



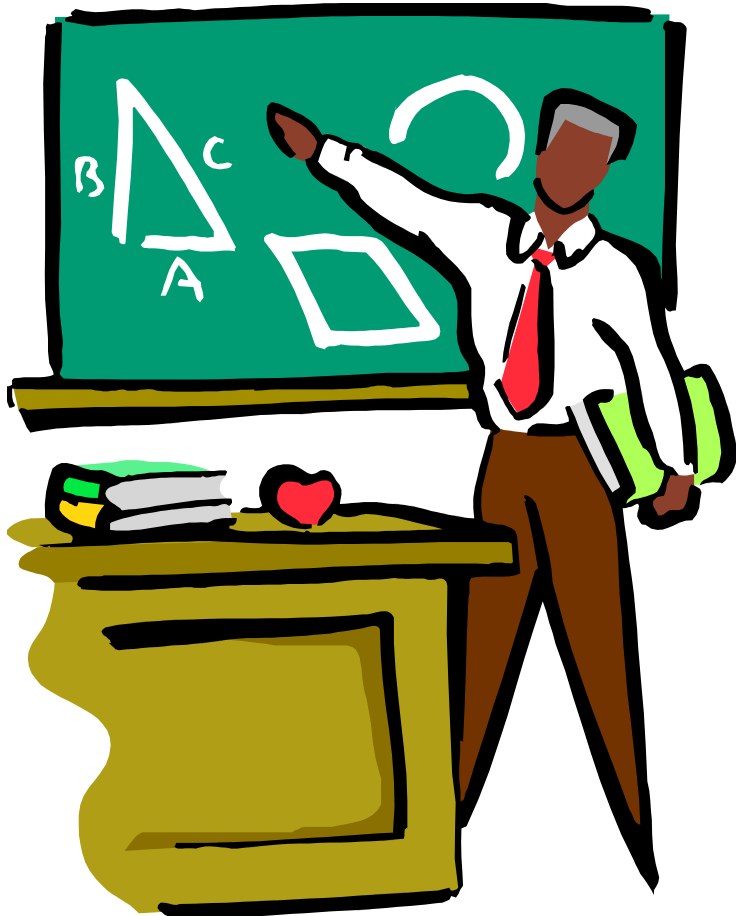
---

# Chapter IX

# Internet Mail Protocols



## Internet Mail Protocols



- 1 - Internet Mail Architecture
- 2 - Simple Mail Transfer Protocol (SMTP)
- 3 - Post Office Protocol (POP)
- 4. Internet Message Access Protocols (IMAP)



# Internet Mail Architecture

## Functional entities

- Mail User Agent (MUA)
  - Interact with end-users via user interfaces
    - Send / read emails
      - » May or may not offer offline interaction possibilities
    - Format message
      - » May encrypt
    - Examples
      - » Web mail (no off-line interaction)
      - » Microsoft proprietary systems (off-line interactions)



# Internet Mail Architecture

## Functional entities

- Split Mail User Agent (Split- MUA)
  - MUA with limitations
    - » May be intermittently connected to Internet
    - » May have limited processing power
      - » Unable to authenticate
      - » Unaware of time zones



# Internet Mail Architecture

## Functional entities

- Mail Transfer Agent (MTA)
  - Provide mail transport services
    - Enable the transportation of mails between source MUA and target MUA.



# Internet Mail Architecture

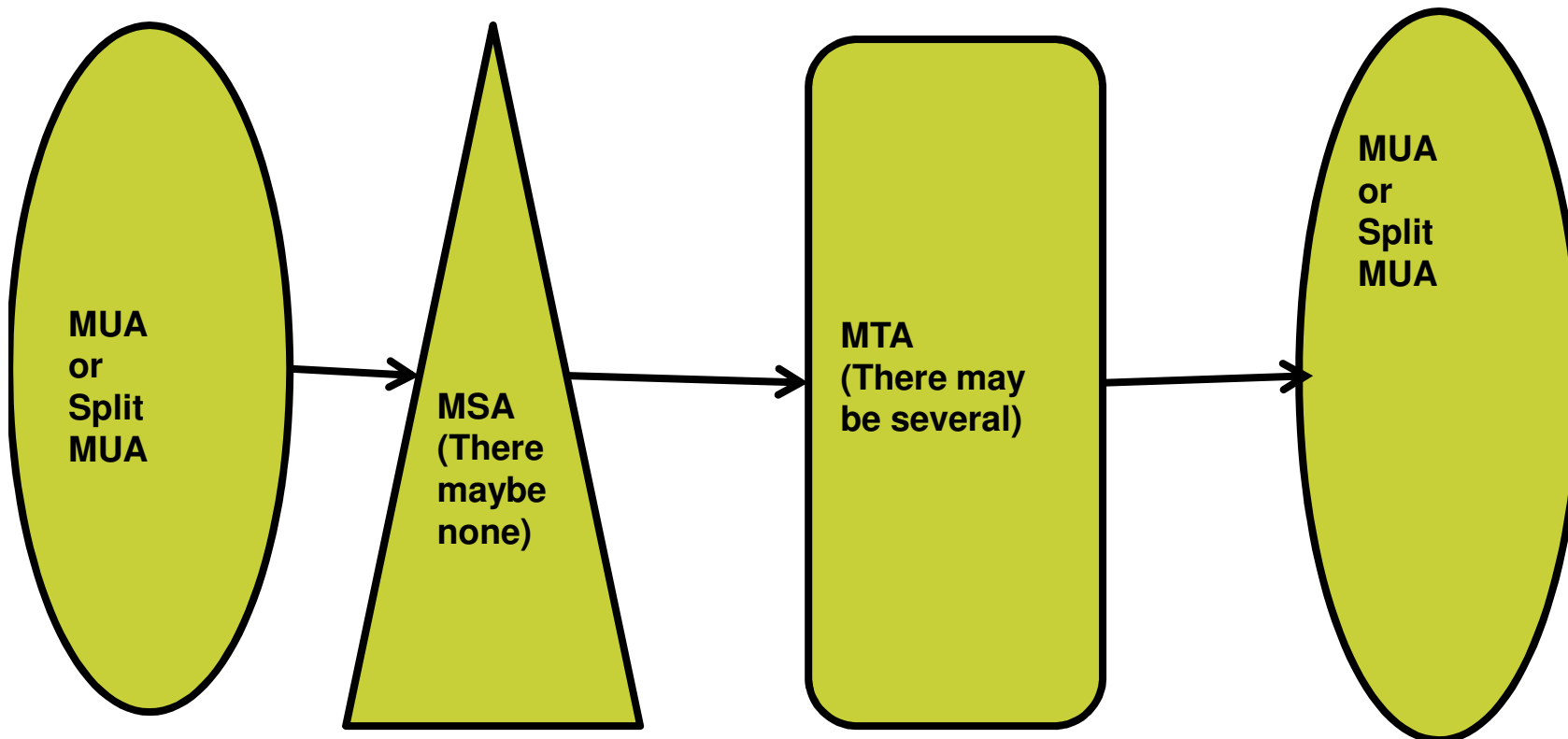
## Functional entities

- Mail Submission Agent (MSA) (Relatively new – 2006)
  - Act as intermediary between MUAs and MTAs
    - Enable functional separation between message submission and message transfer
      - » Objectives
        - » Modularity
        - » Configurable policies
          - » Submission authentication (e.g. off-line submission)
        - » Spam handling
          - » Avoid unauthorized mail relaying
          - » Avoid injection of unsolicited bulk mail



## Internet Mail Architecture

Functional entities (There may be several entities in the same node)





# Internet Mail Architecture

## Main protocols

- MTA/MTA
  - Simple Mail Transfer Protocol (SMTP)
    - May also be used as mail submission protocol between split MUA agent and MTA when in separate nodes
- MUA /MTA
  - Mail retrieval between split MUA and MTA
    - Post Office Protocol (POP)
    - Internet Message Access Protocol (IMAP)





# SMTP

## Primary objective

- Reliable and efficient mail transport and delivery between MTAs
  - Usage as mail submission protocol will gradually disappear

## Transport protocols used

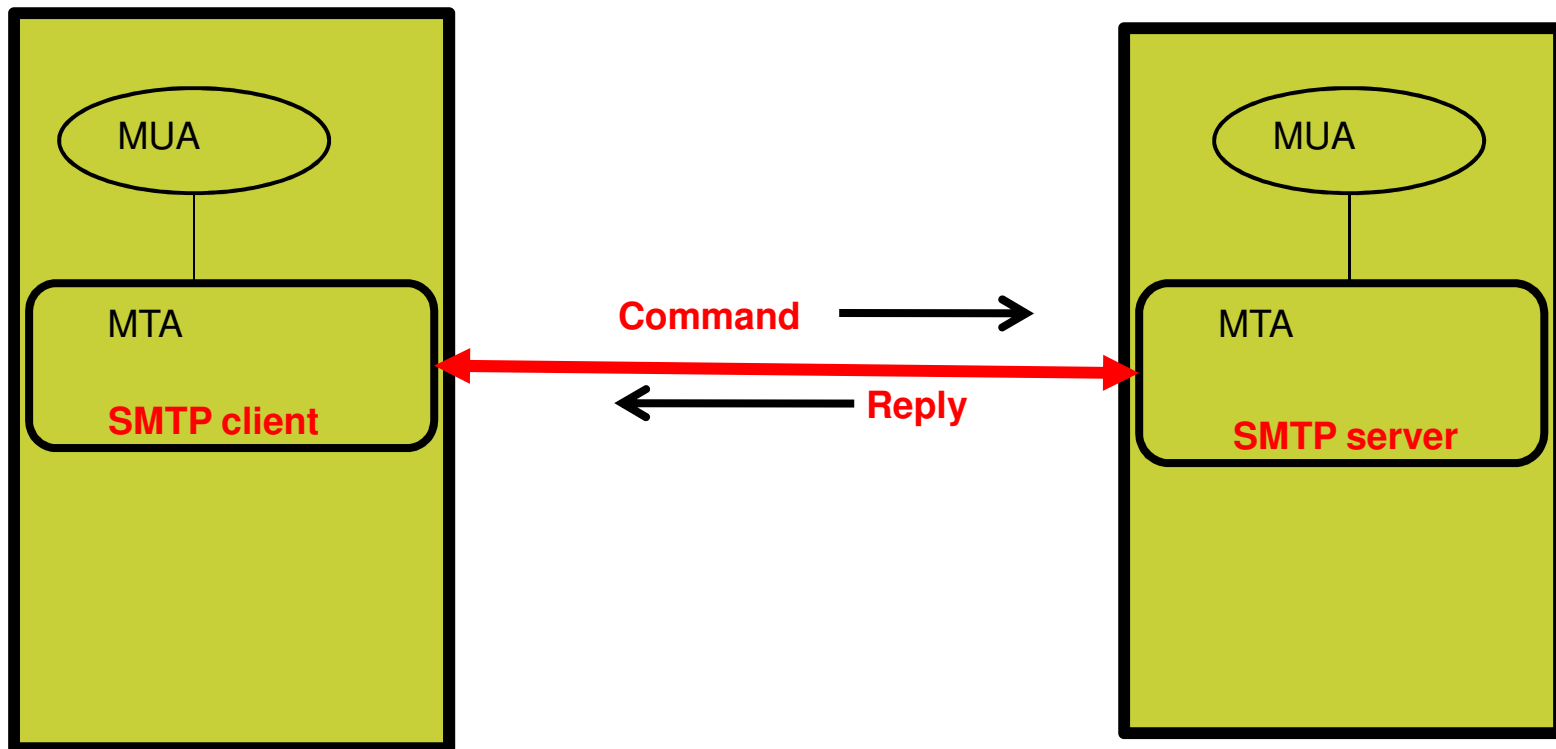
- Runs usually on top of TCP
- May also runs on top of other appropriate transport protocols



# SMTP

## Basic structure

- Mail transport and delivery





# SMTP

## Basic structure

- Mail transport and delivery between MTAs
  - SMTP client
    - Transfer mail messages to SMTP servers
      - » Get SMTP server IP address using DNS
  - SMTP server
    - Ultimate destination or
    - Relay
      - » Act as SMTP client towards another SMTP server
      - » There may be several hops between sources and destinations
    - Gateway
      - » Translation service towards non SMTP servers



# SMTP

## Basic structure

- General characteristics
  - Command / reply
  - Plain text (no encoding)
  - State-full
    - Three states
      - » Session initiation
      - » Transactions
      - » Session termination



# SMTP

## Basic structure

- Three states

### 1. Session initiation

- Three way hand shake
  - » TCP connection (bidirectional) initiated by client, then server talks firsts
  - » Exchange of greetings



# SMTP

## Basic structure

- Three states
  - 2. Mail transactions
    - Series of commands from client to transfer message(s) to server
      - » Specification of sender
      - » Specification of receiver
      - » Transmission of message
    - Series of replies from server (1 reply per command)
      - » Command accepted
      - » Additional commands are expected
      - » Error



# SMTP

## Basic structure

- Three states
- 2. Mail transactions
  - Examples of commands
    - » MAIL FROM
    - » RCPT
    - » DATA
    - » QUIT
  - Examples of reply codes
    - » 250: Requested action OK, completed
    - » 503: bad sequence of commands



# SMTP

## Basic structure

- Three states
  1. Session establishment
  2. Session termination
    - Bidirectional transmission channel tear down





# SMTP

Example: Message sent from [elinor@abcd.com](mailto:elinor@abcd.com) to  
[Carolyn@xyz.cm](mailto:Carolyn@xyz.cm)

– Message

```
From: elinor@abcd.com
To: carolyn@xyz.com
MIME-Version: 1.0
Message-Id: <0704760941.AA00747@abcd.com>
Content-Type: multipart/alternative; boundary=qwertyuiopasdfghjklzxcvbnm
Subject: Earth orbits sun integral number of times
```

This is the preamble. The user agent ignores it. Have a nice day.

```
--qwertyuiopasdfghjklzxcvbnm
Content-Type: text/enriched
```

```
Happy birthday to you
Happy birthday to you
Happy birthday dear <bold> Carolyn </bold>
Happy birthday to you
```

```
--qwertyuiopasdfghjklzxcvbnm
Content-Type: message/external-body;
  access-type="anon-ftp";
  site="bicycle.abcd.com";
  directory="pub";
  name="birthday.snd"
```

```
content-type: audio/basic
content-transfer-encoding: base64
--qwertyuiopasdfghjklzxcvbnm--
```



## SMTP

Example: Message sent from [elinor@abcd.com](mailto:elinor@abcd.com) to  
[Carolin@xyz.cm](mailto:Carolin@xyz.cm)

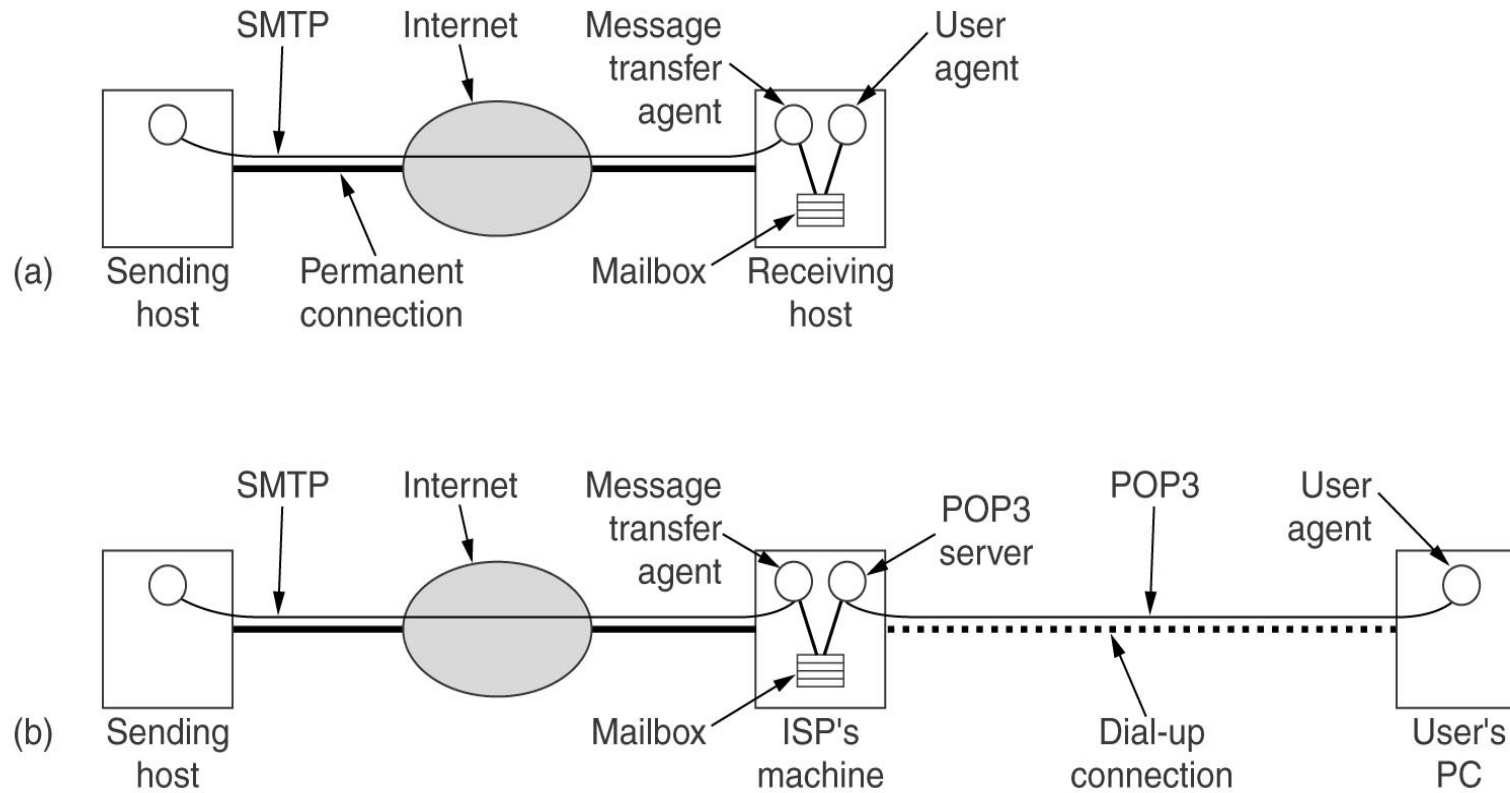
- SMTP messages between SMTP client (elinor domain) and SMTP server (Carolin) domain



# SMTP

```
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
```

# Post Office Protocol (POP) vs. SMTP





# POP

## Primary objective

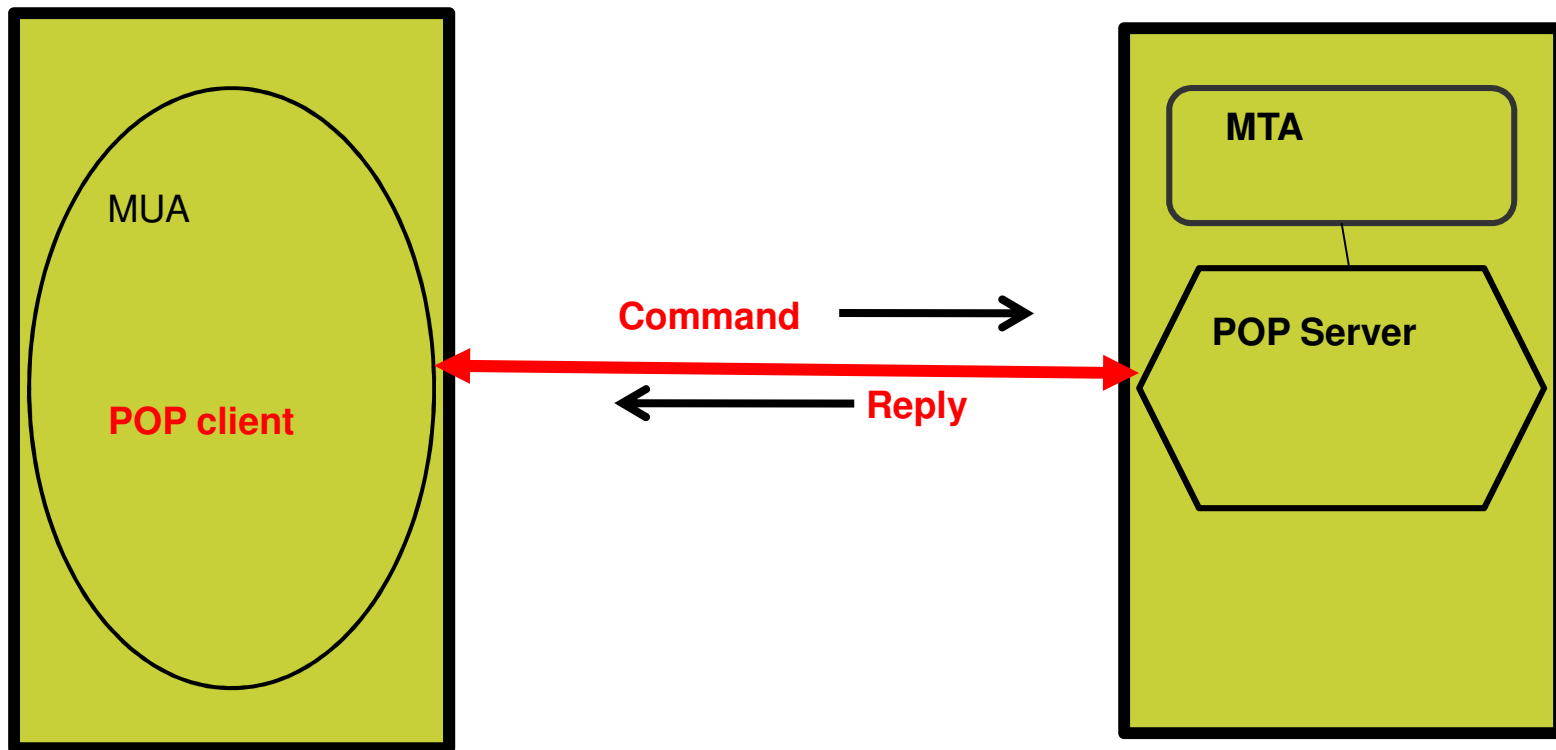
- Enable split-MUA (less endowed MUA) to retrieve mail
  - Dynamic access to a mail drop service for mail retrieval
    - SMTP or other appropriate protocols are used for submissions
  - Runs on top of TCP
  - Basic services
    - Mail downloaded and deleted
    - IMAP offers more comprehensive services



# POP

## Basic structure

- Mail retrieval by split MUA





# POP

## Basic structure

- General characteristics
  - Command / reply
    - Unlike SMTP, there may be several responses to a same command
  - Plain text (no encoding)
  - State-full
- Three states
  - Authorization
  - Transactions
  - Update



# POP

## Basic structure

- Three states

### 1. Authorization state

- TCP connection (bidirectional) initiated by client, then server talks firsts
- Action:
  - » Client identification and authentication
  - » Commands used by client
    - » USER
    - » PASS





# POP

## Basic structure

- Three states

### 2. Transaction state

- Examples of commands:
  - STAT:
    - » Statistics (messages, sizes)
  - LIST:
    - » List messages
  - RETR
    - » Retrieve messages
  - DELE
    - » Delete messages
  - QUIT



# POP

## Basic structure

- Three states

### 3.Update state

- Trigger:
  - » QUIT command given in Transaction mode
- Actions
  - » Deleted messages are actually removed from mailbox
  - » TCP connection is closed



# POP

## An example

```
S: +OK POP3 server ready
C: USER carolyn
S: +OK
C: PASS vegetables
S: +OK login successful
C: LIST
S: 1 2505
S: 2 14302
S: 3 8122
S: .
C: RETR 1
S: (sends message 1)
C: DELE 1
C: RETR 2
S: (sends message 2)
C: DELE 2
C: RETR 3
S: (sends message 3)
C: DELE 3
C: QUIT
S: +OK POP3 server disconnecting
```



# IMAP

## Primary objective

- Enable split-MUA (less endowed MUA) to retrieve mail
  - Dynamic access to a mail drop service for mail retrieval
    - SMTP or other appropriate protocols are used for submissions
  - Runs on top of TCP

•



# IMAP

## Main differences with POP

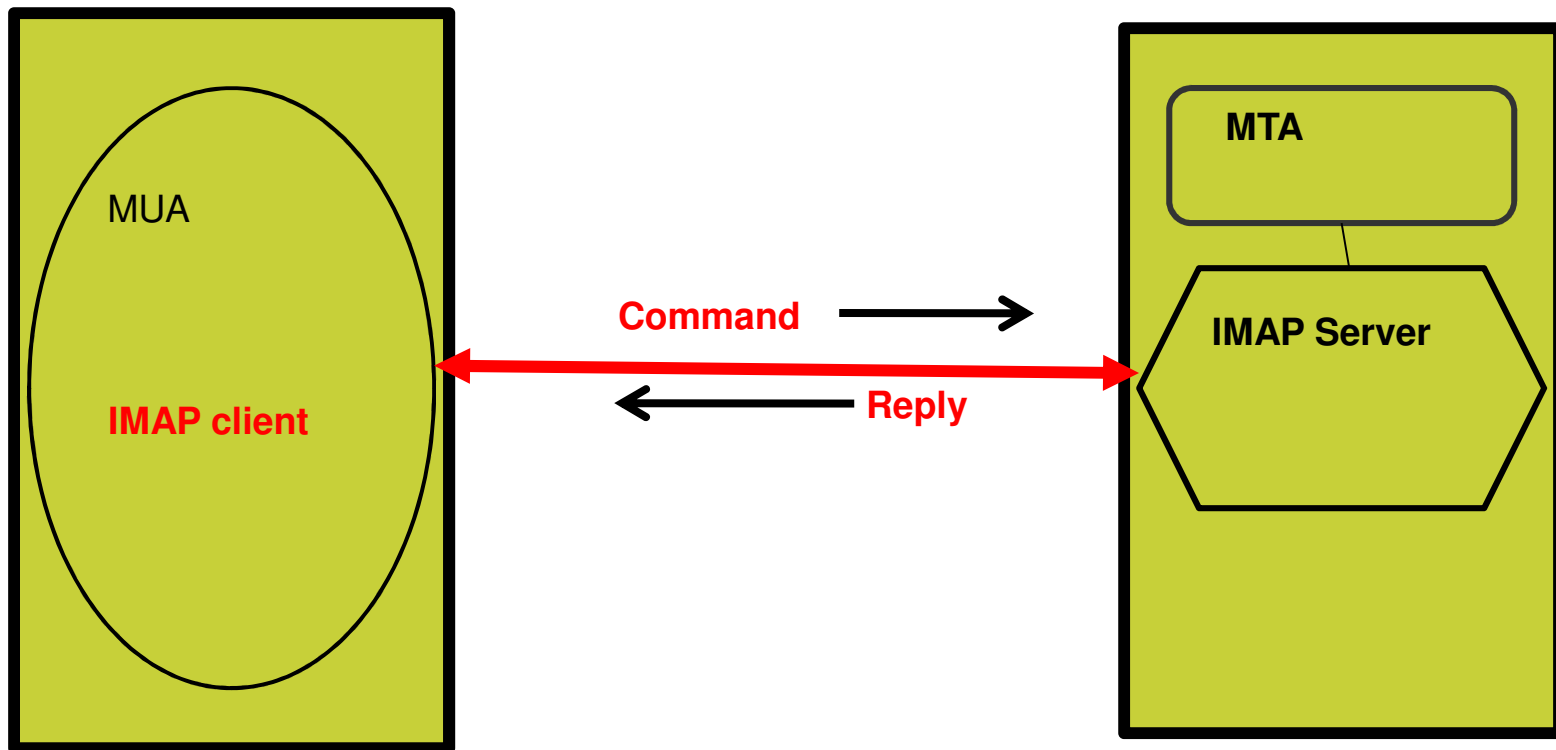
- New services
  - multiple mailboxes manipulation including concurrent access to same mailbox
  - Off-line operations
- Enhancements to POP services
  - Examples:
    - » Optimizations (e.g. header downloading)
    - » Storage including persistent message status storage
      - » Enable access from any computer



# IMAP

## Basic structure

- Mail retrieval by split MUA





# IMAP

## Basic structure

- General characteristics
  - Command / reply
    - Unlike SMTP, there may be several responses to a same command
  - Plain text (no encoding)
  - State-full
- Four states
  - Not authenticated
  - Authenticated
  - Selected
  - Logout



# IMAP

## Four states

### 1. Not authenticated

- First thing after TCP connection and server greetings
- Needed because some clients are pre-authenticated
  - A non pre-authenticated client must supply credentials and this will trigger authenticated state
  - State will automatically move to authenticated if client is pre-authenticated
  - Different greetings are used for pre-authenticated and non pre-authenticated clients
  - Example of commands
    - LOGIN





# IMAP

## Four states

### 2. Authenticated

- After successful client authentication or automatically for pre-authenticated clients
- Examples of valid commands
  - Mailbox manipulation
    - CREATE
    - DELETE
    - APPEND
    - SELECT (Selection of a specific mailbox)



# IMAP

## Four states

### 3. Selected

- A mailbox has been successfully selected
- A very wide range of commands related to the mailbox can now be used including
  - EXPUNGE
    - Delete all messages with the deleted flag
  - SEARCH
  - FETCH



# IMAP

Four states

4. LOGOUT state

Connection termination



---

## References

- 1, RFCs (Make sure to consult the most recent versions)
  - SMTP RFC
  - POP RFC
  - IMAP RFC