

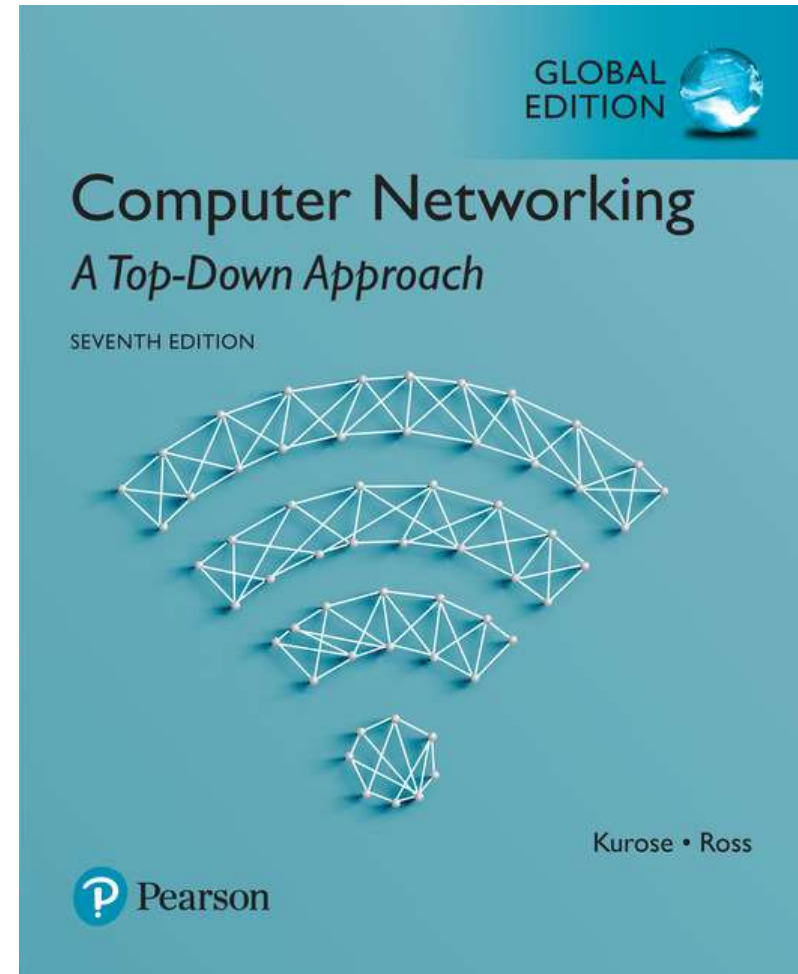
# 제9강 이메일 프로토콜

## *Computer Networking: A Top Down Approach*

컴퓨터 네트워크  
(2019년 1학기)

박승철교수

한국기술교육대학교  
컴퓨터공학부



# Chapter 2: outline

2.1 principles of network applications

2.2 Web and HTTP

2.3 electronic mail

- SMTP, POP3, IMAP

2.4 DNS

2.5 P2P applications

2.6 video streaming and content distribution networks

2.7 socket programming with UDP and TCP

# Pre-study Test :

1) 다음 중 이메일 서비스를 위한 프로토콜이 아닌 것은?

- ① SMTP
- ② POP3
- ③ SNMP
- ④ IMAP

2) 다음 중 이메일 서버가 송신 메시지를 저장하는 자료구조는?

- ① 메일박스
- ② 메시지 큐
- ③ 파일 서버
- ④ 메시지 서버

3) 다음 중 메일 서버와 메일 서버간에 메시지 전송을 위한 프로토콜은?

- ① SMTP
- ② POP3
- ③ HTTP
- ④ IMAP

4) 다음 중 이메일 서비스를 위한 프로토콜이 아닌 것은?

- ① SMTP
- ② POP3
- ③ SNMP
- ④ IMAP

5) 메일 서버가 사용하는 포트 번호는 얼마인가?

- ① 80
- ② 25
- ③ 20
- ④ 21

6) 다음 중 SMTP의 메시지에 대한 설명 중 맞는 것은?

- ① ASCII 문자로만 표현되어야 한다.
- ② ASCII 그래픽 문자로만 표현되어야 한다.
- ③ 모든 문자로 표현될 수 있다.
- ④ 비디오를 포함한 모든 정보로 표현될 수 있다.

7) HTTP와 SMTP의 비교가 틀린 것은?

- ① HTTP는 pull 프로토콜이고, SMTP는 push 프로토콜이다.
- ② SMTP는 다수의 메시지를 포함할 수 있고 HTTP는 단일 메시지를 포함한다.
- ③ SMTP와 HTTP 모두 ASCII 문자로 헤더 정보를 표현한다.
- ④ SMTP와 HTTP 모두 ASCII 문자로 메시지를 표현한다.

8) 다음 중 메일 접근 프로토콜(mail access protocol)이 아닌 것은?

- ① SMTP
- ② POP3
- ③ HTTP
- ④ IMAP

9) 다음 중 SMTP로 멀티미디어 메시지 전송을 가능하게 하는 것은?

- ① HTTP
- ② MIME
- ③ DNS
- ④ CODEC

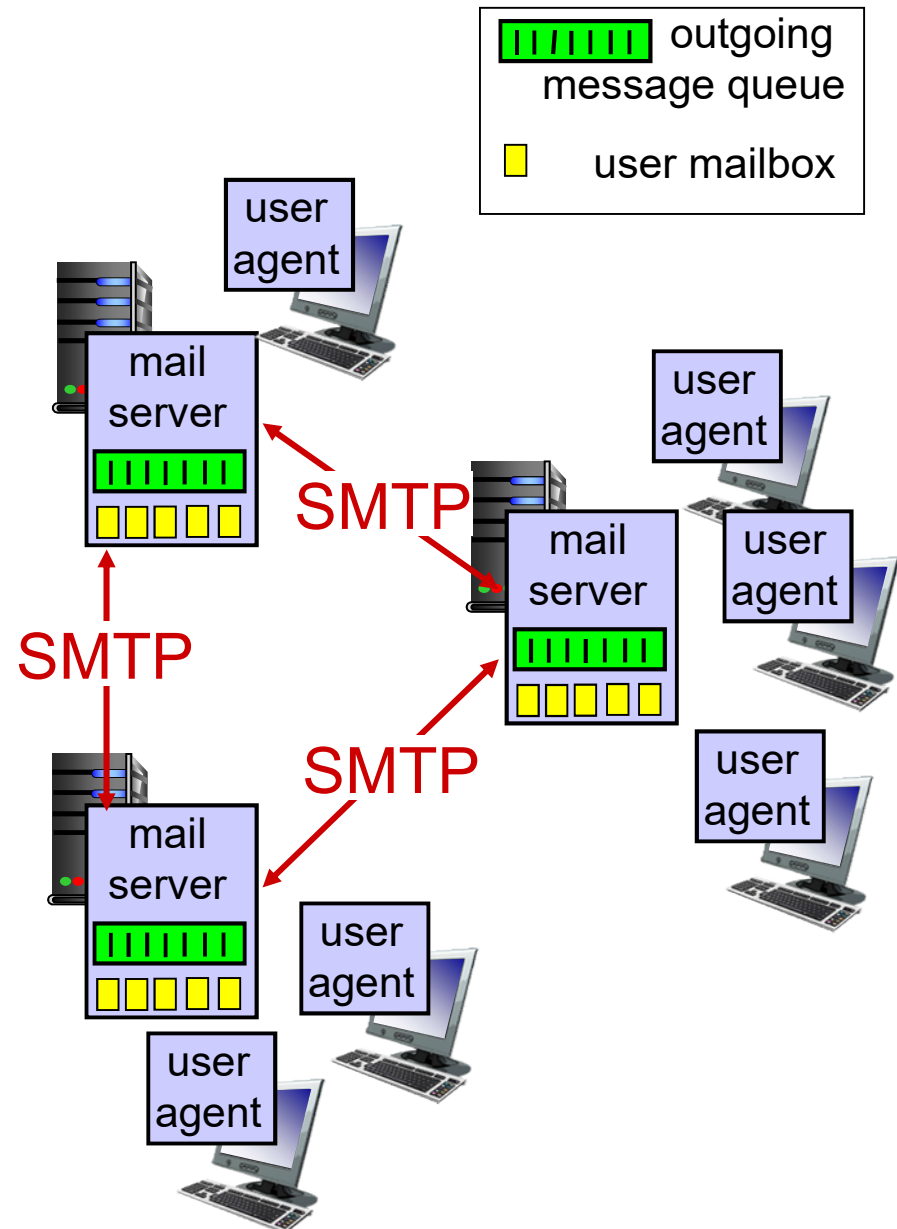
# Electronic mail

## *Three major components:*

- user agents
- mail servers
- simple mail transfer protocol: SMTP

## *User Agent*

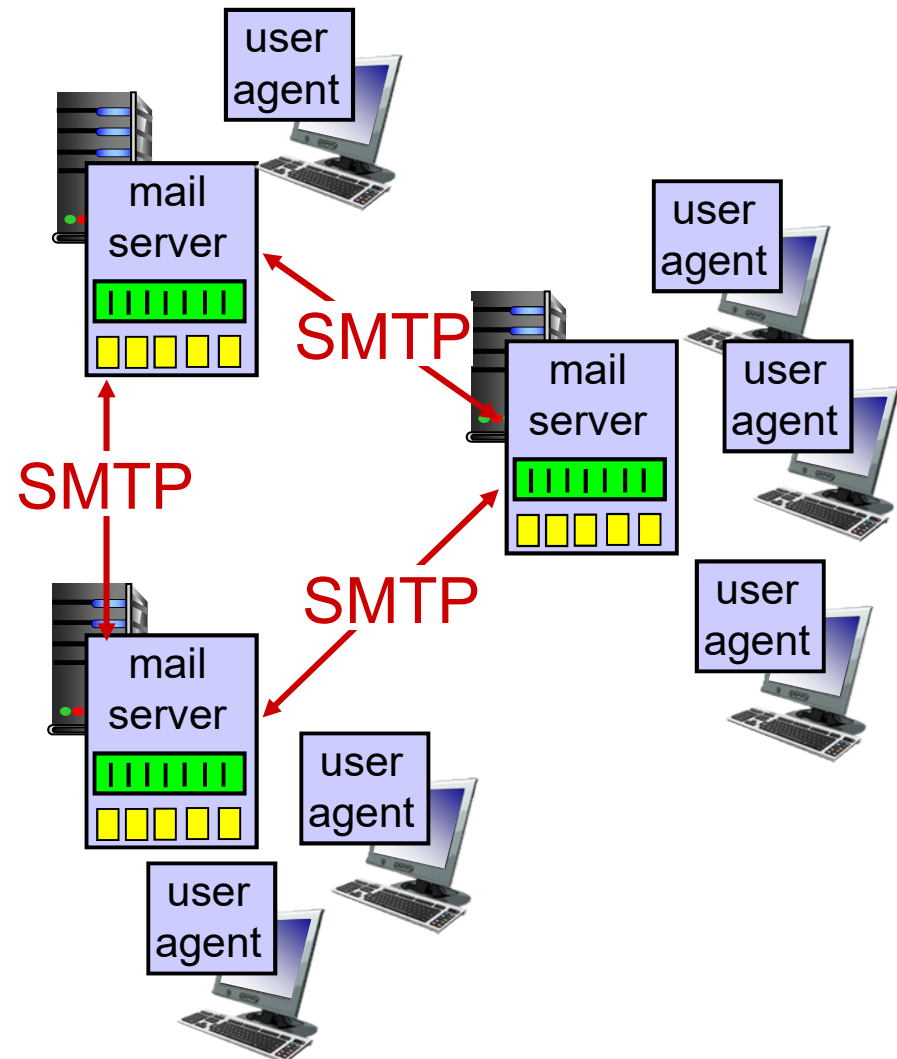
- a.k.a. “mail reader”
- composing, editing, reading mail messages
- e.g., Outlook, Thunderbird, iPhone mail client
- outgoing, incoming messages stored on server



# Electronic mail: mail servers

## mail servers:

- *mailbox* contains incoming messages for user
- *message queue* of outgoing (to be sent) mail messages
- *SMTP protocol* between mail servers to send email messages
  - client: sending mail server
  - “server”: receiving mail server



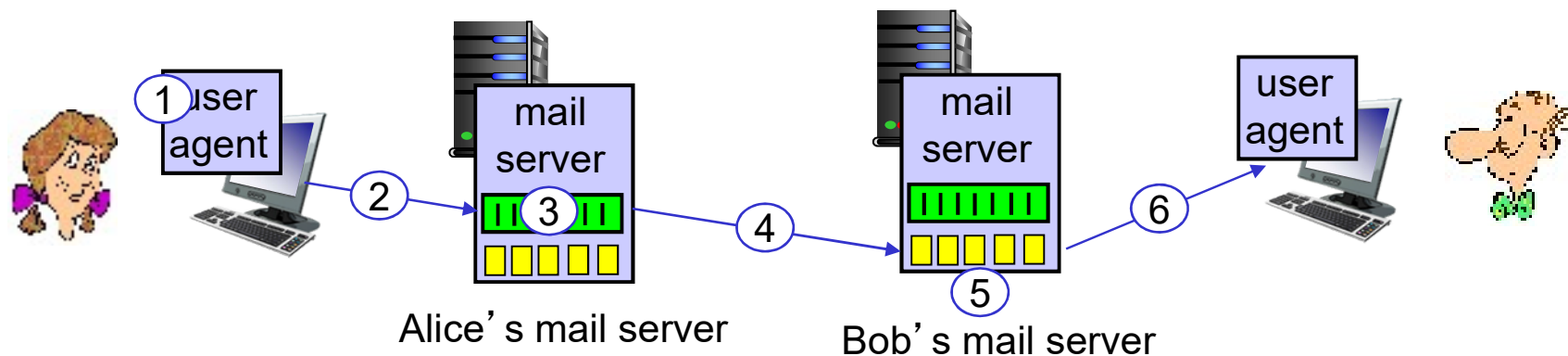
# Electronic Mail: SMTP [RFC 2821]

- 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 (greeting)
  - transfer of messages
  - closure
- command/response interaction (like HTTP)
  - **commands:** ASCII text
  - **response:** status code and phrase
- **messages must be in 7-bit ASCII**



# Scenario: Alice sends message to Bob

- 1) Alice uses UA to compose message "to" `bob@someschool.edu`
- 2) Alice's UA sends message to her mail server; message placed in message queue
- 3) client side of SMTP opens TCP connection with Bob's mail server
- 4) SMTP client sends Alice's message over the TCP connection
- 5) Bob's mail server places the message in Bob's mailbox
- 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
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
C: QUIT
S: 221 hamburger.edu closing connection
```

## Try SMTP interaction for yourself:

- `telnet servername 25`
- see 220 reply from server
- enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

above lets you send email without using email client (reader)

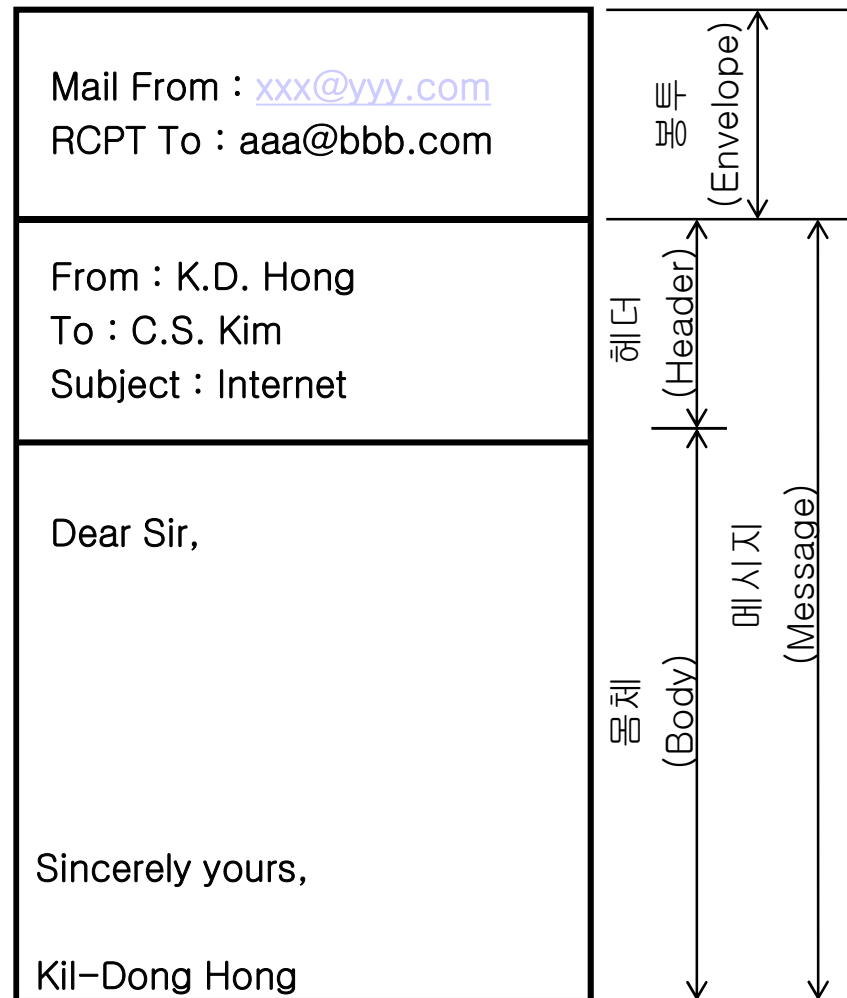
# SMTP: final words

- SMTP uses persistent connections
- SMTP requires message (header & body) to be in 7-bit ASCII
- SMTP server uses CRLF.CRLF to determine end of message

## *comparison with HTTP:*

- HTTP: pull
- SMTP: push
- both have ASCII command/response interaction, status codes
- HTTP: each object encapsulated in its own response message
- SMTP: multiple objects sent in multipart message

# 메일 메시지 포맷



메시지 내에  
다수의 헤더라인  
포함

# 메일 메시지 포맷

- 헤더 키워드(Keyword)
  - From : 송신자
  - To : 수신자
  - Cc : 복사본 수신자
  - Date : 메시지가 전송된 날짜
  - Subject : 메시지 주제
  - Reply-To : 응답이 전송될 주소
  - 기타.
- 메시지 몸체(Message Body)
  - 헤더와 몸체(Body)는 라인 종단(<CR><LF>) 제어 문자만 포함되는 빈 라인(Empty Line)으로 구분
  - 몸체에는 메시지의 내용이 라인 형태로 포함됨
  - 각 라인은 라인 종단 제어 문자(<CR><LF>)로 종료
  - 메시지는 단지 마침표(.)만으로 이루어지는 라인에 의해 종료

# 메일 메시지 포맷

## ■ 멀티파트 메시지 구성 예

```
From: scpark@kut.ac.kr
To: kim@abc.com
Subject: 사진
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=StartOfNextPart
--StartOfNextPart
Dear Kim, Please find the picture enclosed.
--StartOfNextPart
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
base64 encoded data .....
.....base64 encoded data
--StartOfNextPart
How do feel the picture?
```

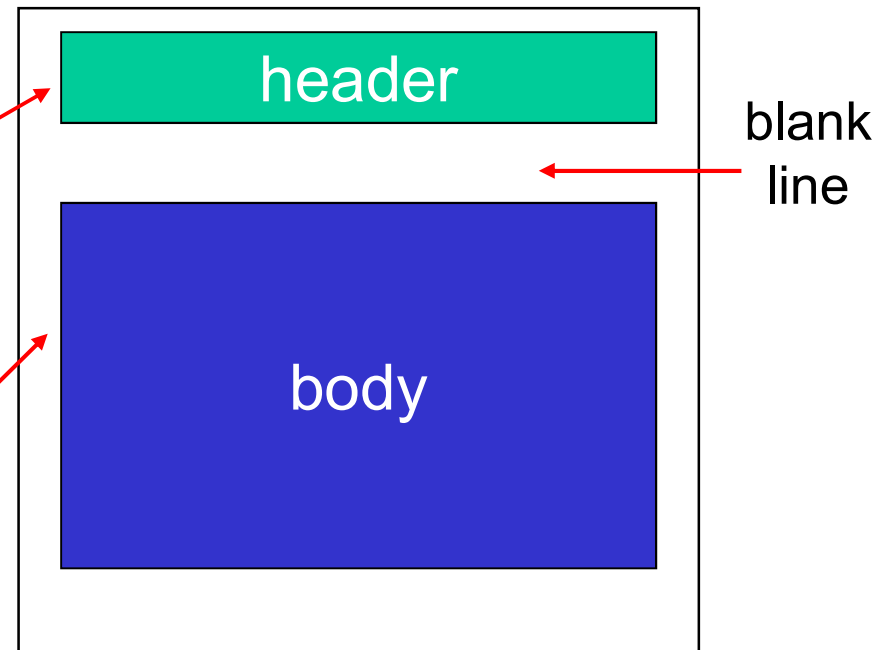
메시지 내에  
다수의 헤더라인  
포함  
→ 메일 내용에  
<CR><LF>와 같은  
내용이 포함된 경우  
어떻게 처리?

# Mail message format

SMTP: protocol for exchanging email messages

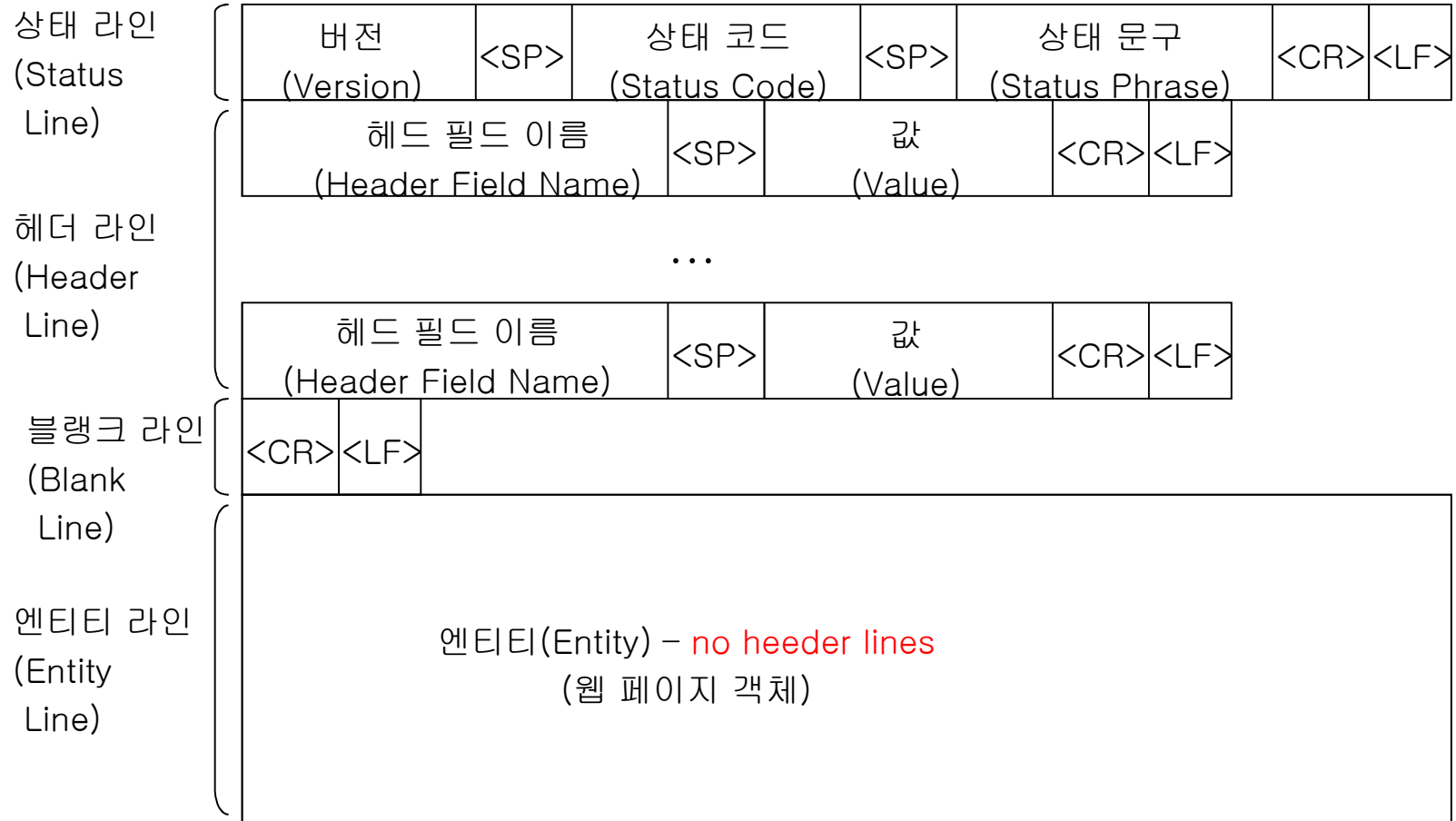
RFC 822: standard for text message format:

- header lines, e.g.,
  - To:
  - From:
  - Subject:
- different from* SMTP MAIL FROM, RCPT TO: commands!
- Body: the “message”
  - ASCII characters only

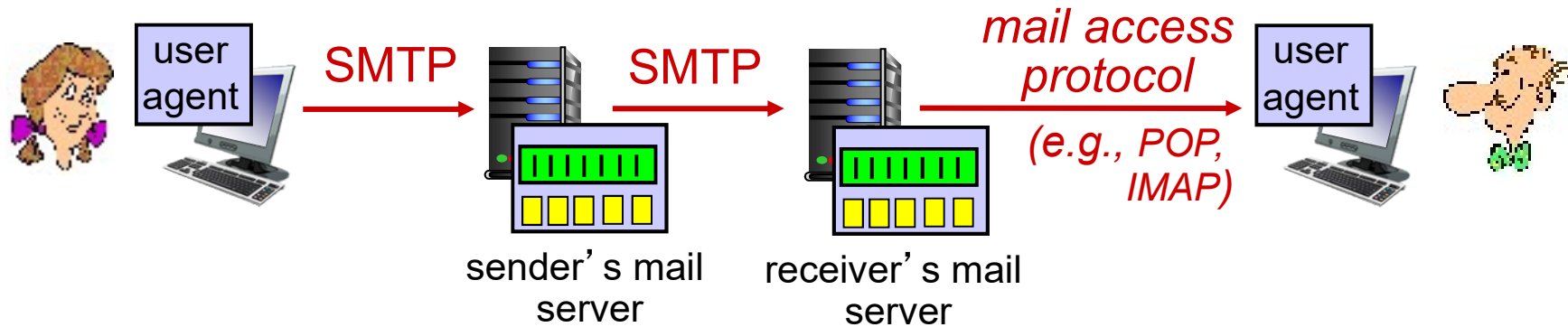




# HTTP 메시지 구성



# Mail access protocols



- **SMTP:** delivery/storage to receiver's server
- mail access protocol: retrieval from server
  - **POP:** Post Office Protocol [RFC 1939]: authorization, download
  - **IMAP:** Internet Mail Access Protocol [RFC 1730]: more features, including manipulation of stored messages on server
  - **HTTP:** gmail, Hotmail, Yahoo! Mail, etc.

# POP3 protocol

## *authorization phase*

- client commands:
  - **user**: declare username
  - **pass**: password
- server responses
  - **+OK**
  - **-ERR**

## *transaction phase, client:*

- **list**: list message numbers
- **retr**: retrieve message by number
- **dele**: delete
- **quit**

```
S: +OK POP3 server ready
C: user bob
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

# POP3 (more) and IMAP

## *more about POP3*

- previous example uses POP3 “download and delete” mode
  - Bob cannot re-read e-mail if he changes client
- POP3 “download-and-keep”: copies of messages on different clients
- POP3 is stateless across sessions

## *IMAP*

- keeps all messages in one place: at server
- allows user to organize messages in folders
- keeps user state across sessions:
  - names of folders and mappings between message IDs and folder name

# After-study Test :

1) 다음 중 이메일 서비스를 위한 프로토콜이 아닌 것은?

- ① SMTP
- ② POP3
- ③ SNMP
- ④ IMAP

2) 다음 중 이메일 서버가 송신 메시지를 저장하는 자료구조는?

- ① 메일박스
- ② 메시지 큐
- ③ 파일 서버
- ④ 메시지 서버

3) 다음 중 메일 서버와 메일 서버간에 메시지 전송을 위한 프로토콜은?

- ① SMTP
- ② POP3
- ③ HTTP
- ④ IMAP

4) 다음 중 이메일 서비스를 위한 프로토콜이 아닌 것은?

- ① SMTP
- ② POP3
- ③ SNMP
- ④ IMAP

5) 메일 서버가 사용하는 포트 번호는 얼마인가?

- ① 80
- ② 25
- ③ 20
- ④ 21

6) 다음 중 SMTP의 메시지에 대한 설명 중 맞는 것은?

- ① ASCII 문자로만 표현되어야 한다.
- ② ASCII 그래픽 문자로만 표현되어야 한다.
- ③ 모든 문자로 표현될 수 있다.
- ④ 비디오를 포함한 모든 정보로 표현될 수 있다.

7) HTTP와 SMTP의 비교가 틀린 것은?

- ① HTTP는 pull 프로토콜이고, SMTP는 push 프로토콜이다.
- ② SMTP는 다수의 메시지를 포함할 수 있고 HTTP는 단일 메시지를 포함한다.
- ③ SMTP와 HTTP 모두 ASCII 문자로 헤더 정보를 표현한다.
- ④ SMTP와 HTTP 모두 ASCII 문자로 메시지를 표현한다.

8) 다음 중 메일 접근 프로토콜(mail access protocol)이 아닌 것은?

- ① SMTP
- ② POP3
- ③ HTTP
- ④ IMAP

9) 다음 중 SMTP로 멀티미디어 메시지 전송을 가능하게 하는 것은?

- ① HTTP
- ② MIME
- ③ DNS
- ④ CODEC