

# **NETWORKS**

The Application Layer

## **Chapter VI**

# **The Application Layer**

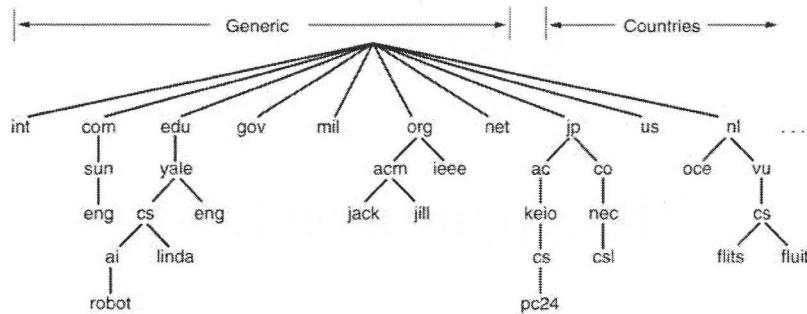
# Chapter 7

## The Application Layer

### DNS – The Domain Name System

- The DNS Name Space
- Resource Records
- Name Servers

## The DNS Name Space



A portion of the Internet domain name space.

## 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.

## Resource Records (2)

```
; Authoritative data for cs.vu.nl
cs.vu.nl.      86400 IN SOA    star.boss (952771,7200,7200,2419200,86400)
                86400 IN TXT    "Divisie Wiskunde en Informatica."
                86400 IN TXT    "Vrije Universiteit Amsterdam."
                86400 IN MX     1 zephyr.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

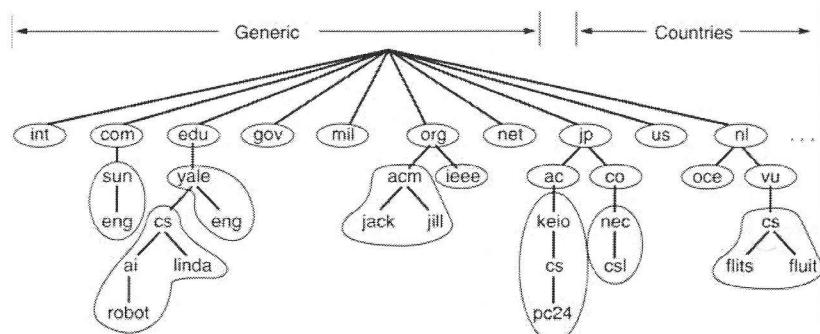
rowboat          IN A      130.37.56.201
                  IN MX    1 rowboat
                  IN MX    2 zephyr
                  IN HINFO  Sun Unix

little-sister    IN A      130.37.62.23
                  IN HINFO  Mac MacOS

laserjet         IN A      192.31.231.216
                  IN HINFO  "HP Laserjet IISi" Proprietary
```

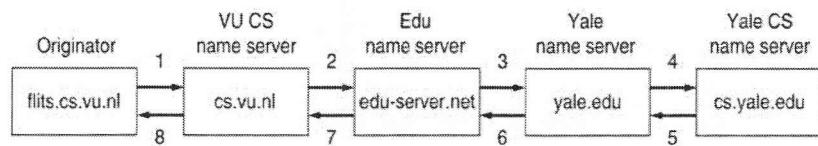
A portion of a possible DNS database for *cs.vu.nl*.

## Name Servers



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

## Name Servers (2)



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

## Electronic Mail

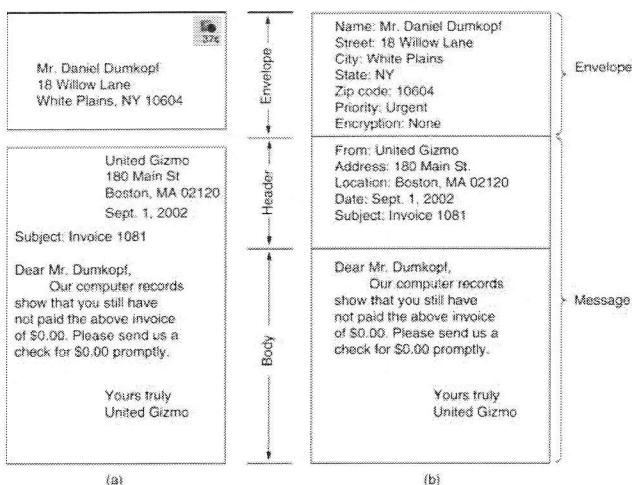
- Architecture and Services
- The User Agent
- Message Formats
- Message Transfer
- Final Delivery

# Architecture and Services

## Basic functions

- Composition
- Transfer
- Reporting
- Displaying
- Disposition

## The User Agent



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

## 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

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 (2)

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 (3)

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 <b>Carolyn</b> Carolyn </b>  
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--

A multipart message containing enriched and audio alternatives.

## Message Transfer

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

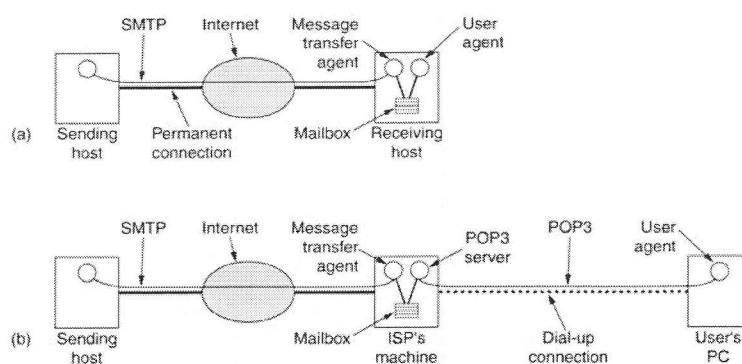
```

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 "."
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 <b>Carolyn</b> Carolyn </b>
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: QUIT
S: 250 message accepted
C: QUIT
S: 221 xyz.com closing connection

```

## Final Delivery



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

## POP3

```
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
```

Using POP3 to fetch three messages.

## IMAP

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

A comparison of POP3 and IMAP.

# The World Wide Web

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

## 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

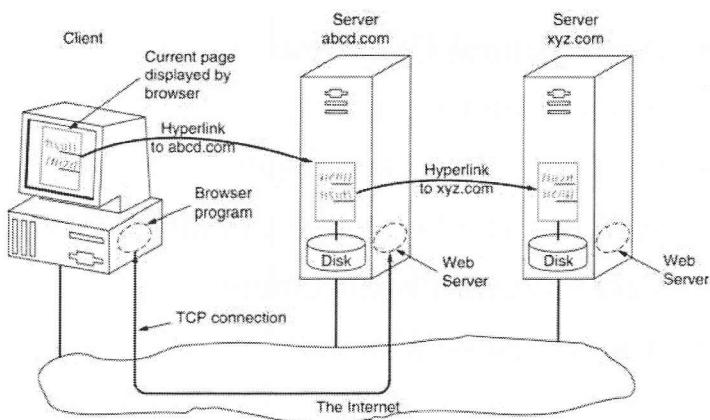
- Information for prospective majors
- Personnel
  - Faculty members
    - Graduate students
    - Nonacademic staff
- Research Projects
- Positions available
- Our most popular courses
  - Dealing with herbivores
  - Horse management
  - Negotiating with your pet
  - User-friendly doghouse construction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

Webmaster@animalpsy.eastpodunk.edu

(b)

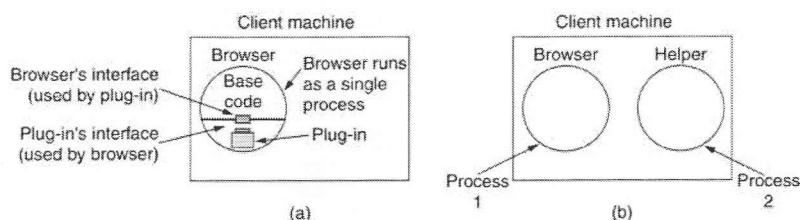
(a) A Web page (b) The page reached by clicking on Department of Animal Psychology.

## Architectural Overview (2)



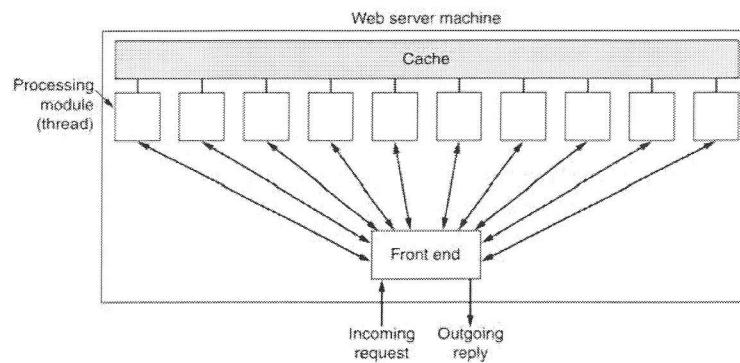
The parts of the Web model.

## The Client Side



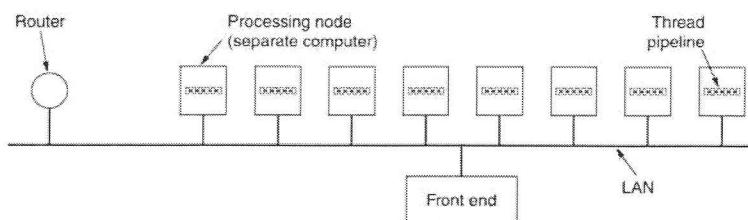
(a) A browser plug-in. (b) A helper application.

## The Server Side



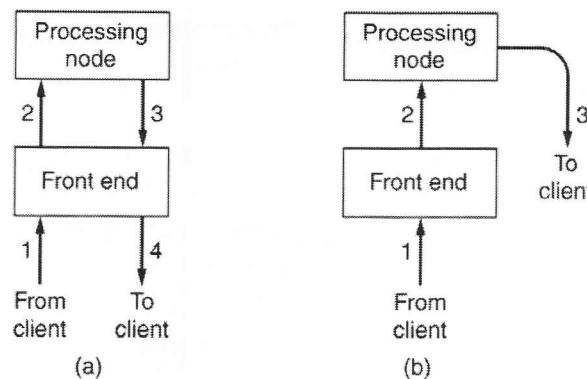
A multithreaded Web server with a front end and processing modules.

## The Server Side (2)



A server farm.

## The Server Side (3)



- (a) Normal request-reply message sequence.  
(b) Sequence when TCP handoff is used.

## URLs – Uniform Resource Locators

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.

## Statelessness and Cookies

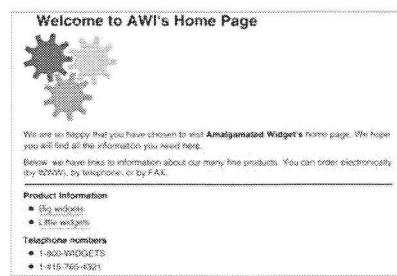
Domain	Path	Content	Expires	Secure
toms-casino.com	/	CustomerID=497793521	15-10-02 17:00	Yes
joes-store.com	/	Cart=1-00501;1-07031;2-13721	11-10-02 14:22	No
aportal.com	/	Prefs=Stk:SUNW+ORCL;Spt:Jets	31-12-10 23:59	No
sneaky.com	/	UserID=3627239101	31-12-12 23:59	No

Some examples of cookies.

## HTML – HyperText Markup Language

```
<html><head> AMALGAMATED WIDGET, INC </head> <body>
<h1> Welcome to AWI's Home Page!</h1>

```



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

## HTML (2)

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

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

## 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=centers>
<col align=centers>
<col align=centers>
<col align=centers>
<tr> <th> Item </th> <th> HTML 1.0 </th> <th> HTML 2.0 </th> <th> HTML 3.0 </th> <th> HTML 4.0 </th>
<tr> <td> Hyperlinks </td> <x> <td> x <td> x <td> x <td> x </td>
<tr> <td> Images </td> <x> <td> x <td> x <td> x <td> x </td>
<tr> <td> Lists </td> <x> <td> x <td> x <td> x <td> x <td> x </td>
<tr> <td> Active Maps and Images </td> &nbsp; <td> x <td> x <td> x <td> x <td>
<tr> <td> Forms </td> &nbsp; <td> x <td> x <td> x <td> x <td>
<tr> <td> Equations </td> &nbsp; <td> x &nbsp; <td> x <td> x <td> x <td>
<tr> <td> Toolbars </td> &nbsp; <td> x &nbsp; <td> x <td> x <td>
<tr> <td> Tables </td> &nbsp; <td> x &nbsp; <td> x <td> x <td>
<tr> <td> Accessibility features </td> &nbsp; <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x <td>
<tr> <td> Object embedding </td> &nbsp; <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x <td>
<tr> <td> Scripting </td> &nbsp; <td> &nbsp; <td> &nbsp; <td> &nbsp; <td> x <td>
</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)

## 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=48> </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 <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)

Widget Order Form

Name

Street address

City  State  Country

Credit card #  Expires  M/C  Visa

Widget size Big  Little  Ship by express courier

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

(b)

## 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.

## XML and XSL

A simple Web page  
in XML.

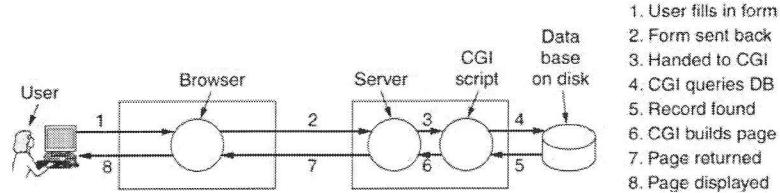
```
<?xml version="1.0" ?>
<?xml-stylesheet type="text/xsl" href="b5.xsl"?>
<book_list>
  <book>
    <title> Computer Networks, 4/e </title>
    <author> Andrew S. Tanenbaum </author>
    <year> 2003 </year>
  </book>
  <book>
    <title> Modern Operating Systems, 2/e </title>
    <author> Andrew S. 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>
```

## 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>
```

## Dynamic Web Documents



Steps in processing the information from an HTML form.

## Dynamic Web Documents (2)

```
<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 (3)

```
<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)

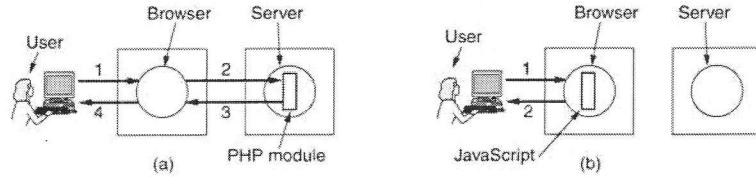
(a) A Web page containing a form. (b) A PHP script for handling the output of the form. (c) Output from the PHP script when the inputs are "Barbara" and 24 respectively.

## 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>
```

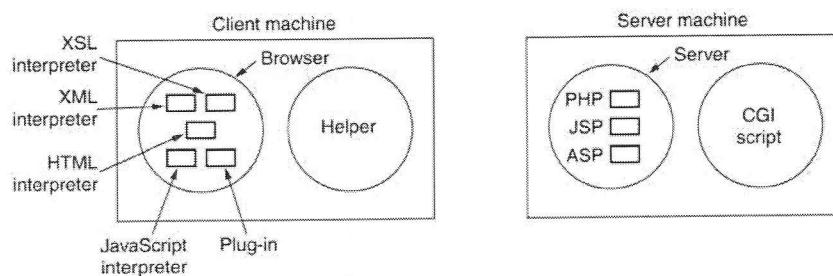
A JavaScript program for computing and printing factorials.

## 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.

## 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.

## HTTP Methods (2)

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.

## 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](http://www.ietf.org/rfc.html).

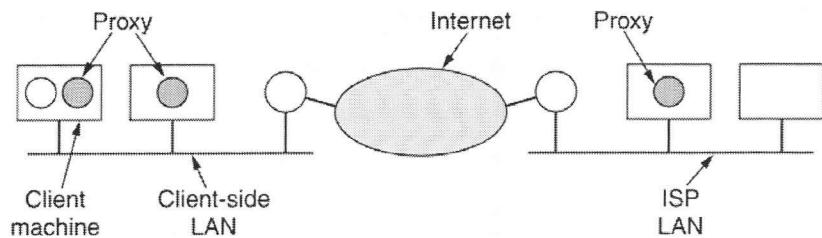
```
Trying 4.17.168.6...
Connected to www.ietf.org.
Escape character is '^].
HTTP/1.1 200 OK
Date: Wed, 08 May 2002 22:54:22 GMT
Server: Apache/1.3.20 (Unix) mod_ssl/2.8.4 OpenSSL/0.9.5a
Last-Modified: Mon, 11 Sep 2000 13:56:29 GMT
ETag: "2a79d-c8b-39bce48d"
Accept-Ranges: bytes
Content-Length: 3211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

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

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

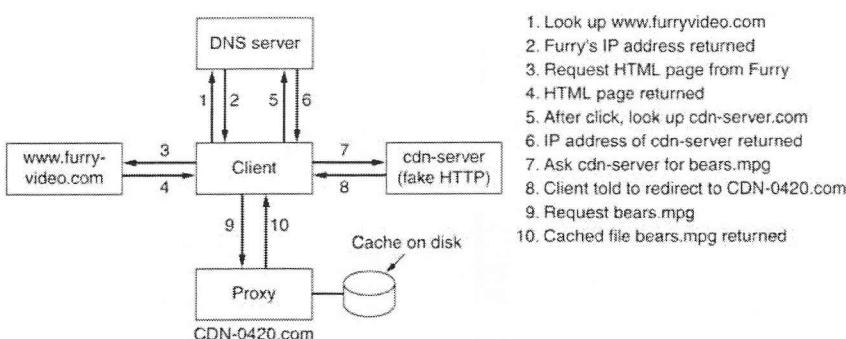
</head>
```

## Caching



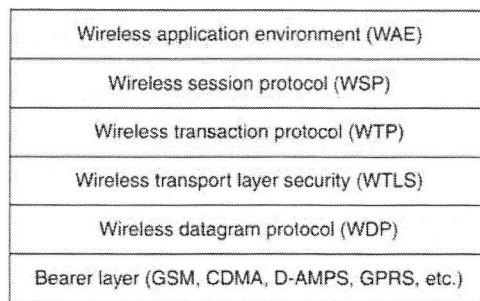
Hierarchical caching with three proxies.

## The Wireless Web



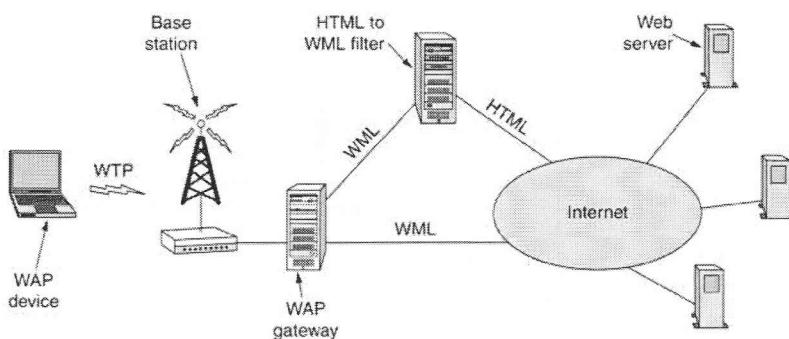
Steps in looking up a URL when a CDN is used.

## WAP – The Wireless Application Protocol



The WAP protocol stack.

## WAP (2)



The WAP architecture.

## 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)

XHTML	
WSP	HTTP
WTP	TLS
WTLS	TCP
WDP	IP
Bearer layer	Bearer layer

WAP 1.0 protocol stack                    WAP 2.0 protocol stack

WAP 2.0 supports two protocol stacks.

## Second-Generation Wireless Web (4)

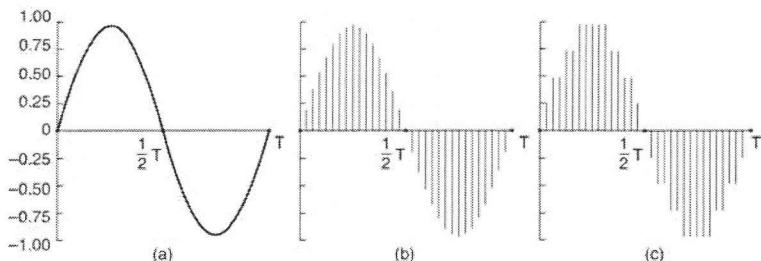
Module	Req.?	Function	Example tags
Structure	yes	Doc. structure	body, head, html, title
Text	yes	Information	br, code, dfn, em, h <sub>n</sub> , kbd, p, strong
Hypertext	yes	Hyperlinks	a
List	yes	Itemized lists	dl, dt, dd, ol, ul, li
Forms	No	Fill-in forms	form, input, label, option, textarea
Tables	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.

## Multimedia

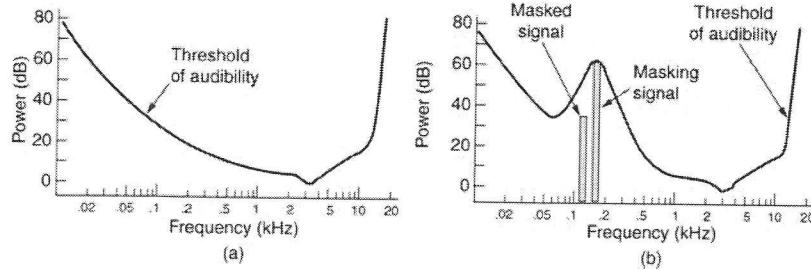
- Introduction to Audio
- Audio Compression
- Streaming Audio
- Internet Radio
- Voice over IP
- Introduction to Video
- Video Compression
- Video on Demand
- The MBone – The Multicast Backbone

## Introduction to Audio



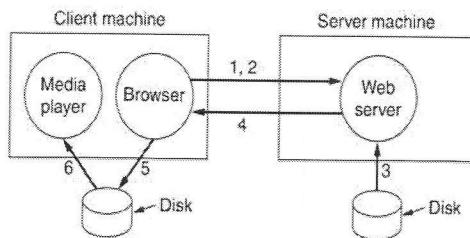
(a) A sine wave. (b) Sampling the sine wave.  
(c) Quantizing the samples to 4 bits.

## Audio Compression



- (a) The threshold of audibility as a function of frequency.  
(b) The masking effect.

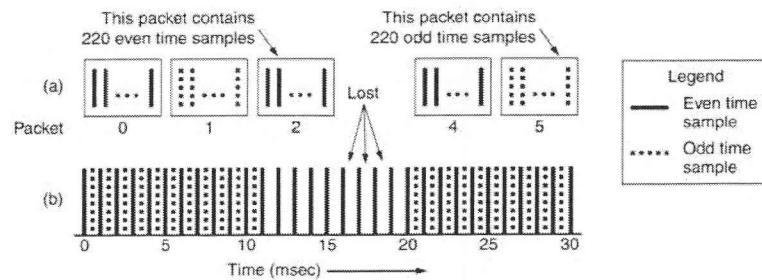
## Streaming Audio



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

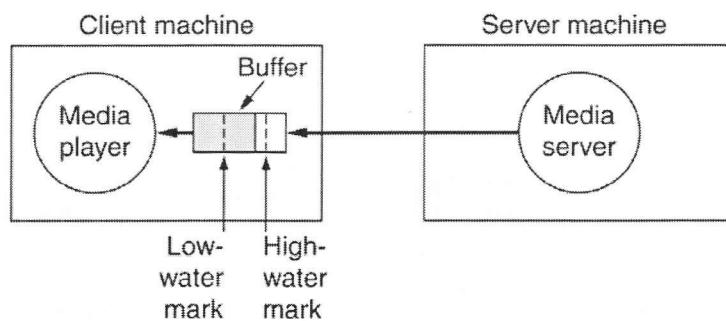
A straightforward way to implement clickable music on a Web page.

## Streaming Audio (2)



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

## Streaming Audio (3)



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

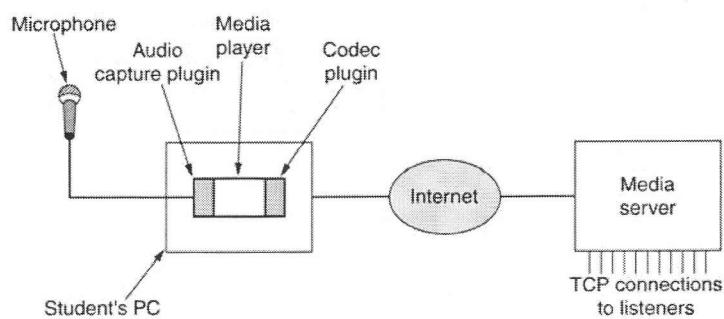
## Streaming Audio (4)

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

Real Time Streaming Protocol commands from the player to the server.

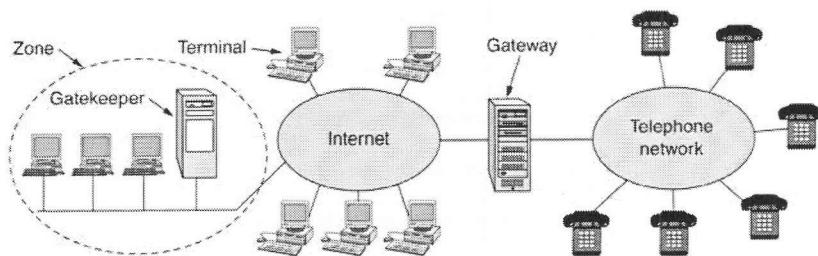
R T S P

## Internet Radio



A student radio station.

## Voice over IP



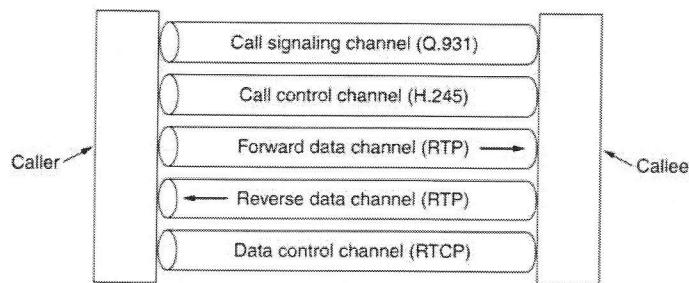
The H323 architectural model for Internet telephony.

## Voice over IP (2)

Speech	Control								
G.7xx	RTCP	H.225 (RAS)	Q.931 (Call signaling)	H.245 (Call control)					
RTP									
UDP		TCP							
IP									
Data link protocol									
Physical layer protocol									

The H323 protocol stack.

## Voice over IP (3)



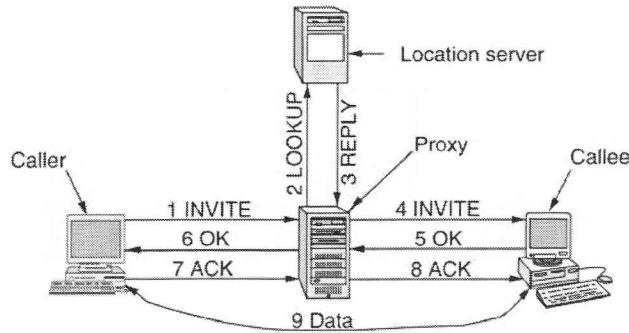
Logical channels between the caller and callee during a call.

## 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.

## SIP (2)

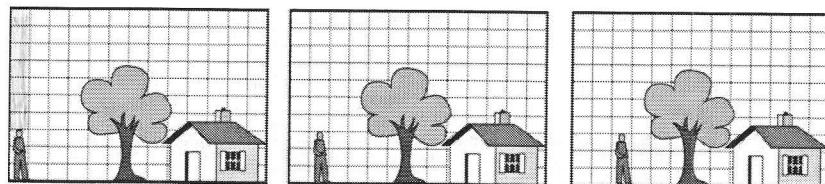


Use a proxy and redirection servers with SIP.

## Comparison of H.323 and SIP

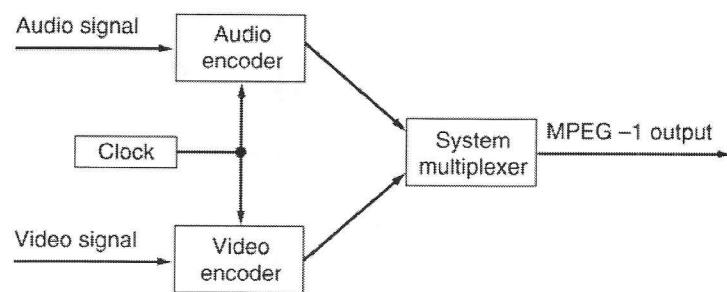
Item	H.323	SIP
Designed by	ITU	IETF
Compatibility with PSTN	Yes	Largely
Compatibility with Internet	No	Yes
Architecture	Monolithic	Modular
Completeness	Full protocol stack	SIP just handles setup
Parameter negotiation	Yes	Yes
Call signaling	Q.931 over TCP	SIP over TCP or UDP
Message format	Binary	ASCII
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

## Video



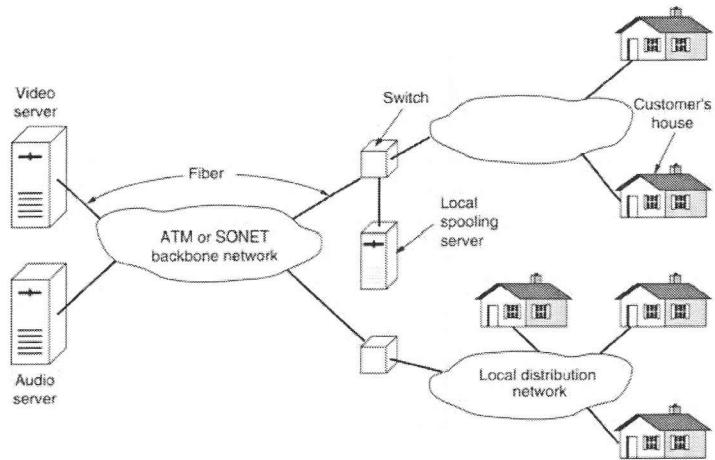
Three consecutive frames.

## The MPEG Standard



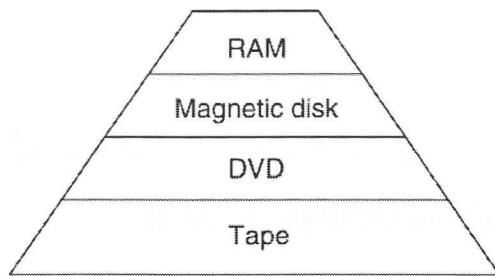
Synchronization of the audio and video streams in MPEG-1.

## Video on Demand



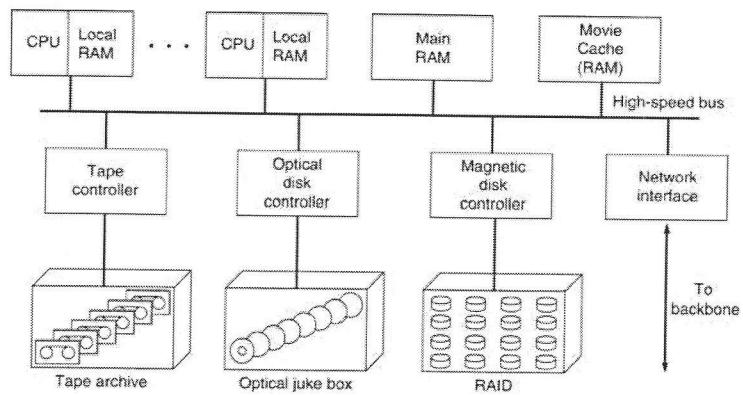
Overview of a video-on-demand system.

## Video Servers



A video server storage hierarchy.

## Video Servers (2)

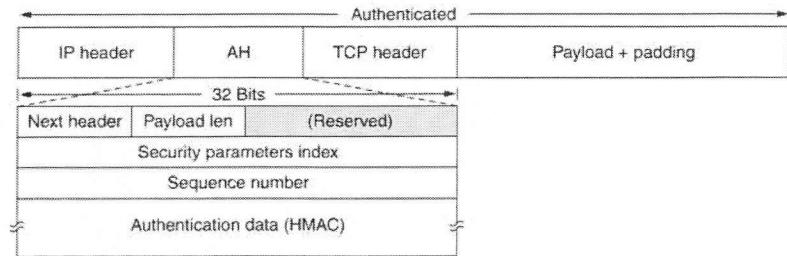


The hardware architecture of a typical video server.

## Communication Security

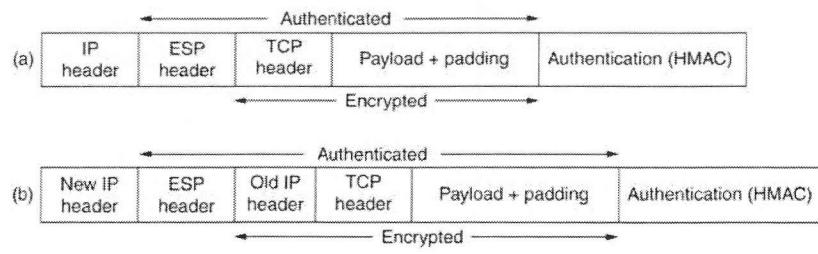
- a) IPsec
- b) Firewalls
- c) Virtual Private Networks
- d) Wireless Security

# IPsec



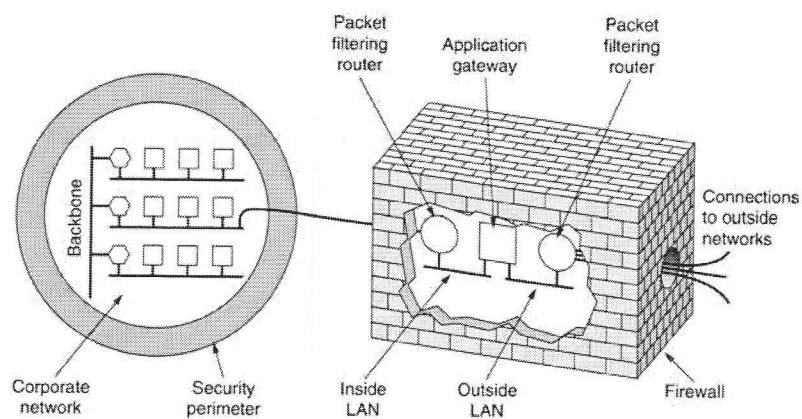
a) The IPsec authentication header in transport mode for IPv4.

# IPsec (2)



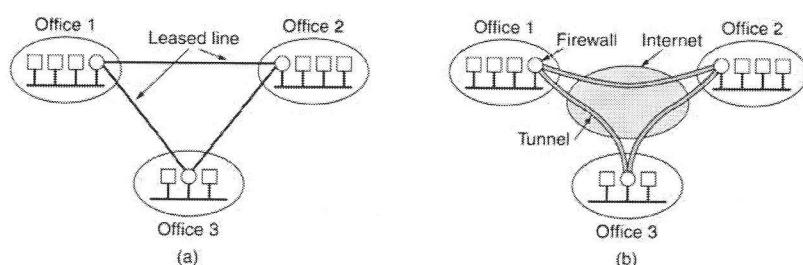
(a) ESP in transport mode. (b) ESP in tunnel mode.

## Firewalls



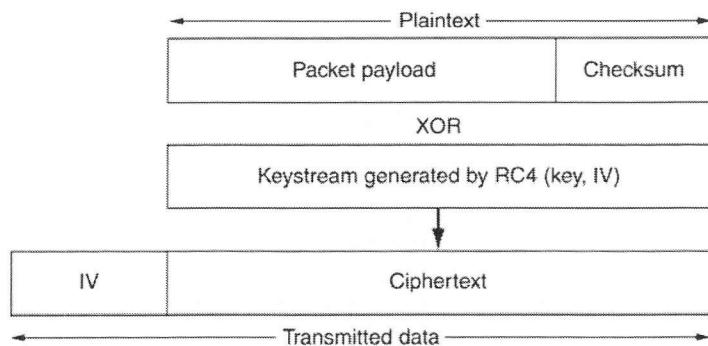
- a) A firewall consisting of two packet filters and an application gateway.

## Virtual Private Networks



- a) (a) A leased-line private network. (b) A virtual private network.

## 802.11 Security

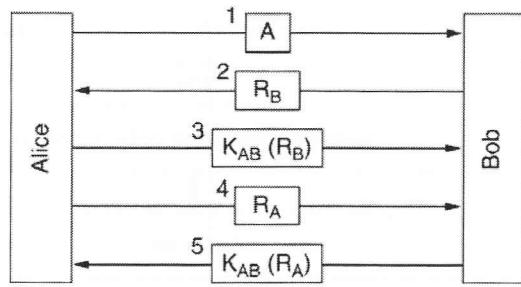


- a) Packet encryption using WEP.

## Authentication Protocols

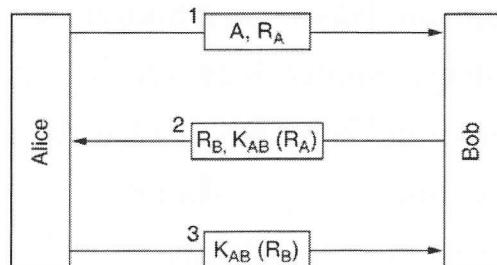
- a) Authentication Based on a Shared Secret Key
- b) Establishing a Shared Key: Diffie-Hellman
- c) Authentication Using a Key Distribution Center
- d) Authentication Using Kerberos
- e) Authentication Using Public-Key Cryptography

## Authentication Based on a Shared Secret Key



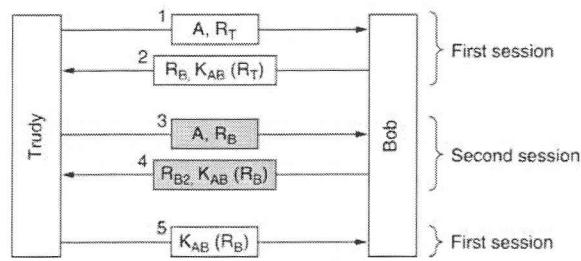
a) Two-way authentication using a challenge-response protocol.

## Authentication Based on a Shared Secret Key (2)



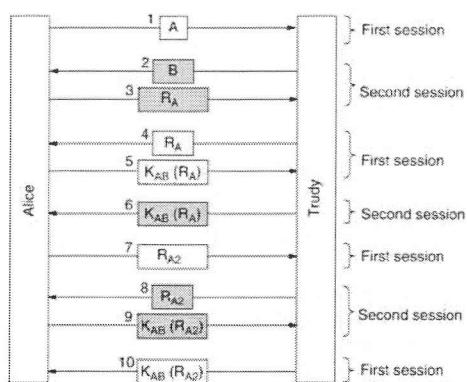
a) A shortened two-way authentication protocol.

### Authentication Based on a Shared Secret Key (3)



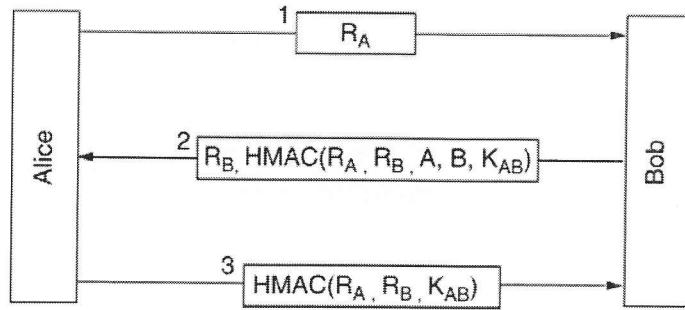
a) The reflection attack.

### Authentication Based on a Shared Secret Key (4)



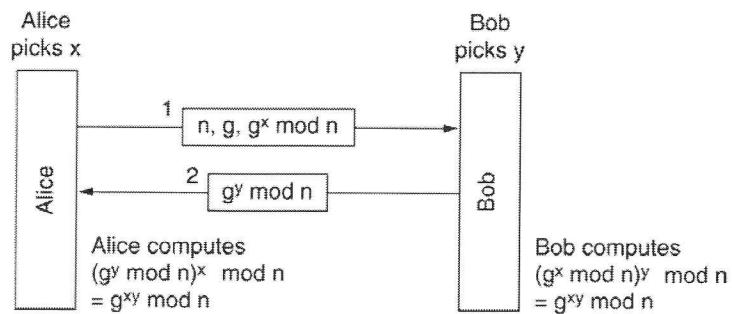
a) A reflection attack on the protocol slide 41

### Authentication Based on a Shared Secret Key (5)



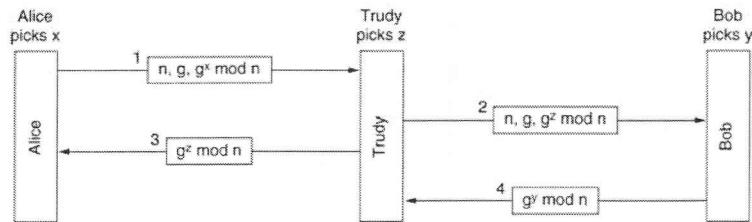
a) Authentication using HMACs.

### Establishing a Shared Key: The Diffie-Hellman Key Exchange



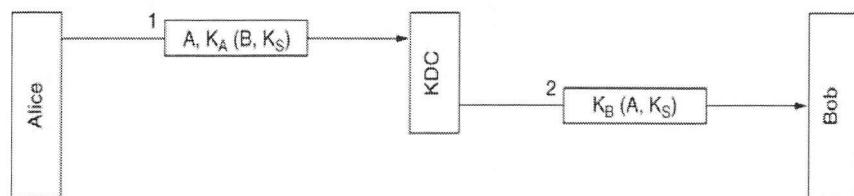
a) The Diffie-Hellman key exchange.

## Establishing a Shared Key: The Diffie-Hellman Key Exchange



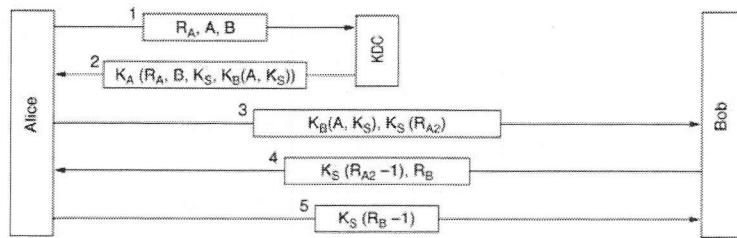
a) The bucket brigade or man-in-the-middle attack.

## Authentication Using a Key Distribution Center



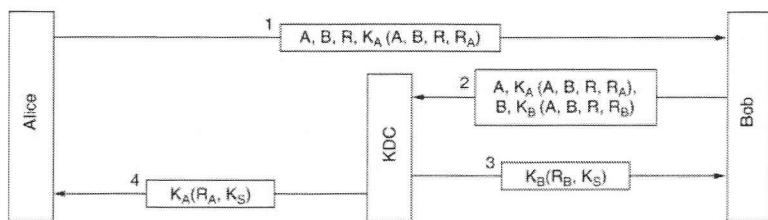
a) A first attempt at an authentication protocol using a KDC.

## Authentication Using a Key Distribution Center (2)



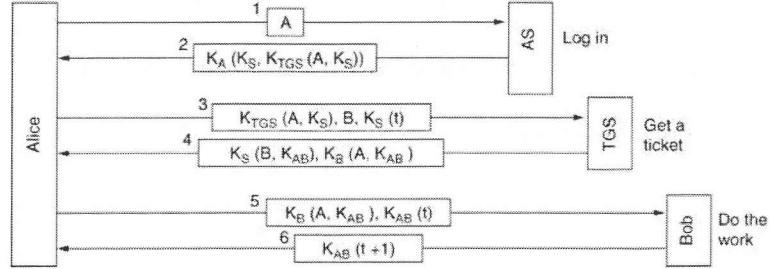
a) The Needham-Schroeder authentication protocol.

## Authentication Using a Key Distribution Center (3)



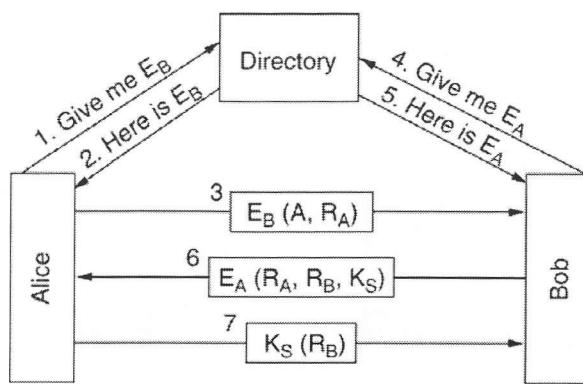
a) The Otway-Rees authentication protocol (slightly simplified).

## Authentication Using Kerberos



a) The operation of Kerberos V4.

## Authentication Using Public-Key Cryptography

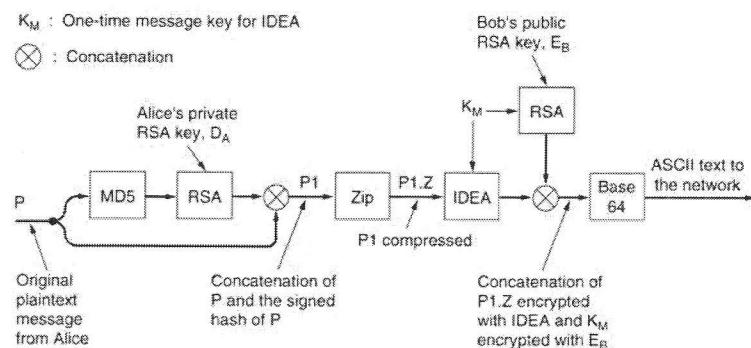


a) Mutual authentication using public-key cryptography.

## E-Mail Security

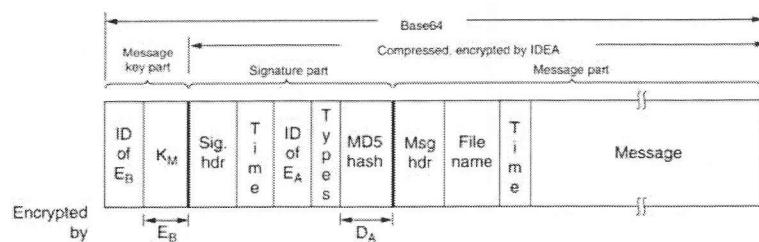
- a) PGP – Pretty Good Privacy
- b) PEM – Privacy Enhanced Mail
- c) S/MIME

### PGP – Pretty Good Privacy



- a) PGP in operation for sending a message.

## PGP – Pretty Good Privacy (2)

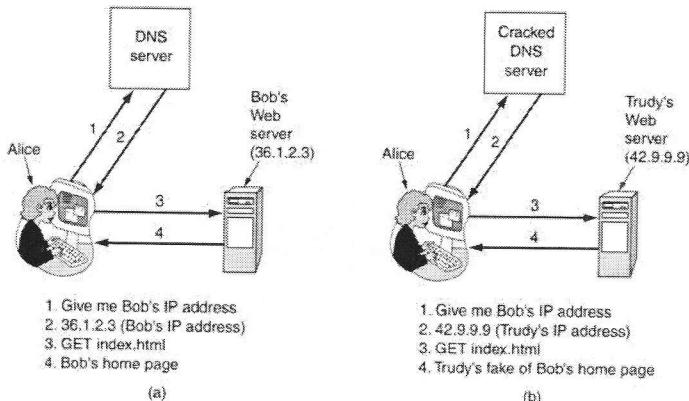


a) A PGP message.

## Web Security

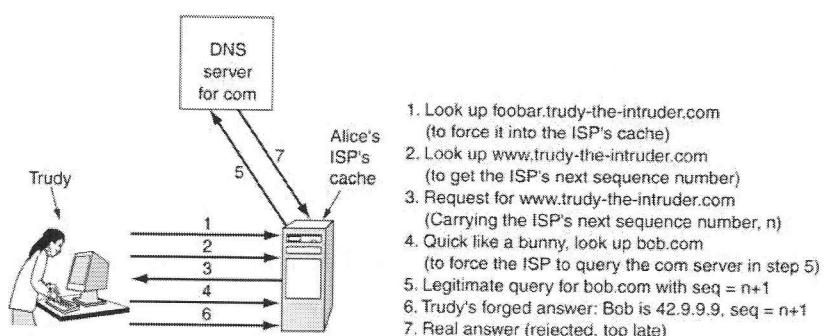
- a) Threats
- b) Secure Naming
- c) SSL – The Secure Sockets Layer
- d) Mobile Code Security

## Secure Naming



- a) (a) Normal situation. (b) An attack based on breaking into DNS and modifying Bob's record.

## Secure Naming (2)



- a) How Trudy spoofs Alice's ISP.

## Secure DNS

Domain name	Time to live	Class	Type	Value
bob.com.	86400	IN	A	36.1.2.3
bob.com.	86400	IN	KEY	3682793A7B73F731029CE2737D...
bob.com.	86400	IN	SIG	86947503A8B848F5272E53930C...

An example RRSet for *bob.com*. The *KEY* record is Bob's public key. The *SIG* record is the top-level *com* server's signed has of the *A* and *KEY* records to verify their authenticity.

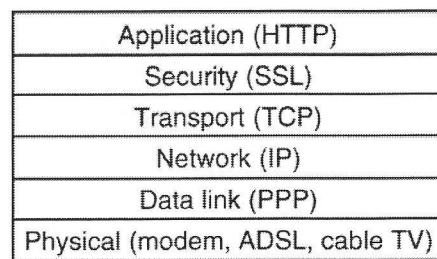
## Self-Certifying Names

http://www.bob.com:2g5hd8bfjkc7mf6hg8dgany23xds4pe6/photos/bob.jpg

Server      SHA-1 (Server, Server's Public key)      File name

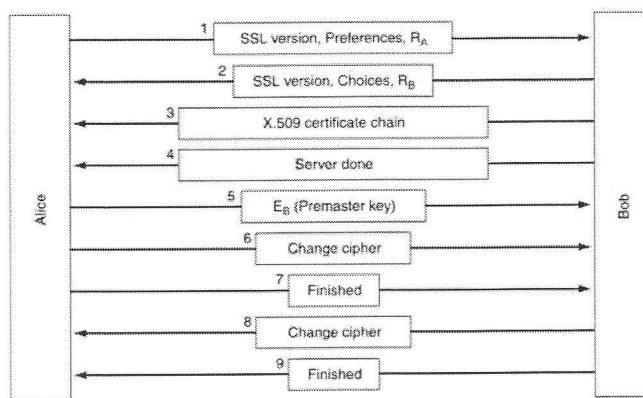
- a) A self-certifying URL containing a hash of server's name and public key.

## SSL—The Secure Sockets Layer



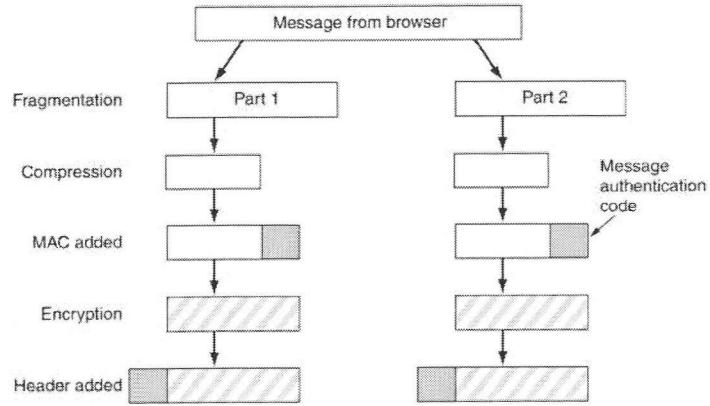
- a) Layers (and protocols) for a home user browsing with SSL.

## SSL (2)



- a) A simplified version of the SSL connection establishment subprotocol.

### SSL (3)



a) Data transmission using SSL.





ISP: PAI

CDN:

Spooling Server;

ATU, SNET

buchanan network