Application Layer: E-mail, SMTP, and IMAP$_s$

Study-Ready Notes

Andrew Photinakis

November 15, 2025

Contents

1 Introduction to Application Layer

- Application layer is the top layer in network protocol stack
- Provides services directly to user applications
- Key protocols: HTTP, SMTP, IMAP, DNS
- Socket programming enables application development

[Summary: The application layer serves as the interface between network services and user applications, implementing protocols like HTTP for web, SMTP for email, and DNS for name resolution.]

2 E-mail Infrastructure

2.1 Three Major Components

1. **User Agents**
   - Also known as ”mail readers”
   - Functions: composing, editing, reading mail messages
   - Examples: Outlook, iPhone mail client
   - Stores outgoing and incoming messages on server
2. **Mail Servers**
   - Contains user mailboxes for incoming messages
   - Maintains message queue for outgoing mail
   - Multiple servers communicate via SMTP
3. **Simple Mail Transfer Protocol (SMTP)**
   - Protocol for transferring email between servers
   - Uses TCP for reliable delivery
   - Operates on port 25

2.2 E-mail Architecture Diagram

[Mnemonic: USA - User agents, Servers, SMTP Architecture] [Summary: Email infrastructure consists of user agents for interface, mail servers for storage and queuing, and SMTP protocol for server-to-server communication.]

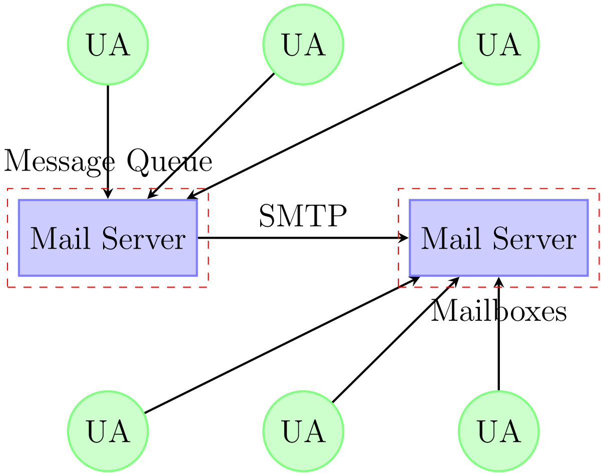

Figure 1: E-mail system architecture showing user agents (UA), mail servers, and SMTP protocol flow

3 Simple Mail Transfer Protocol (SMTP)

3.1 SMTP Protocol Overview

- Defined in RFC 5321
- Uses TCP for reliable transfer on port 25
- Direct transfer: sending server acts as client to receiving server
- Three-phase transfer process:
  1. SMTP handshaking (greeting)
  2. SMTP transfer of messages
  3. SMTP closure
- Command/response interaction using ASCII text

3.2 SMTP Session Timeline

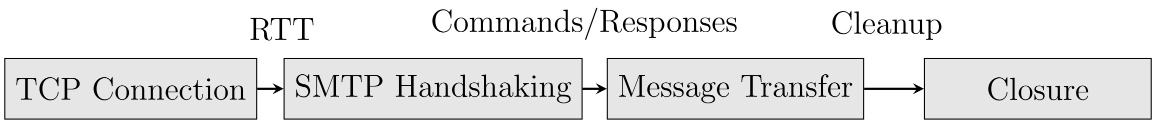

Figure 2: SMTP protocol phases showing the sequence of operations during email transfer

3.3 Email Delivery Scenario: Alice to Bob

1. Alice uses UA to compose email to bob@someschool.edu
2. Alice's UA sends message to her mail server using SMTP
3. Message placed in outgoing message queue
4. SMTP client at Alice's server opens TCP connection with Bob's server
5. SMTP client sends Alice's message over TCP connection
6. Bob's server places message in Bob's mailbox
7. Bob uses UA to read message via mail access protocol

[Concept Map: User Agent → Outgoing Server → SMTP → Incoming Server → Mailbox → User Agent]

4 SMTP Protocol Details

4.1 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
```

4.2 SMTP Observations and Comparisons

4.2.1 Comparison with HTTP

4.2.2 Key SMTP Characteristics

- **Push Protocol**: Sending server initiates transfer
- **7-bit ASCII**: Messages must be in 7-bit ASCII format

SMTP	HTTP
Client push protocol	Client pull protocol
ASCII command/response interaction	ASCII command/response interaction
Multiple objects in multipart message	Each object in separate response message
Persistent connections	Both persistent and non-persistent
7-bit ASCII requirement for messages	No encoding restrictions
CRLF.CRLF marks end of message	Content-Length or chunked encoding

Table 1: Comparison between SMTP and HTTP protocols

- **Persistent Connections**: Multiple messages can be sent in single connection
- **Message Delimiter**: CRLF.CRLF indicates end of message

[Summary: SMTP uses a push model with persistent connections, requires 7-bit ASCII encoding, and follows a strict command-response sequence for reliable email transfer between servers.]

5 Mail Message Format

5.1 RFC Standards

- **RFC 5321**: Defines SMTP protocol (equivalent to RFC 7231 for HTTP)
- **RFC 2822**: Defines syntax for email message format (equivalent to HTML for web)

5.2 Message Structure

```
Header
To: bob@hamburger.edu
From: alice@crepes.fr
Subject: Lunch plans

Body
Let's meet at the usual place at noon.
.
```

5.2.1 Header Fields

- **To**: Recipient's email address

- **From**: Sender's email address
- **Subject**: Brief description of message content
- Note: These are different from SMTP commands MAIL FROM and RCPT TO

5.2.2 Body Format

- Contains the actual message content
- Originally ASCII characters only
- Separated from headers by blank line

[Mnemonic: HBS - Headers, Blank line, Body Structure]

6 Mail Access Protocols

6.1 Email Retrieval Architecture

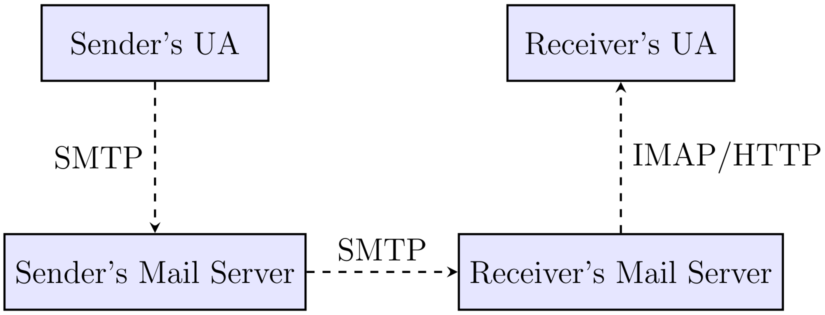

Figure 3: Email flow showing SMTP for delivery and access protocols for retrieval

6.2 Mail Access Protocols

- **SMTP**: Used for delivery and storage to receiver's server
- **Mail Access Protocols**: Used for retrieval from server to user agent

6.2.1 IMAP (Internet Mail Access Protocol)

- Defined in RFC 3501
- Messages stored on server
- Provides retrieval, deletion, and folder management
- Maintains messages on server after reading

6.2.2 HTTP-based Email

- Used by webmail services: Gmail, Hotmail, Yahoo!Mail
- Web-based interface on top of SMTP (sending) and IMAP/POP (retrieval)
- Provides rich user interface through web browser

[Summary: While SMTP handles server-to-server delivery, mail access protocols like IMAP and HTTP handle retrieval from servers to user agents, with IMAP maintaining server storage and HTTP providing web interfaces.]

7 Key Concepts Review

7.1 Important Definitions

- **User Agent**: Mail client software for composing, reading, and managing email
- **Mailbox**: Storage area on mail server for incoming user messages
- **Message Queue**: Temporary storage for outgoing messages awaiting transfer
- **SMTP**: Application layer protocol for reliable email transfer between servers
- **IMAP**: Protocol for accessing and managing email on remote servers

7.2 Protocol Comparison Summary

- **SMTP**: Push protocol, server-to-server, port 25, 7-bit ASCII
- **IMAP**: Pull protocol, server-to-client, port 143, server storage
- **HTTP**: Pull protocol, web interface, ports 80/443, rich content

[Concept Map: Email Flow: Composition → UA → SMTP → Server Queue → SMTP → Receiver Server → Mailbox → Access Protocol → UA → Reading]

8 Study Aids

8.1 Mnemonics

- USA: User agents, Servers, SMTP Architecture
- HBS: Headers, Blank line, Body Structure
- PPS: Push (SMTP) vs Pull (HTTP/IMAP) Services

8.2 Key Points to Remember

- SMTP is used between mail servers, not between user agent and server
- Email has two types of addressing: SMTP commands and message headers
- IMAP keeps messages on server, POP3 typically downloads to local machine
- HTTP-based email adds web interface layer to traditional email protocols