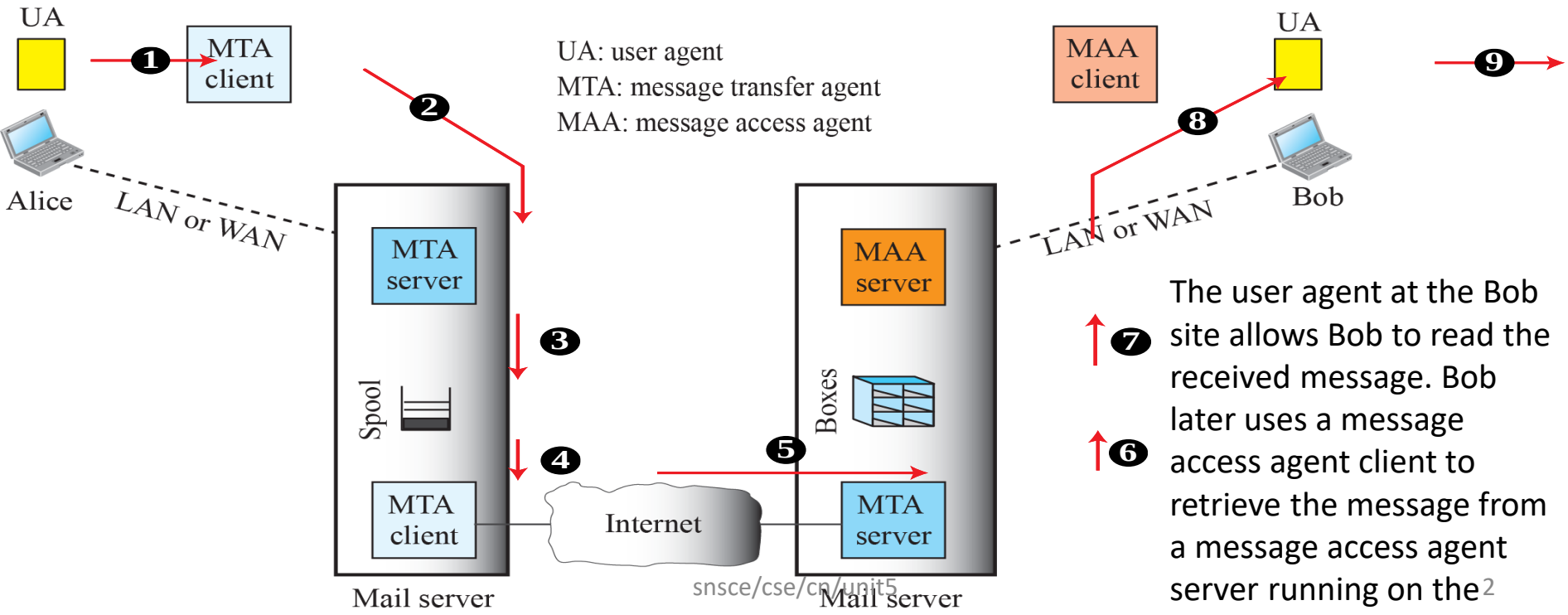
ELECTRONIC MAIL

Electronic mail (or e-mail) allows users to exchange messages. The nature of this application is different from other applications discussed so far. This means that the idea of client/server programming should be implemented in another way: using some intermediate computers (servers).

Architecture

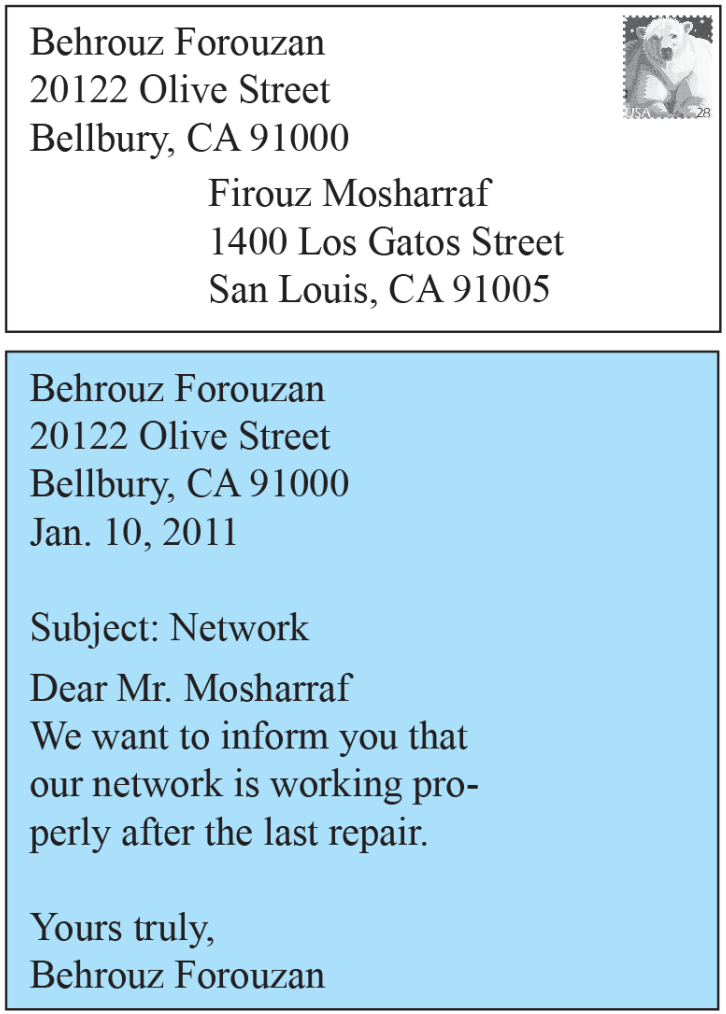
The sender and the receiver, Alice and Bob, are connected to two mail servers. The administrator has created one mailbox for each user. A mailbox is part of a server hard drive, a special file with permission restrictions. Only the owner of the mailbox has access to it.

When Alice needs to send a message to Bob, she runs a user agent (UA) program to prepare the message and send it to her mail server. The mail server at her site uses a queue (spool) to store messages waiting to be sent. The message needs to be sent through the Internet from Alice's site to Bob's site using a message transfer agent (MTA). Here two message transfer agents are needed: one client and one server. The server needs to run all the time because it does not know when a client will ask for a connection. The client, on the other hand, can be triggered by the system when there is a message in the queue to be sent.

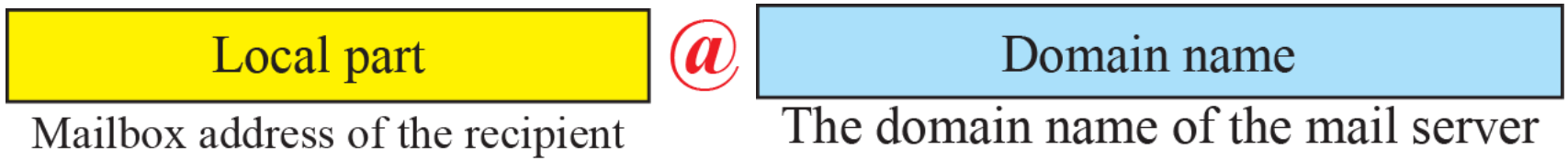


The user agent at the Bob site allows Bob to read the received message. Bob later uses a message access agent client to retrieve the message from a message access agent server running on the second server.

Format of an e-mail



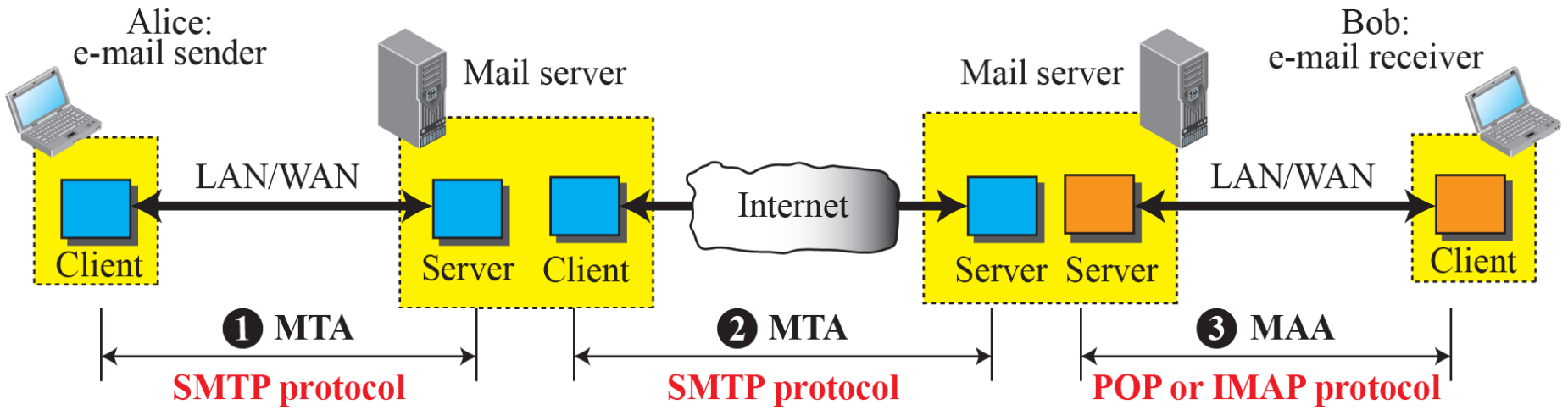
E-mail address



In the Internet, the address consists of two parts:

1. ***Local part:*** defines the name of a special file, called the user mailbox, where all the mail received for a user is stored for retrieval by the message access agent.
2. ***Domain name:*** An organization usually selects one or more hosts to receive and send e-mail; they are sometimes called *mail servers* or *exchangers*.

Protocols used in electronic mail



For MTA, the message needs to be pushed from the client to the server (need a push protocol) → Simple Mail Transfer Protocol (SMTP).

For MAA, the client must pull messages from the server (need a pull protocol) →POP and IMAP.

SMTP Commands

The command is from an MTA client to an MTA server.

Keyword	*Argument(s)*	*Description*
HELO	Sender's host name	Identifies itself
MAIL FROM	Sender of the message	Identifies the sender of the message
RCPT TO	Intended recipient	Identifies the recipient of the message
DATA	Body of the mail	Sends the actual message
QUIT		Terminates the message
RSET		Aborts the current mail transaction
VRFY	Name of recipient	Verifies the address of the recipient
NOOP		Checks the status of the recipient
TURN		Switches the sender and the recipient
EXPN	Mailing list	Asks the recipient to expand the mailing list.
HELP	Command name	Asks the recipient to send information about the command sent as the argument
SEND FROM	Intended recipient	Specifies that the mail be delivered only to the terminal of the recipient, and not to the mailbox
SMOL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *or* the mailbox of the recipient
SMAL FROM	Intended recipient	Specifies that the mail be delivered to the terminal *and* the mailbox of the recipient

SMTP responses (Continued)

The response is from an MTA server to the MTA client

Code	*Description*
	Positive Completion Reply
211	System status or help reply
214	Help message
220	Service ready
221	Service closing transmission channel
250	Request command completed
251	User not local; the message will be forwarded
	Positive Intermediate Reply
354	Start mail input
	Transient Negative Completion Reply
421	Service not available
450	Mailbox not available
451	Command aborted: local error
452	Command aborted; insufficient storage
	Permanent Negative Completion Reply
500	Syntax error; unrecognized command
501	Syntax error in parameters or arguments
502	Command not implemented
503	Bad sequence of commands
504	Command temporarily not implemented
550	Command is not executed; mailbox unavailable
551	User not local
552	Requested action aborted; exceeded storage location
553	Requested action not taken; mailbox name not allowed
554	Transaction failed

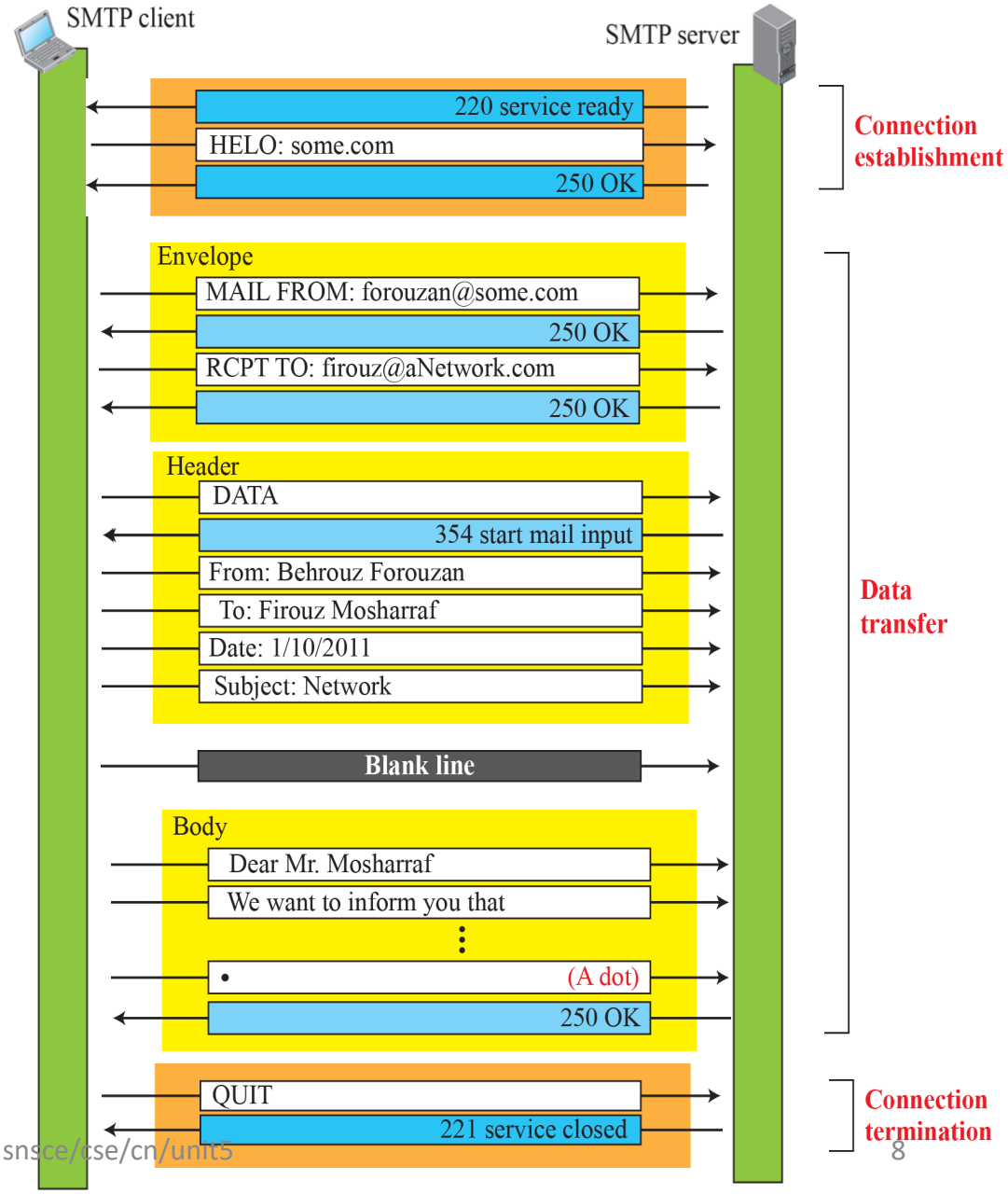
Example

The process of transferring a mail message occurs in three phases: connection establishment, mail transfer, and connection termination. In the figure, we have separated the messages related to the envelope, header, and body in the data transfer section.

Note that the steps in this figure are repeated two times in each e-mail transfer:

- once from the e-mail sender to the local mail server
- once from the local mail server to the remote mail server.

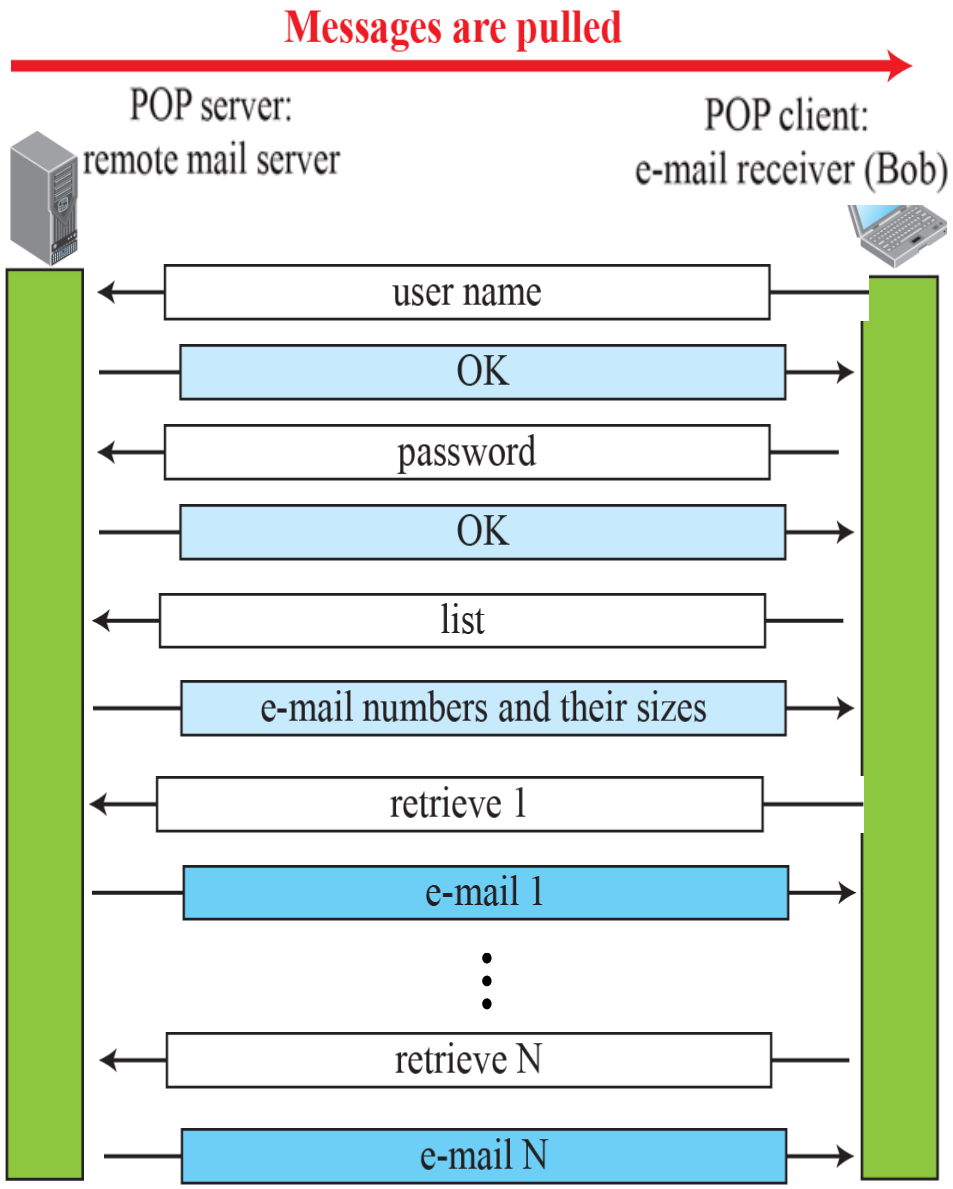
The local mail server, after receiving the whole e-mail message, may spool it and send it to the remote mail server at another time.



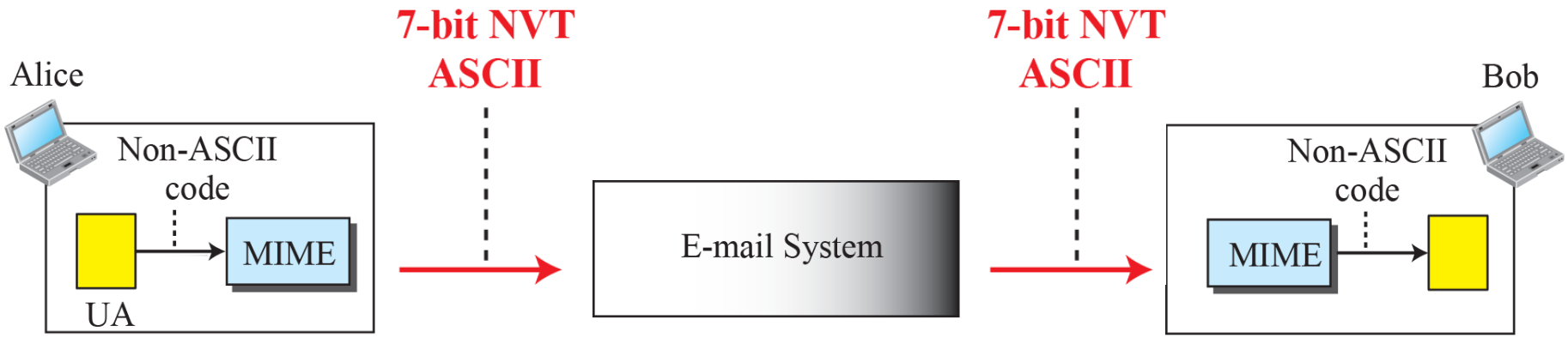
POP3

Post Office Protocol, version 3 (POP3)

Mail access starts with the client when the user needs to download its e-mail from the mailbox on the mail server. The client opens a connection to the server on TCP port 110. It then sends its user name and password to access the mailbox. The user can then list and retrieve the mail messages, one by one.



MIME

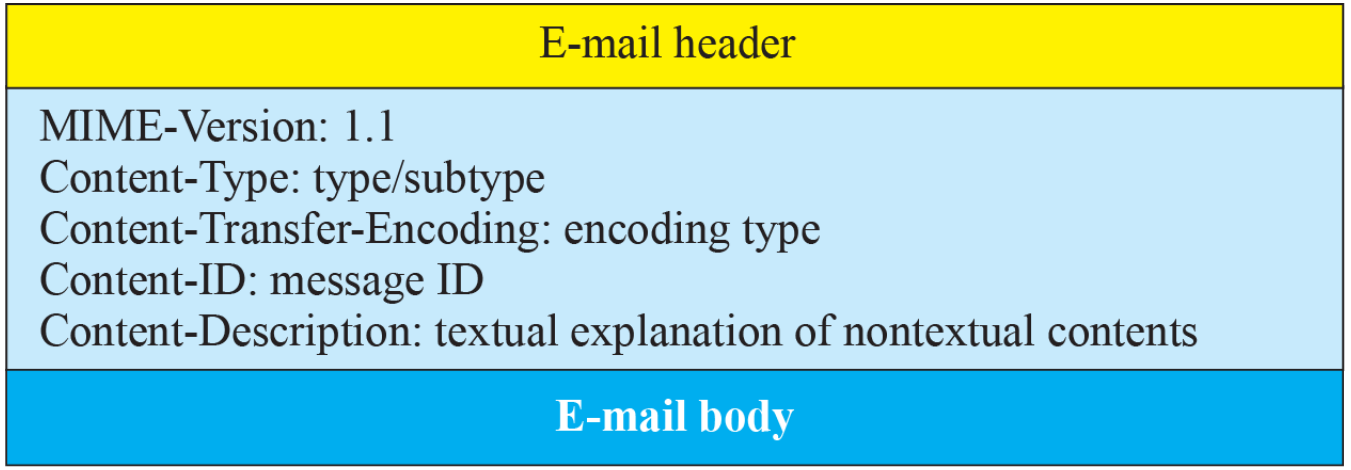


Electronic mail can send messages only in NVT 7-bit ASCII format. It cannot be used for languages other than English or to send binary files or video or audio data.

Multipurpose Internet Mail Extensions (MIME) is a supplementary protocol that transforms non-ASCII data at the sender site to NVT ASCII data and delivers it to the client MTA to be sent through the Internet. The message at the receiving site is transformed back to the original data.

MIME header \

MIME headers

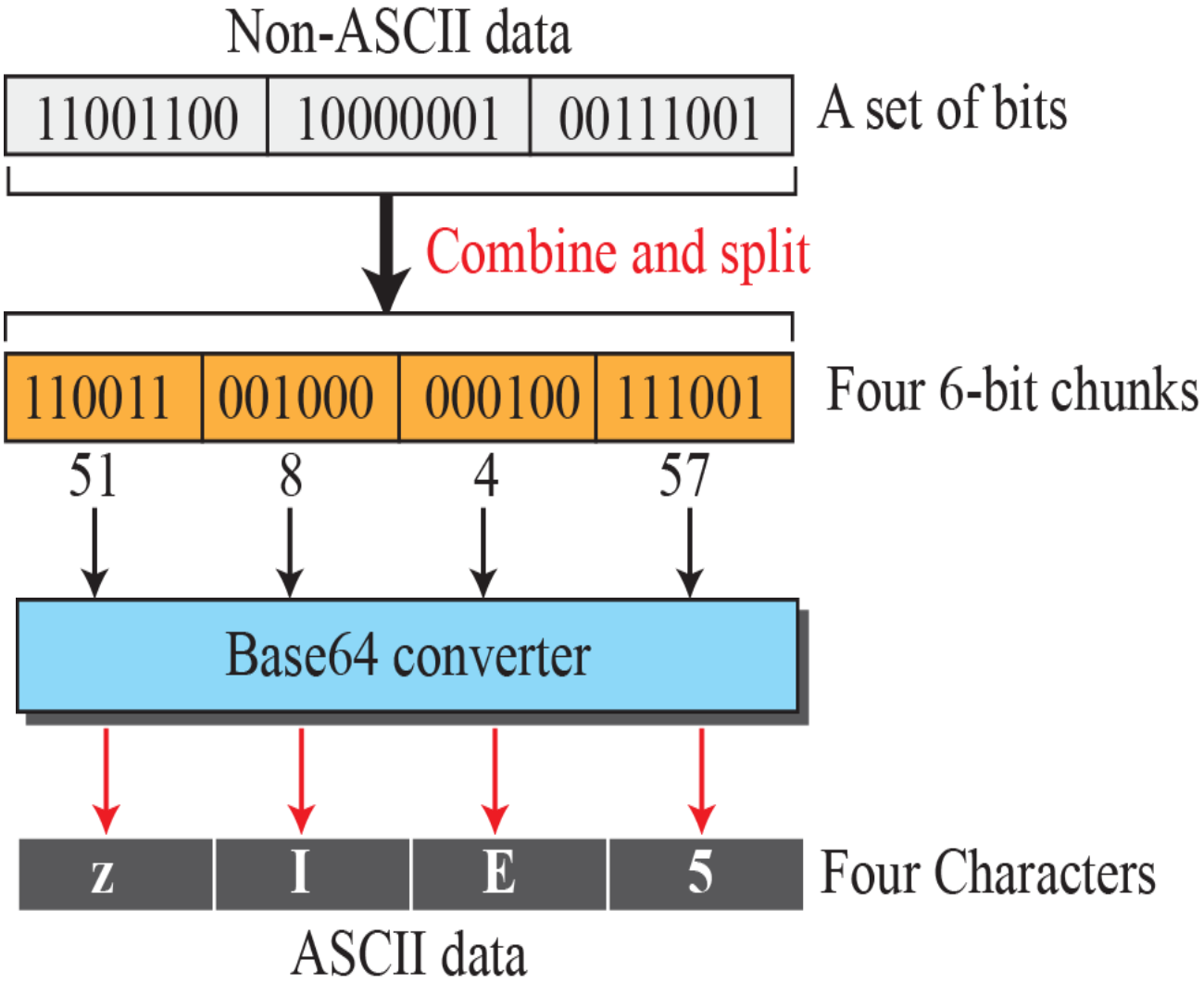


Data Types and Subtypes in MIME

Type	*Subtype*	*Description*
Text	Plain	Unformatted
	HTML	HTML format (see Appendix C)
Multipart	Mixed	Body contains ordered parts of different data types
	Parallel	Same as above, but no order
	Digest	Similar to Mixed, but the default is message/RFC822
	Alternative	Parts are different versions of the same message
Message	RFC822	Body is an encapsulated message
	Partial	Body is a fragment of a bigger message
	External-Body	Body is a reference to another message
Image	JPEG	Image is in JPEG format
	GIF	Image is in GIF format
Video	MPEG	Video is in MPEG format
Audio	Basic	Single channel encoding of voice at 8 KHz
Application	PostScript	Adobe PostScript
	Octet-stream	General binary data (eight-bit bytes)

Base64 conversion

In the Base64 encoding, data is divided into 6-bit chunks. Each 6-bit section is then converted into an ASCII character according to Table 26.10.

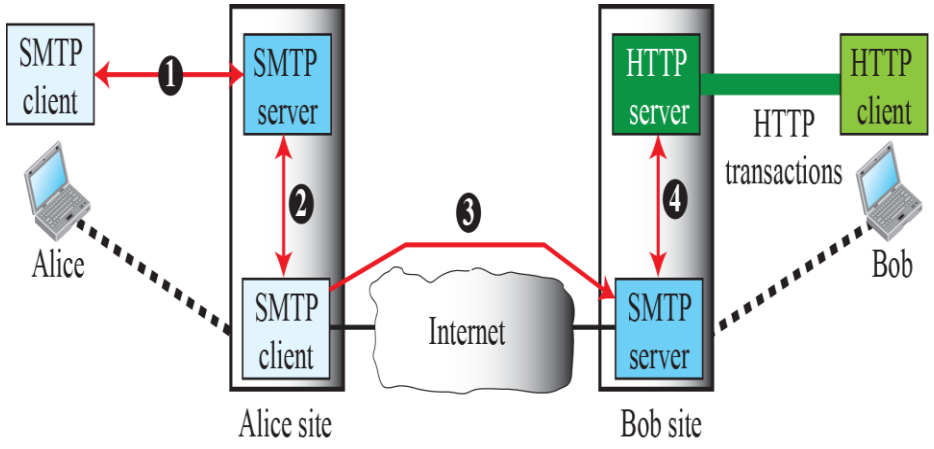


Web-Based Mail

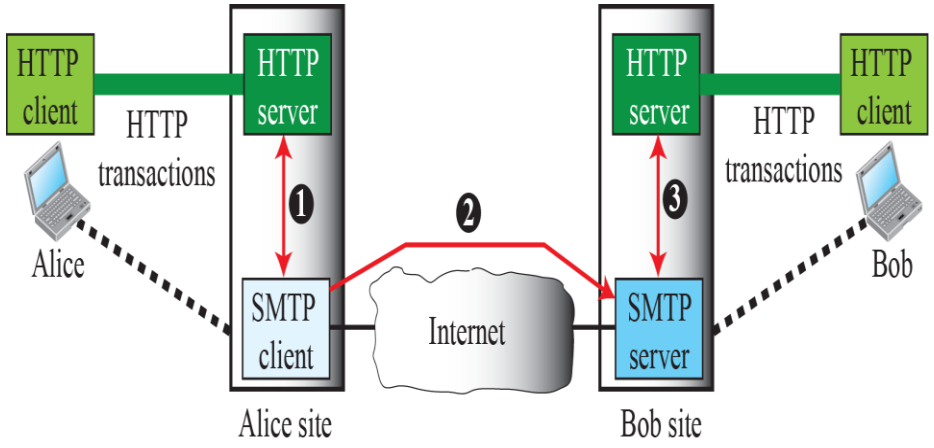
E-mail is such a common application that some websites today provide this service to anyone who accesses the site.
Three common sites are Hotmail, Yahoo, and Google mail. The idea is very simple. Figure 26.22 shows two cases.

Web-based e-mail, cases I and II

Case I: Alice, uses a mail server; Bob, the receiver, has an account on a web-based server.
Mail transfer from Alice's browser to her traditional mail serve, and from the sending mail server to the receiving mail server is through SMTP. However, the message from the receiving server (the web server) to Bob's browser is done through HTTP., instead of using POP3 or IMAP4.
When Bob needs to retrieve his e-mails, he sends a request HTTP message to the website (Hotmail, for example). The website sends a form to be filled in by Bob (the log-in name and the password).
If the log-in name and password match, the list of e-mails is transferred from the web server to Bob's browser in HTML format.
Case 2: In the second case, both Alice and Bob use web servers. Alice sends an HTTP request message to her web server using the name and address of Bob's mailbox as the URL. The server at the Alice site passes the message to the SMTP client and sends it to the server at the Bob site using SMTP protocol. Bob receives the message using HTTP transactions. SMTP protocol. Is used to transfer the message from the server at the Alice site to the server at the Bob site.



Case 1: Only receiver uses HTTP



Case 2: Both sender and receiver use HTTP

E-Mail Security

E-mail exchanges can be secured using two application-layer securities designed in particular for e-mail systems. Two of these protocols, Pretty Good Privacy (PGP) and Secure/Multipurpose Internet Mail Extensions (S/MIME).