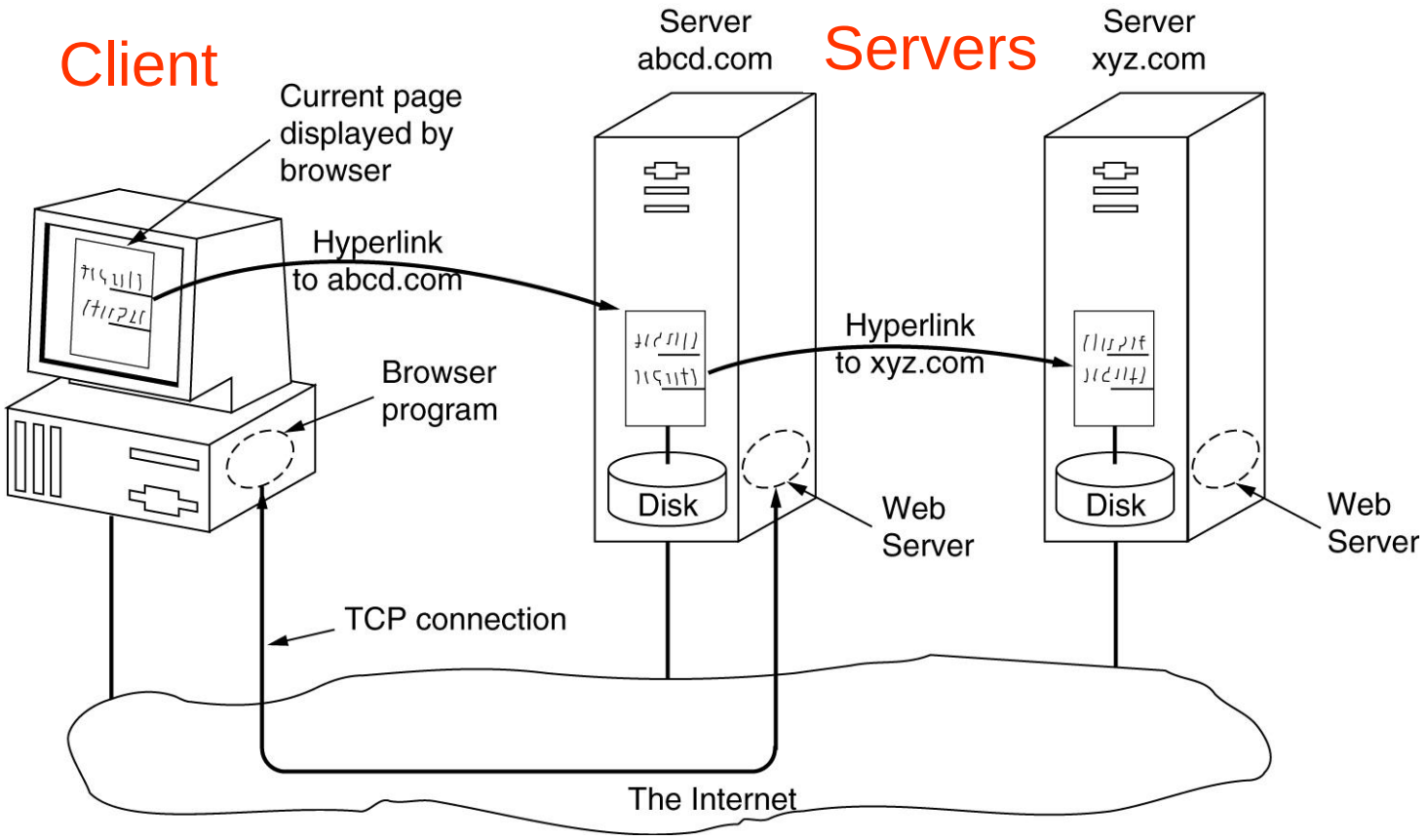
Alternative 2

External sound file

(Will be played if client is capable)

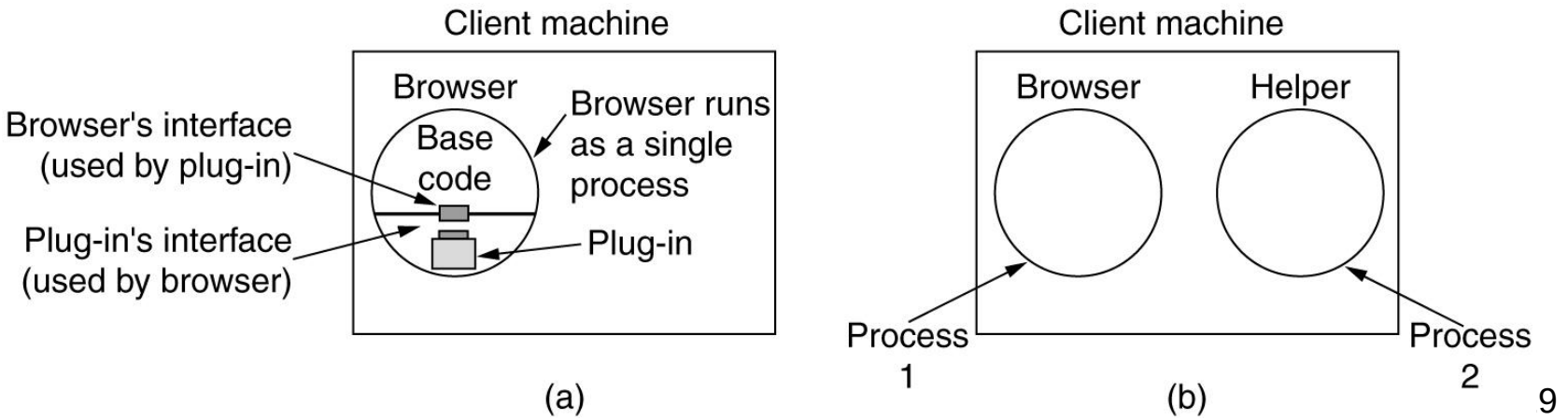
```
From: elinor@abcd.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times

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--
```

World Wide Web



Client (browser)

- User accesses Internet via a browser program (eg MS Explorer or Netscape)
- Browser understands *html*
  - hypertext mark-up language
- Browser uses plug-ins or helper applications to handle specific file types referenced by html web pages
  eg audio, video



URLs

- Uniform Resource Locators
- Part before :// specifies application/protocol
- Part after :// specifies target host (and file)

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

HTTP protocol (RFC 2616)

Method	Description
GET	Request to read a Web page
HEAD	Request to read a Web page's header
PUT	Request to store a Web page
POST	Append to a named resource (e.g., a Web page)
DELETE	Remove the Web page
TRACE	Echo the incoming request
CONNECT	Reserved for future use
OPTIONS	Query certain options

HTTP protocol (2)

Reply codes

Code	Meaning	Examples
1xx	Information	100 = server agrees to handle client's request
2xx	Success	200 = request succeeded; 204 = no content present
3xx	Redirection	301 = page moved; 304 = cached page still valid
4xx	Client error	403 = forbidden page; 404 = page not found
5xx	Server error	500 = internal server error; 503 = try again later

HTTP protocol (3)

Header fields - (do not memorize!)

Sent following a request such as GET

Sent following a reply such as "page not found"

Beware!

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

Cookies

- Used to store state information so a returning user can *resume* a session
- Stores login IDs, passwords, credit card numbers
- eg amazon.com, barclays.co.uk
- Can track Internet usage (click-trails) for marketing (or other) purposes

Static web pages

- Html pages describe both content and formatting
- Makes it difficult to achieve consistent look and feel across a large website
- XML/XSL is an attempt to separate content and formatting

Static web pages (2)

- XML = eXtensible Mark-up Language
- XML page describes content only with formatting tags

```
<?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>
```

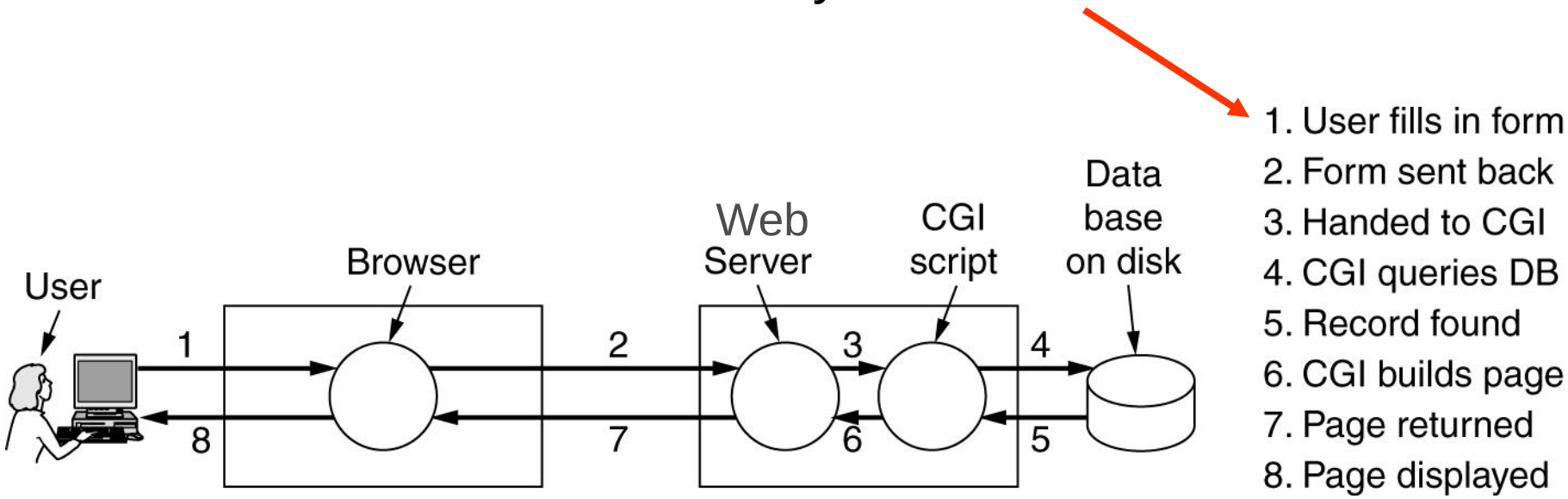
Static web pages (3)

- XSL = XML Style sheet
- XSL page describes meaning of formatting tags

```
<?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>
```

Dynamic web pages

Some information may come from cookies



Pages generated dynamically by server then sent to client for display

CGI = Common Gateway Interface
A generic API for web servers to communicate with back–end programs

Dynamic web pages (2)

(a) Server-side scripting with PHP (a script embedded within the html page and executed by the server to generate the page dynamically)

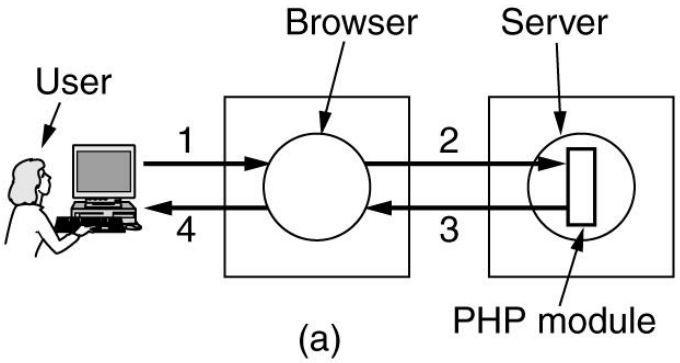
Scripts interact mainly with servers (eg database)

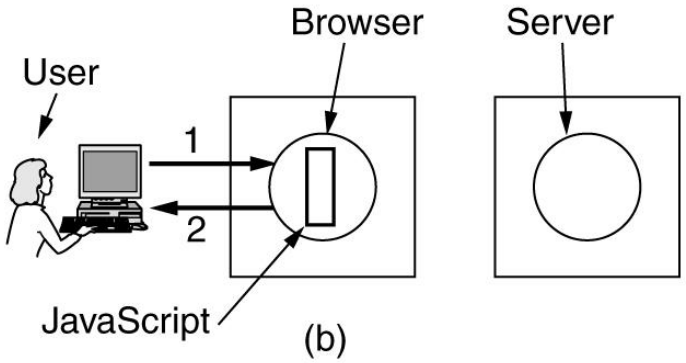
Can also use JSP (Java) or ASP (Visual Basic Script)

(b) Client-side scripting with JavaScript (a script embedded within the html page and executed by the client to generate the page dynamically)

Scripts interact mainly with user (eg mouse clicks)

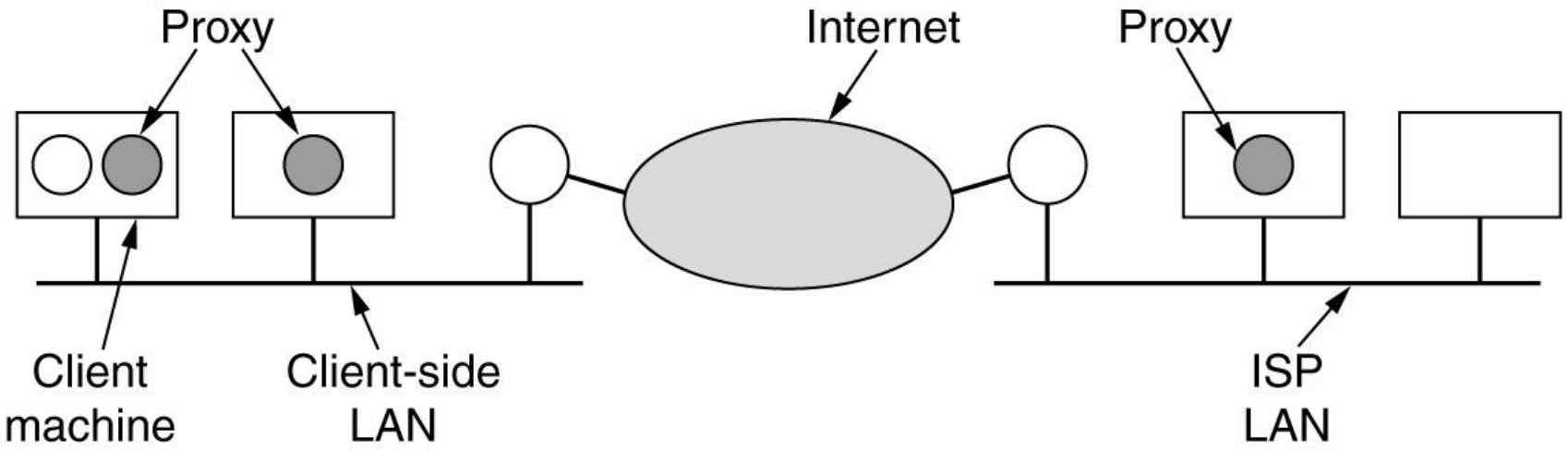
Can also use applets (Java) or ActiveX (native compiled)



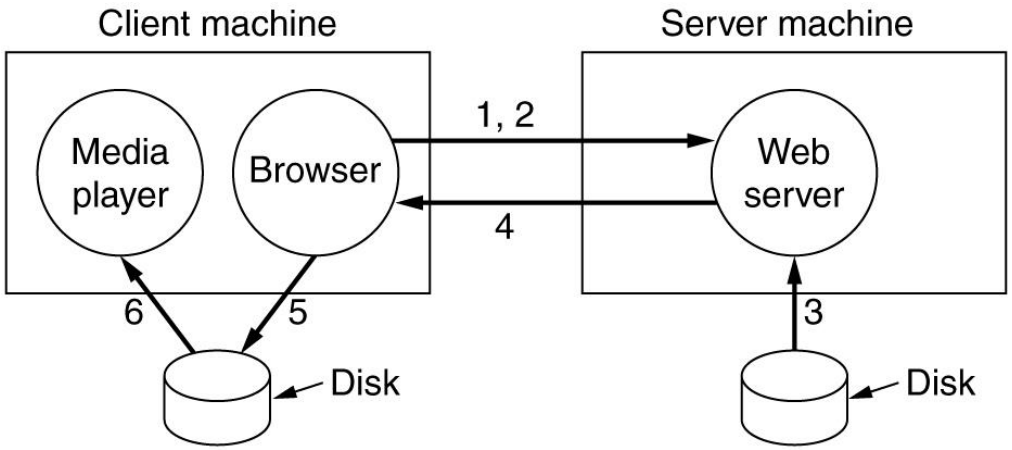


Caching

- Copies of frequently used web pages stored on client or local proxy host or ISP proxy host
- 3-level cache shown
- Caching improves performance but content could be old (big issue with dynamic pages)
- HTTP includes commands for proxies to determine if a page has changed without having to GET the page



Streaming Audio



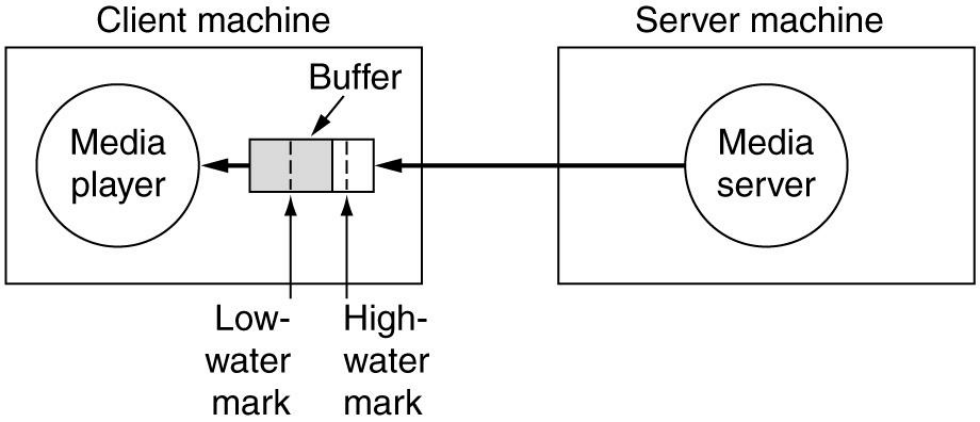
1. Establish TCP connection
2. Send HTTP GET request
3. Server gets file from disk
4. File sent back
5. Browser writes file to disk
6. Media player fetches file block by block and plays it

Above method transfers whole file to client disk before playing

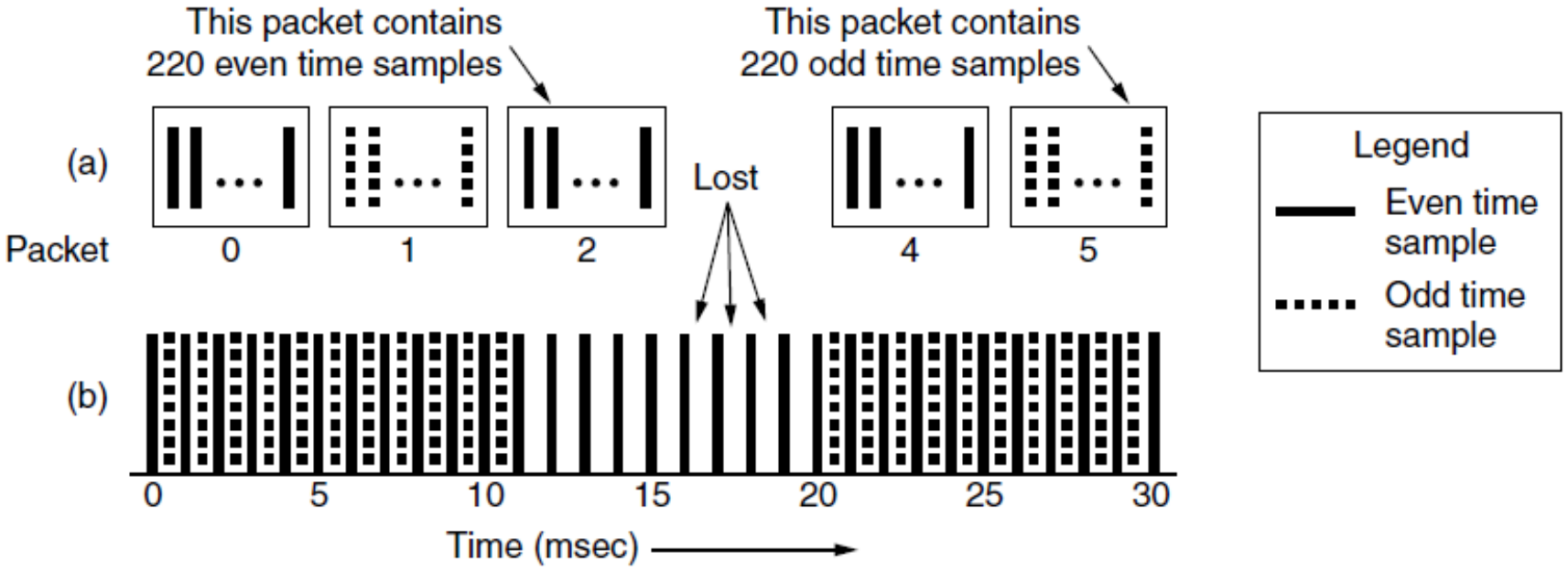
For steaming audio, file does not contain music but instead contains a URL (eg rtsp://ncl.ac.uk/mysong.mp3)

RTSP Real Time Streaming Protocol – RFC 2326

Media player fetches music in small packets and can play after a few have been buffered



Streaming Stored Media



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