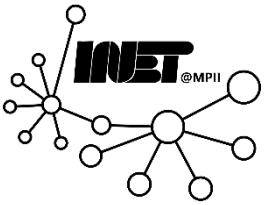
Email

Prof. Anja Feldmann, Ph.D.

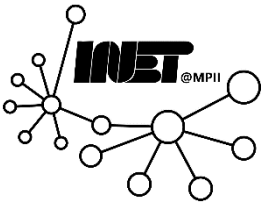
Balakrishnan Chandrasekaran, Ph.D.

(Based on slide deck of Computer Networking, 7th ed., Jim Kurose and Keith Ross.)



Email: Agenda

- Brief introduction
- Accessing emails
  - User agents and mail access protocols
- Storage and retrieval
  - Mail servers and mail server protocols



Email: mode of communication

Lifewire | **The Number of Emails Sent Per Day in 2018 (and 20+ Ot...**

Statistics, extrapolations, and counts by the Radicati Group in March 2018 estimated the number of email accounts worldwide at 3.8 billion — and the number of consumer and business email[illegible]rch firm projects the latter figure[illegible]

281 billion emails!

By contrast, the Radicati Group's estimate for 2015 was 205 billion emails per day, and the estimate for 2009 was 247 billion emails sent per day.

Email: First Electronic Missive

Ray Tomlinson

1971

To: **Ray Tomlinson**

"something like QWERTYUIOP."

(http://www.computinghistory.org.uk/det/6116/First-e-mail-sent-by-Ray-Tomlinson/)

Email: First Email

Questions

[Did you send the first network email?](#)
[Why did you do it?](#)
[Why did you choose the at sign?](#)
[What was the first message?](#)

Did you send the first network email?

As far as I know, yes. However, there are a few qualifications. *Network* should be included because there were many earlier instances of email within a single machine. Computer networks, in any real sense, didn't exist until the ARPANET was built starting in 1969. Dick Watson proposed a form of email in July 1971 (RFC 196). I don't think that was ever implemented. It differed in that the mail was directed to numeric mailboxes. RFC 196 also suggests that the final product would be a printer output (i.e. ink on paper). SNDMSG sent messages to named individuals (computer users).

Why did you do it?

Mostly because it seemed like a neat idea. There was no directive to "go forth and invent email". The ARPANET was a solution looking for a

During the summer and autumn of 1971, I was part of a small group of programmers who were developing a time-sharing system called TENEX that ran on Digital PDP-10 computers. We were supporting a larger group working on natural language. Earlier, I had worked on the Network Control Protocol (NCP) for TENEX and network programs such as an experimental file transfer program called CPYNET.

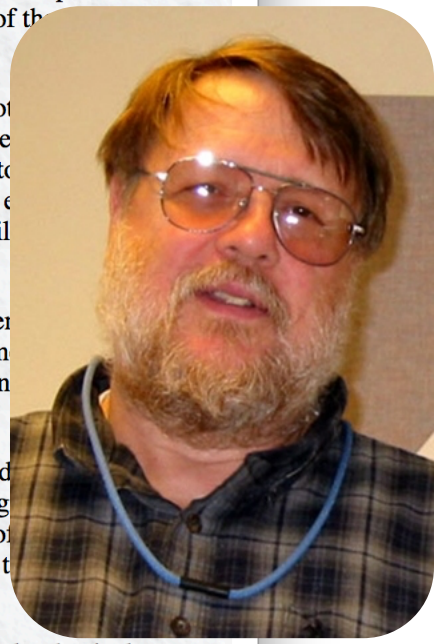
I was making improvements to the local inter-user mail program called SNDMSG. Single-computer electronic mail had existed since at least the early 1960's and SNDMSG was an example of th
allowed a user to compos address, and send a message to other users' mailboxes.

A mailbox was si articular name. It's only special property was its pro
allowed otl That is, they could write more material onto the
mailbox, bu ite what was already there. The idea occurred t
could appen ust as readily as SNDMSG could. SNDMSG could
the code hrough a network connection to remote mail
to appendin ies.

The missin tal CPYNET protocol had no provision for appe
could just send a ding the missing piece was a no-brainer -- just a min
protocol. I don't r etails, but appending to a file was the same as writin
for the mode in which the e was opened.

Next, the CPYNET code was incorporated into SNDMSG. It remained to provide a way to d
mail from network mail. I chose to append an at sign and the host name to the user's (log
frequently asked why I chose the at sign, but the at sign just makes sense. The purpose o
English) was to indicate a unit price (for example, 10 items @ $1.95). I used the at sign t
user was "at" some other host rather than being local.

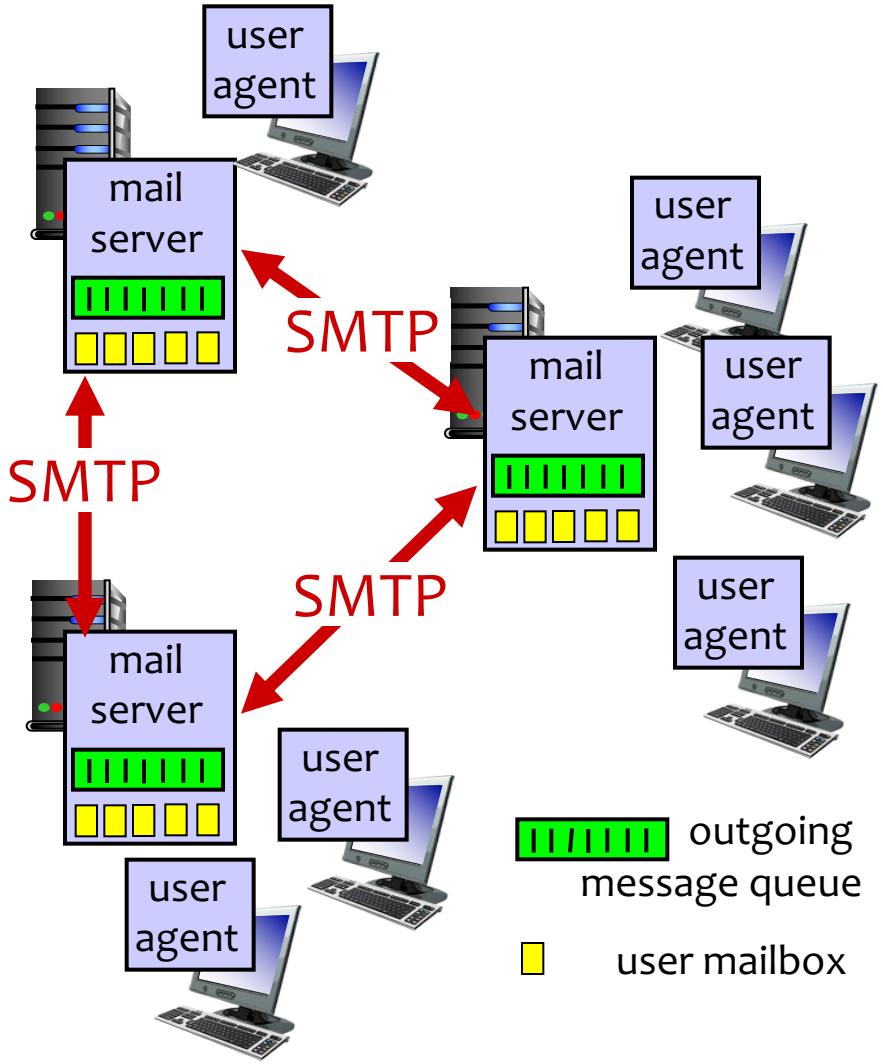
The first message was sent between two machines that were literally side by side. The only physical connection they had (aside from the floor they sat on) was through the ARPANET. I sent a number of test



http://openmap.bbn.com/~tomlinso/ray/firstemailframe.html

Email: Components

- *User Agents*
- *Mail Servers*
- *Simple Mail Transfer Protocol (SMTP)*

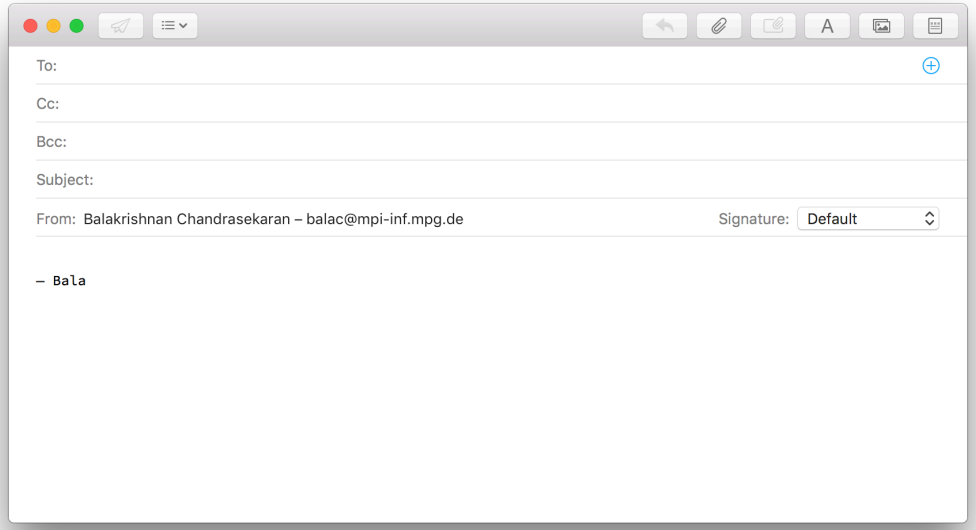


Email: User Agent

"Mail Reader"

- Composing, editing, reading mail messages

  (e.g., Thunderbird, Outlook, Mail)



Outgoing & incoming messages stored on server

Email: Mail Servers

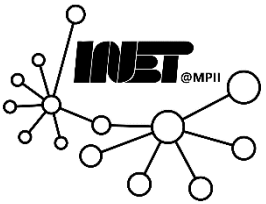
- ***Mailbox*** contains incoming messages for user
- ***Message queue*** of outgoing mail

- ***SMTP protocol*** between mail servers to send email messages
  - *client*: sending mail server
  - *server*: receiving mail server

Email: SMTP

- Uses ***TCP*** to ***reliably*** transfer email message from client to server
  - *port 25*

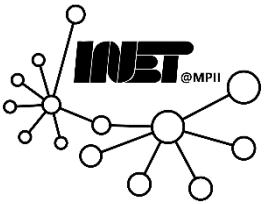
- *direct* transfer: sending server to receiving server
- Three phases of transfer
  - ***Handshaking***
  - ***Transfer of messages***
  - ***Closure***



Email: SMTP

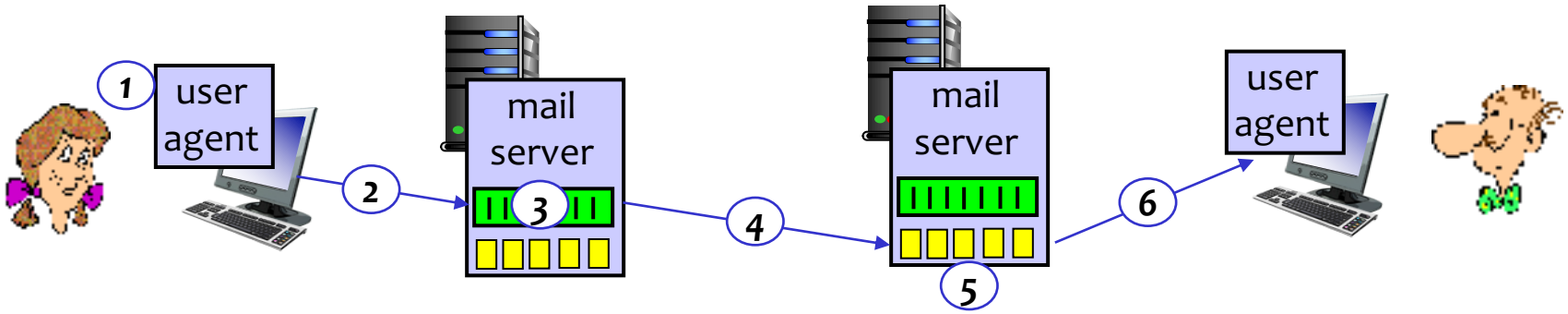
- Command/response interaction (like HTTP)
  - commands: ASCII text
  - response: status code and phrase

- Messages *must* be in *7-bit* ASCII



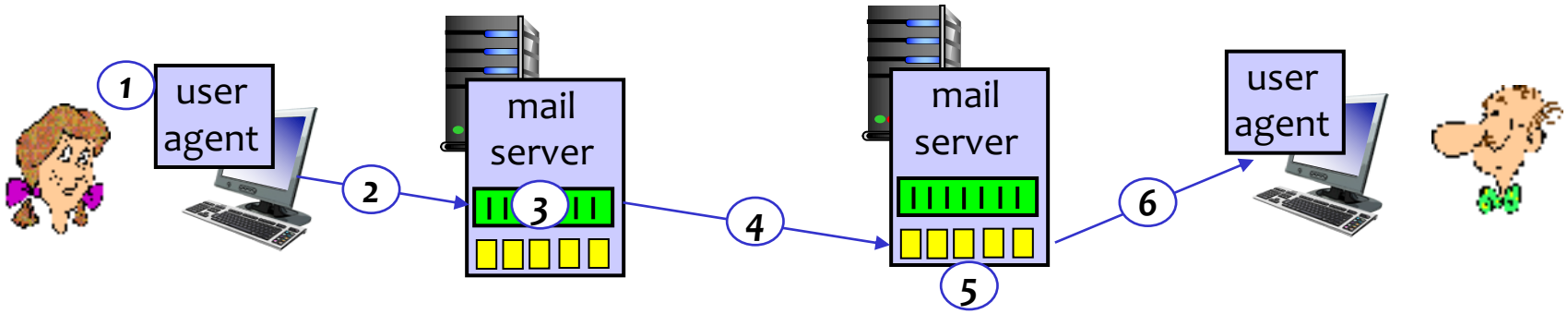
Email: Walkthrough

1. Alice uses *user agent* to compose message *"to" bob@someschool.edu*



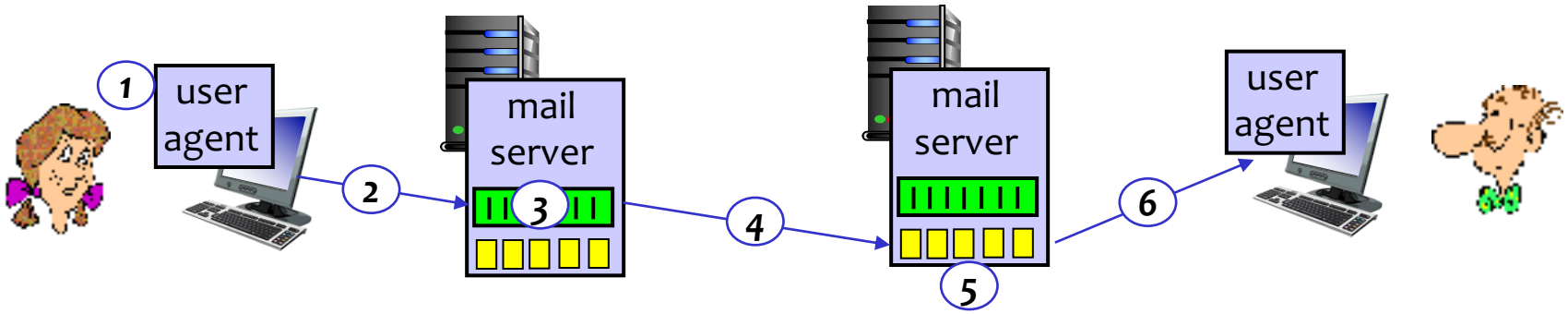
Email: Walkthrough

2. Alice's *user agent* sends message to her *mail server*; message placed in *message queue*.



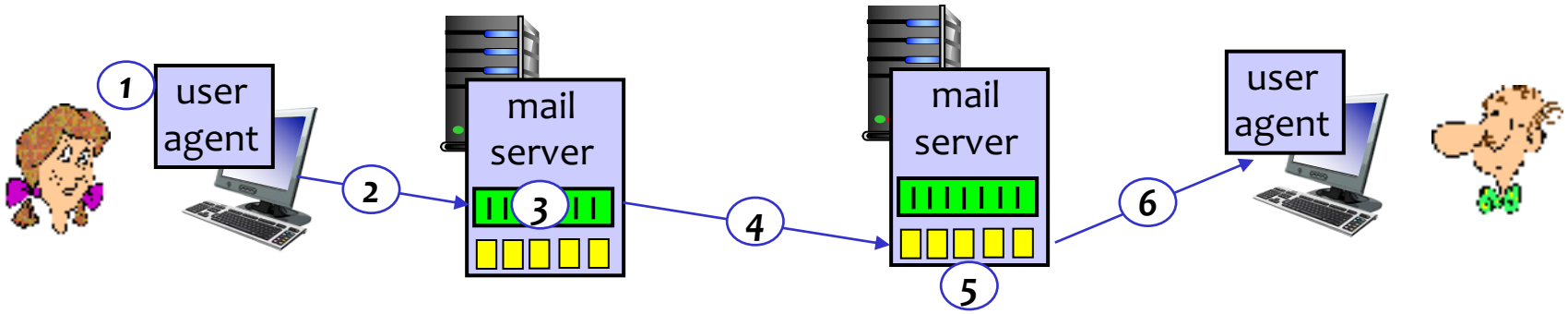
Email: Walkthrough

3. *Client* side of *SMTP* opens *TCP* connection with Bob's *mail server*.



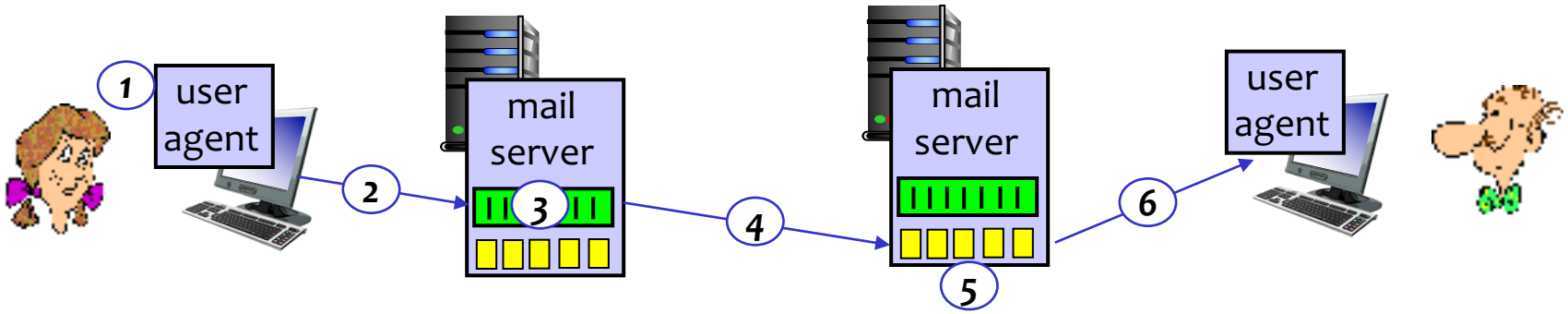
Email: Walkthrough

4. *SMTP client* sends Alice's *message* over the *TCP* connection.



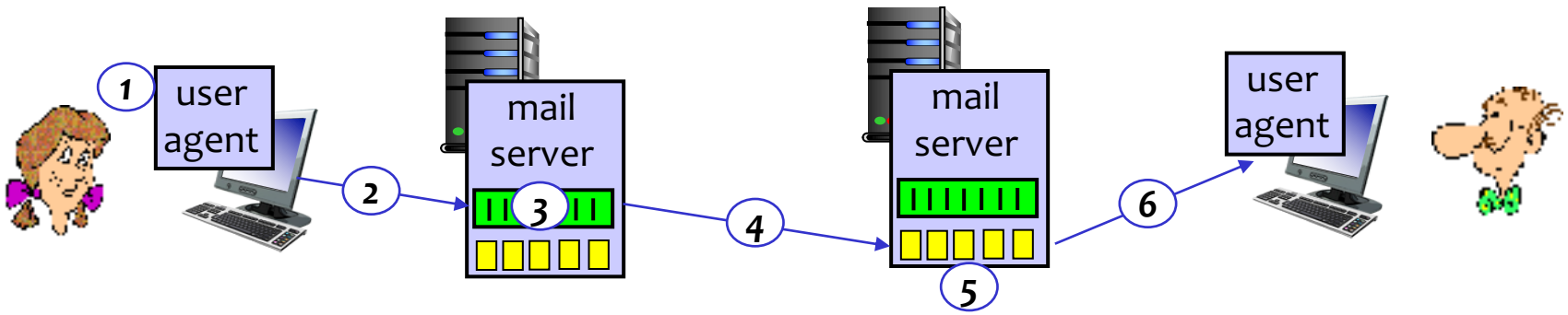
Email: Walkthrough

5. Bob's *mail server* places the *message* in Bob's *mailbox*.



Email: Walkthrough

6. Bob invokes his *user agent* to read *message*.

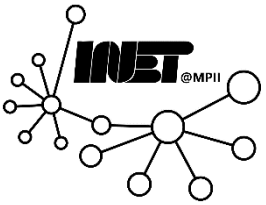


Sample SMTP Interaction

S: 220 hamburger.edu

C: HELO crepes.fr

S: 250 Hello crepes.fr, pleased to meet you



Sample SMTP Interaction

C: *MAIL FROM: <alice@crepes.fr>*

S: *250 alice@crepes.fr... Sender ok*

C: *RCPT TO: <bob@hamburger.edu>*

S: *250 bob@hamburger.edu ... Recipient ok*

C: *DATA*

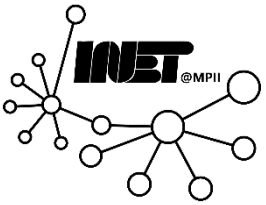
S: *354 Enter mail, end with "." on a line by itself*

C: *Do you like ketchup?*

C: *How about pickles?*

C: *.*

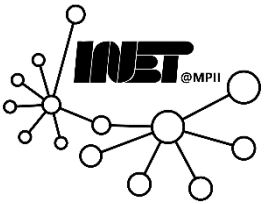
S: *250 Message accepted for delivery*



Sample SMTP Interaction

C: *QUIT*

S: *221 hamburger.edu closing connection*



SMTP: Try it out

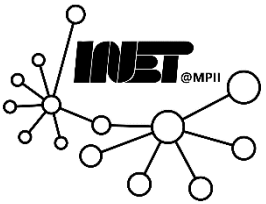
telnet <server-name> 25

- see 220 reply from server
- enter *HELO*, *MAIL FROM*, *RCPT TO*, *DATA*, *QUIT* commands

Lets you send email without using email client (reader)

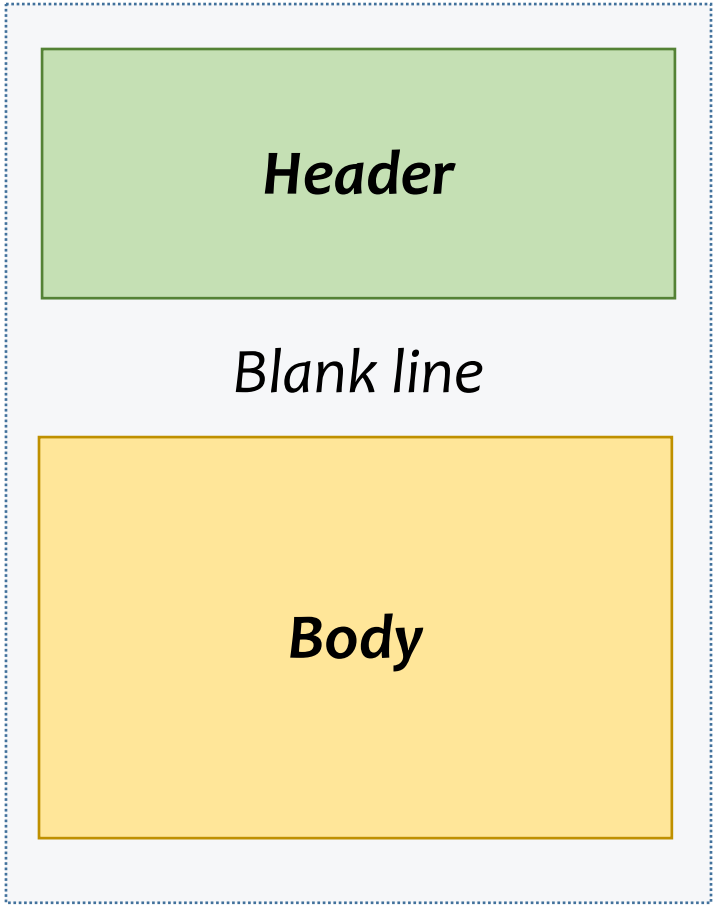
SMTP: Summary

- Uses *persistent* connections
- Requires *message (header & body)* to be in *7-bit ASCII*
- Server uses *CRLF.CRLF* to determine end of message



Mail Message Format

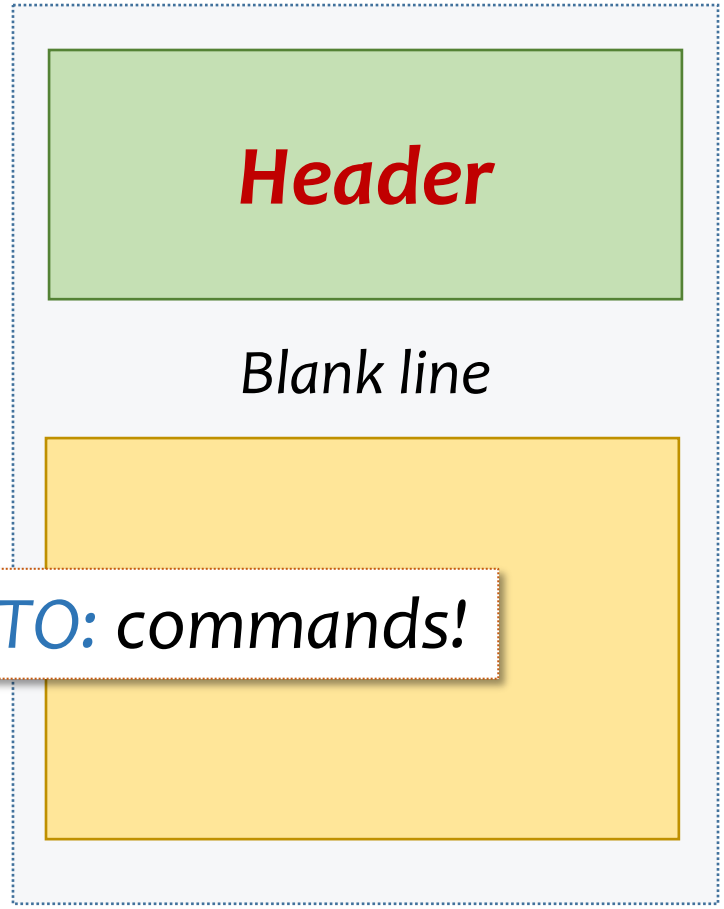
- *RFC 822*
  - standard for text message format



Mail Message Format

Header lines

- *To:*
- *From:*
- *Subject:*

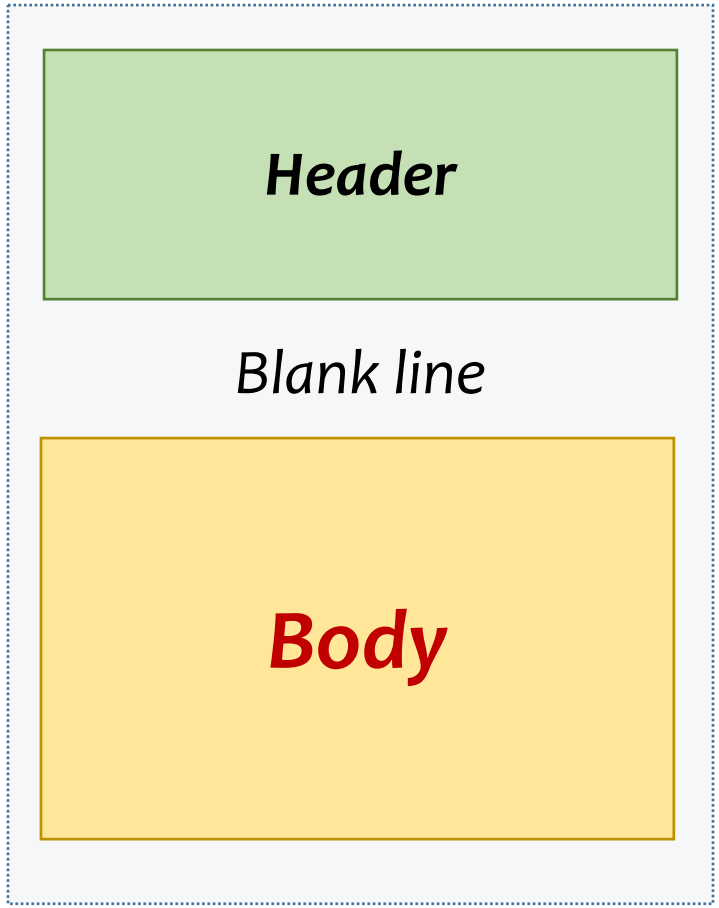


Different from SMTP's MAIL FROM, RCPT TO: commands!

Mail Message Format

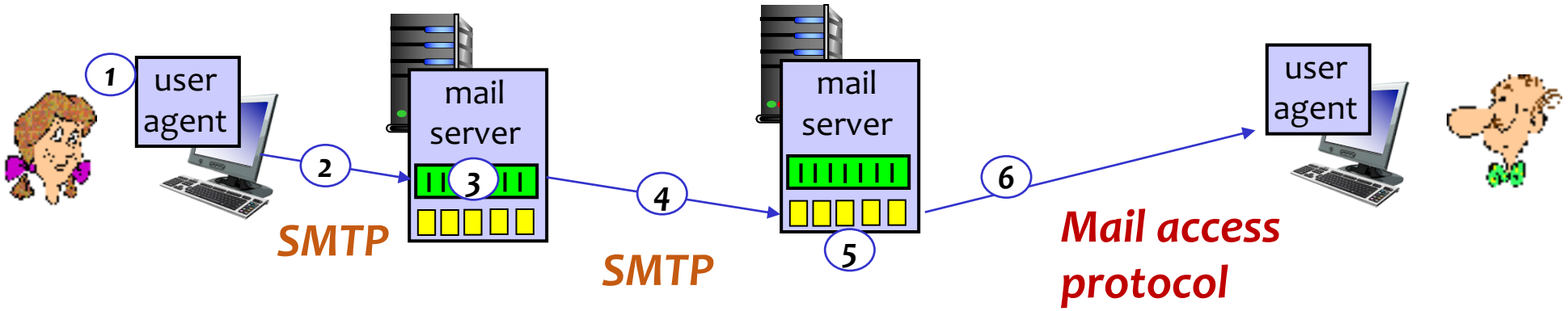
Body

- *"Message"*
- ASCII characters *only*



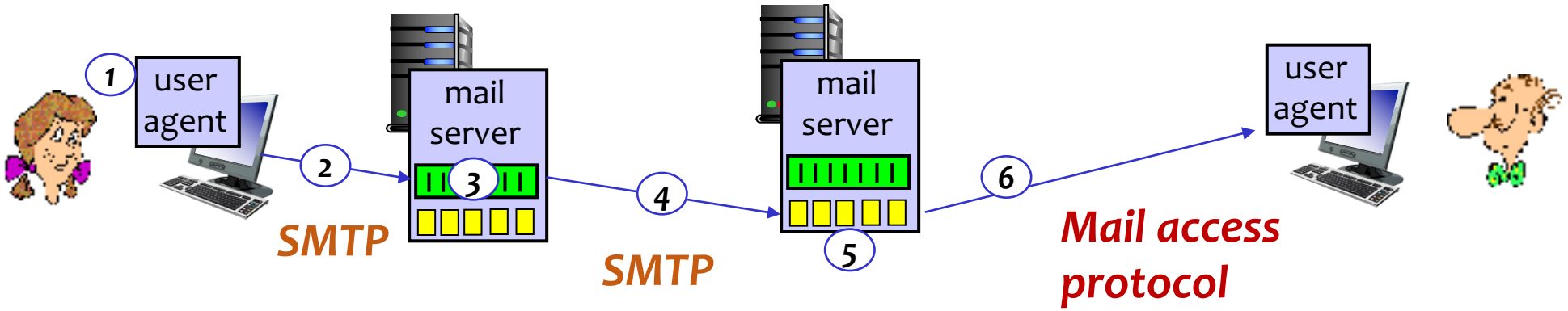
Mail access protocols

- *SMTP*: delivery/storage to receiver's server
- ***Mail access protocol***: retrieval from server



Mail access protocols

- ***Post Office Protocol (POP)*** *[RFC 1939]*
  - authorization, download
- ***Internet Mail Access Protocol (IMAP)*** *[RFC 1730]*
  - more features, including manipulation of stored messages on server
- ***HTTP***
  - gmail, Hotmail, Yahoo! Mail, etc.



POP3 vs IMAP

POP3

Download-and-delete mode

- cannot re-read e-mail if user changes client

Download-and-keep mode

- copies of messages on different clients

Stateless across sessions

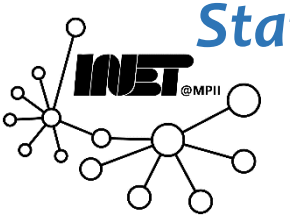
IMAP

Keeps all messages in one place— at server

Organize messages in folders

Keeps user state across sessions

- names of folders & mappings between message IDs and folder names



POP3 vs IMAP

POP3

Download-and-delete mode

- cannot re-read e-mail if user changes client

Download-and-keep mode

- copies of messages on different clients

Stateless across sessions

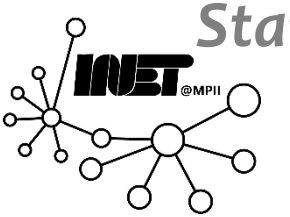
IMAP

Keeps all messages in one place— at server

Organize messages in folders

Keeps user state across sessions

- names of folders & mappings between message IDs and folder names



Email: Recap

- Important form of communication
- User agents and mail servers
- Mail server protocol
  - *SMTP*
- Mail access protocols
  - *POP3 and IMAP*