Introduction to LAN/WAN

Application Layer (Part II)

Application Layer Topics

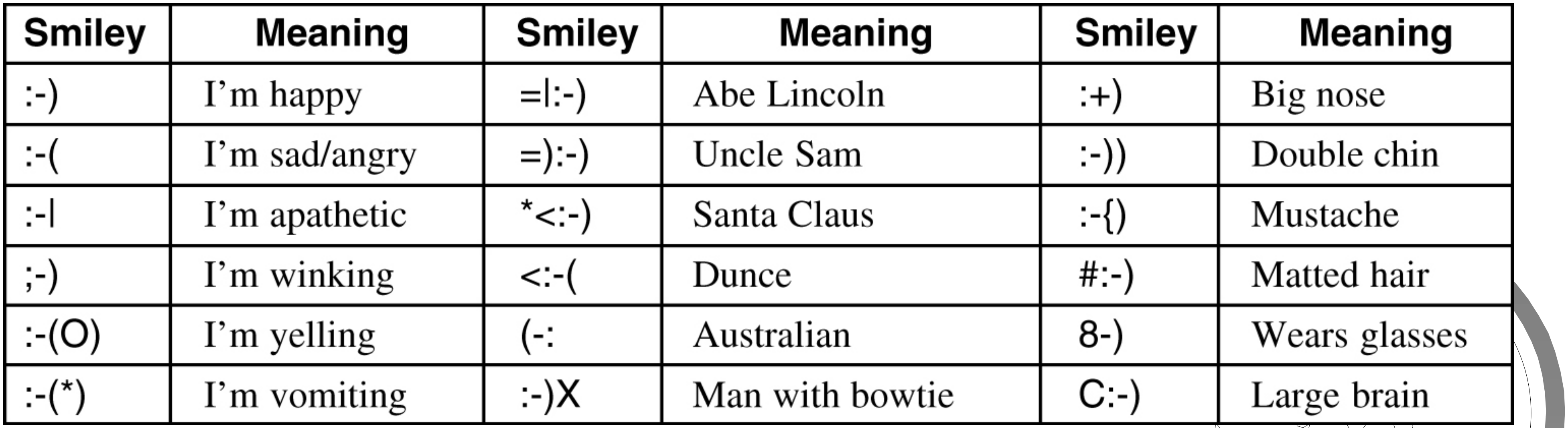
- ☞ Domain Name System (DNS) (7.1)
- ☞ Electronic Mail (Email) (7.2) ←
- ☞ World Wide Web (WWW) (7.3)

Electronic Mail (Email)

- Mostly used in academia before 1990
- 1990s:
  - Became widely used by public
  - Grew exponentially, now exceeds snail mail
- 1997 quote:
  - Amount of information on Internet per day in terabytes (more than library of Congress)
- Email relatively informal with conventions
  - BTW (By The Way), ROFL, IMHO, etc

Email: Smileys

- Smileys popular (Sanderson and Dougherty, 1993)
- Usually read by rotating 90 degrees clockwise



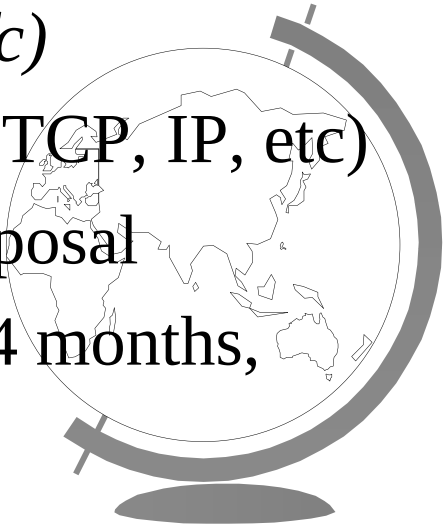
Smiley	Meaning	Smiley	Meaning	Smiley	Meaning
:-)	I’m happy	=|:-)	Abe Lincoln	:+)	Big nose
:-(	I’m sad/angry	=):-)	Uncle Sam	:-))	Double chin
:-|	I’m apathetic	*<:-)	Santa Claus	:-{)	Mustache
;-)	I’m winking	<:-(	Dunce	#:-)	Matted hair
:-(O)	I’m yelling	(-:	Australian	8-)	Wears glasses
:-(*)	I’m vomiting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

Electronic Mail (Email)

- ☞ Email systems initially simple
  - – Used FTP with recepient address as first line
- ☞ Major limitations
  - – Inconvenient to send to group
  - – Messages: no structure, processing difficult. E.g. extracting forwarded portion
  - – Sender not sure of delivery
  - – Secretary couldn’t handle absent boss’ mail
  - – Poor interface: edit first, exit editor, send
  - – Impossible to send mixed media mail (text, images, voice, video, fax, etc)

Email Standards

- 1982: ARPANET proposals published
  - RFC 821: transmission protocol
  - RFC 822: message format
- Later: RFC 2821 and 2822 became standards
- Most people still refer to RFC 821 and 822
- Note: Internet standardization process
  - RFC are Request For Proposals *(www.ietf.org/rfc)*
  - Tech report, describes a protocol or idea (email, TCP, IP, etc)
  - Internet culture informal, everyone critiques proposal
  - If RFC has merit, working implementations for 4 months, becomes standard

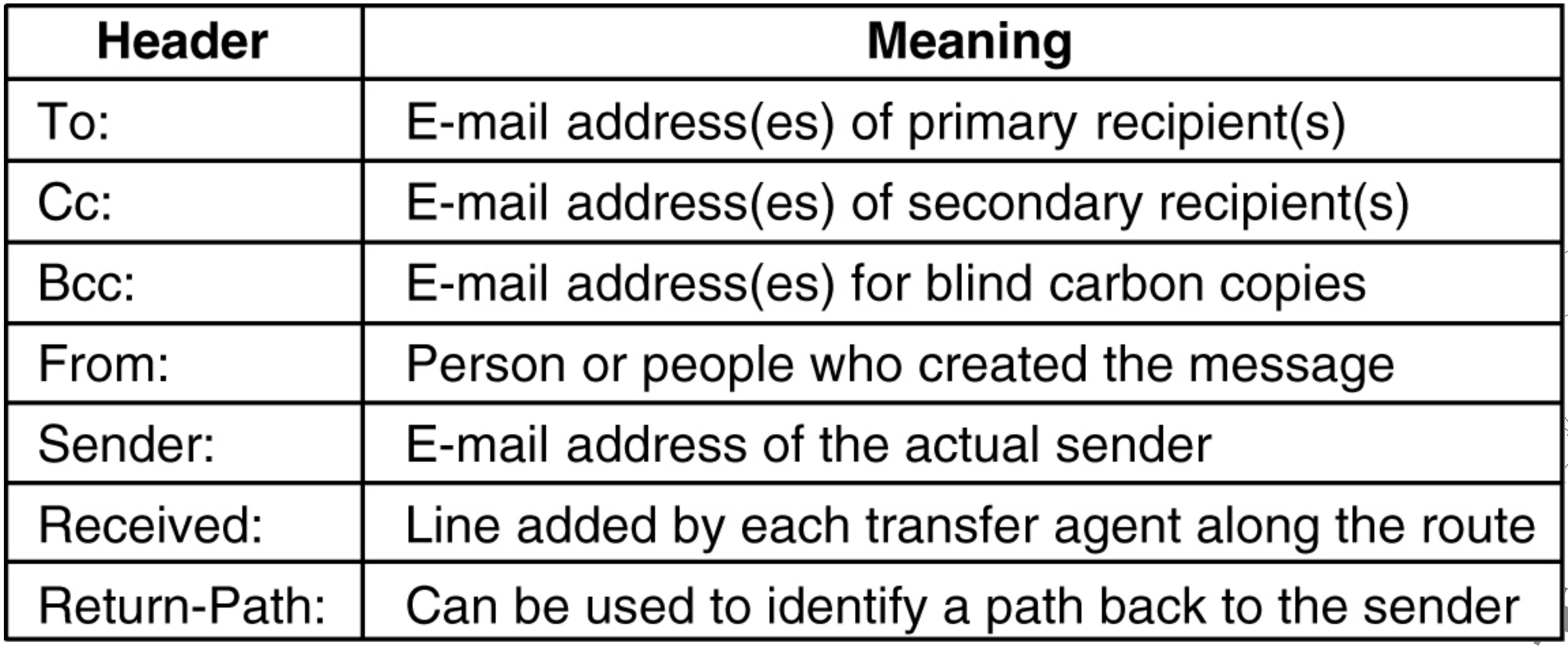


Email Standards

- ☞ Email system parts:
  - – *User agent*: (mail reader) is program used for composing, editing and manipulating email. E.g. *pine* (RFC 822 or 2822)
  - – *Message transfer agent*: concerned with relaying email message from originator to recepient (RFC 821 or 2821)
- ☞ Note RFC 822 deals with ASCII text
- ☞ MIME deals with multimedia extensions

RFC 822

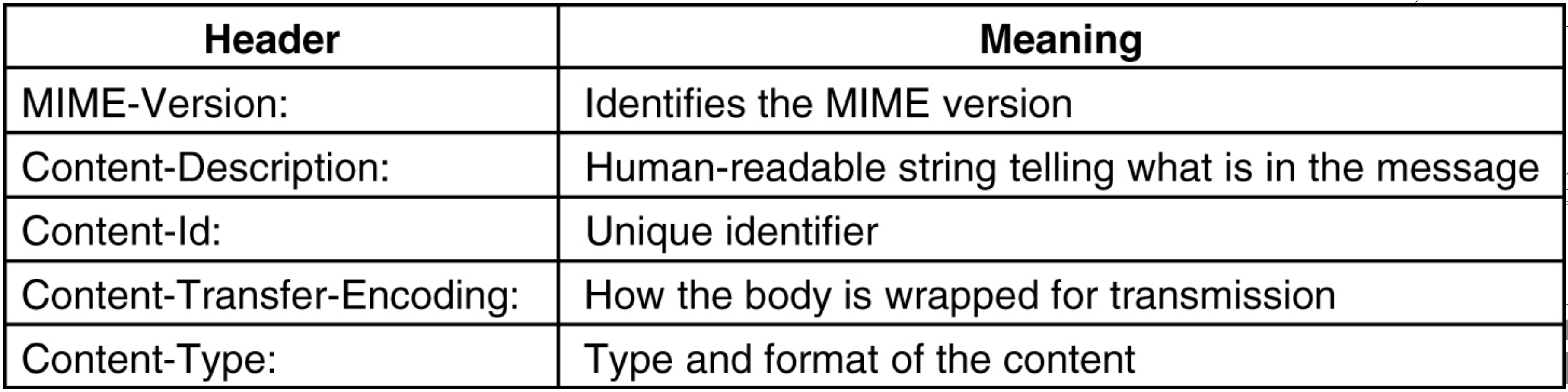
- ☞ *To:* field gives DNS address of primary recipient
- ☞ RFC 822 header fields related to message transport:



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

Multipurpose Internet Mail Extensions (MIME)

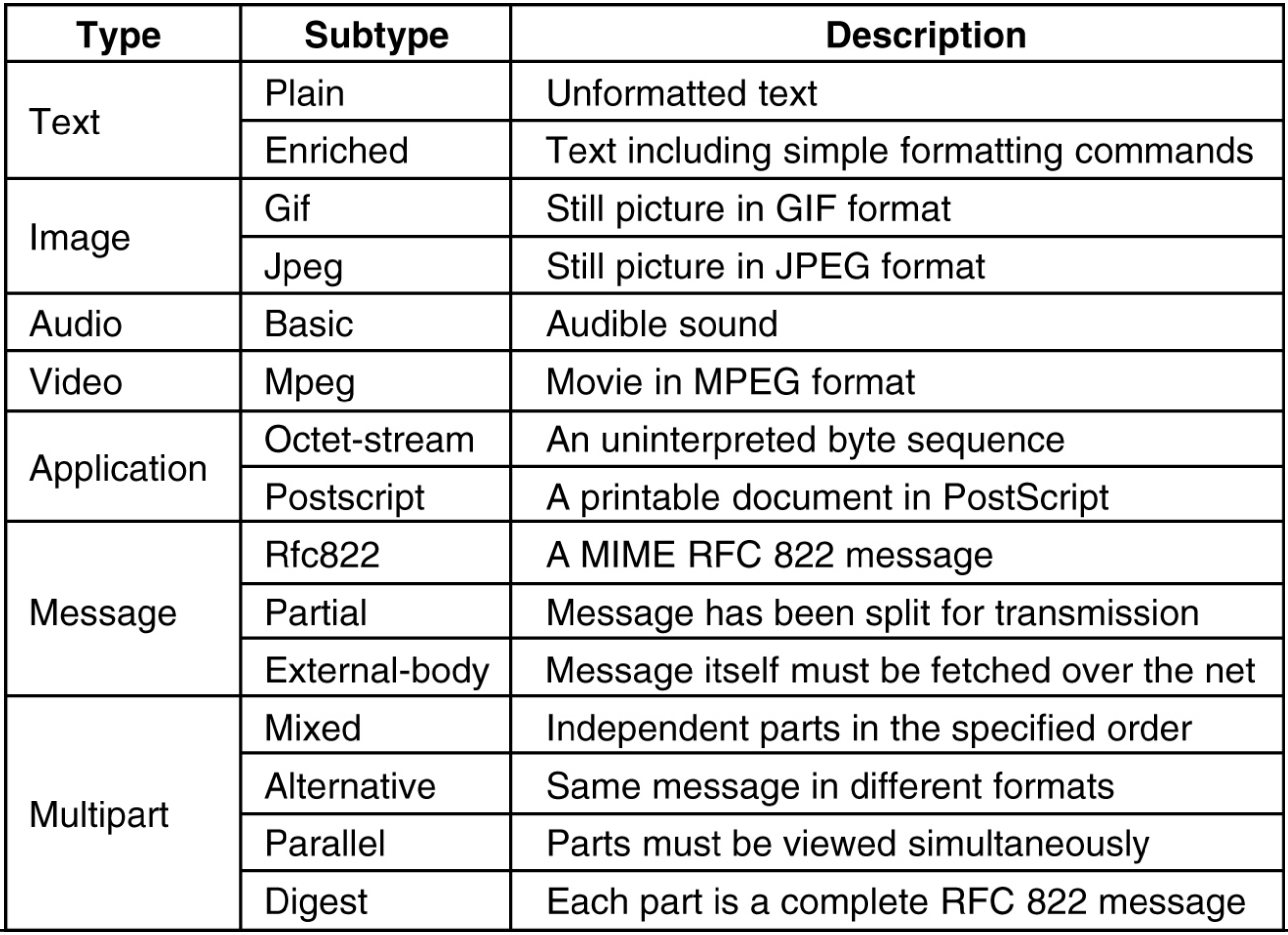
- RFC 822 had problems with international languages with:
  - accents (French, German), non-Latin alphabets (Hebrew, Russian) and without alphabets (Chinese, Japanese)
- Also with messages containing NO text (audio, images)
- Basic idea: continue RFC 822 format, but add structure to message body and define encoding rules for non-ASCII
- Five new (RFC 822) headers added:



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

MIME

- Most interesting MIME header is *Content-Type*
- 7 types in RFC 2045. E.g: *Content-Type: video/mpeg*
- MIME types and subtypes defined in RFC 2045
- Properly designed user agents must interprete these types



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

SMTP

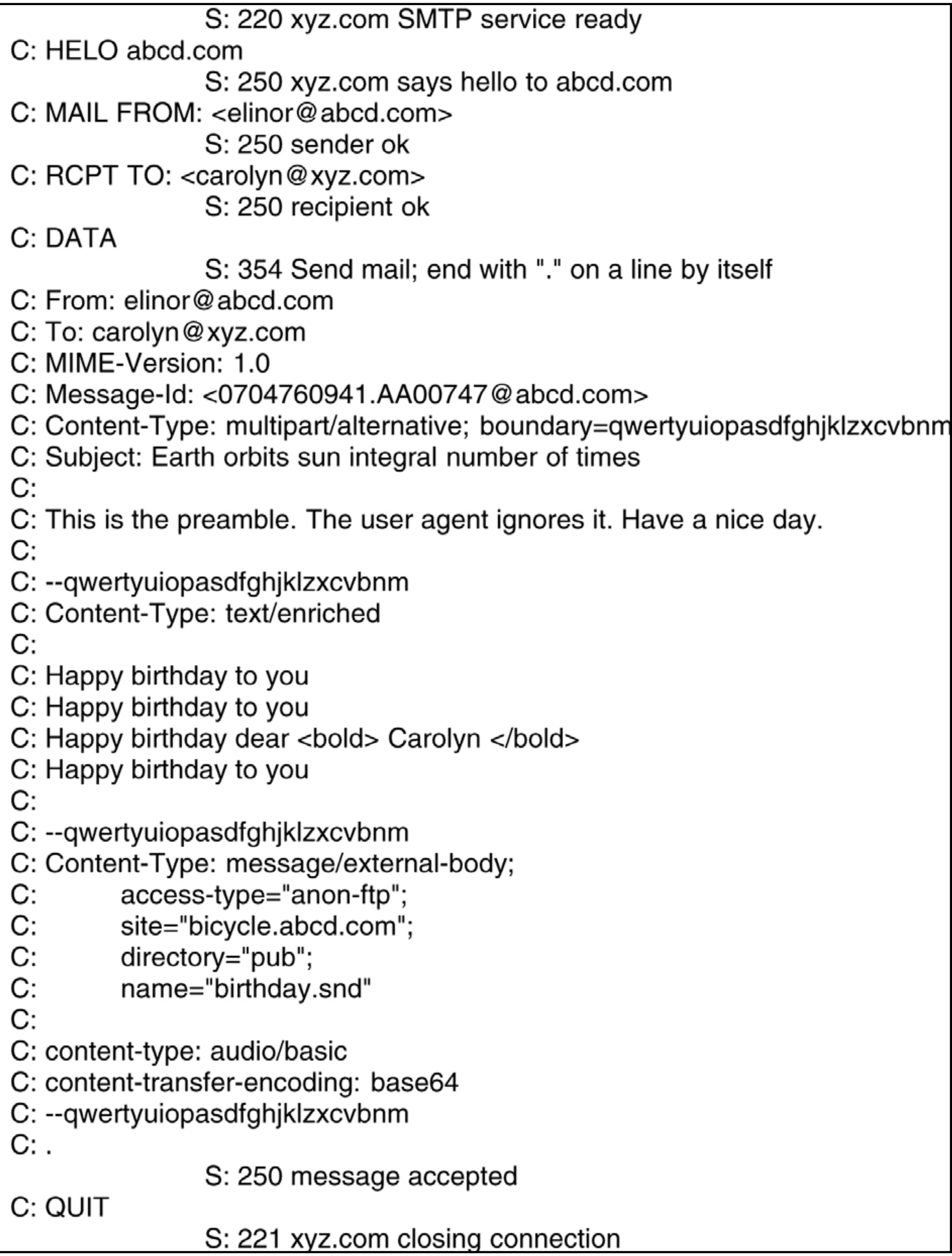
☞ Internet email delivery:

- – Source machine establishes TCP connection to port 25 of destination machine
- – E-mail daemon (process) listening at this port speaks SMTP (Simple Mail Transfer Protocol), RFC 821
- – Daemon accepts incoming connections, copies messages to appropriate mailboxes
- – Error generated for undeliverable message

☞ SMTP is simple ASCII protocol

Message Transfer

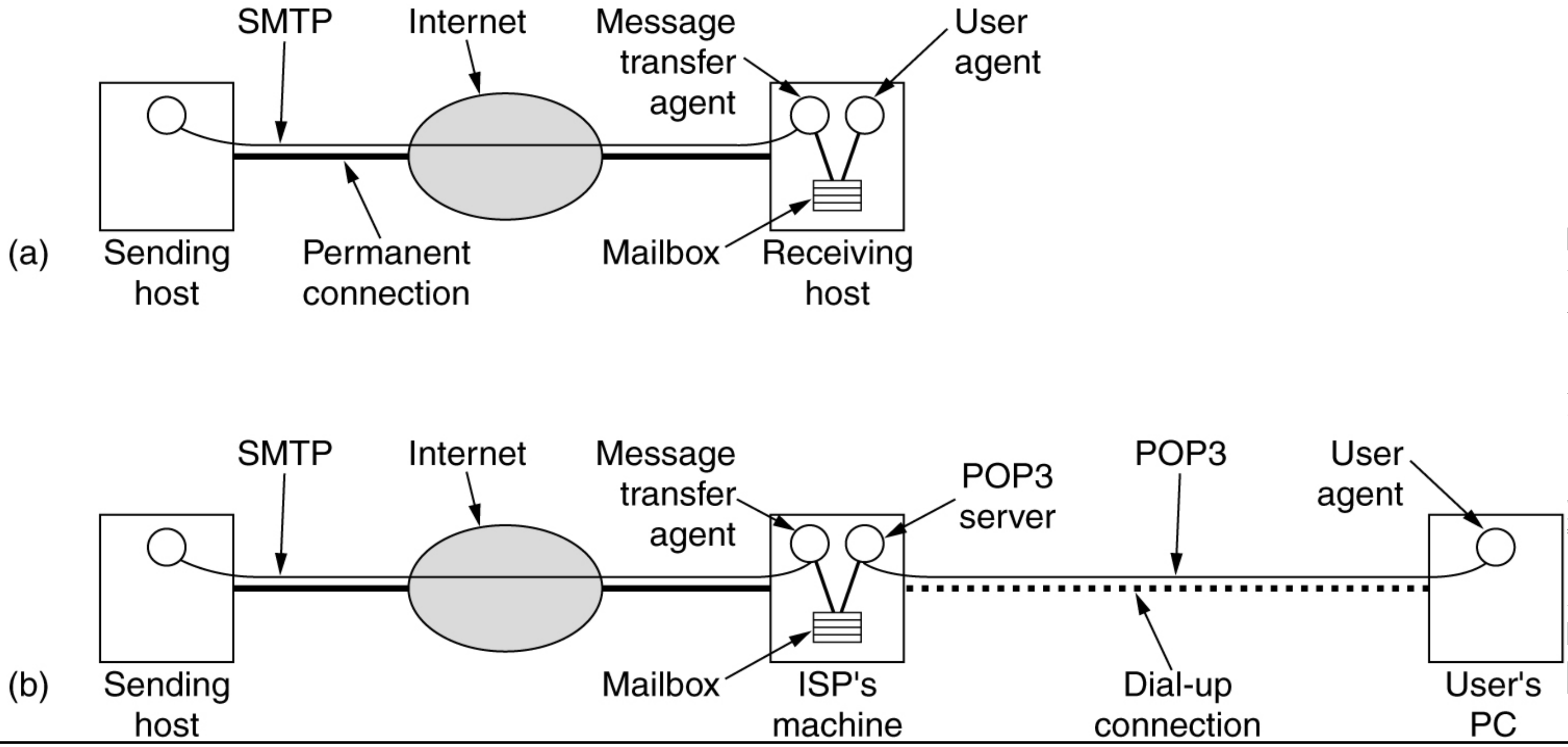
- ☞ Example: Transferring a message from *elinore@abc.com* to *carolyn@xyz.com.*
- ☞ Try telnet *garden* 25



```
                    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
```

Message Transfer with Temporary Connections

- *Case 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.
- *Case b:* Reading e-mail when the receiver has a dial-up connection to an ISP.



Disconnected Users

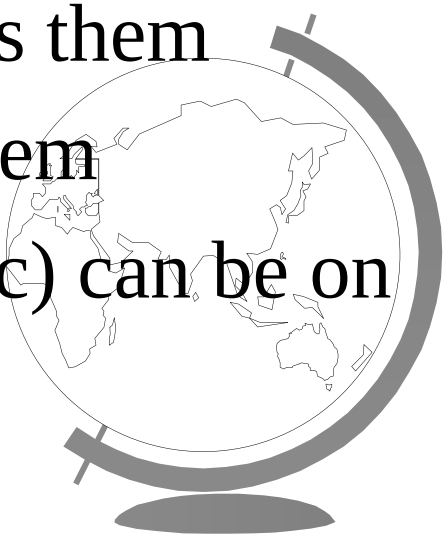
- Post Office Protocol (Pop3) (RFC 1939)
  - ISP accepts mail for disconnected subscribers (USPS??)
  - Disconnected user calls up ISP, transfers using port 110
  - Pop3 protocol goes through 3 states: *authorization*, *transaction* and *update*
- Internet Message Access Protocol (IMAP) (RFC 2060)
  - One user can have multiple IMAP accounts (WPI, AOL, etc)
  - Pop3 assumes use will download all mail once connected
  - IMAP allows manipulation of message parts on server
- Webmail (hotmail, yahoo, etc)
  - Uses SMTP on port 25
  - Web interface, similar process as before

World Wide Web

- Architectural framework for accessing linked documents
- Linked pages spead all over the Internet, on many machines
- Innovator: Tim Berners-Lee, *CERN* (European Nuclear Physics Center), March 1989
- In 10 years, went from means to distributing high-energy physics to pervasive application
- Most people (meatheads??) think of the web as the Internet
- Milestones:
  - First public demo’ of WWW at *Hypertext ’91* conference
  - Mosaic, first web browser, free, Marc Andreessen, UIUC, 1993
  - Netscape, first commercial browser, 1995, IPO, mania!!!
  - WWW consortium formed 1994 (MIT, CERN), *www.w3.org*

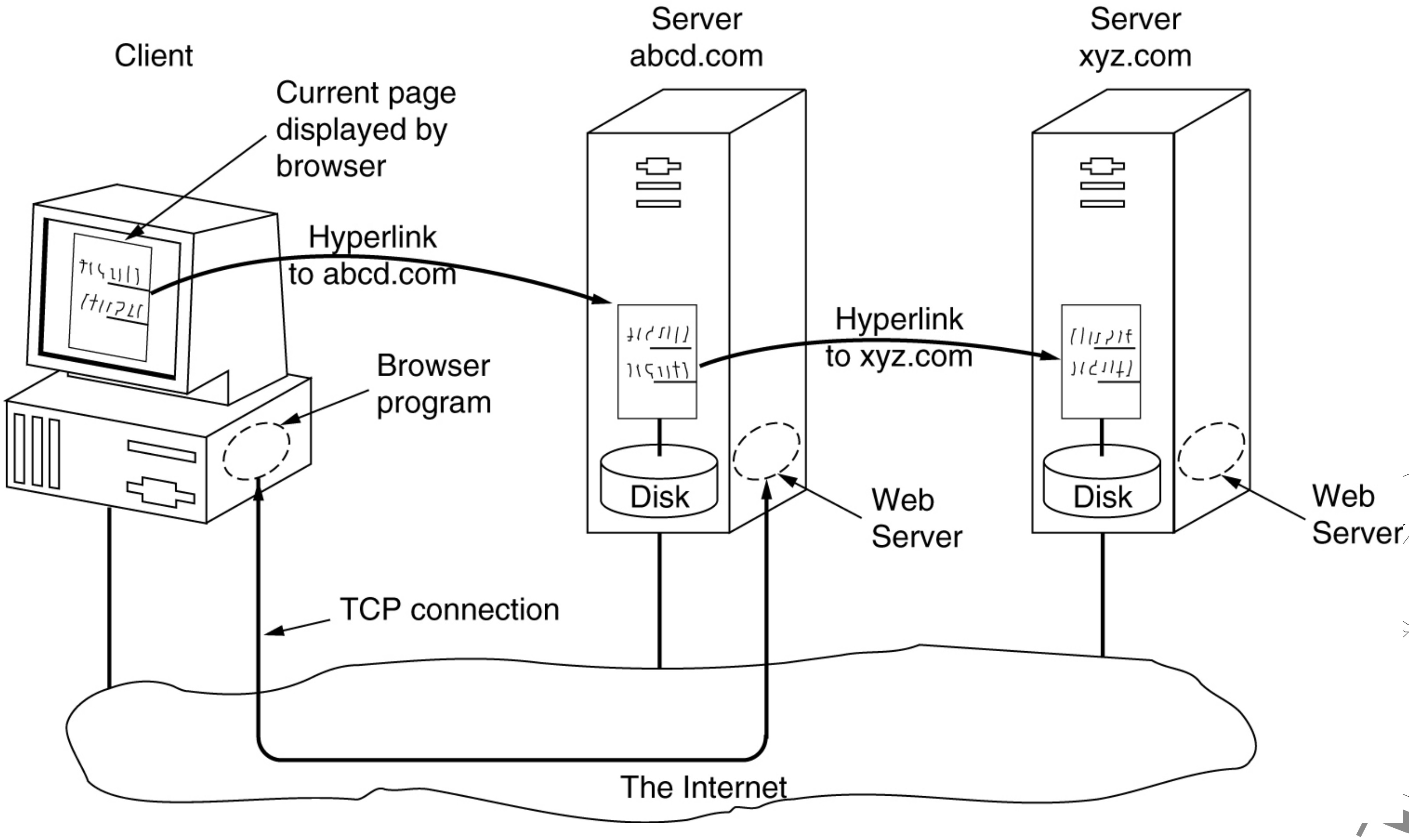
Web Architecture

- User perceives web of worldwide collection of documents or *web pages*
- Original hypertext vision: Vanavar Bush, MIT prof, US presidential advisor, 1945
- Web pages viewed using browsers (Netscape navigator, Internet Explorer)
- Browser fetches pages, interpretes and displays them
- User can access linked pages by clicking on them
- Linked pages or pieces (images, video, text, etc) can be on same host or in Australia



Web Architecture

- Parts of the web model:

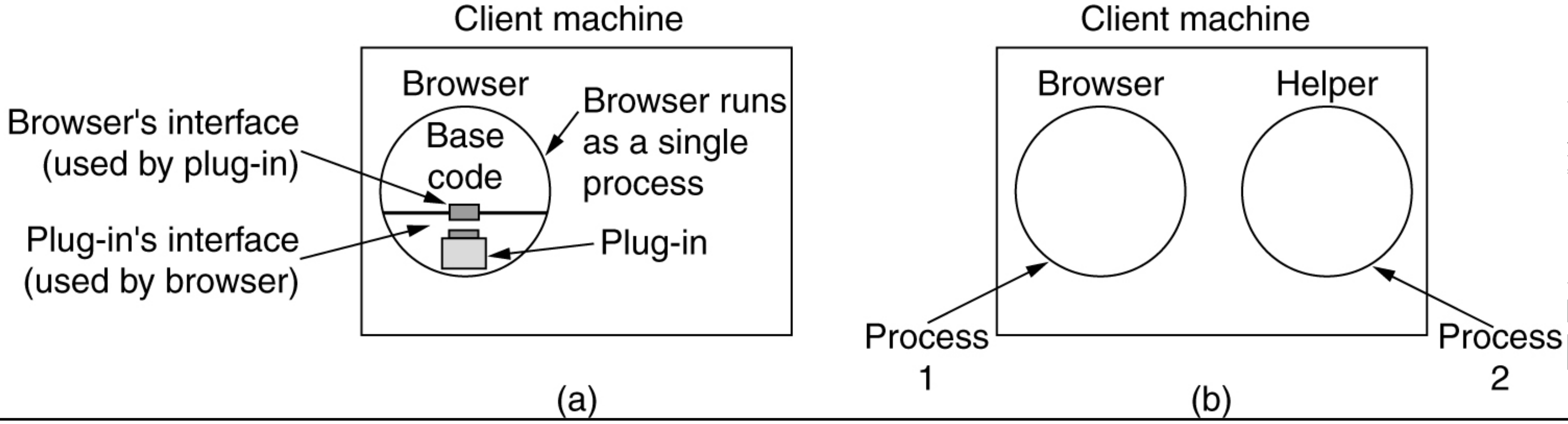


Client Side

- ☞ Client side (browser) steps:
  - – User enters or click on URL (e.g. *www.itu.org*
  - – Browser asks DNS server for IP address of *www.itu.org*
  - – DNS replies with IP address (*e.g. 156.106.192.32*)
  - – Browser makes TCP connection (port 80) with *156.106.192.32*
  - – Asks for file (web page) on server's directory */home/index.html*
  - – *www.itu.org* server sends file */home/index.html*
  - – TCP connection is released
  - – Browser displays all text in web page
  - – Browser fetches and displays all images in this file

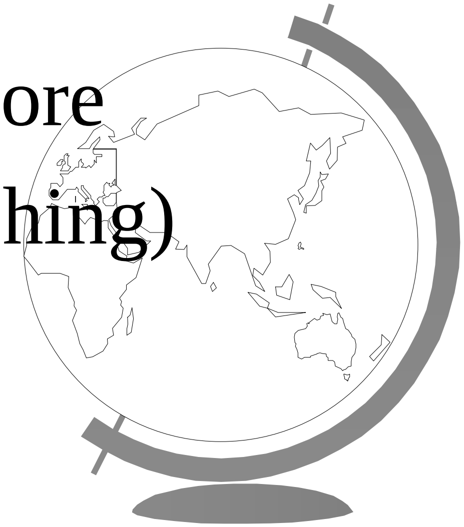
Client Side

- Web pages written in HTML,
- HTML: interpreted markup language derived from SGML
- Browsers support HTML as well as MIME types
- Ever increasing number of file formats
- Instead of building larger browsers, use:
  - *Plug-ins:* software extension (module) that browser loads to run new file. Usually runs MIME types
  - *Helper applications*: complete program, separate process. E.g. *application/pdf* specified in browser option will load Adobe Acrobat reader. Can be non-MIME type


Client machine
Browser
Base code
Browser runs as a single process
Browser's interface (used by plug-in)
Plug-in's interface (used by browser)
Plug-in
(a)
Client machine
Browser
Helper
Process 1
Process 2
(b)

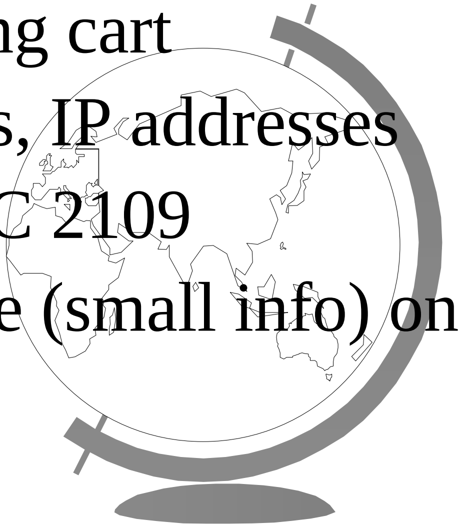
Server Side

- Web server steps:
  - Waits for connections coming in on port 80
  - Accepts a TCP connection from client (browser)
  - Get name of requested file
  - Fetch requested file from disk
  - Return file to client (similar to FTP)
  - Release TCP connection
- Above is basic web server, modern ones do more
- Server could be innundated with requests (caching)



Web Architecture

- Universal Resource Locator (URL)
  - A web page’s unique name (worldwide)
  - Three parts (protocol, DNS name of host, local file name)
  - E.g: *http:www.cs.nu.nl/video/index-en.html* breaks down to *protocol* (http), *www.cs.vu.nl* (DNS name) and *video/index-en.html* (file name)
  - Other web protocols: *ftp:*, *file:*, *news*, *gopher*, *mailto:*, *telnet:*
- Statelessness and cookies
  - Stateless server, doesn’t remember previous requests or clients
  - Sometimes want to track. E.g e-Commerce shopping cart
  - Can’t use IP address, many people share computers, IP addresses
  - Netscape devised *cookies* (No!! can’t eat ‘em), RFC 2109
  - Cookies: server returns requested file, plants cookie (small info) on client
  - Next time client logs in, sends cookies back

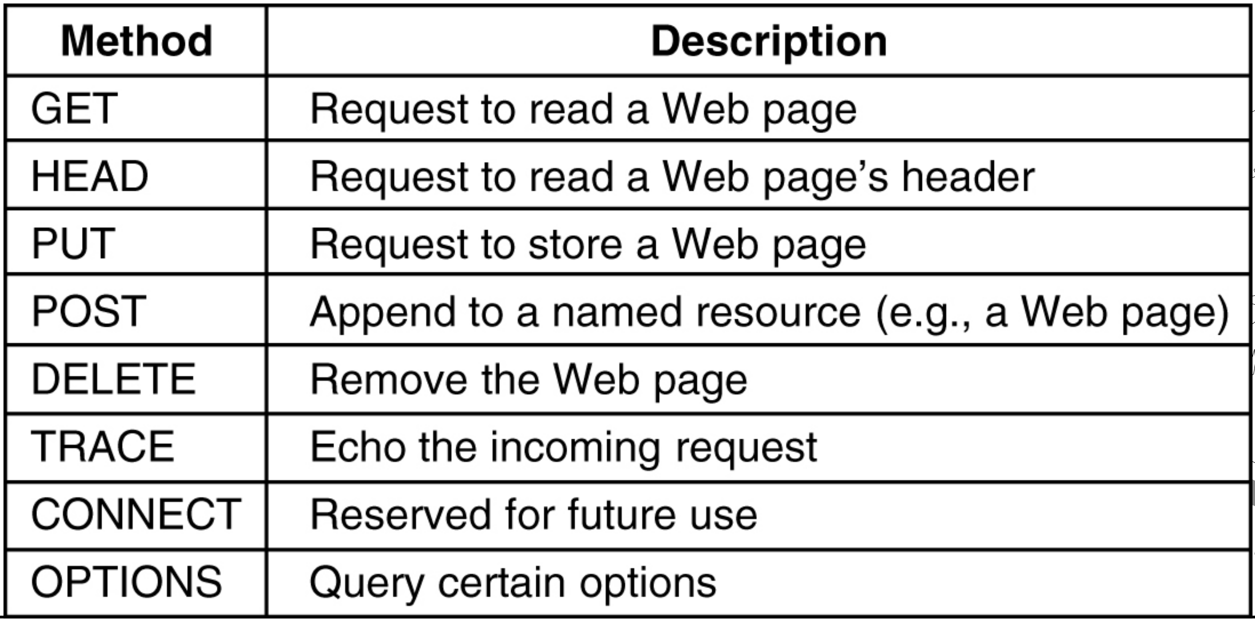


Web Pages

- Two types: *static* and *dynamic*
- *Static:*
  - Simple, same exact page (file) every time
  - Examples: HTML, forms, XML
- *Dynamic:*
  - Web page or parts of it are generated on demand
  - Content generation can either take place on server or client
  - Server side usually involves scripting (CGI, PHP, JSP) and maybe database access
  - Client side is for interactive (and kewl) websites, with features like rollover buttons. E.g. Javascript

HTTP

- Web transfer protocol, *Hypertext Transfer Protocol (HTTP)*
- RFC 2616, specifies what messages a client sends and responses server returns
- TCP used so that servers don’t worry about lost messages
- Initially, one TCP connection per web object (part)
- Too much eye candy these days, try *persistent connections*
- 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