Internet infrastructure

Prof. dr. ir. André Mariën
email
SMTP

- Simple Mail Transfer Protocol
- RFC 2821
- Successor of RFC 821 (and others)
- Objective: transfer mail reliably and efficiently
Basic structure

(c) A. Mariën -
Determine proper email server

- Determine postbox domain
  - postbox@domain
- Look up SMTP server in DNS MX records
  - nslookup: Querytype=mx
  - Search ‘domain’
- Set up TCP connection to server, port 25
- Transfer mail using SMTP protocol

- DNS contains Mail eXchange (MX) resource records
- MX records contain server and priority
- example:
  smallcom.com MX mail.smallcom.com 10
  smallcom.com MX mail.hoster.com 20
- if possible, deliver to mail.smallcom.com, otherwise to mail.hoster.com

(c) A. Mariën - Ubizen
Set up connection to server

- **SMTP**: typically on top of TCP/IP
- **IANA assigned number**
  - SMTP port: 25
    - To test:
      - telnet `host 25`

- **Note**
  - Typical configuration: via a dedicated SMTP gateway
  - Configured in email client
  - No DNS lookup, no direct connection

```plaintext
HELO marien.com
MAIL FROM: andre@marien.com
RCPT TO: an@an.com
RCPT TO: bea@bea.com
...
DATA
ILOVEYOU
QUIT
```

(c) A. Mariën -
Mail object structure

• Mail envelop
  – MAIL FROM
  – RCPT TO
  – DATA

numeric completion code
• 1yz:
  – positive preliminary
• 2yz:
  – positive completion
• 3yz:
  – positive intermediate
• 4yz:
  – transient negative completion
• 5yz:
  – permanent negative completion

(c) A. Mariën -
Additional commands

Verify
- Verify: checking of mailbox existence
- OK: 250 response
- If ambiguous: (553-[name] <address>)* 553 [name] <address>
- 500 / 502 : NOK
- Often this functionality is disactivated for security reasons (account enumeration)

Expand
- Expand: support for mailing lists
- OK: (250-[name] <address>)* 250 [name] <address>
- 500 / 502 : NOK
- Often disactivated for security reasons
- Alternatives
  - User agent definitions
  - Server side group expansion

(c) A. Mariën -
“Postmaster” address

• Mandatory: ‘postmaster’ must exist
• Used for:
  – error reporting
  – abuse reporting (SPAM)
• Exception: mail can be send without domain, implies domain of SMTP server

(c) A. Mariën -
POP3
POP3

• Post Office Protocol version 3
• Protocol for mailldrop (pick-up) service
• Simple protocol
  – advanced requirements: use IMAP
  – delivery: use SMTP
• RFC 1939, STD 53
POP3 set-up

(c) A. Mariën
POP3 connection

- TCP/IP connection
- Server assigned port number: 110
- Reply: two options
  - +OK
  - -ERR
- Multi-line response:
  - <CRLF>.<CRLF> terminated

```
telnet pop.somewhere.else 110
  - USER andre
  - PASS andre
  - LIST
  - RETR 1
  - DELE 1
  - QUIT
```
Commands

Minimal:
• Always:
  – QUIT
• Authorization state:
  – USER name / PASS string
• Transaction state
  – STAT / LIST [msg] / RETR msg / DELE msg / NOOP / RSET
  where msg is the message ordinal number

Optional:
• Authorization
  – APOP name digest
• Transaction
  – TOP msg n
  – UIDL [msg]
POP commands

• STAT
  – Reply:
    +OK number of messages, total number of bytes

• POP cmd: LIST
  – LIST [message ordinal number]
  – Response:
    +OK
    <Msg number> <nr of bytes>
    ...

• POP cmd: RETR
  – RETR <message ordinal number>
  – Response:
    +OK
    Message
    .
POP commands

• DELE & RSET
  – DELE <message ordinal number>
  – Response
    +OK
  – Note
    • Message is only marked for deletion
    • Subsequent LIST or RETR commands fail
    • Actual removal: after QUIT command ONLY
    • RSET: clears all deletion markings

• TOP
  – TOP <message ordinal number> <lines>
  – Response
    +OK
    Header
    First <lines> lines of the message
  .
POP commands

• UIDL
  – UIDL [<message ordinal number>]
  – Response
    +OK
    1 jdkodoiezjfrj
    2 uikjendsiojdj
    3 jdsoioizehjkl

• APOP
  – APOP <digest>
  – Response: +OK or –ERR
  – The greeting banner must contain the challenge
  – The client computes md5(challenge+secret)
  – Challenge has msg-id format
    +OK POP3 server ready <12.34@x.com>
Mail client

• Multi-protocol client
  – SMTP: sending mail
  – POP3: receiving mail
• But possibly also
  – DNS
    • Mail relay name to IP
    • SMTP with target servers
  – LDAP: user and group look-up
Free software

• UNIX systems:
  – Qpopper: http://qpopper.sourceforge.net/

• All
  – Xmail: http://xmailserver.org/
    • Ssmtp server
    • Pop3 server
IMAP4
IMAP4

- Internet Message Access Protocol version 4
- Extensive, comprehensive protocol for post office access
- Richer but more complicated than POP
- RFC 2060
Connection oriented

• TCP/IP connection
• server port number: 143
• reply: OK or NO or BAD or BYE
• Reply:
  – tagged (tag of request)
  – untagged
• request/reply, but also unsolicited
• Asynchronous operation possible
Tags in the reply

- **Request ID:**
  - Response to a request, the request ID is repeated in the reply

- **“*”:**
  - untagged server message
  - Not a reply; for instance quota notifications

- **“+”:**
  - continuation required (for instance: authentication)

---

Tag usage:

> Request007 “command”
< * ...  
< * ...  
< Request007 OK “command completed”

(c) A. Mariën -
Message attributes

• Numbers
  – Unique ID
  – Sequential ID
• Flags
• IMAP server reception timestamp
• RFC822 size
• RFC822 Envelope structure
• Body structure (MIME)
Message flags: system

- \texttt{seen}: Message has been “read”
- \texttt{answered}: Message has been answered
- \texttt{flagged}: Message is flagged
- \texttt{deleted}: Message is marked ‘to-be-deleted’
- \texttt{draft}: message partially composed
- \texttt{recent}: new message notification flag
IMAP operation

• Commands depend on state
• State transition diagram
• Basic flow:
  – Initial
  – Not authenticated
  – Authenticated
  – Selected
  – End
State diagram (from RFC 2060)

1. Initial connection and server greeting
2. Non-authenticated
3. Authenticated
4. Selected
5. Logout and close connection

(c) A. Mariën -
Commands:

- any state
  - CAPABILITY: which authentication mechanisms?
  - NOOP:
    - polling command to trigger unsolicited (!) status info
    - Session keep-alive (against time-out)
  - LOGOUT: end session

- non-authenticated state
  - AUTHENTICATE “authentication mechanism”
  - LOGIN “user name” “password”
Example from RFC222: AUTHENTICATE

- S: * OK IMAP4 Server
- C: A001 AUTHENTICATE SKEY
- S: +
- C: bW9yZ2Fu
- S: + OTUgUWE1ODMwOA==
- C: Rk9VUiBNQU5OIFNPT04[...]TUFTSA==
- S: A001 OK S/Key authentication successful
Commands: authenticated:

• managing mailboxes
  – CREATE “mailbox name”
  – DELETE “mailbox name”
  – RENAME “existing mailbox name” “new mailbox name”

• browsing
  – LIST “reference name” “wildcard mailbox name”
    • Browsing + metainfo (hierarchy separator)
  – STATUS “mailbox name” ( “status data item names” )
    • Number of messages
Commands: authenticated: selecting current mailbox

• SELECT “mailbox name”:
  – open read/write
  – Responses: untagged
    • * FLAGS (...)
    • * <n> EXISTS
    • * <n> RECENT
    • * OK [UNSEEN 134]

• EXAMINE “mailbox name”:
  – open read-only, rest: see SELECT
Commands: selected state

• CLOSE
  – Remove (silently) all messages marked for deletion

• EXPUNGE
  – Remove (silently) all messages marked for deletion; unmarked responses:
    • * <msg number> EXPUNGE
    • ...

(c) A. Mariën -
FETCH command

• Fetch <message set> <data item names>
  – Message set:
    • <nr>
    • “*” (highest number)
    • <start>:<end>
    • <nr>,<set>
  – Examples:
    • 7
    • 5:100
    • 7,5:100,120,130:*
FETCH: Data item names

• BODY [<section>][<selection>]
  – <section>: RFC822 decomposition
  – <selection>: “<“<frombyte>”.”<nrbytes>”>”
  • Example, first 100 bytes: <0.100>
Fetch command: “BODY” & “BODY.PEEK”

- BODY[HEADER]
- BODY[TEXT]
- BODY[HEADER.FIELDS (field-name field-name)]
- BODY[HEADER.FIELDS (DATE FROM)]

(c) A. Mariën -
FETCH command: RFC822 decomposition

• Top-level: HEADER & TEXT
• MIME: multipart/*
  – 1.HEADER & 1.TEXT & 1.MIME
  – 2.HEADER & 2.TEXT & 2.MIME
  – ...

(c) A. Mariën -
FETCH: other elements

- BODYSTRUCTURE
- FLAGS
- ENVELOPE
- UID
- INTERNALDATE

(c) A. Mariën -
FETCH command example

• X007 FETCH 2:8 (FLAGS BODY[HEADER.FIELDS (DATE FROM)] BODY[TEXT]<0.100>)
  • * 2 FETCH ...
  • ...
  • * 8 FETCH ...
• X007 OK Fetch completed
SEARCH command

• Conditions: examples:
  – <Select by number>; ALL;
  – To <string>; BCC <string>; Cc <string>
  – BODY <string>; SUBJECT <string>
  – Before <date>; sentbefore <date>; sentsince <date>; senton <date>
  – Answered; Deleted; ...
  – Unanswered; Undeleted; ...
  – Smaller <size>
Search reply

• X007 search From student unanswered
• * search 5 8 12 45 77 123 453
• X007 OK search completed
IMAP URLs

• URL type: IMAP
• imap://imaphost?imap command