7

THE APPLICATION LAYER

Adversary	Goal
Student	To have fun snooping on people's email
Hacker	To test out someone's security system; steal data
Sales rep	To claim to represent all of Europe, not just Andorra
Businessman	To discover a competitor's strategic marketing plan
Ex-employee	To get revenge for being fired
Accountant	To embezzle money from a company
Stockbroker	To deny a promise made to a customer by email
Con man	To steal credit card numbers for sale
Spy	To learn an enemy's military strength
Terrorist	To steal germ warfare secrets

Fig. 7-1. Some people who cause security problems and why.

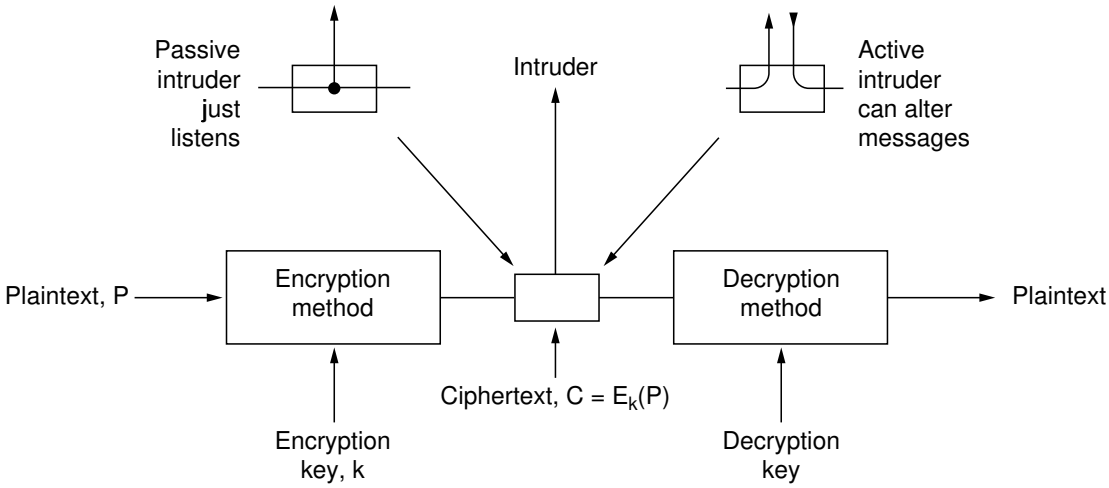


Fig. 7-2. The encryption model.

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

Plaintext

pleasetransferonemilliondollarsto
myswissbankaccountsixtwotwo

Ciphertext

AFLLSKSOSELAWAIATOOSSCTCLNMOMANT
ESILYNTWRNNTSOWDPAEDOBUOERIRICXB

Fig. 7-3. A transposition cipher.

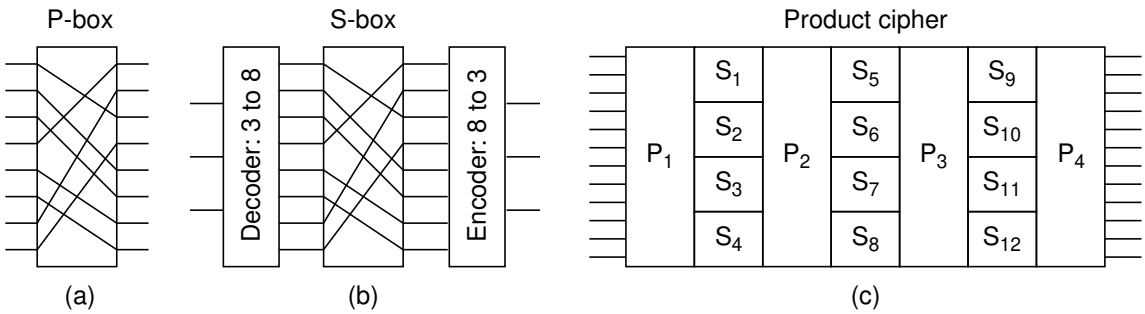


Fig. 7-4. Basic elements of product ciphers. (a) P-box. (b) S-box. (c) Product.

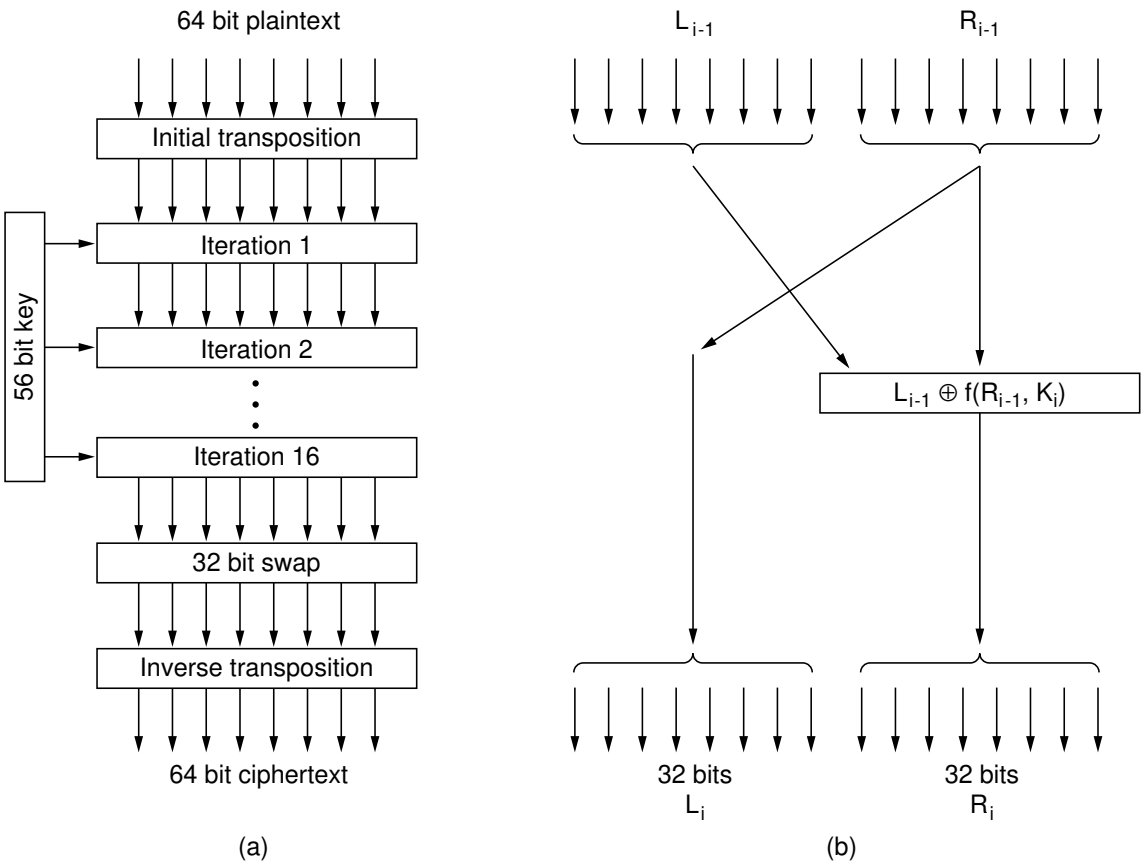


Fig. 7-5. The data encryption standard. (a) General outline. (b) Detail of one iteration.

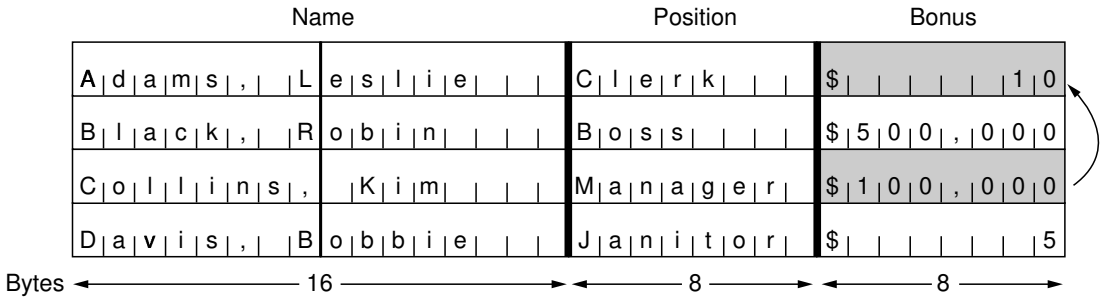


Fig. 7-6. The plaintext of a file encrypted as 16 DES blocks.

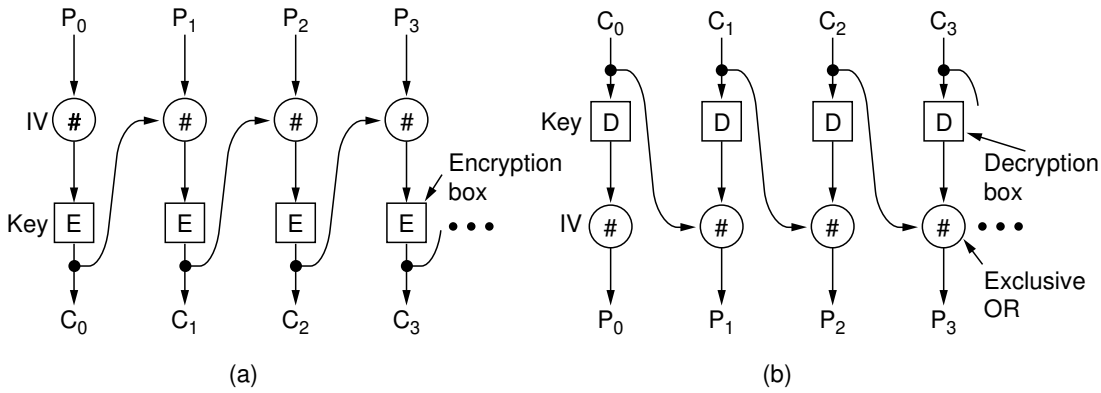


Fig. 7-7. Cipher block chaining

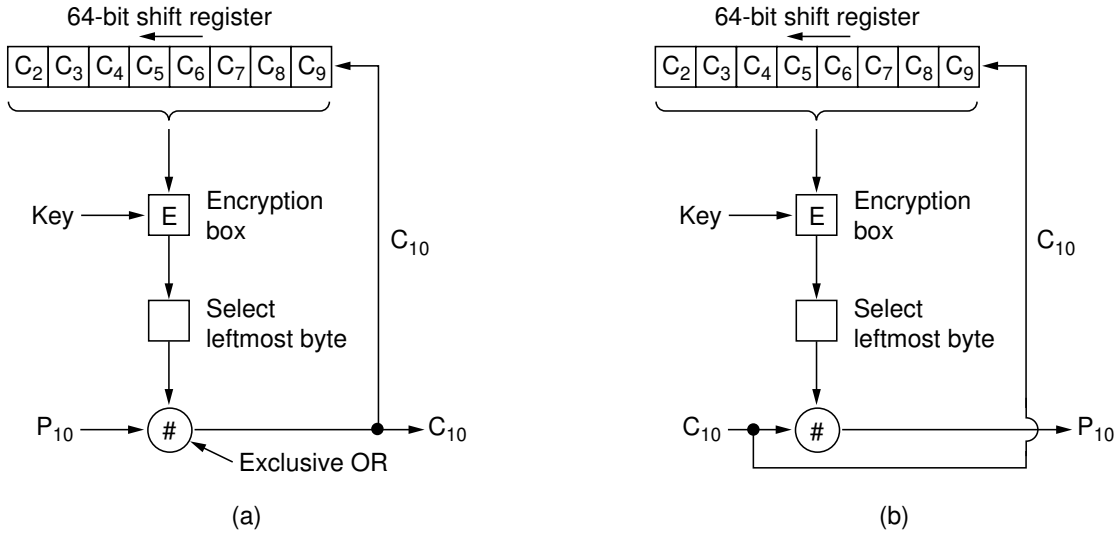


Fig. 7-8. (a) Cipher feedback mode. (b) Output feedback mode.

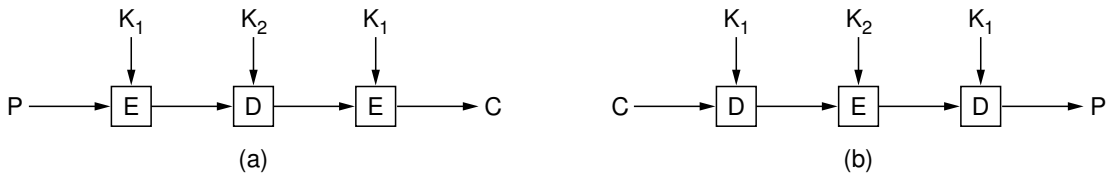


Fig. 7-9. Triple encryption using DES.

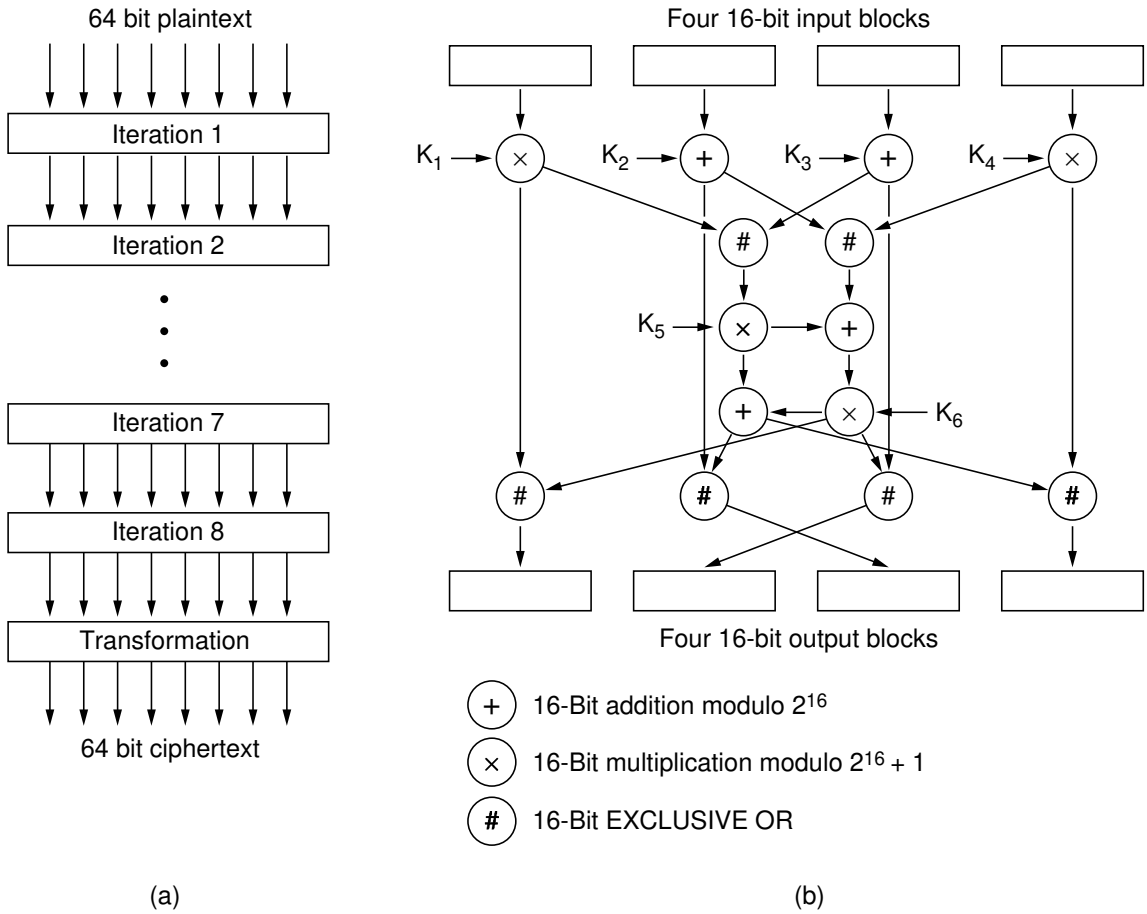


Fig. 7-10. (a) IDEA. (b) Detail of one iteration.

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	1	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	5	E
Sender's computation				Receiver's computation		

Fig. 7-11. An example of the RSA algorithm.

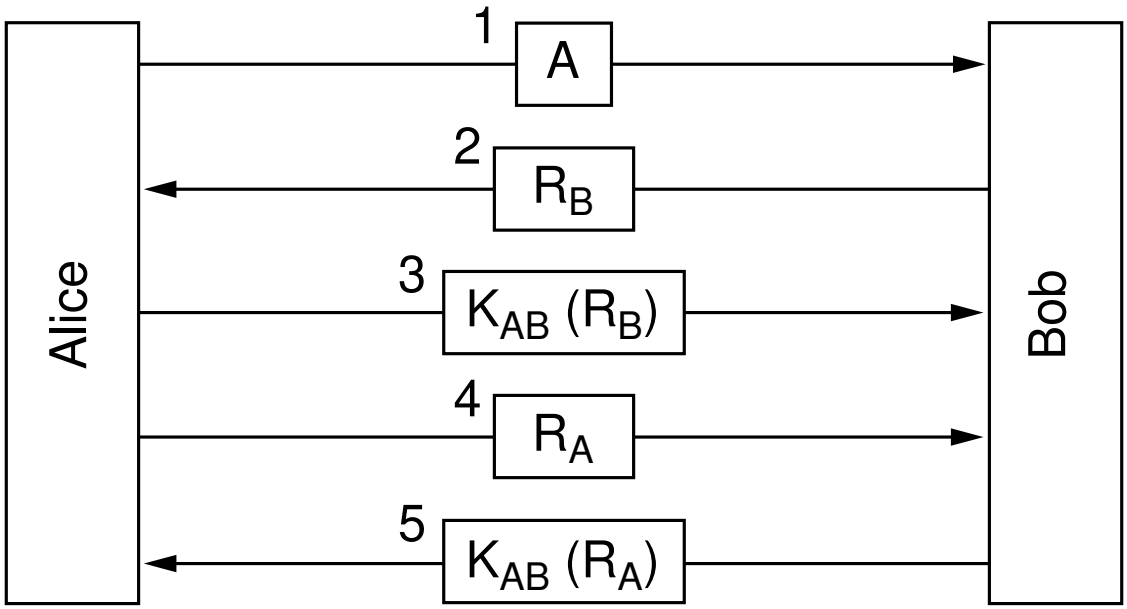


Fig. 7-12. Two-way authentication using a challenge-response protocol.

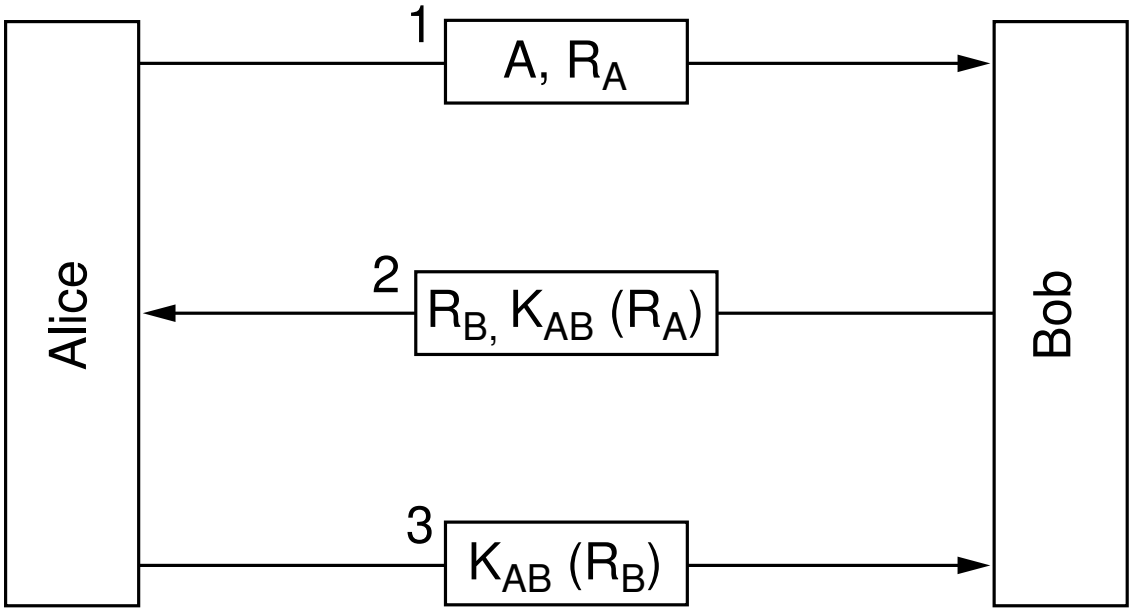


Fig. 7-13. A shortened two-way authentication protocol.

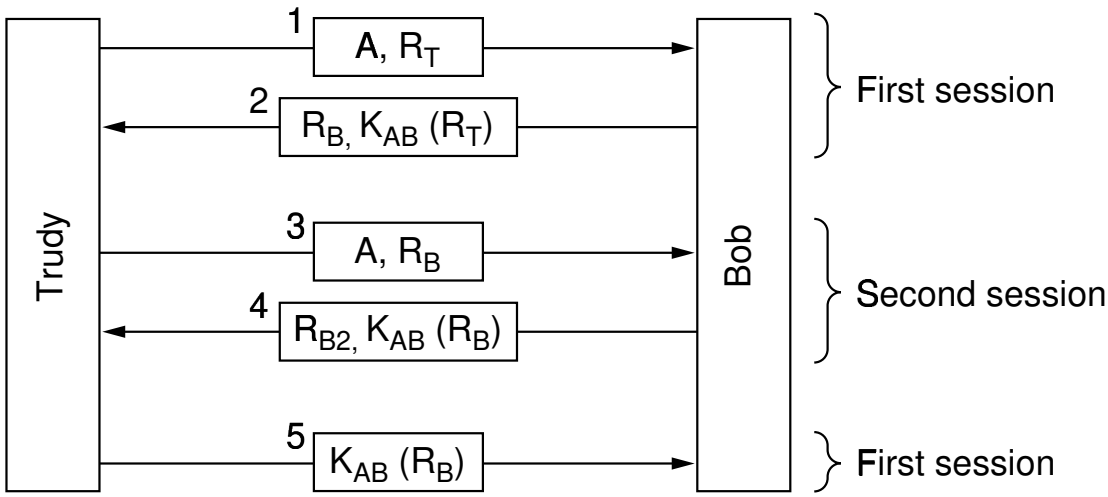


Fig. 7-14. The reflection attack.

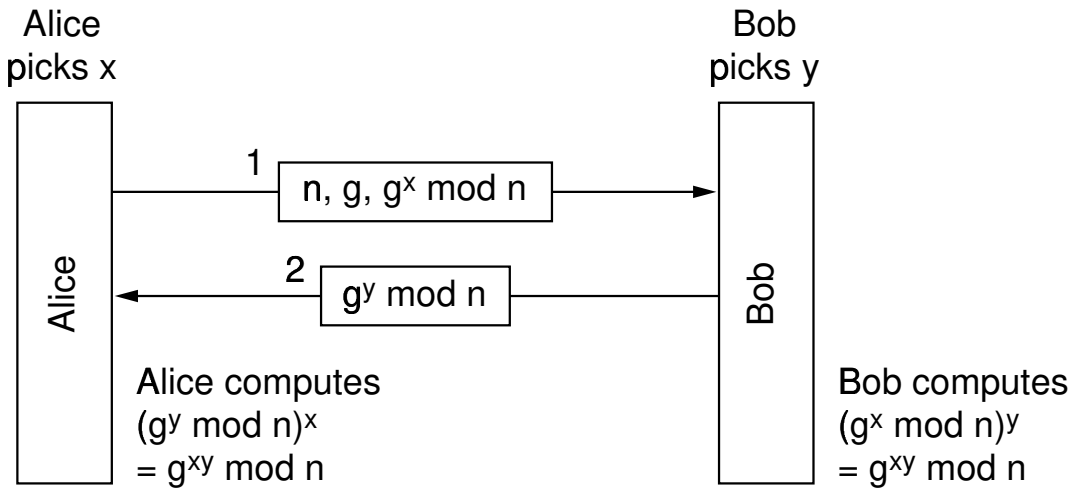


Fig. 7-15. The Diffie-Hellman key exchange.

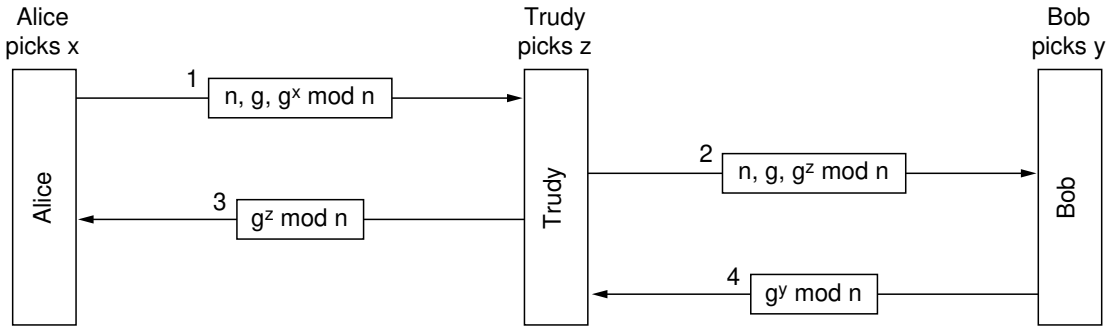


Fig. 7-16. The bucket brigade attack.

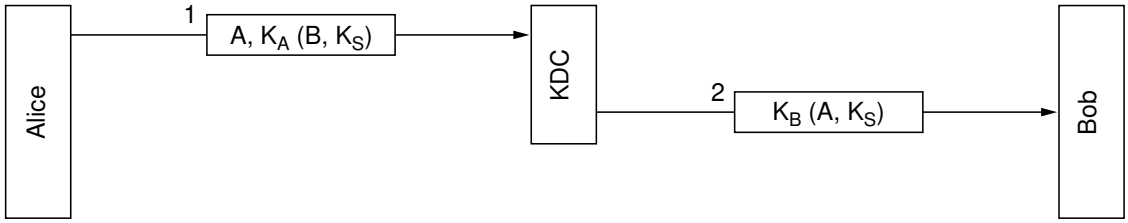


Fig. 7-17. A first attempt at an authentication protocol using a KDC.

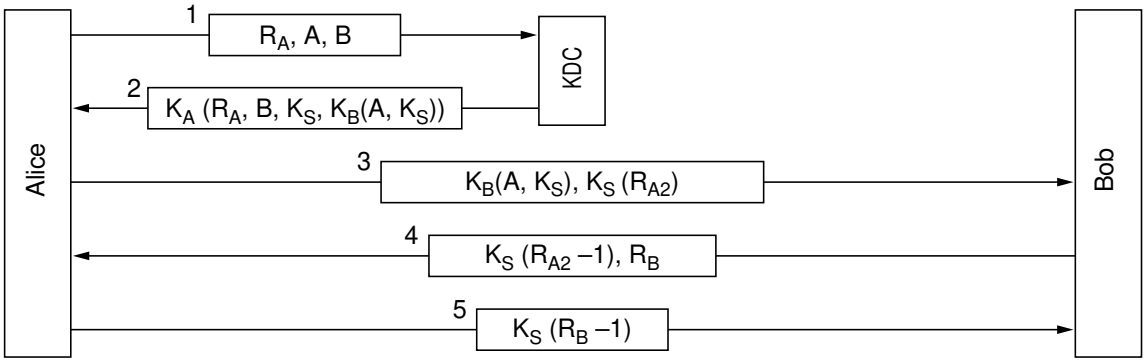


Fig. 7-18. The Needham-Schroeder authentication protocol.

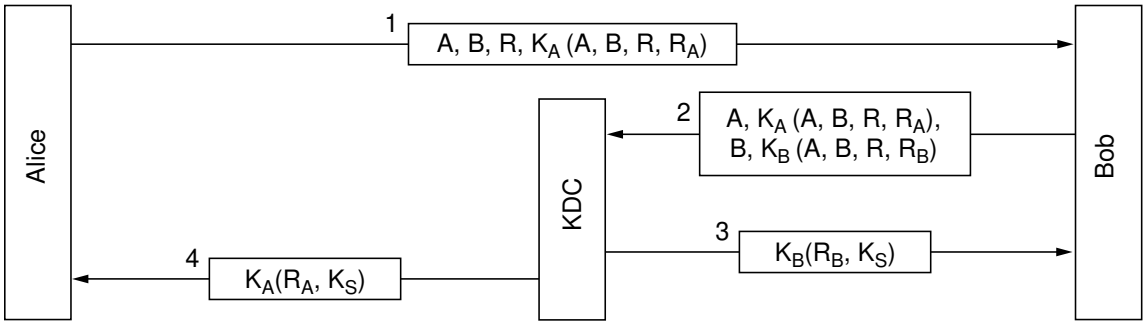


Fig. 7-19. The Otway-Rees authentication protocol (slightly simplified).

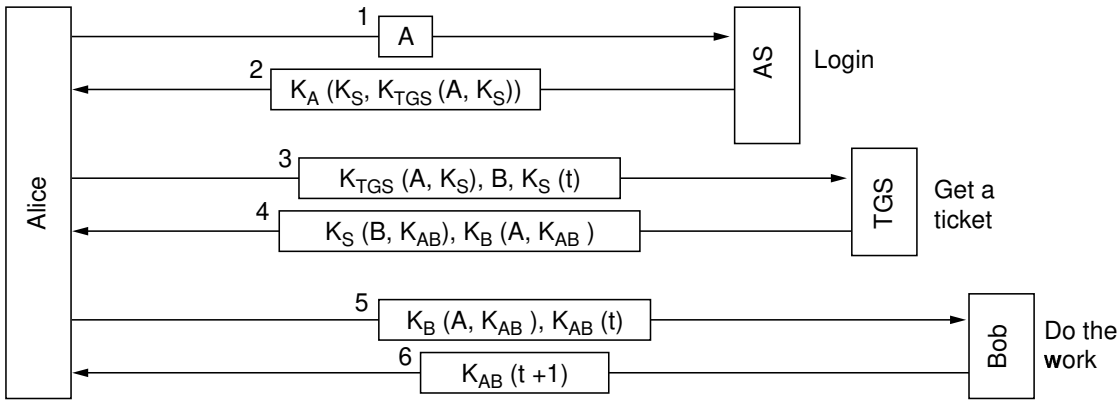


Fig. 7-20. The operation of Kerberos V4.

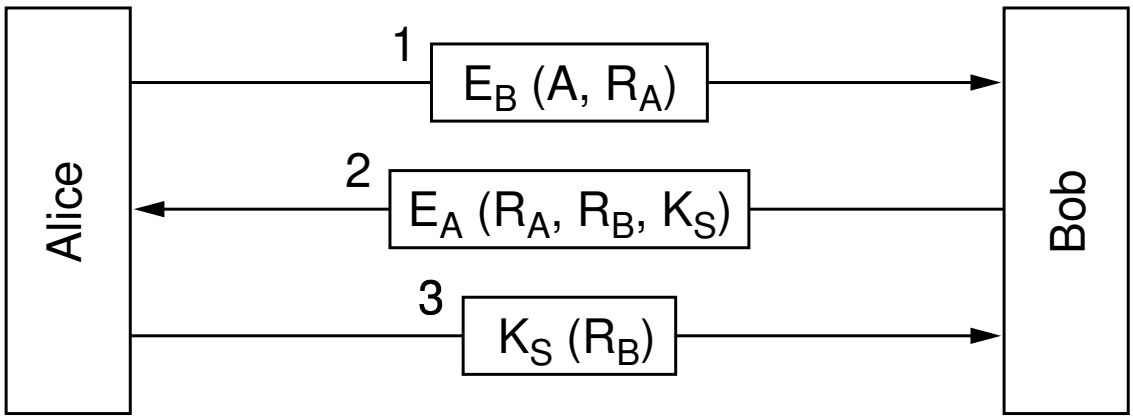


Fig. 7-21. Mutual authentication using public-key cryptography.

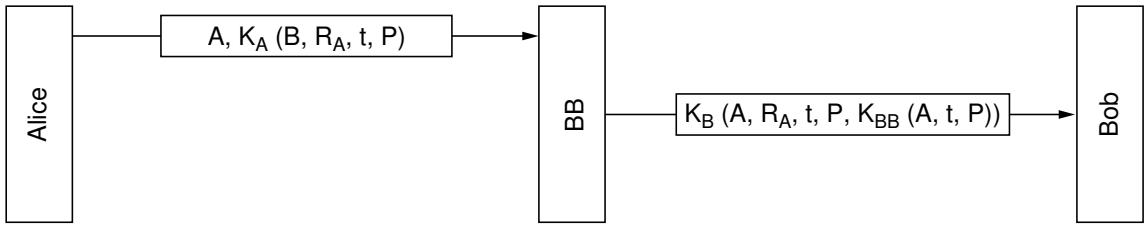


Fig. 7-22. Digital signatures with Big Brother.

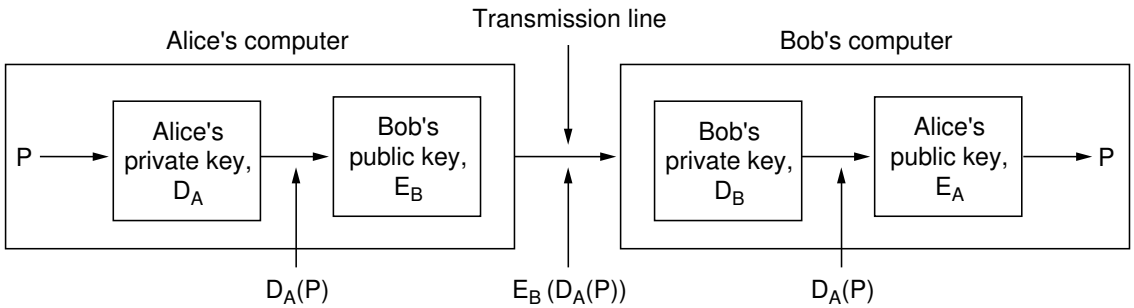


Fig. 7-23. Digital signatures using public-key cryptography.

Fig. 7-24. Digital signatures using message digests.

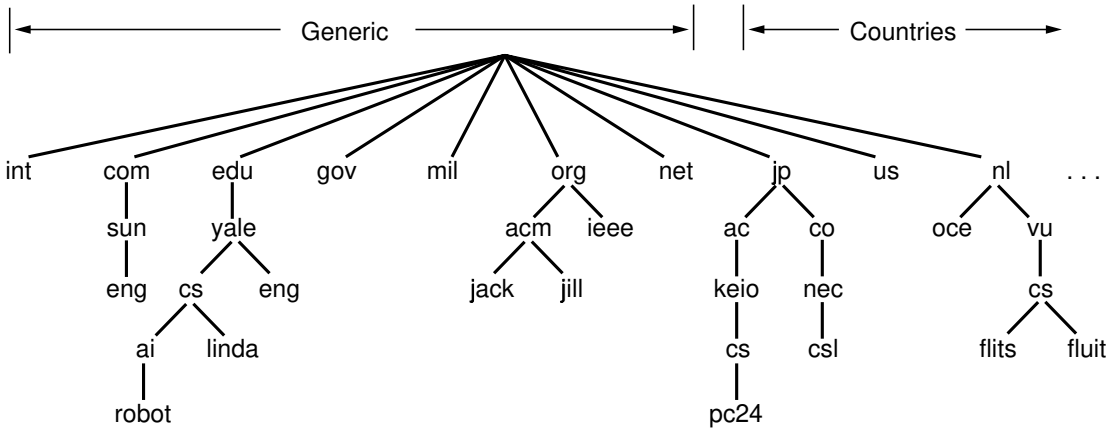


Fig. 7-25. A portion of the Internet domain name space.

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 email
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

Fig. 7-26. The principal DNS resource record types.

```
; 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    "Faculteit 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
```

Fig. 7-27. A portion of a possible DNS database for *cs.vu.nl*

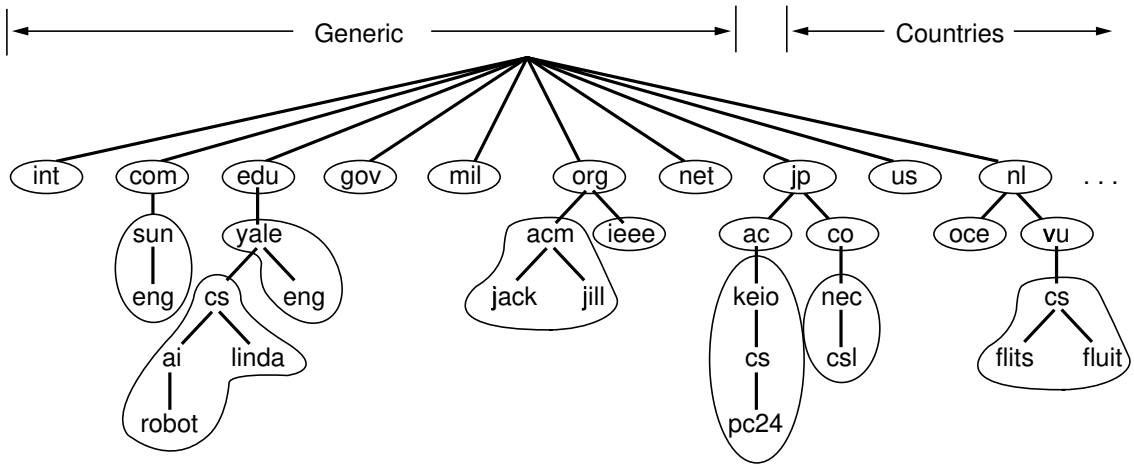


Fig. 7-28. Part of the DNS name space showing the division into zones.

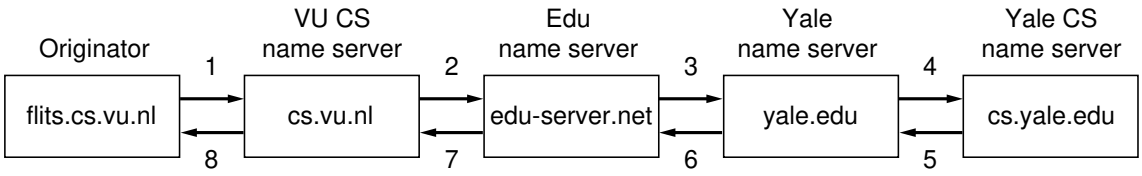


Fig. 7-29. How a resolver looks up a remote name in eight steps.

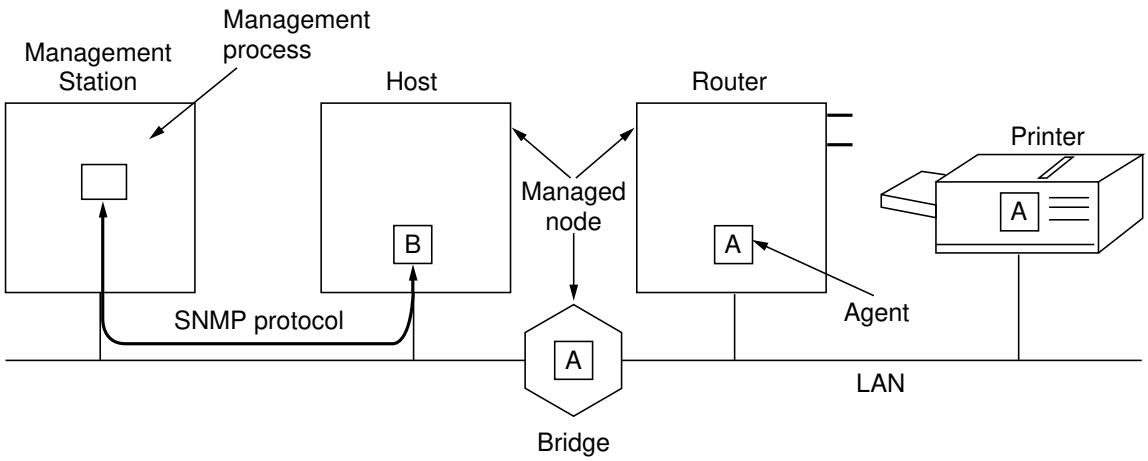


Fig. 7-30. Components of the SNMP management model.

Primitive type	Meaning	Code
INTEGER	Arbitrary length integer	2
BIT STRING	A string of 0 or more bits	3
OCTET STRING	A string of 0 of more unsigned bytes	4
NULL	A place holder	5
OBJECT IDENTIFIER	An officially defined data type	6

Fig. 7-31. The ASN.1 primitive data types permitted in SNMP.

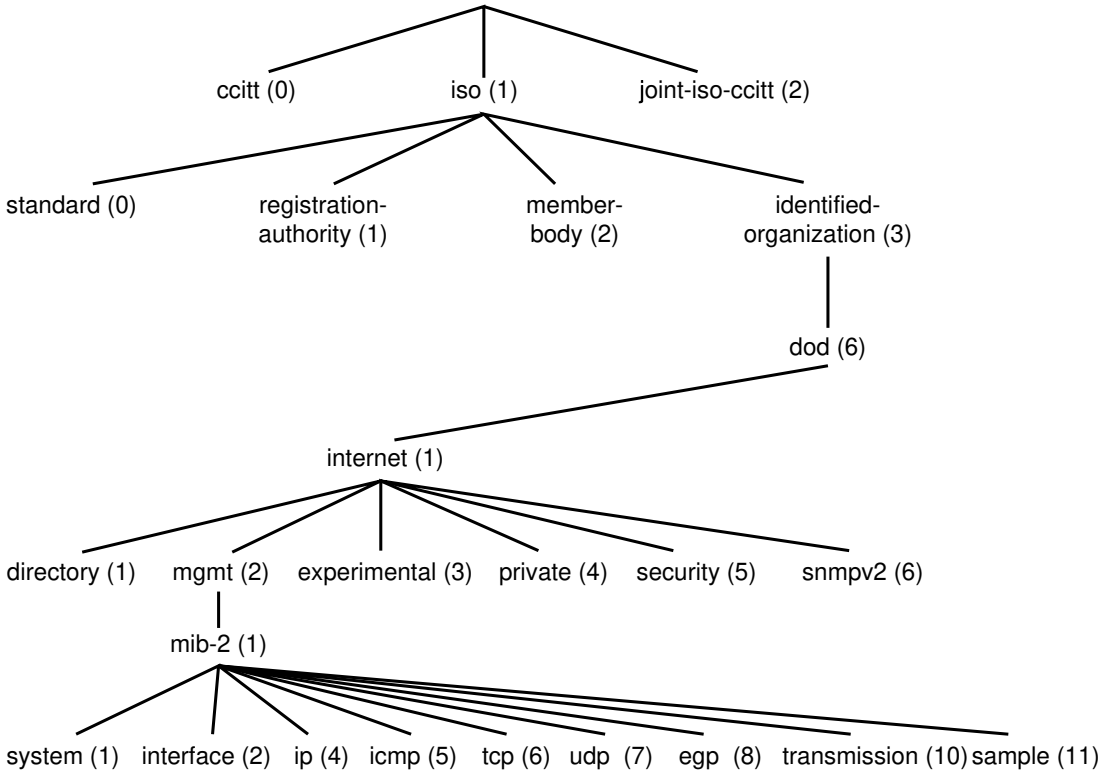


Fig. 7-32. Part of the ASN.1 object naming tree.

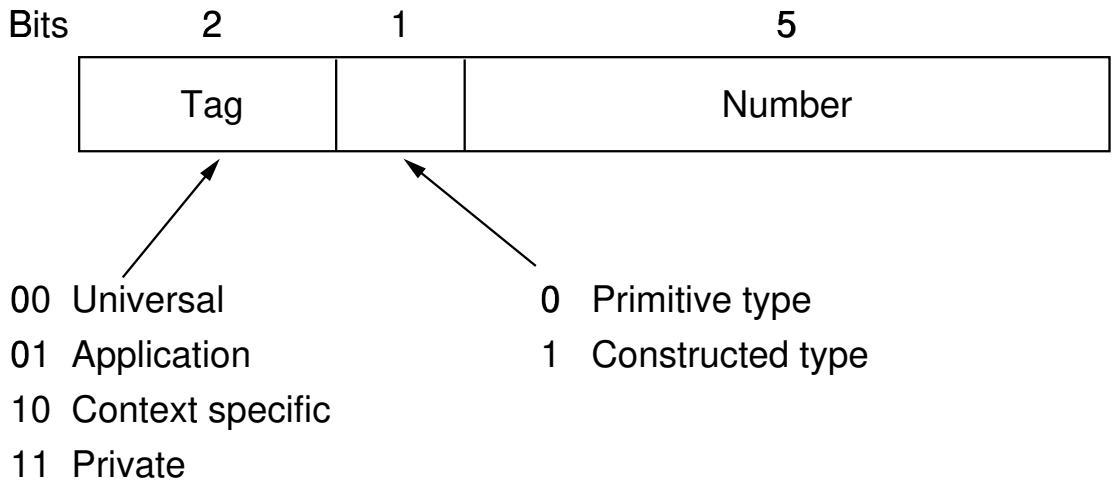


Fig. 7-33. The first byte of each data item sent in the ASN.1 transfer syntax.

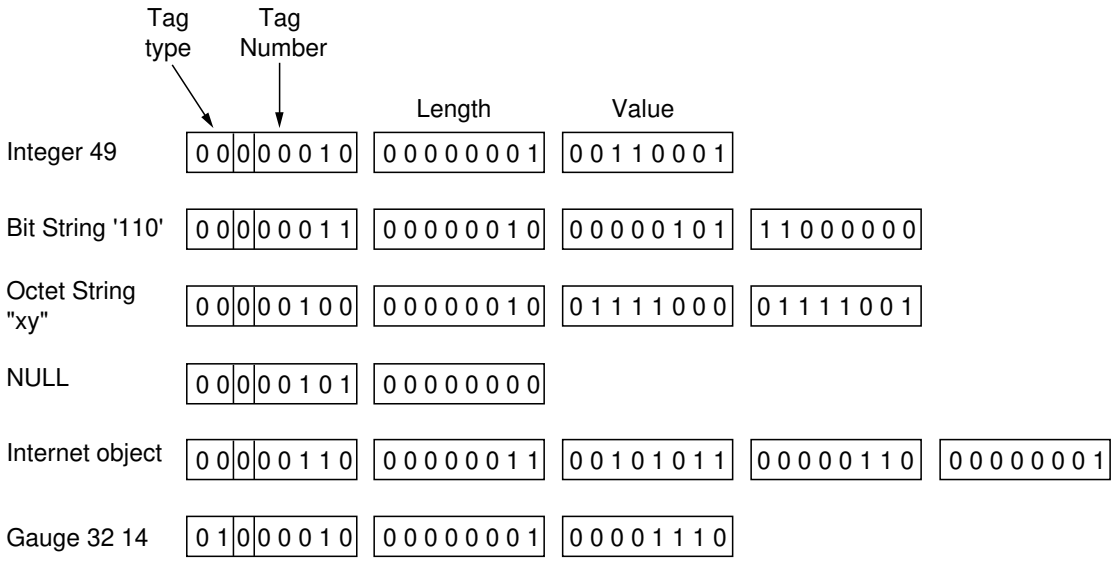


Fig. 7-34. ASN.1 encoding of some example values.

Name	Type	Bytes	Meaning
INTEGER	Numeric	4	Integer (32 bits in current implementations)
Counter32	Numeric	4	Unsigned 32-bit counter that wraps
Gauge32	Numeric	4	Unsigned value that does not wrap
Integer32	Numeric	4	32 Bits, even on a 64-bit CPU
UInteger32	Numeric	4	Like Integer32, but unsigned
Counter64	Numeric	8	A 64-bit counter
TimeTicks	Numeric	4	In hundredths of a second since some epoch
BIT STRING	String	4	Bit map of 1 to 32 bits
OCTET STRING	String	≥ 0	Variable length byte string
Opaque	String	≥ 0	Obsolete; for backward compatibility only
OBJECT IDENTIFIER	String	>0	A list of integers from Fig. 7-0
IpAddress	String	4	A dotted decimal Internet address
NsapAddress	String	< 22	An OSI NSAP address

Fig. 7-35. Data types used for SNMP monitored variables.

```
lostPackets OBJECT TYPE
    SYNTAX Counter32-- use a 32-bit counter
    MAX-ACCESS read-only-- the management station may not change
    STATUS current    -- this variable is not obsolete (yet)
    DESCRIPTION
        "The number of packets lost since the last boot"
    ::= {experimental 20}
```

Fig. 7-36. An example SNMP variable.

Group	# Objects	Description
System	7	Name, location, and description of the equipment
Interfaces	23	Network interfaces and their measured traffic
AT	3	Address translation (deprecated)
IP	42	IP packet statistics
ICMP	26	Statistics about ICMP messages received
TCP	19	TCP algorithms, parameters, and statistics
UDP	6	UDP traffic statistics
EGP	20	Exterior gateway protocol traffic statistics
Transmission	0	Reserved for media-specific MIBs
SNMP	29	SNMP traffic statistics

Fig. 7-37. The object groups of the Internet MIB-II.

Message	Description
Get-request	Requests the value of one or more variables
Get-next-request	Requests the variable following this one
Get-bulk-request	Fetches a large table
Set-request	Updates one or more variables
Inform-request	Manager-to-manager message describing local MIB
SnmpV2-trap	Agent-to-manager trap report

Fig. 7-38. SNMP message types.

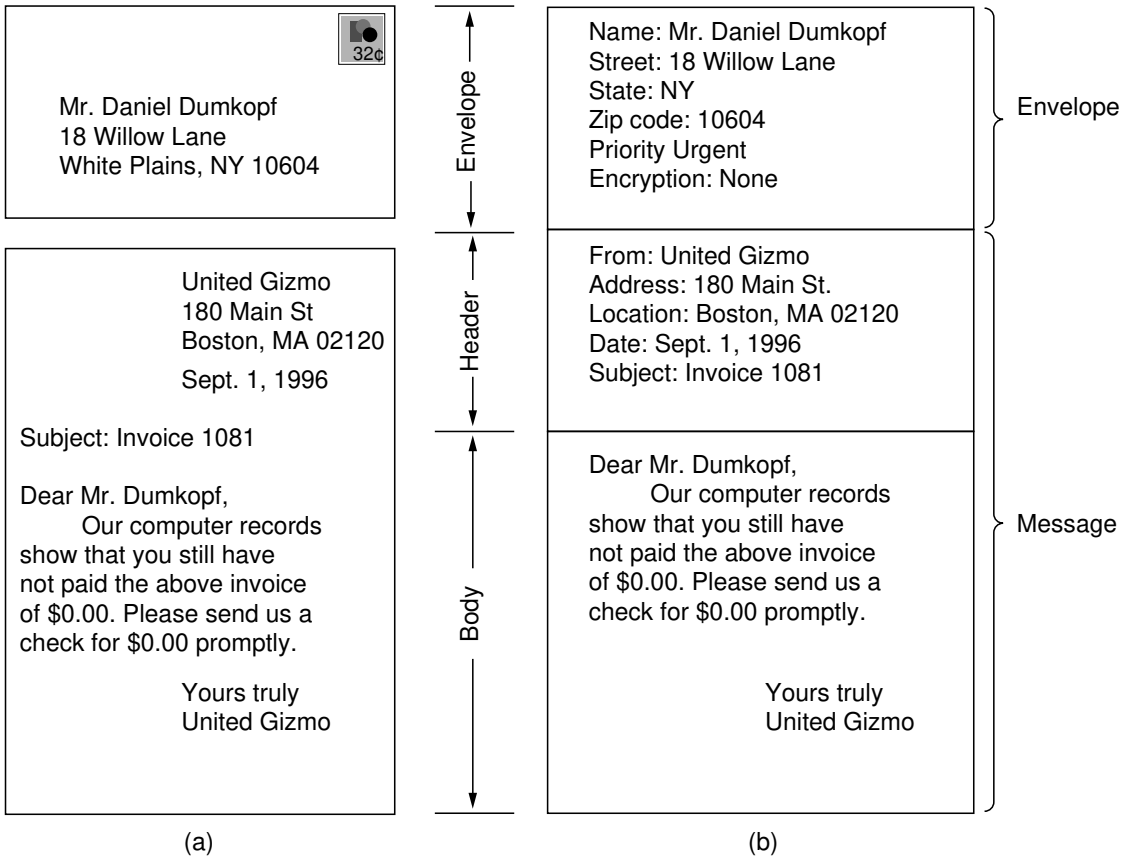


Fig. 7-39. Envelopes and messages. (a) Postal email. (b) Electronic mail.

#	Flags	Bytes	Sender	Subject
1	K	1030	asw	Changes to MINIX
2	KA	6348	radia	Comments on material you sent me
3	K F	4519	Amy N. Wong	Request for information
4		1236	bal	Deadline for grant proposal
5		103610	kaashoek	Text of DCS paper
6		1223	emily E.	Pointer to WWW page
7		3110	saniya	Referee reports for the paper
8		1204	dmr	Re: My student's visit

Fig. 7-40. An example display of the contents of a mailbox.

Command	Parameter	Description
h	#	Display header(s) on the screen
c		Display current header only
t	#	Type message(s) on the screen
s	address	Send a message
f	#	Forward message(s)
a	#	Answer message(s)
d	#	Delete message(s)
u	#	Undelete previously deleted message(s)
m	#	Move message(s) to another mailbox
k	#	Keep message(s) after exiting
r	mailbox	Read a new mailbox
n		Go to the next message and display it
b		Backup to the previous message and display it
g	#	Go to a specific message but do not display it
e		Exit the mail system and update the mailbox

Fig. 7-41. Typical mail handling commands.

Header	Meaning
To:	Email address(es) of primary recipient(s)
Cc:	Email address(es) of secondary recipient(s)
Bcc:	Email address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	Email 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

Fig. 7-42. RFC 822 header fields related to message transport.

Header	Meaning
Date:	The date and time the message was sent
Reply-To:	Email 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

Fig. 7-43. Some fields used in the RFC 822 message header.

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:	Nature of the message

Fig. 7-44. RFC 822 headers added by MIME.

Type	Subtype	Description
Text	Plain	Unformatted text
	Richtext	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

Fig. 7-45. The MIME types and subtypes defined in RFC 1521.

```
From: elinor@abc.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abc.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvb
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/richtext

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.abc.com";
        directory="pub";
        name="birthday.snd"

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

Fig. 7-46. A multipart message containing richtext and audio alternatives.

```
                    S: 220 xyz.com SMTP service ready
C: HELO abc.com
                    S: 250 xyz.com says hello to abc.com
C: MAIL FROM: <elinor@abc.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@abc.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abc.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/richtext
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.abc.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
```

Fig. 7-47. Transferring a message from *elinor@abc.com* to *carolyn@xyz.com*.

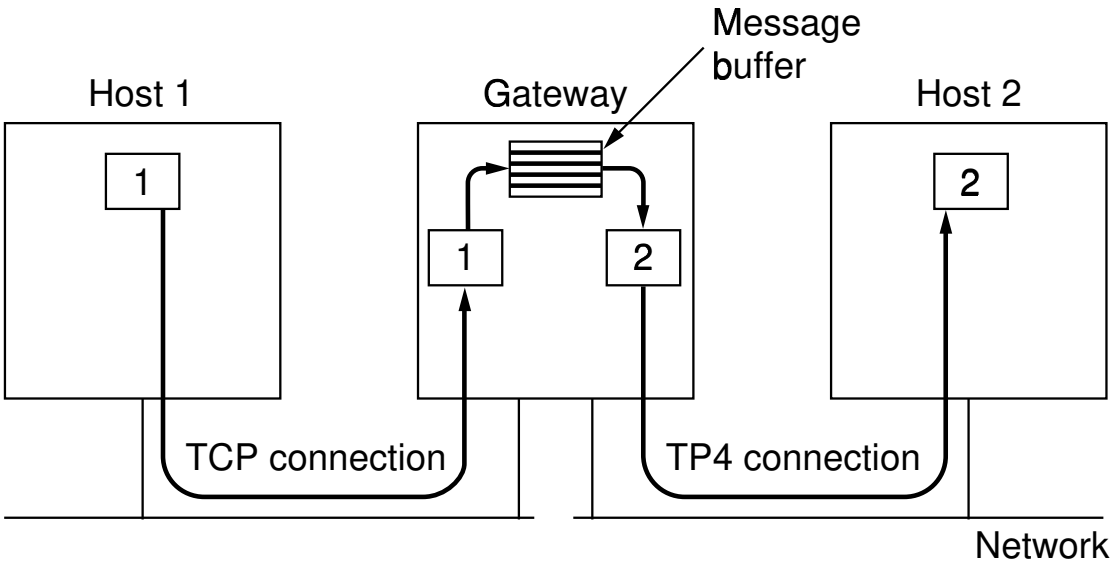


Fig. 7-48. Transferring email using an application layer email gateway.

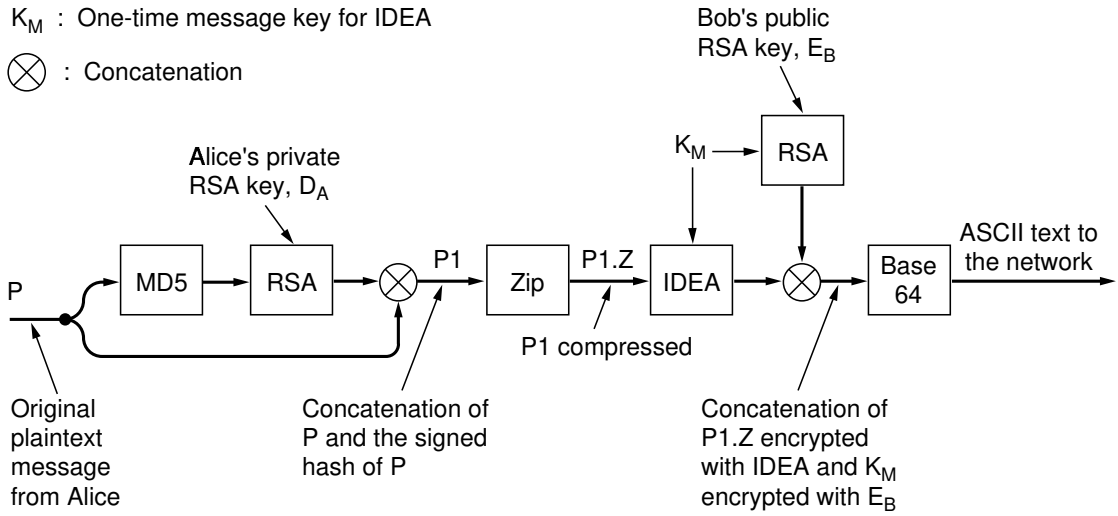


Fig. 7-49. PGP in operation for sending a message.

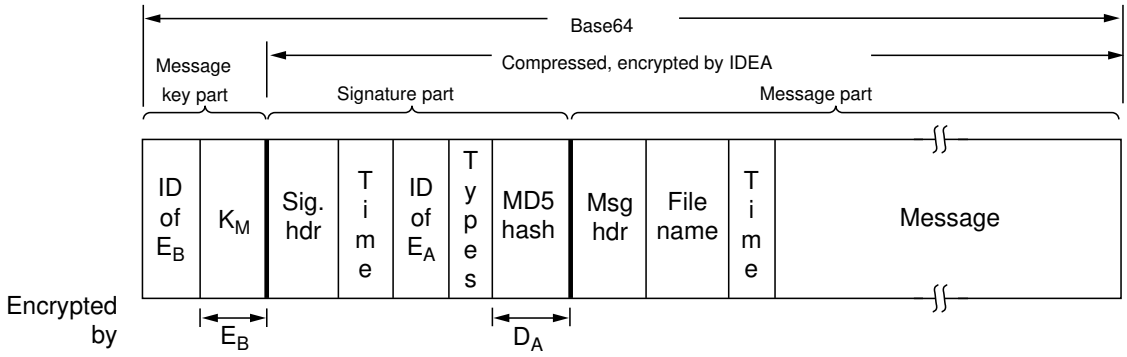


Fig. 7-50. A PGP message.

Item	PGP	PEM
Supports encryption?	Yes	Yes
Supports authentication?	Yes	Yes
Supports nonrepudiation?	Yes	Yes
Supports compression?	Yes	No
Supports canonicalization?	No	Yes
Supports mailing lists?	No	Yes
Uses base64 coding?	Yes	Yes
Current data encryption algorithm	IDEA	DES
Key length for data encryption (bits)	128	56
Current algorithm for key management	RSA	RSA or DES
Key length for key management (bits)	384/512/1024	Variable
User name space	User defined	X.400
X.509 conformant?	No	Yes
Do you have to trust anyone?	No	Yes (IPRA)
Key certification	Ad hoc	IPRA/PCA/CA hierarchy
Key revocation	Haphazard	Better
Can eavesdroppers read messages?	No	No
Can eavesdroppers read signatures?	No	Yes
Internet Standard?	No	Yes
Designed by	Small team	Standards committee

Fig. 7-51. A comparison of PGP and PEM.

Name	Topics covered
Comp	Computers, computer science, and the computer industry
Sci	The physical sciences and engineering
Humanities	Literature and the humanities
News	Discussion of USENET itself
Rec	Recreational activities, including sports and music
Misc	Everything that does not fit in somewhere else
Soc	Socializing and social issues
Talk	Diatribes, polemics, debates and arguments galore
Alt	Alternative tree covering virtually everything

Fig. 7-52. USENET hierarchies in order of decreasing signal-to-noise ratio.

Name	Topics covered
Comp.ai	Artificial intelligence
Comp.databases	Design and implementation of database systems
Comp.lang.c	The C programming language
Comp.os.minix	Tanenbaum's educational MINIX operating system
Comp.os.ms-windows.video	Video hardware and software for Windows
Sci.bio.entomology.lepidoptera	Research on butterflies and moths
Sci.geo.earthquakes	Geology, seismology, and earthquakes
Sci.med.orthopedics	Orthopedic surgery
Humanities.lit.authors.shakespeare	Shakespeare's plays and poetry
News.groups	Potential new newsgroups
News.lists	Lists relating to USENET
Rec.arts.poems	Free poetry
Rec.food.chocolate	Yum yum
Rec.humor.funny	Did you hear the joke about the farmer who ...
Rec.music.folk	Folks discussing folk music
Misc.jobs.offered	Announcements of positions available
Misc.health.diabetes	Day-to-day living with diabetes
Soc.culture.estonia	Life and culture in Estonia
Soc.singles	Single people and their interests
Soc.couples	Graduates of soc.singles
Talk.abortion	No signal, all noise
Talk.rumors	This is where rumors come from
Alt.alien.visitors	Place to report flying saucer rides
Alt.bermuda.triangle	If you read this, you vanish mysteriously
Alt.sex.voyeurism	Take a peek and see for yourself
Alt.tv.simpsons	Bart et al.

Fig. 7-53. A small selection of the newsgroups.

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

Fig. 7-54. Some smileys.

From: Vogel@nyu.edu
Message-Id: <54731@nyu.edu>
Subject: Bird Sighting
Path: cs.vu.nl!sun4nl!EU.net!news.sprintlink.net!in2.uu.net!pc144.nyu.edu
Newsgroups: rec.birds
Followup-To: rec.birds
Distribution: world
Nntp-Posting-host: nuthatch.bio.nyu.edu
References:
Organization: New York University
Lines: 4
Summary: Guess what I saw

I just saw an ostrich on 52nd St. and Fifth Ave. in New York. Is this their n
season? Did anybody else see it?

Jay Vogel

Fig. 7-55. A sample news article.

Command	Meaning
LIST	Give me a list of all newsgroups and articles you have
NEWGROUPS date time	Give me a list of newsgroups created after date/time
GROUP grp	Give me a list of all articles in grp
NEWNEWS grps date time	Give me a list of new articles in specified groups
ARTICLE id	Give me a specific article
POST	I have an article for you that was posted here
IHAVE id	I have article id. Do you want it?
QUIT	Terminate the session

Fig. 7-56. The principal NNTP commands for news diffusion.

```
      S: 200 feeder.com NNTP server at your service   (response to new connection)
C: NEWNEWS soc.couples 960901 030000  (any new news in soc.couples?)
      S: 230 List of 2 articles follows
      S: <13281@psyc.berkeley.edu>        (article 1 of 2 in soc.couples is from Berkeley)
      S: <162721@aol.com>                 (article 2 of 2 in soc.couples is from AOL)
      S: .                                (end of list)
C: ARTICLE <13281@psyc.berkeley.edu>   (please give me the Berkeley article)
      S: 220 <13281@psyc.berkeley.edu> follows
      S: (entire article <13281@psyc.berkeley.edu> is sent here)
      S: .                                (end of article)
C: ARTICLE <162721@aol.com>            (please give me the AOL article)
      S: 220  <162721@aol.com> follows
      S: (entire article <162721@aol.com> is sent here)
      S: .                                (end of article)
C: NEWNEWS misc.kids 960901 030000     (any new news in misc.kids?)
      S: 230 List of 1 article follows
      S: <43222@bio.rice.edu>             (1 article from Rice)
      S: .                                (end of list)
C: ARTICLE <43222@bio.rice.edu>        (please give me the Rice article)
      S: 220  <43222@bio.rice.edu> follows
      S: (entire article <43222@bio.rice.edu> is sent here)
      S: .                                (end of article)
C: NEWGROUPS 960901 030000
      S: 231 2 new groups follow
      S: rec.pets
      S: rec.nude
      S: .
C: NEWNEWS rec.pets 0 0                (list everything you have)
      S: 230 List of 1 article follows
      S: <124@fido.net>                   (1 article from fido.net)
      S: .                                (end of list)
C: ARTICLE <124@fido.net>              (please give me the fido.net article)
      S: 220 <124@fido.net> follows
      S: (entire article is sent here)
      S: .
C: POST
      S: 340                              (please send your posting)
C:    (article posted on wholesome.com sent here)
      S: 240                              (article received)
C: IHAVE <5321@foo.com>
      S: 435                              (I already have it, please do not send it)
C: QUIT
      S: 205                              (Have a nice day)
```

Fig. 7-57. How *wholesome.com* might acquire news articles from its newsfeed.

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)

Fig. 7-58. (a) A Web page. (b) The page can be reached by clicking on Department of Animal Psychology

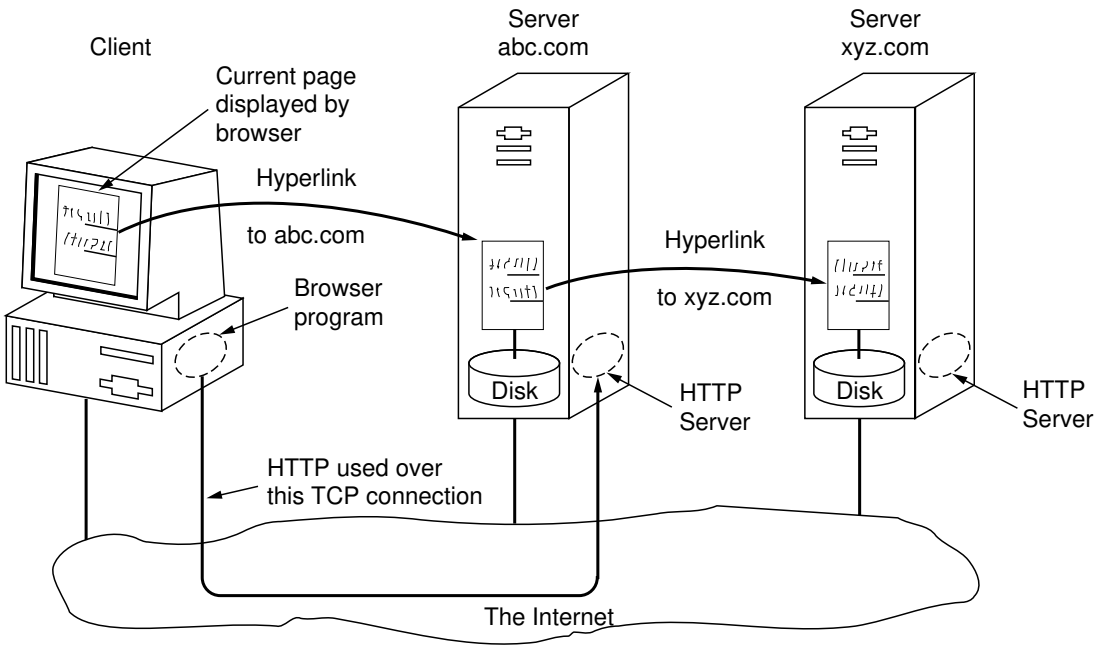


Fig. 7-59. The parts of the Web model.

```
C: telnet  www.w3.org  80
T: Trying 18.23.0.23 ...
T: Connected to www.w3.org.
T: Escape character is '^]'.
C: GET  /hypertext/WWW/TheProject.html  HTTP/1.0
C:
      S: HTTP/1.0 200 Document follows
      S: MIME-Version: 1.0
      S: Server: CERN/3.0
      S: Content-Type: text/html
      S: Content-Length: 8247
      S:
      S: <HEAD> <TITLE> The World Wide Web Consortium (W3C) </TITLE> </HEAD>
      S: <BODY>
      S: <H1> <IMG ALIGN=MIDDLE ALT="W3C" SRC="Icons/WWW/w3c_96x67.gif">
      S: The World Wide Web Consortium </H1> <P>
      S:
      S: The World Wide Web is the universe of network-accessible information.
      S: The <A HREF="Consortium/"> World Wide Web Consortium </A>
      S: exists to realize the full potential of the Web. <P>
      S:
      S: W3C works with the global community to produce
      S: <A HREF="#Specifications"> specifications </A> and
      S: <A HREF="#Reference"> reference software </A> .
      S: W3C is funded by industrial
      S: <A HREF="Consortium/Member/List.html"> members </A>
      S: but its products are freely available to all. <P>
      S:
      S: In this document:
      S: <menu>
      S: <LI> <A HREF="#Specifications"> Web Specifications and Development Areas </A>
      S: <LI> <A HREF="#Reference"> Web Software </A>
      S: <LI> <A HREF="#Community"> The World Wide Web and the Web Community </A>
      S: <LI> <A HREF="#Joining"> Getting involved with the W3C </A>
      S: </menu>
      S: <P> <HR>
      S: <P> W3C is hosted by the
      S: <A HREF="http://www.lcs.mit.edu/"> Laboratory for Computer Science </A> at
      S: <A HREF="http://web.mit.edu/"> MIT </A> , and
      S: in Europe by <A HREF="http://www.inria.fr/"> INRIA </A> .
      S: </BODY>
```

Fig. 7-60. A sample scenario for obtaining a Web page.

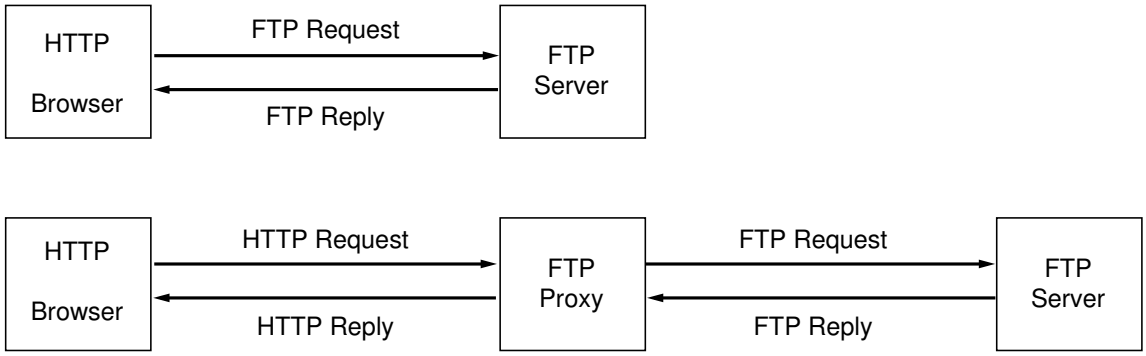


Fig. 7-61. (a) A browser that speaks FTP. (b) A browser that does not.

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
LINK	Connects two existing resources
UNLINK	Breaks an existing connection between two resources

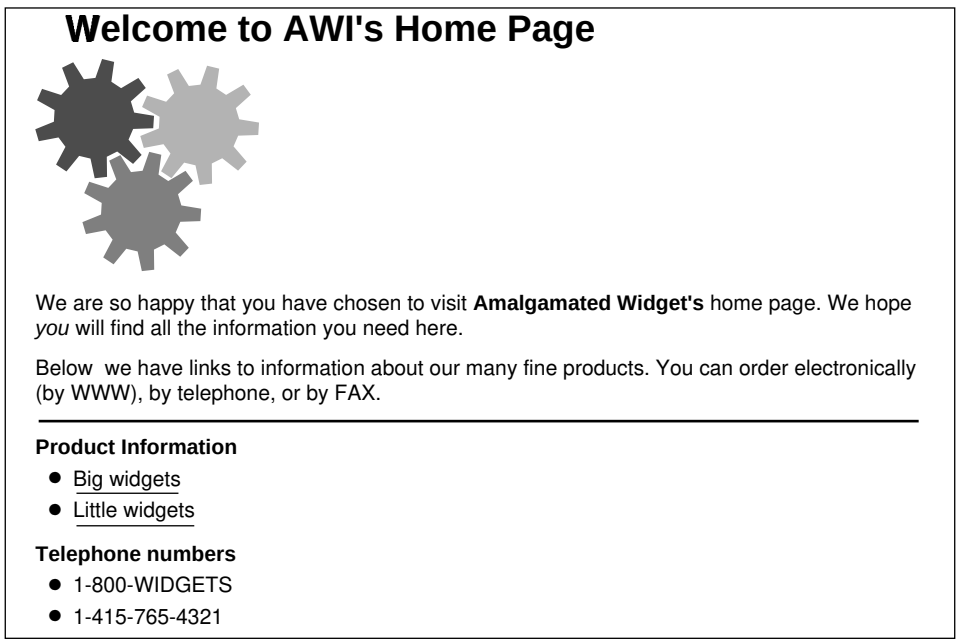
Fig. 7-62. The built-in HTTP request methods.

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	/usr/suzanne/prog.c
news	News group	news:comp.os.minix
news	News article	news:AA0134223112@cs.utah.edu
gopher	Gopher	gopher://gopher.tc.umn.edu/11/Libraries
mailto	Sending email	mailto:kim@acm.org
telnet	Remote login	telnet://www.w3.org:80

Fig. 7-63. Some common URLs.

```
<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. <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)

Fig. 7-64. (a) The HTML for a sample Web page. (b) The formatted page.

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
<B> ... </B>	Set ... in boldface
<I> ... </I>	Set ... in italics
<UL> ... </UL>	Brackets an unordered (bulleted) list
<OL> ... </OL>	Brackets a numbered list
<MENU> ... </MENU>	Brackets a menu of <LI> items
<LI>	Start of a list item (there is no </LI>)
 	Force a break here
<P>	Start of paragraph
<HR>	Horizontal rule
<PRE> ... </PRE>	Preformatted text; do not reformat
<IMG SRC="...">	Load an image here
<A HREF="..."> ... </A>	Defines a hyperlink

Fig. 7-65. A selection of common HTML tags. Some have additional parameters.

```
<HTML> <HEAD> <TITLE> A sample page with a table </TITLE> </HEAD>
<BODY>
<TABLE BORDER=ALL RULES=ALL>
<CAPTION> Some Differences between HTML Versions </CAPTION>
<COL ALIGN=LEFT>
<COL ALIGN=CENTER>
<COL ALIGN=CENTER>
<COL ALIGN=CENTER>
<TR> <TH>Item  <TH>HTML 1.0  <TH>HTML 2.0  <TH>HTML 3.0
<TR> <TH> Active Maps and Images <TD> <TD> x <TD> x
<TR> <TH> Equations <TD> <TD> <TD> x
<TR> <TH> Forms <TD> <TD> x <TD> x
<TR> <TH> Hyperlinks x <TD> <TD> x <TD> x
<TR> <TH> Images <TD> x <TD> x <TD> x
<TR> <TH> Lists <TD> x <TD> x <TD> x
<TR> <TH> Toolbars <TD> <TD> <TD> x
<TR> <TH> Tables <TD> <TD> <TD> x
</TABLE> </BODY> </HTML>
```

(a)

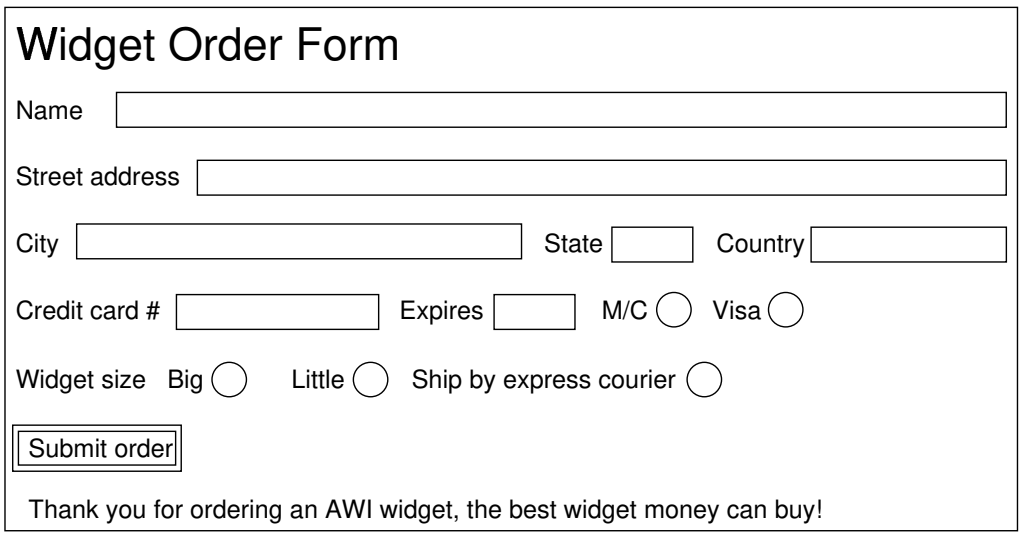
Some Differences between HTML Versions

Item	HTML 1.0	HTML 2.0	HTML 3.0
Active Maps and Images		x	x
Equations			x
Forms		x	x
Hyperlinks	x	x	x
Images	x	x	x
Lists	x	x	x
Toolbars			x
Tables			x

Fig. 7-66. (a) An HTML table. (b) A possible rendition of this table.

```
<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>
Name <INPUT NAME="customer" SIZE=46> <P>
Street Address <INPUT NAME="address" SIZE=40> <P>
City <INPUT NAME="city" SIZE=20> State <INPUT NAME="state" SIZE =4>
Country <INPUT NAME="country" SIZE=10> <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>
Widget size Big <INPUT NAME="product" TYPE=RADIO VALUE="expensive">
Little <INPUT NAME="product" TYPE=RADIO VALUE="cheap">
Ship by express courier <INPUT NAME="express" TYPE=CHECKBOX> <P>
<INPUT TYPE=SUBMIT VALUE="Submit order"> <P>
Thank you for ordering an AWI widget, the best widget money can buy!
</FORM> </BODY> </HTML>
```

(a)



(b)

Fig. 7-67. (a) The HTML for an order form. (b) The formatted page.

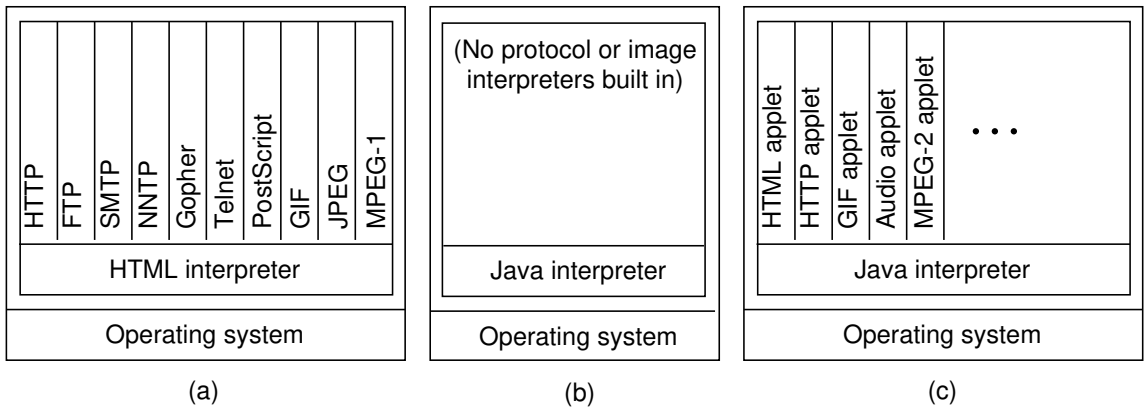


Fig. 7-68. (a) A first generation browser. (b) A Java-based browser at startup. (c) The browser of (b) after running for a while.

Type	Size	Description
Byte	1 Byte	A signed integer between −128 and +127
Short	2 Bytes	A signed 2-byte integer
Int	4 Bytes	A signed 4-byte integer
Long	8 Bytes	A signed 8-byte integer
Float	4 Bytes	A 4-byte IEEE floating-point number
Double	8 Bytes	An 8-byte IEEE floating-point number
Boolean	1 Bit	The only values are true and false
Char	2 Bytes	A character in Unicode

Fig. 7-69. The basic Java data types.

Statement	Description	Example
Assignment	Assign a value	n = i + j;
If	Boolean choice	if (k < 0) k = 0; else k = 2*k;
Switch	Select a case	switch (b) {case 1: n++; case 2: n––;}
For	Iteration	for (i = 0; i < n; i++) a[i] = b[i];
While	Repetition	while (n < k) n += i;
Do	Repetition	do {n = n + n} while (n < m);
Break	Exit statement	break label;
Return	Return	return n;
Continue	Next iteration	continue label;
Throw	Raise exception	throw new IllegalArgumentException();
Try	Exception scoping	try { ... } catch (Exception e) {return –1};
Synchronized	Mutual exclusion	synchronized void update(int s) { ... }

Fig. 7-70. The Java statements. The notation { ... } indicates a block of code.

```
class Factorial { /* This program consists of a single class with two metho

  public static void main (int argc, String args[]) {// main program
  long i, f, lower = 1, upper = 20; // declarations of four longs

  for (i = lower; i <= upper; i++) { // loop from lower to upper
        f = factorial(i);             // f = i!
        System.out.println(i + " " + f);      // print i and f
  }
}

  static long factorial (long k) {    // recursive factorial function
        if (k == 0)
            return 1;                 // 0! = 1
        else
            return k * factorial(k-1);   // k! = k * (k-1)!
  }
}
```

Fig. 7-71. A Java program for computing and printing 0! to 20!.

```
class ComplexNumber {    // Define a subclass of Object called Complexl
  // Hidden data.
  protected double re, im;   // real and imaginary parts

  // Five methods that manage the hidden data.
  public void Complex(double x, double y) {re = x; im = y;}
  public double Real() {return re;}
  public double Imaginary() {return im;}
  public double Magnitude() {return Math.sqrt(re*re + im*im);}
  public double Angle() {return Math.atan(im/re);}
}

class test {                // A second class, for testing ComplexNumber
  public static void main (String args[]) {
     ComplexNumber c;       // declare an object of class ComplexNumber

     c = new ComplexNumber();  // actually allocate storage for c
     c.Complex(3.0, 4.0);      // invoke the Complex method to initialize c
     System.out.println("The magnitude of c is " + c.Magnitude() );
  }
}
```

Fig. 7-72. A package defining two classes.

```
import ComplexNumber;    // import the ComplexNumber package


class HairyNumber extends ComplexNumber {// define a new class
  public void AddTo(ComplexNumber z) {// with one method
      re = re + z.Real();
      im = im + z.Imaginary();
  }
}


class test2 {              // test program for HairyNumber
  public static void main(String args[]) {
    HairyNumber a, h; // declare two HairyNumbers

    a = new HairyNumber();// allocate storage for a
    h = new HairyNumber();// allocate storage for h
    a.Complex(1.0, 2.0);      // assign a value to a
    h.Complex(-1.5, 4.0);     // assign a value to h
    h.AddTo(a);           // invoke the AddTo method on h
    System.out.println("h = (" + h.Real() + "," + h.Imaginary() + ")"  );
  }
}
```

Fig. 7-73. A subclass of *ComplexNumber* defining a new method.

Package	Example functionality
Java.lang	Classes, threads, exceptions, math, strings
Java.io	I/O on streams and random access files, printing
Java.net	Sockets, IP addresses, URLs, datagrams
Java.util	Stacks, hash tables, vectors, time, date
Java.applet	Getting and displaying Web pages, audio, Object class
Java.awt	Events, dialog, menus, fonts, graphics, window management
Java.awt.image	Colors, image cropping, filtering, and conversion
Java.awt.peer	Access to the underlying window system

Fig. 7-74. The packages included in the standard API.

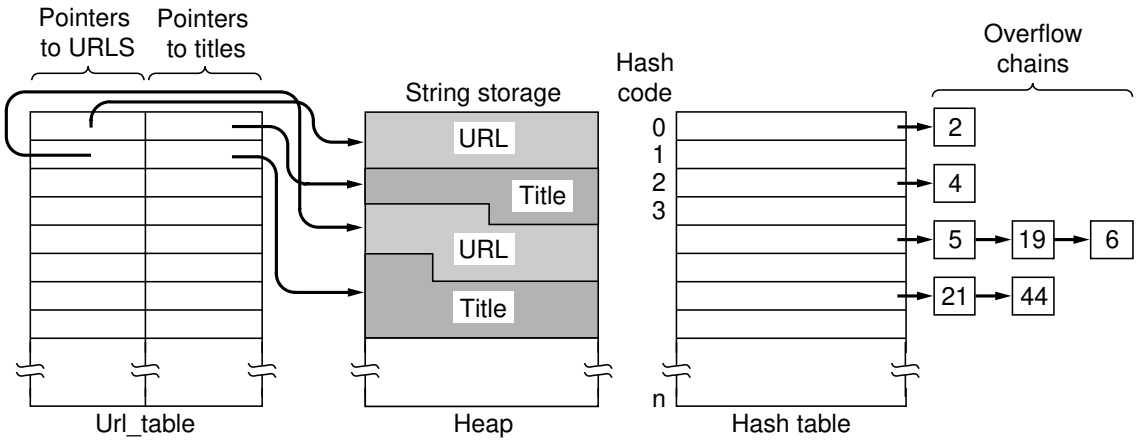


Fig. 7-75. Data structures used in a simple search engine.

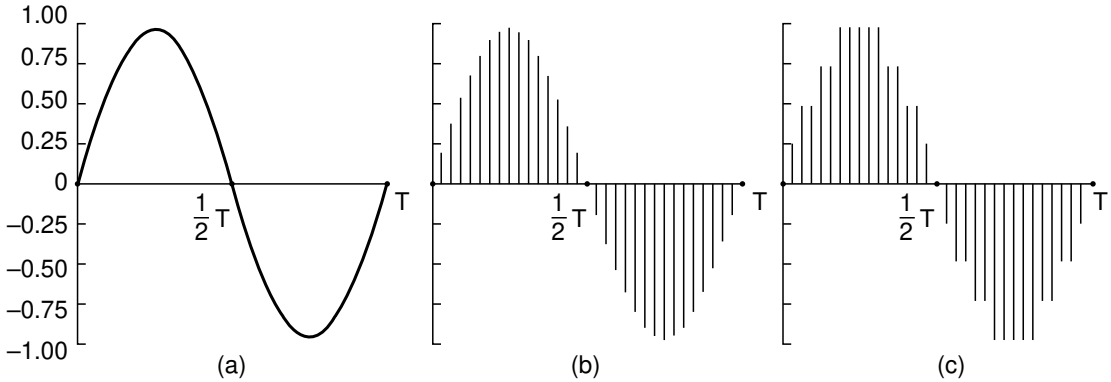


Fig. 7-76. (a) A sine wave. (b) Sampling the sine wave. (c) Quantizing the samples to 3 bits.

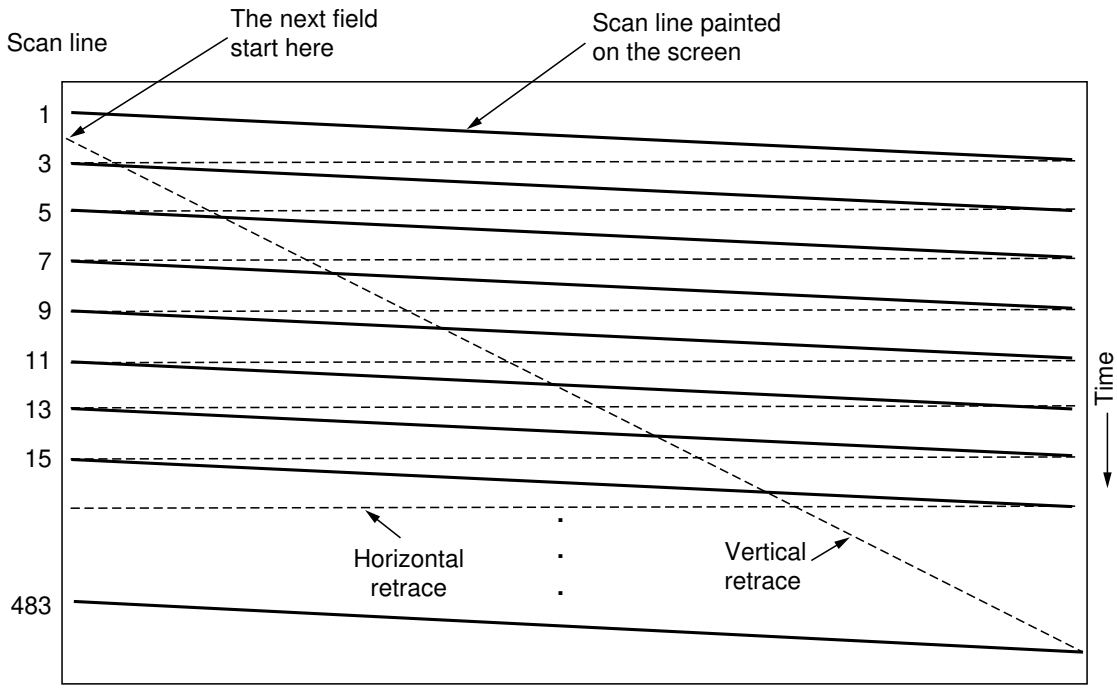


Fig. 7-77. The scanning pattern used for NTSC video and television.

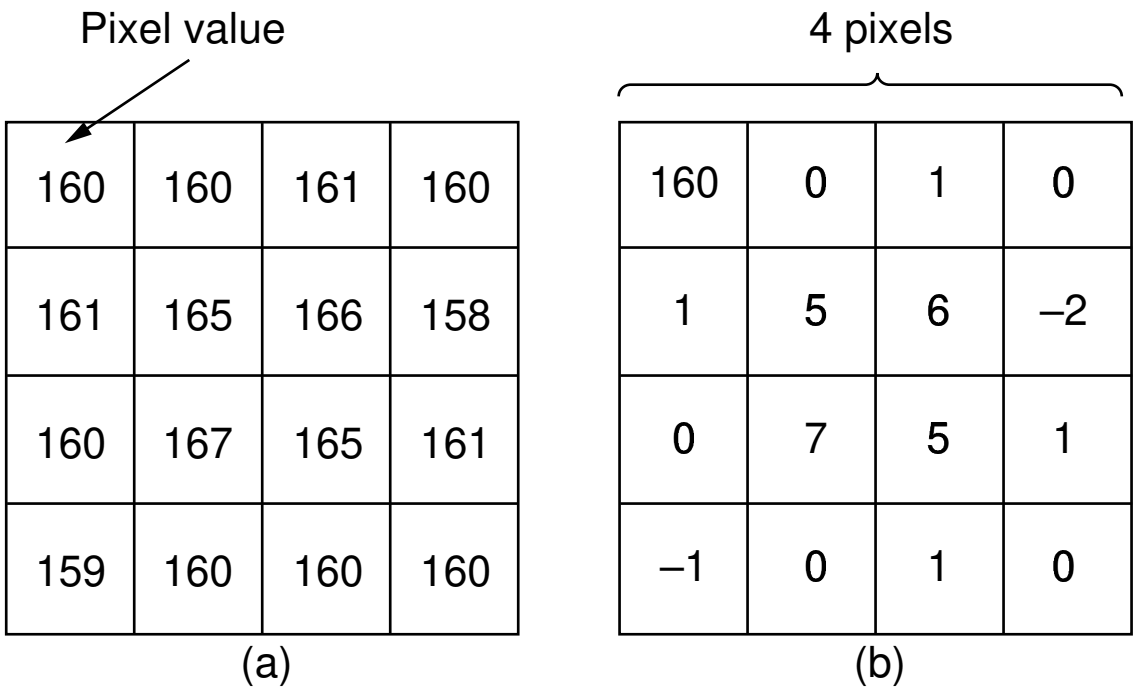


Fig. 7-78. (a) Pixel values for part of an image. (b) A transformation in which the upper left-hand element is subtracted from all elements except itself.

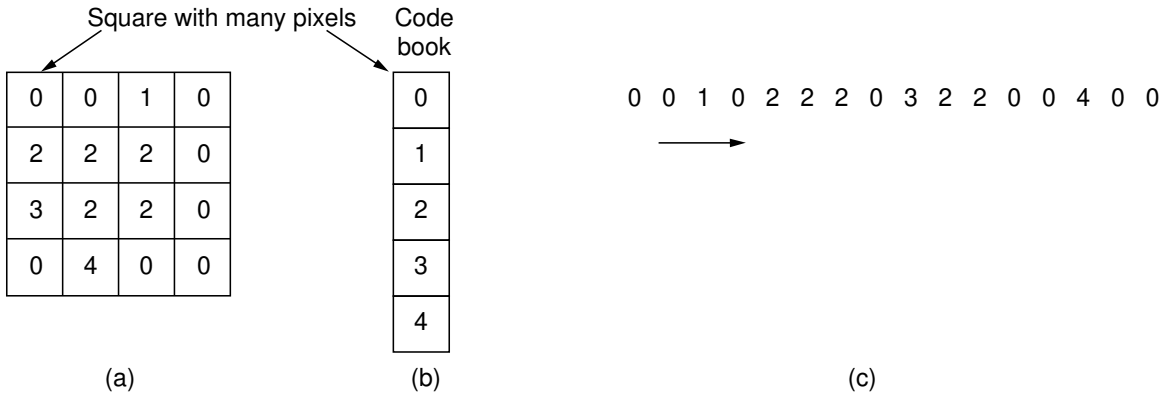


Fig. 7-79. An example of vector quantization. (a) An image divided into squares. (b) A code book for the image. (c) The encoded image.

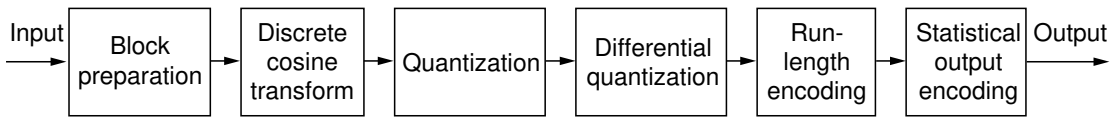


Fig. 7-80. The operation of JPEG in lossy sequential mode.

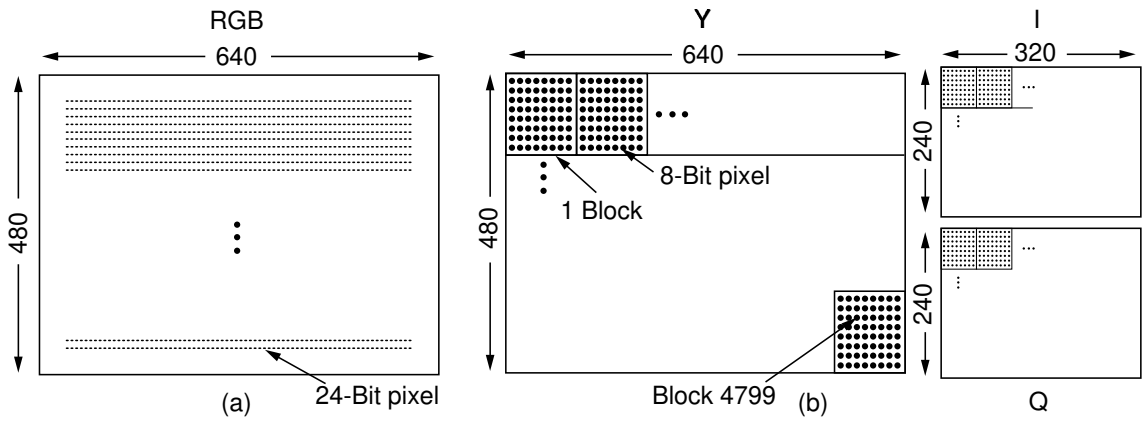


Fig. 7-81. (a) RGB input data. (b) After block preparation.

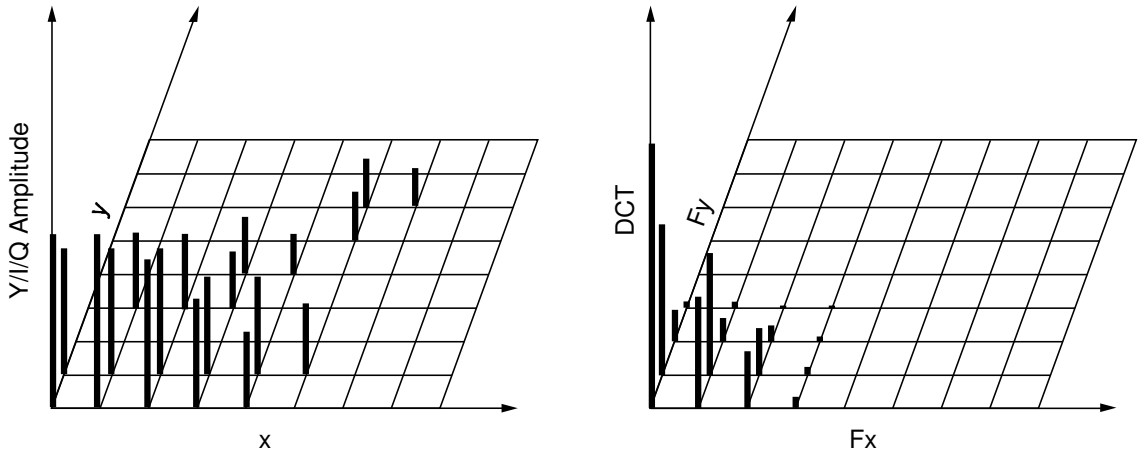


Fig. 7-82. (a) One block of the *Y* matrix. (b) The DCT coefficients.

DCT Coefficients

150	80	40	14	4	2	1	0
92	75	36	10	6	1	0	0
52	38	26	8	7	4	0	0
12	8	6	4	2	1	0	0
4	3	2	0	0	0	0	0
2	2	1	1	0	0	0	0
1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantized coefficients

150	80	20	4	1	0	0	0
92	75	18	3	1	0	0	0
26	19	13	2	1	0	0	0
3	2	2	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantization table

1	1	2	4	8	16	32	64
1	1	2	4	8	16	32	64
2	2	2	4	8	16	32	64
4	4	4	4	8	16	32	64
8	8	8	8	8	16	32	64
16	16	16	16	16	16	32	64
32	32	32	32	32	32	32	64
64	64	64	64	64	64	64	64

Fig. 7-83. Computation of the quantized DCT coefficients.

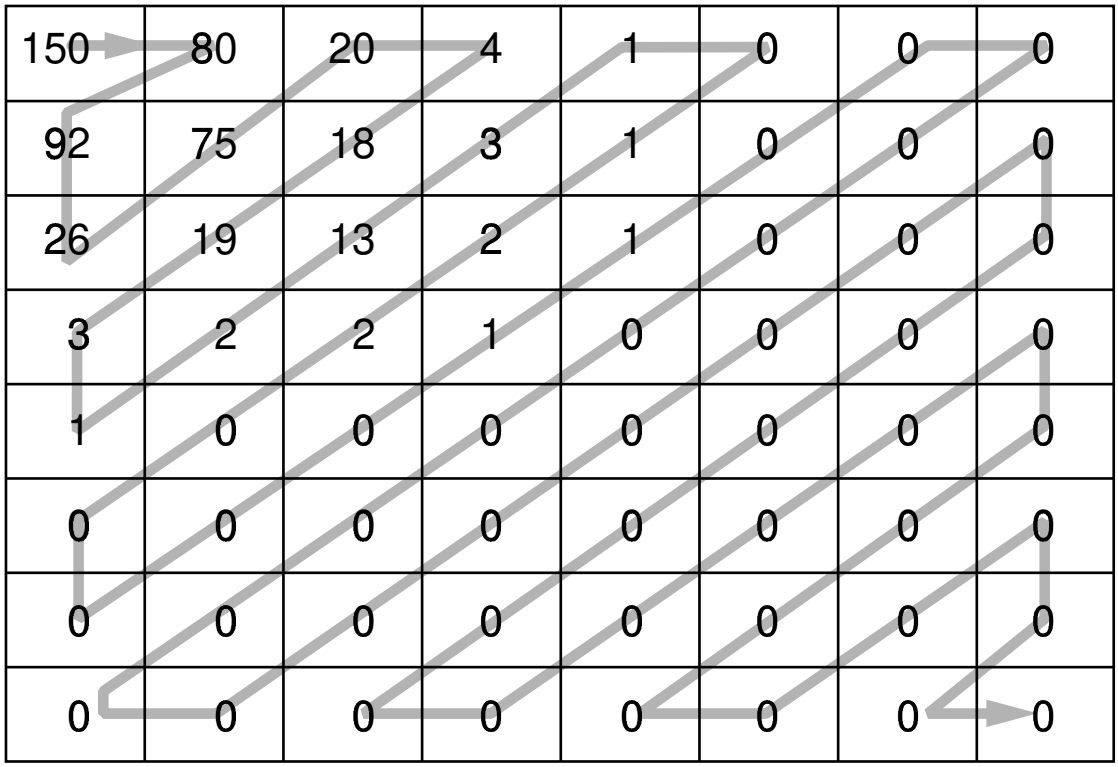


Fig. 7-84. The order in which the quantized values are transmitted.

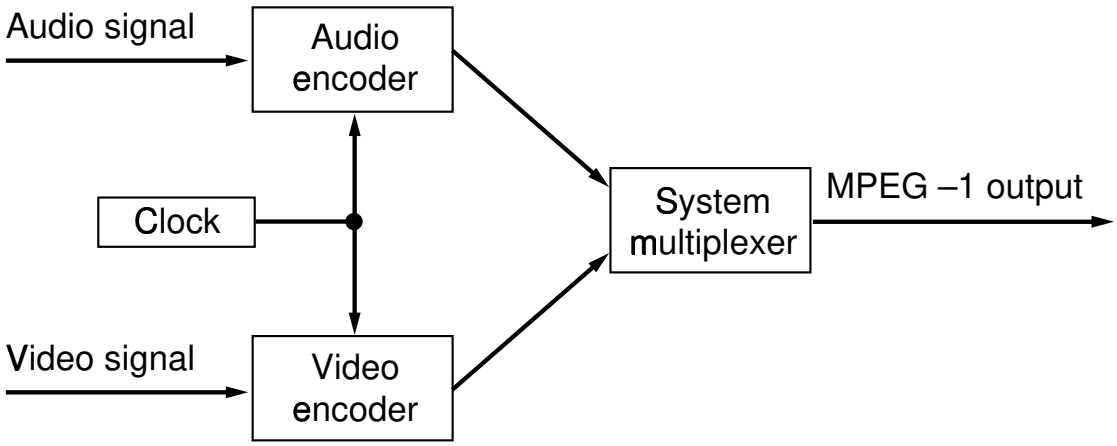


Fig. 7-85. Synchronization of the audio and video streams in MPEG-1.

Fig. 7-86. Three consecutive frames.

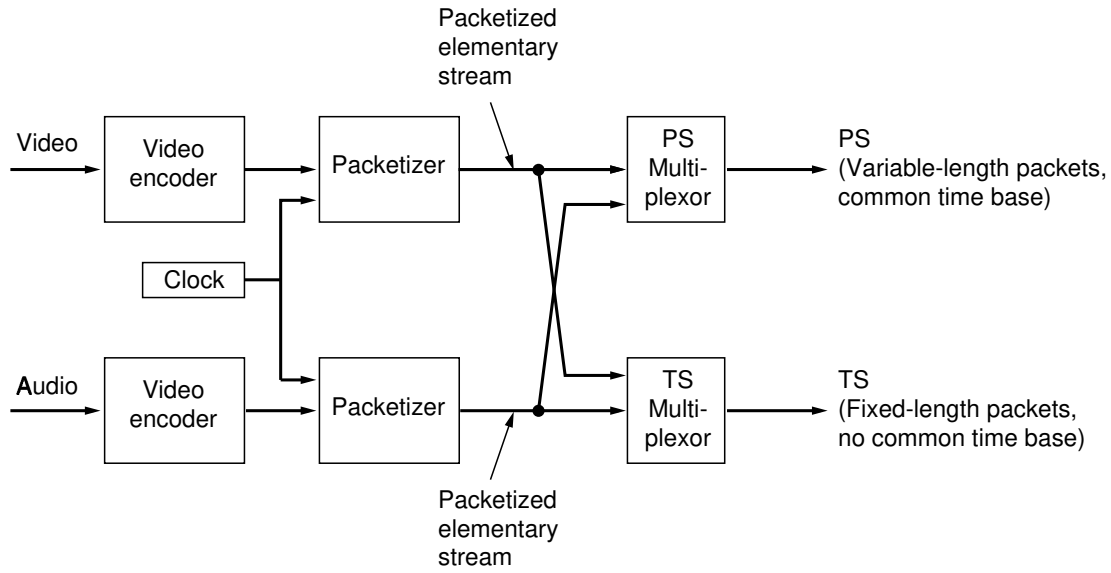


Fig. 7-87. Multiplexing of two streams in MPEG-2.

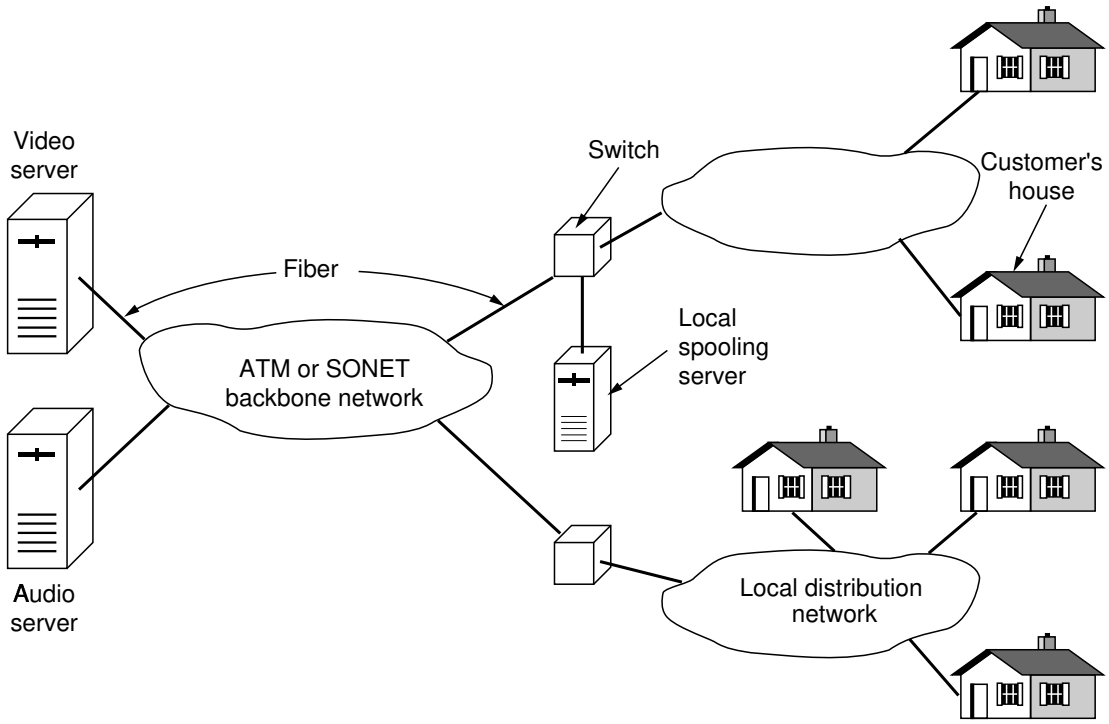


Fig. 7-88. Overview of a video-on-demand system.

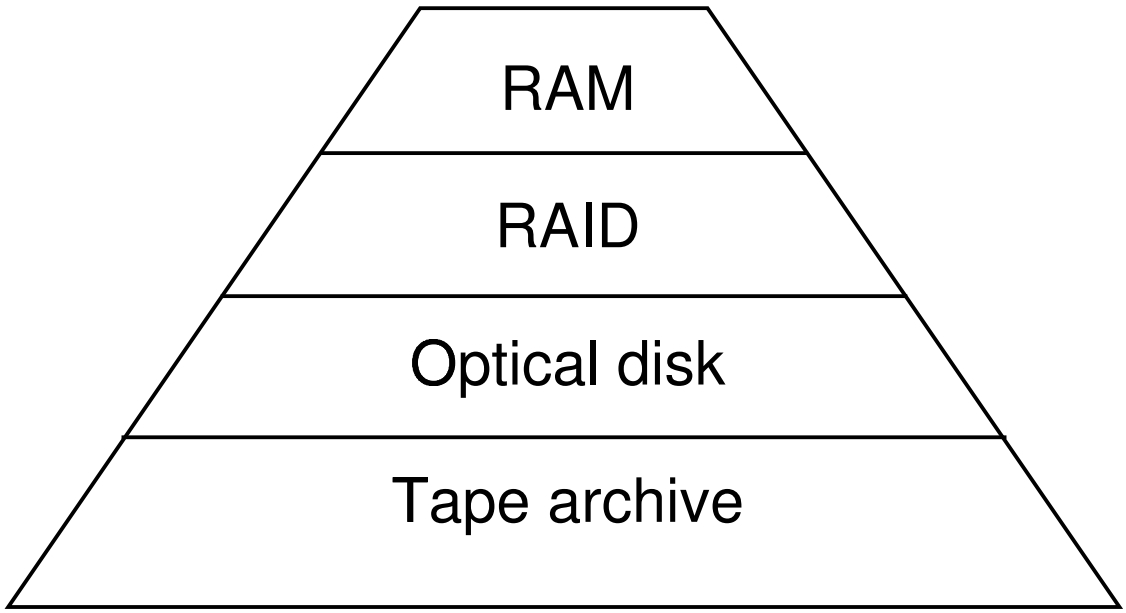


Fig. 7-89. A video server storage hierarchy.

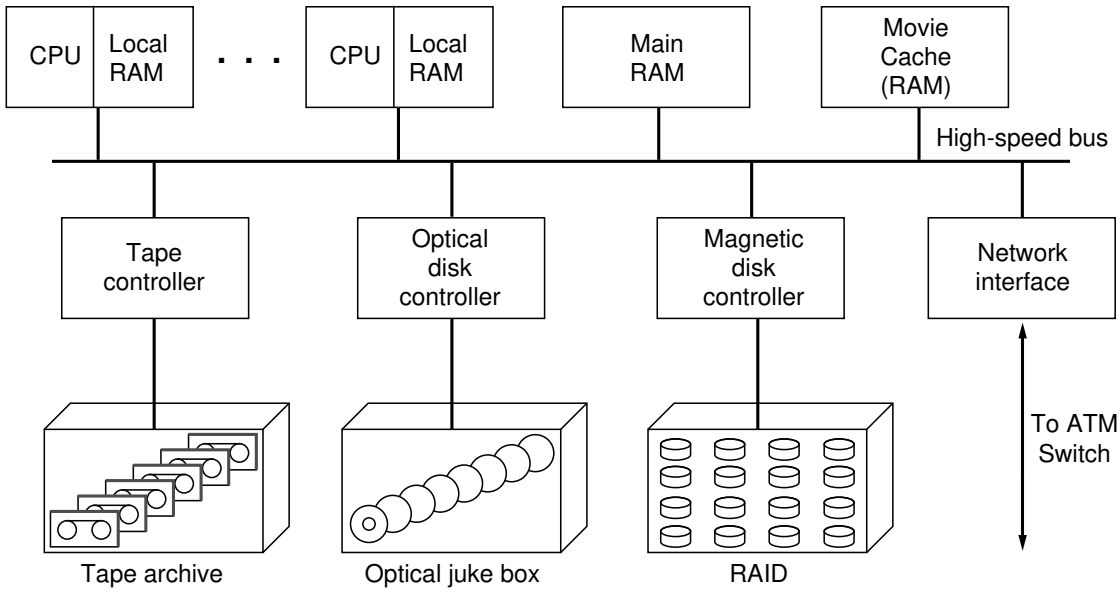


Fig. 7-90. The hardware architecture of a typical video server.

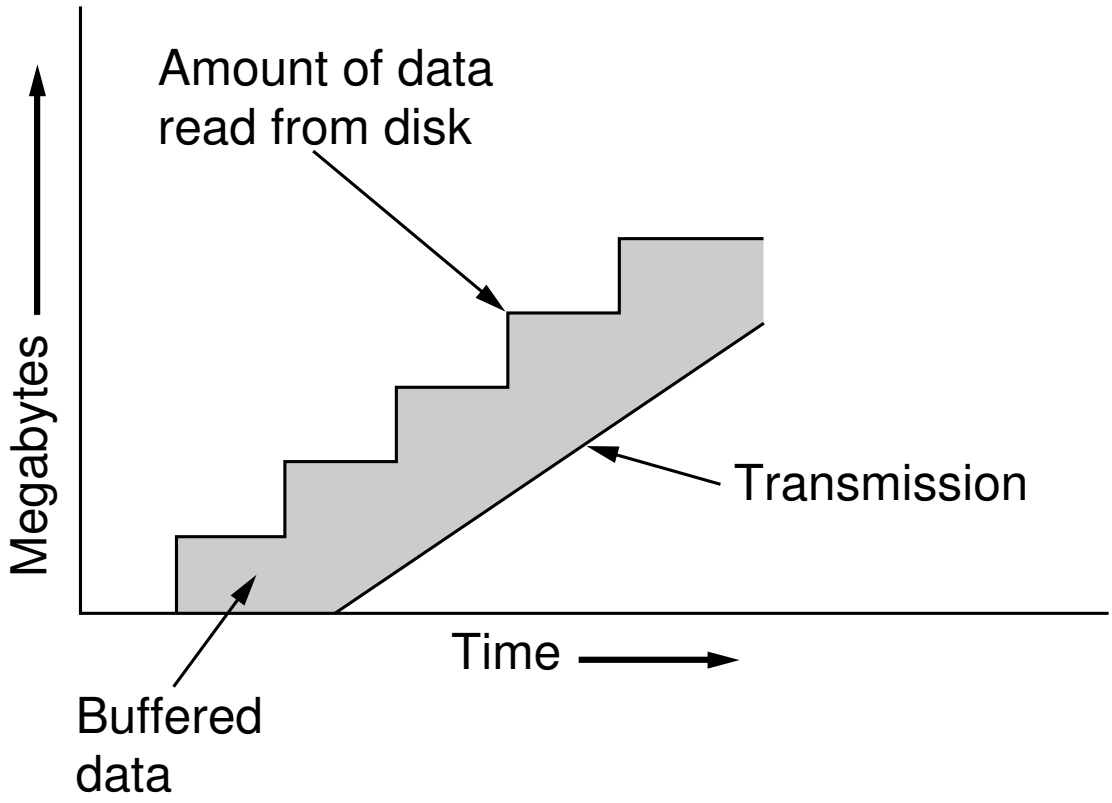


Fig. 7-91. Disk buffering at the server.

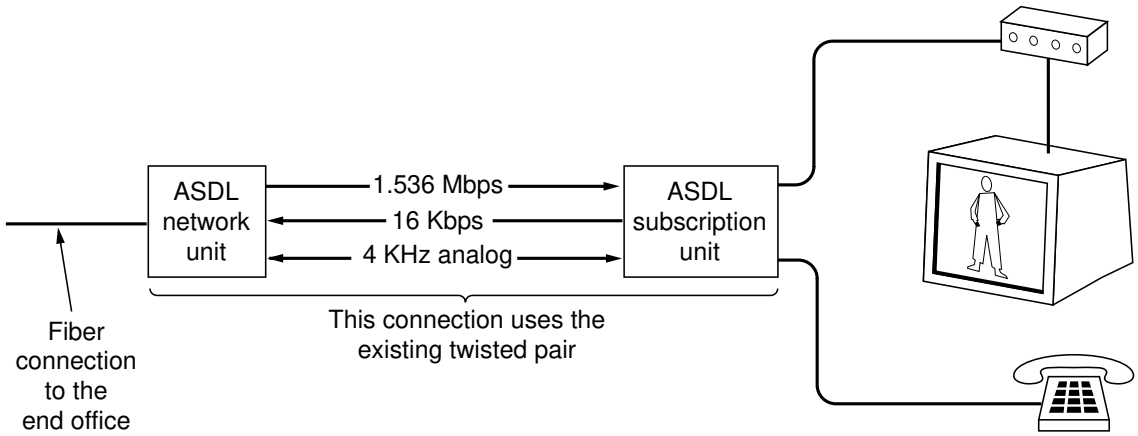


Fig. 7-92. ADSL as the local distribution network.

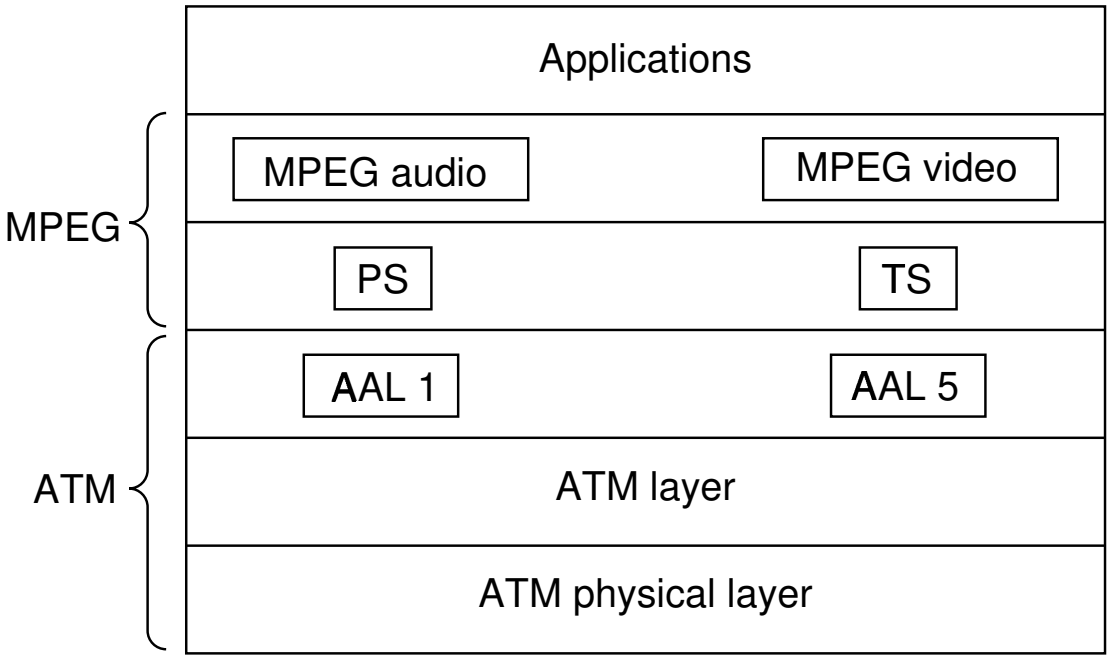


Fig. 7-93. A video-on-demand protocol stack.

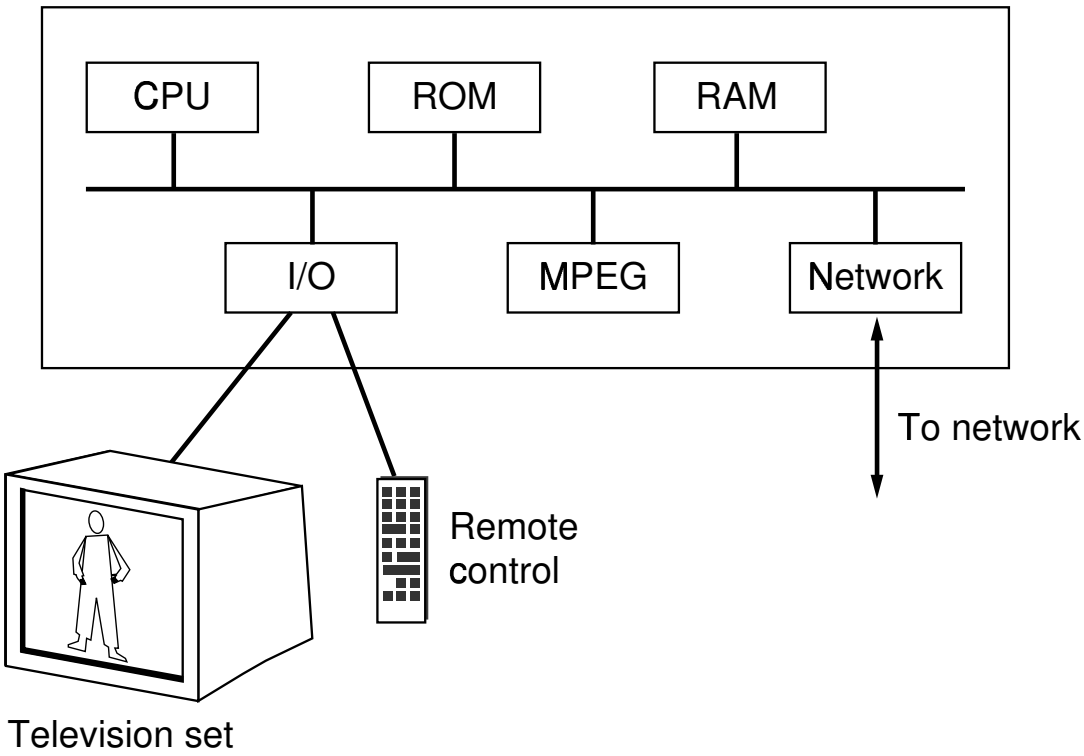


Fig. 7-94. The hardware architecture of a simple set-top box.

What technology will the backbone use (SONET, ATM, SONET + ATM)?
What speed will the backbone run at (OC-3, OC-12)?
How will local distribution be done (HFC, FTTC)?
How much upstream bandwidth will there be (16 kbps, 1.5 Mbps)?
Will movies be encrypted, and if so, how?
Will error correction be present (mandatory, optional, absent)?
Who will own the set-top box (user, network operator)?
Will telephony be part of the system (analog, N-ISDN)?
Will high-resolution hypertext applications be supported (e.g., WWW)?

Fig. 7-95. A few areas in which standards are needed.

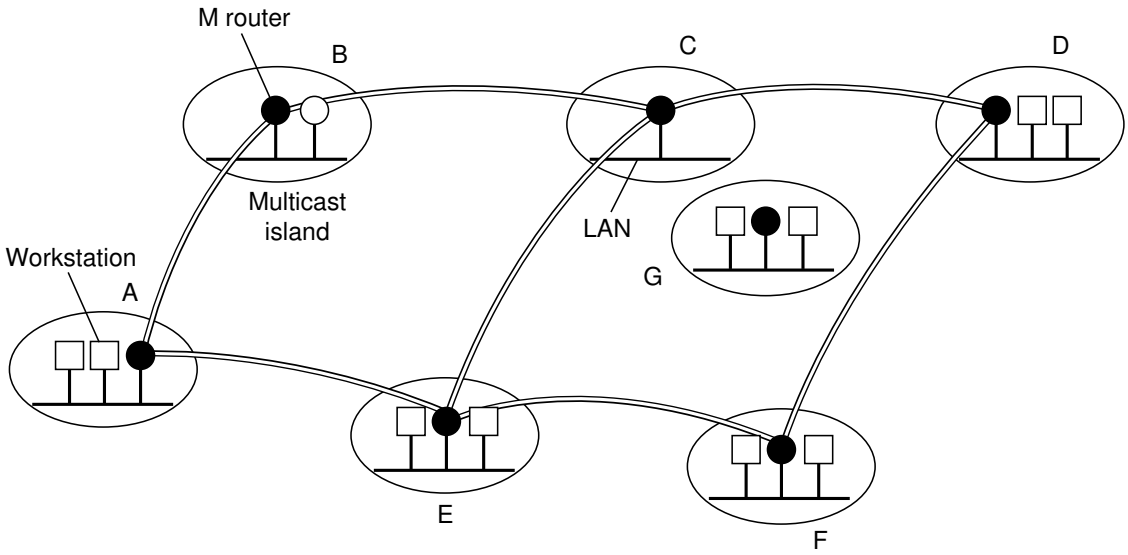


Fig. 7-96. MBone consists of multicast islands connected by tunnels.