Chapter 7

(Week 13)

The Application Layer

ANDREW S. TANENBAUM
COMPUTER NETWORKS
FOURTH EDITION
PP. 579-611

PREVIOUS LAYERS

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

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

The Application Layer

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

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

The Application Layer

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

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

The Application Layer's topics

7.1. DNS – The Domain Name System

7.2. Electronic Mail

7.3. The World Wide Web

7.4. Multimedia

7.5. Summary

7.1. DNS – The Domain Name System

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

7.1. DNS – The Domain Name System

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

7.1. DNS – The Domain Name System

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

7.1. DNS – The Domain Name System

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

7.1. DNS – The Domain Name System

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

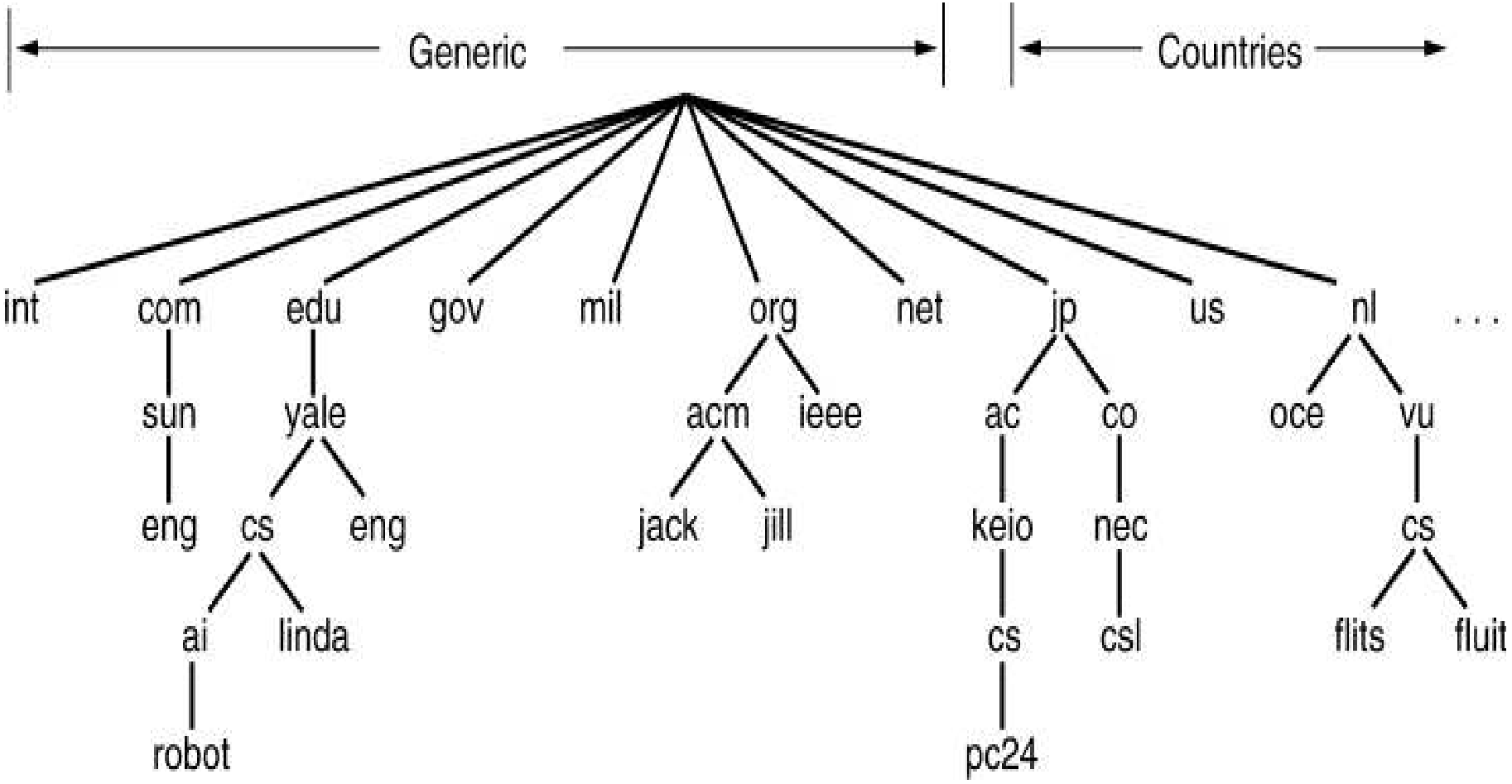
7.1. DNS – The Domain Name System

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space



A portion of the Internet domain name space.

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.1. The DNS Name Space

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

7.1.2. Resource Records

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

7.1.2. Resource Records

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

7.1.2. Resource Records

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

7.1.2. Resource Records

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

7.1.2. Resource Records

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

7.1.2. Resource Records

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

7.1.2. Resource Records

Type	Meaning	Value
SOA	Start of Authority	Parameters for this zone
A	IP address of a host	32-Bit integer
MX	Mail exchange	Priority, domain willing to accept e-mail
NS	Name Server	Name of a server for this domain
CNAME	Canonical name	Domain name
PTR	Pointer	Alias for an IP address
HINFO	Host description	CPU and OS in ASCII
TXT	Text	Uninterpreted ASCII text

The principal DNS resource records types.

7.1.2. Resource Records

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

7.1.2. Resource Records

```
; Authoritative data for cs.vu.nl
cs.vu.nl.          86400   IN   SOA      star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.          86400   IN   TXT      "Divisie Wiskunde en Informatica."
cs.vu.nl.          86400   IN   TXT      "Vrije Universiteit Amsterdam."
cs.vu.nl.          86400   IN   MX       1 zephyr.cs.vu.nl.
cs.vu.nl.          86400   IN   MX       2 top.cs.vu.nl.

flits.cs.vu.nl.    86400   IN   HINFO    Sun Unix
flits.cs.vu.nl.    86400   IN   A        130.37.16.112
flits.cs.vu.nl.    86400   IN   A        192.31.231.165
flits.cs.vu.nl.    86400   IN   MX       1 flits.cs.vu.nl.
flits.cs.vu.nl.    86400   IN   MX       2 zephyr.cs.vu.nl.
flits.cs.vu.nl.    86400   IN   MX       3 top.cs.vu.nl.
www.cs.vu.nl.      86400   IN   CNAME    star.cs.vu.nl
ftp.cs.vu.nl.      86400   IN   CNAME    zephyr.cs.vu.nl

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

little-sister              IN   A        130.37.62.23
                           IN   HINFO    Mac MacOS

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

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

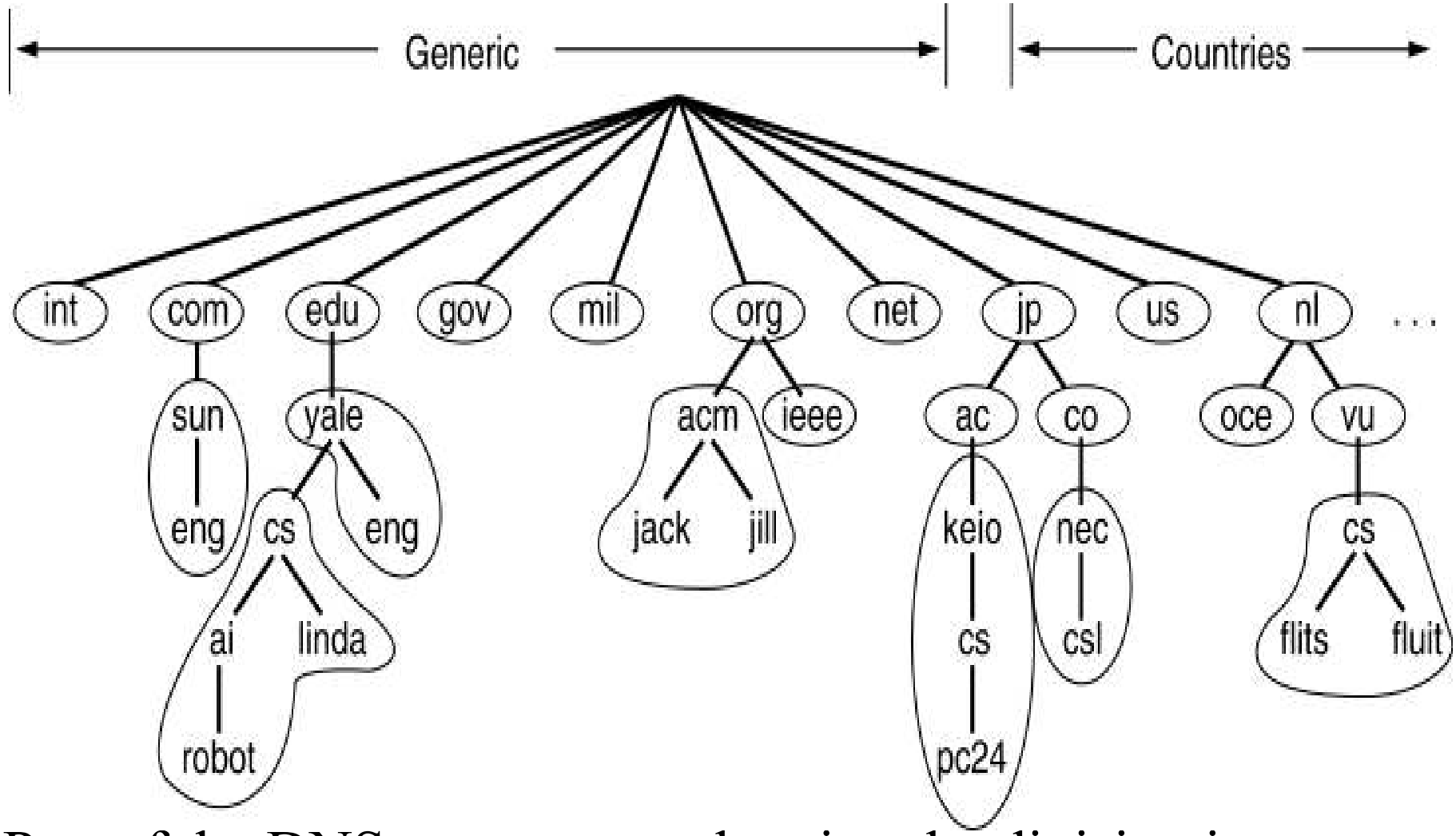
7.1.3. Name Servers

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

7.1.3. Name Servers

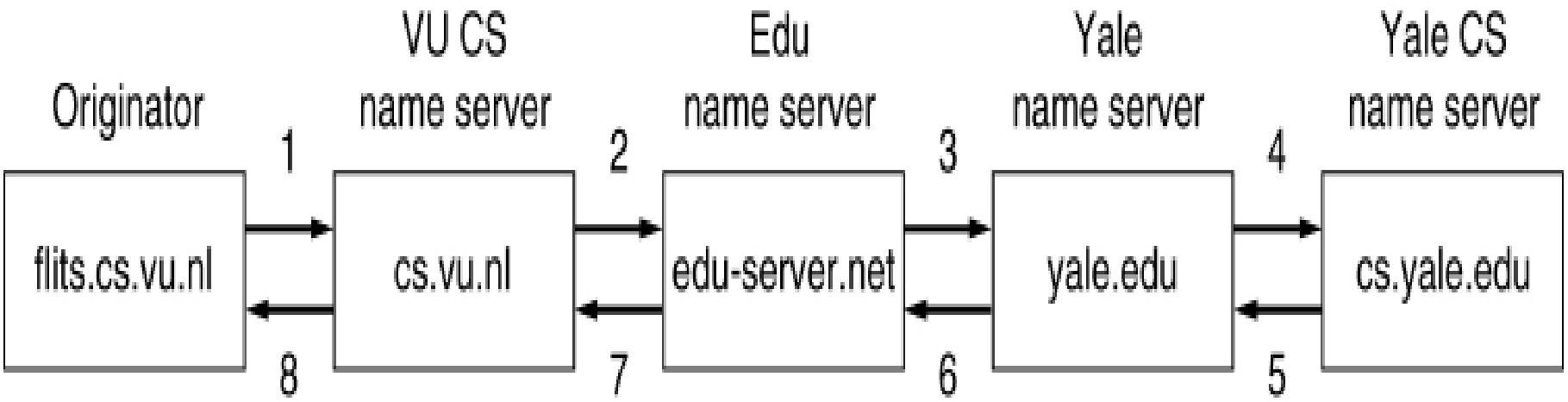
- To avoid the problems associated with having only a single source of information, the DNS name space is divided into nonoverlapping zones.
- One possible way to divide the name space on slide 12 is shown in following figure.

7.1.3. Name Servers



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

7.1.3. Name Servers



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

7.2. Electronic Mail

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

7.2. Electronic Mail

Smiley	Meaning	Smiley	Meaning	Smiley	Meaning
:-)	I'm happy	=|:-)	Abe Lincoln	:+)	Big nose
:-(	I'm sad/angry	=):-)	Uncle Sam	:-))	Double chin
:-|	I'm apathetic	*<:-)	Santa Claus	:-{)	Mustache
;-)	I'm winking	<:-(	Dunce	#:-)	Matted hair
:-(O)	I'm yelling	(-:	Australian	8-)	Wears glasses
:-(*)	I'm vomiting	:-)X	Man with bowtie	C:-)	Large brain

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

7.2. Electronic Mail

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

7.2. Electronic Mail

- Some of the complaints were as follows:

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

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

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

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

7.2. Electronic Mail

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

7.2. Electronic Mail

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

7.2.1. Architecture and Services

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

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

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

7.2.1. Architecture and Services

e-mail systems support five basic functions

- Composition
- Transfer
- Reporting
- Displaying
- Disposition

7.2.1. Architecture and Services

A few of advanced features:

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

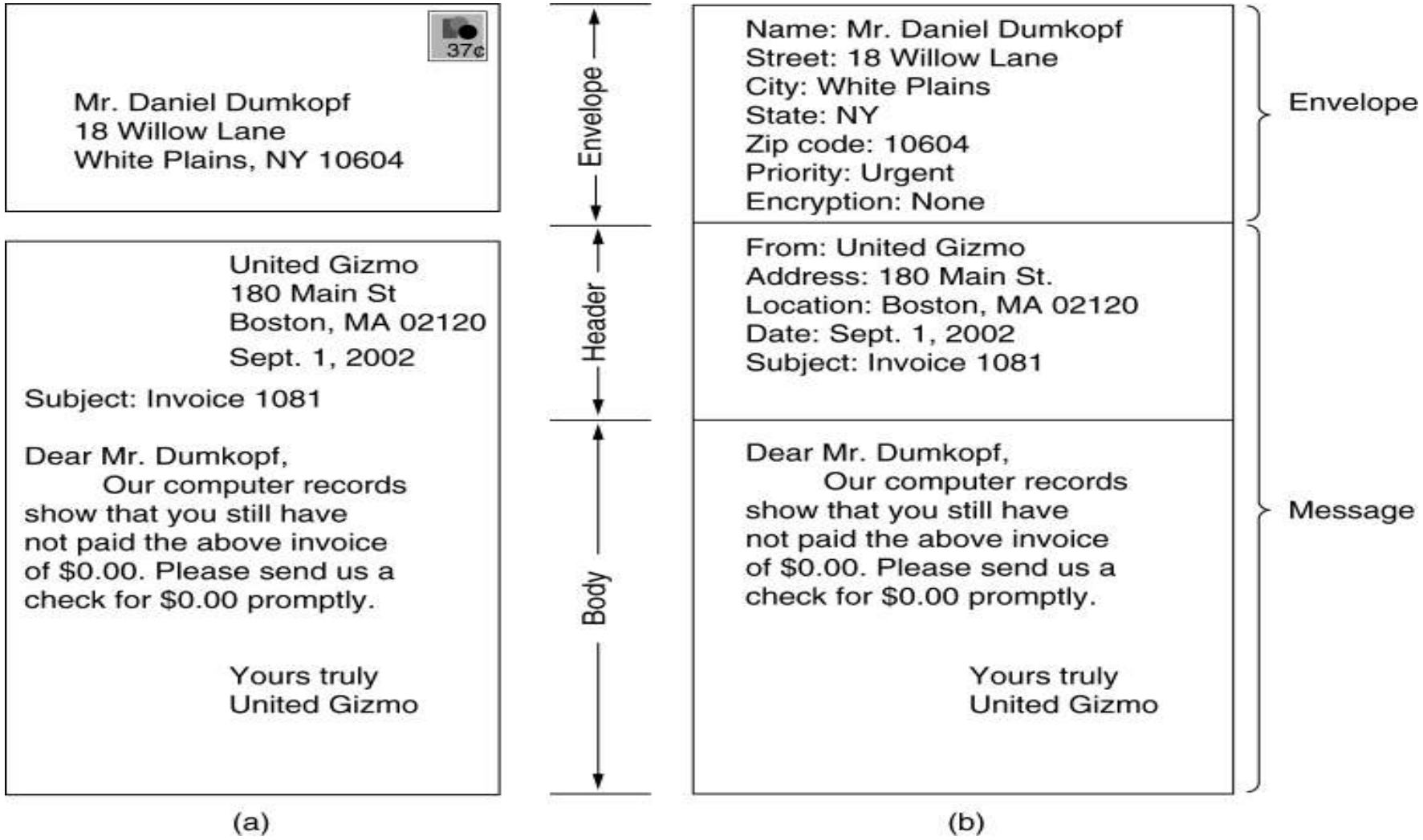
7.2.1. Architecture and Services

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

7.2.1. Architecture and Services

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

7.2.1. Architecture and Services



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

7.2.2. The User Agent

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

7.2.2. The User Agent

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

7.2.2. The User Agent

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

7.2.2. Reading E-mail

#	Flags	Bytes	Sender	Subject
1	K	1030	asw	Changes to MINIX
2	KA	6348	trudy	Not all Trudys are nasty
3	K F	4519	Amy N. Wong	Request for information
4		1236	bal	Bioinformatics
5		104110	kaashoek	Material on peer-to-peer
6		1223	Frank	Re: Will you review a grant proposal
7		3110	guido	Our paper has been accepted
8		1204	dmr	Re: My student's visit

An example display of the contents of a mailbox.

7.2.3 Message Formats – RFC 822

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

7.2.3 Message Formats – RFC 822

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

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

7.2.3 Message Formats – RFC 822

Header	Meaning
To:	E-mail address(es) of primary recipient(s)
Cc:	E-mail address(es) of secondary recipient(s)
Bcc:	E-mail address(es) for blind carbon copies
From:	Person or people who created the message
Sender:	E-mail address of the actual sender
Received:	Line added by each transfer agent along the route
Return-Path:	Can be used to identify a path back to the sender

RFC 822 header fields related to message transport.

Message Formats – RFC 822 (2)

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

Some fields used in the RFC 822 message header.

MIME – Multipurpose Internet Mail Extensions

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

MIME – Multipurpose Internet Mail Extensions

Problems with international languages:

- Languages with accents (French, German).
- Languages in non-Latin alphabets (Hebrew, Russian).
- Languages without alphabets (Chinese, Japanese).
- Messages not containing text at all (audio or images).

MIME – Multipurpose Internet Mail Extensions

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

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 <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--
```

A multipart message containing enriched and audio alternatives.

7.2.4. Message Transfer

- The message transfer system is concerned with relaying messages from the originator to the recipient.

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

7.2.4. Message Transfer

- SMTP – The Simple Mail Transfer Protocol
- Within the Internet, e-mail is delivered by having the source machine establish a TCP connection to port 25 of the destination machine.
- Listening to this port is an e-mail daemon that speaks SMTP.

7.2.4. Message Transfer

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

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

7.2.4. Message Transfer

- SMTP is a simple ASCII protocol.

- After establishing the TCP connection to port 25, the sending machine, operating as the client, waits for the receiving machine, operating as the server, to talk first.

7.2.4. Message Transfer

- The server starts by sending a line of text giving its identity and telling whether it is prepared to receive mail.

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

7.2.4. Message Transfer

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

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

7.2.4. Message Transfer

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

Message Transfer

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

```
                S: 220 xyz.com SMTP service ready
C: HELO abcd.com
                S: 250 xyz.com says hello to abcd.com
C: MAIL FROM: <elinor@abcd.com>
                S: 250 sender ok
C: RCPT TO: <carolyn@xyz.com>
                S: 250 recipient ok
C: DATA
                S: 354 Send mail; end with "." on a line by itself
C: From: elinor@abcd.com
C: To: carolyn@xyz.com
C: MIME-Version: 1.0
C: Message-Id: <0704760941.AA00747@abcd.com>
C: Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
C: Subject: Earth orbits sun integral number of times
C:
C: This is the preamble. The user agent ignores it. Have a nice day.
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: text/enriched
C:
C: Happy birthday to you
C: Happy birthday to you
C: Happy birthday dear <bold> Carolyn </bold>
C: Happy birthday to you
C:
C: --qwertyuiopasdfghjklzxcvbnm
C: Content-Type: message/external-body;
C:          access-type="anon-ftp";
C:          site="bicycle.abcd.com";
C:          directory="pub";
C:          name="birthday.snd"
C:
C: content-type: audio/basic
C: content-transfer-encoding: base64
C: --qwertyuiopasdfghjklzxcvbnm
C: .
                S: 250 message accepted
C: QUIT
                S: 221 xyz.com closing connection
```

7.2.5. Final Delivery

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

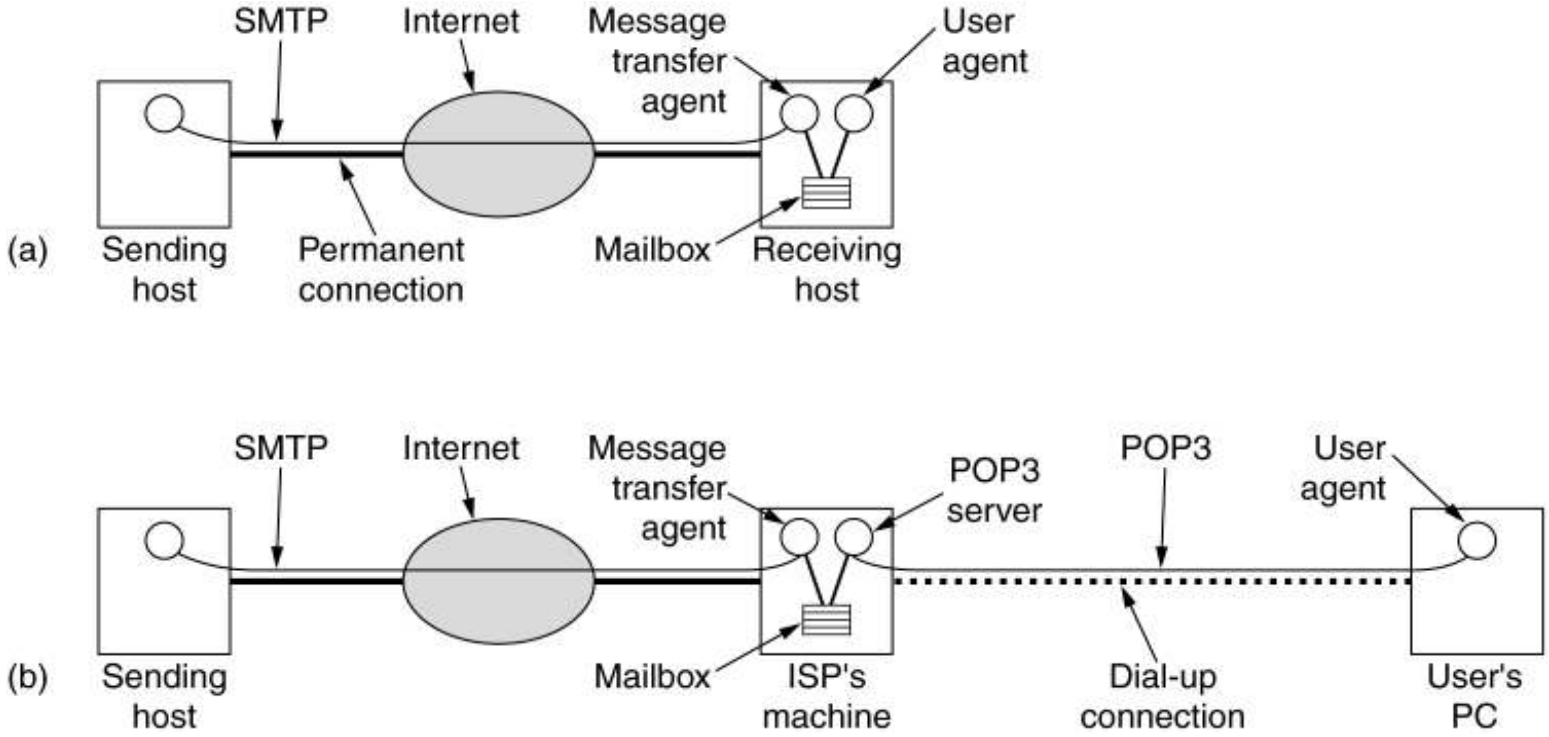
7.2.5. Final Delivery

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

7.2.5. Final Delivery - POP3

- How does the user get the e-mail from the ISP's message transfer agent?
- The solution to this problem is to create another protocol that allows user transfer agents (on client PCs) to contact the message transfer agent (on ISP's machine) and allow e-mail to be copied from the ISP to the user.
- POP3 – Post Office Protocol Version 3

7.2.5. Final Delivery



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

7.2.5. Final Delivery - POP3

```
                    S: +OK POP3 server ready
C: USER carolyn
                    S: +OK
C: PASS vegetables
                    S: +OK login successful
C: LIST
                    S: 1 2505
                    S: 2 14302
                    S: 3 8122
                    S: .
C: RETR 1
                    S: (sends message 1)
C: DELE 1
C: RETR 2
                    S: (sends message 2)
C: DELE 2
C: RETR 3
                    S: (sends message 3)
C: DELE 3
C: QUIT
                    S: +OK POP3 server disconnecting
```

Using POP3 to fetch three messages.

7.2.5. Final Delivery - IMAP

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