SMTP: Protocol and Format

Phil Regnauld <regnauld@nsrc.org>

PacNOG 10

Noumea, New Caledonia

Summary

- Presentation of the SMTP model
- Evolution of SMTP
- Architecture of the software

Presentation of the SMTP model

- Structure of the messages
- Addressing
- Transport Protocols
- Message routing

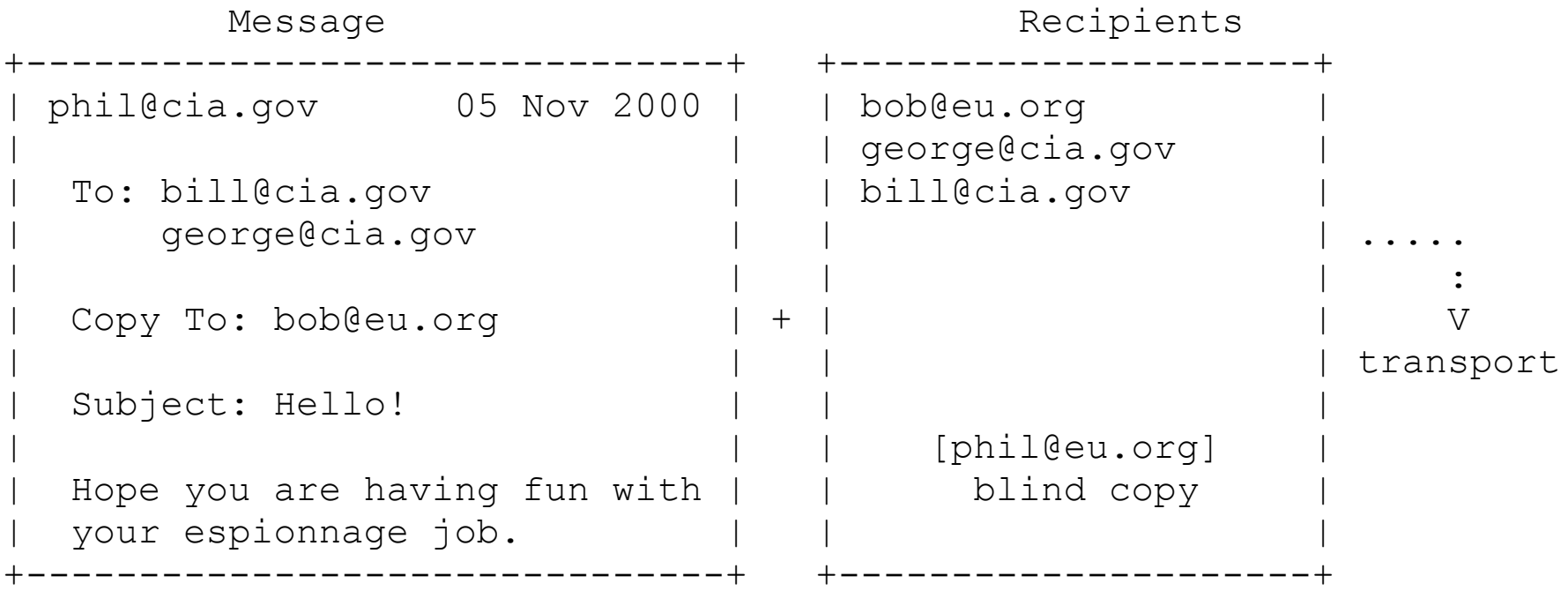
What is SMTP ?

- Simple Mail Transfer Protocol (RFC 2821, originally 821)
- Protocol allows to transport messages in the RFC 2822 (formerly 2821) format
- Historically, could be used over other transports than TCP/IP, but this is not used anymore.

Structure of the messages

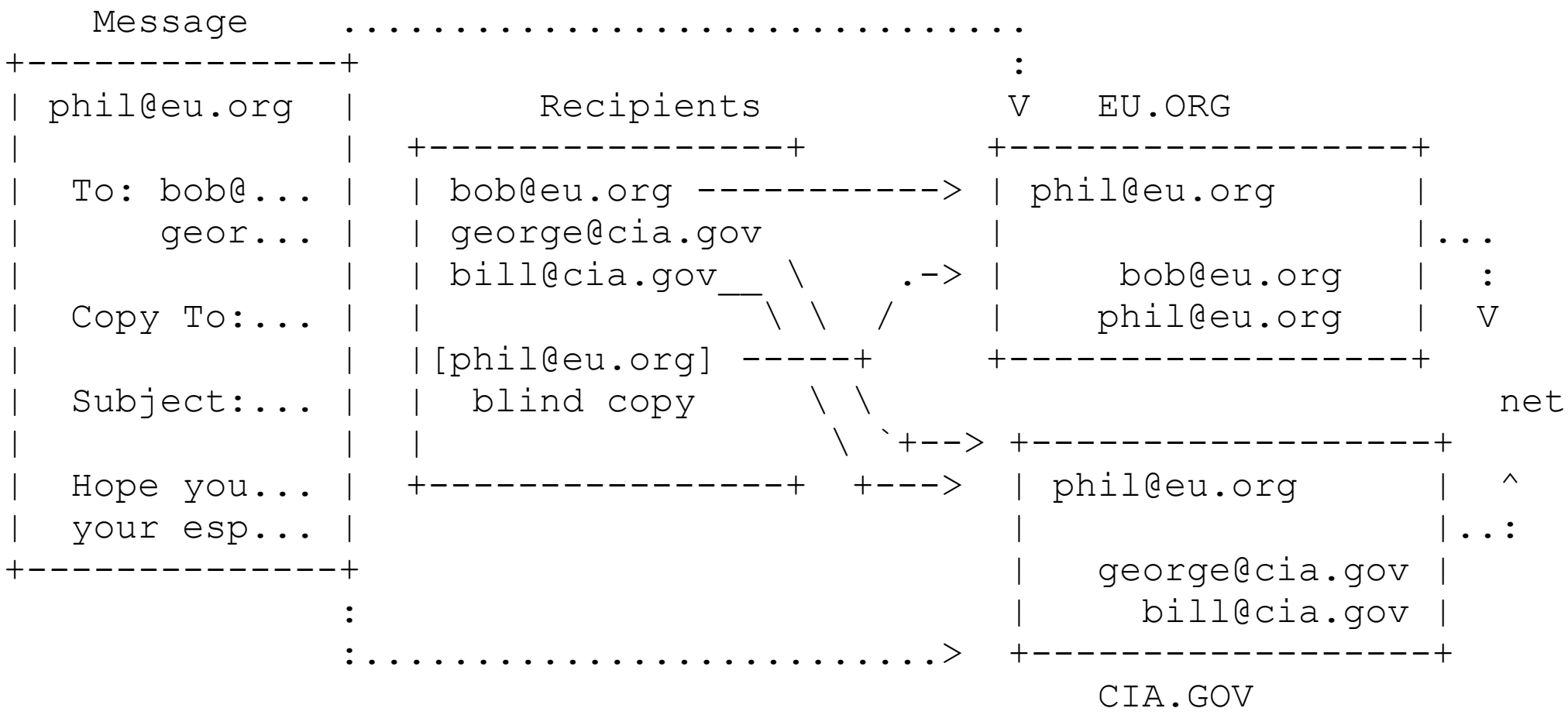
- Envelope
  - Routing information for the "postman"
- Message Header
  - Sender
  - Recipients (simple, lists, copies, blind copies)
  - Other fields of control (date, subject)
- Message Body
  - Free text
  - Structured document (i.e.: MIME)

Content of a message



A message = envelope + content

- The transport agent ensures message duplication and "packing" into envelopes: one envelope per destination domain (NOT per person)



SMTP addressing

- Identifies in a unique way each mailbox
- Independant from any Internet provider
- Simple and readable
- Syntax: *local-part@remote-part*
- Examples:
  - *regnauld@eu.org, webmaster@nsrc.org*

RFC 822 Addressing

- Defines the grammar of addresses
  - Addresses are analyzed lexicographically
  - A routing decision is taken, and the message is sent
- Complicated addresses are the biggest problems for mail routers.

RFCs allow very complex addresses:

@eu.org:"Phil Regnauld"%nsrc.org@(Just a comment)x0.dk

- The software must analyze the addresses then decide on routing
- Without this complexity, email routing would be easy

Not all characters are allowed

- RFC 2822 allows a lot
- But the RFCs of the DNS forbid certain characters in domain names
- For example: _ (underscore) is authorized by RFC 2822, but forbidden in the DNS
- Certain systems drop messages with hostnames containing '_'
- Advice: use letters, numbers, and – (dash) only.

RFC 822 header (1/4)

One or more fields in the format :

Name of field: content

- Long lines continue on the next which must begin with whitespace
- Example:

To: "Phil Regnauld" <regnauld@eu.org>,

webmaster@x0.dk, bob@cia.gov

- An empty line marks the end of the header and the beginning of the message body

Main fields

- From: sender
- To: recipients
- Reply-To: reply address
- Cc: carbon copy
- Bcc: blind carbon copy
- Subject: the topic of this message
- Date: date message was sent
- Message-ID: unique message identifier (optional)
- Received: transport information (one per relay)

ASCII characters only

- Representation of international characters using MIME
- The representation is defined in RFC 1522
- This is a historical leftover (SMTP can without problems carry pure 8 bit)
- Many accept 8 bit, but not everyone. SMTP agents are allowed to throw away non-conforming messages

Letter case is InDifFeReNt.

·FROM = From = FroM

- For addresses: the remote part is indifferent (to the right of @), but can matter for the local part (to the left of @)
  - *pr@eu.org = pr@EU.org*
  - *pr@eu.ORG != PR@eu.org*
- Many systems do not make a difference between upper and lower case on the local part

ASCII characters. Other representations: RFC 1521 (MIME) strongly recommended.

- But sending pure, non ASCII characters works often (once again, the transport agent is allowed to reject 8bit if 8bit MIME was not negotiated)

- A message is structured
  - groups of lines
  - maxmimum length of each line: 1000 chars (RFC-821)
  -

- Sending raw binary is therefore not possible without some preliminary processing.

Example of conforming message

```
>From chlor@inet.uni2.dk  Thu Jun 17 02:04:28 1999
Return-Path: <chlor@inet.uni2.dk>
Received: from ns.eu.org (valerian.glou.eu.org [193.56.58.251])
        by mail.ftf.dk (8.9.3/8.9.3/gw-ftf-1.2) with ESMTP id DAA02180
        for <regnauld@i23.eu.org>; Thu, 17 Jun 1999 03:28:20 +0200
(CEST)
X-Authentication-Warning: mail.ftf.dk: Host valerian.glou.eu.org
[193.56.58.251]
+claimed to be ns.eu.org
Received: from appleton.uni2.dk (appleton.uni2.dk [129.142.7.23])
        by ns.eu.org (8.8.8/8.8.8/19980611) with ESMTP id DAA29606
        for <regnauld@eu.org>; Thu, 17 Jun 1999 03:26:49 +0200 (CEST)
Received: from hertz.uni2.dk (hertz.uni2.dk [129.142.7.6])
        by appleton.uni2.dk (8.9.1a/8.9.1) with ESMTP id DAA05767
        for <regnauld@eu.org>; Thu, 17 Jun 1999 03:26:44 +0200 (MET
DST)
Received: from neccesary (vlb-210-75.ppp.uni2.dk [129.142.210.75])
        by hertz.uni2.dk (8.9.2/8.9.1) with SMTP id DAA21921
        for <regnauld@eu.org>; Thu, 17 Jun 1999 03:26:42 +0200
(METDST)
Message-ID: <37685D2A.2F64@inet.uni2.dk>
Date: Thu, 17 Jun 1999 03:27:54 +0100
From: Hans Schou <chlor@inet.uni2.dk>
Reply-To: chlor@schou.dk
Organization: The Linux Community
MIME-Version: 1.0
To: regnauld@eu.org
Subject: Painting the corner
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
Content-Transfer-Encoding: 8bit

Hi Phil

Nice article - can it be found somewhere in electronic form ?
```

The Date: field

Indicates the date the message was sent

Examples:

- *Tue, 19 Sep 1995 11:10:48 +0200 (MET DST)*

- Only US timezones are normalized -- elsewhere: offset from UTC

- Because of the often non-synchronized clocks, it happens that one receives a message before the date on which it it was sent!

The From:, To:, Cc:, Bcc:, Reply-To: fields

- Contain RFC 2822 addresses
  - Can be prefixed with "Resent-". Indicates a bounce.
  - The Bcc: field usually doesn't appear to recipients
  - Reply-To: is not always presented well enough to the user's attention (users rarely verify the addresses they are replying to)

Each SMTP relay add a Received: line

- It mentions origin, destination, date, and sometimes the address of the envelope and of the destination
- Very useful for debugging and troubleshooting
- Is used to avoid loops: if the message has more than a certain number of Received: (17-25), bounce it
- Removing this header does not improve security

Free form (ASCII)

- Subject: identifies the purpose of the message -- optional but recommended
- Comment: allows for adding any comments
- And fields starting with X-... are left up to the user and are not normalized

Message identifier

- Format: Message-Id : <addr-spec>

- In general:
  - *something@local-machine*

- Example:

Message-Id: <200011042005.8827@vinyl.catpipe.net>

Should be unique, but not garanteed !

The SMTP protocol: example

```
220 server.bluepipe.net ESMTP Postfix
HELO macbook.catpipe.net
250 server.bluepipe.net
MAIL From: <regnauld@x0.dk>
250 2.1.0 Ok
RCPT To: <regnauld@nsrc.org>
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: hello
.
250 2.0.0 Ok: queued as 41A8B4F5C94
QUIT
221 2.0.0 Bye
```

- Structured dialogue between client and server
  - Each command provokes a response
  - Programs are only interested in the numerical part of the answer, the rest being provided to human readers for troubleshooting and legibility
  - SMTP line: terminated with CR+LF
  - Specific options are negotiated in the initial dialog phase

SMTP: response codes

- 1xx:positive preliminary answer (action to be continued in subsequent command)
- 2xx:positive response indicating that processing has been carried out as requested
- 3xx:positive partial response: the client must give more data for processing to continue
- 4xx:negative answer, processing is refused, but the command can be tried again later
- 5xx:negative answer, processing cannot be carried out

- No option negotiation
  - Only allows for transporting ASCII (or MIME messages with ASCII encoding)
  - Is often used in 8bit transparent mode since many programs allow it
  - Interactive delivery: delivery delays if the list of recipients is long
  - Need for extensions and MIME have given birth to extended SMTP: ESMTP

Local rules defined by the site administrator

- Use of MX (Mail eXchanger) in the DNS if there are no specific rules
  - the MX indicates a list of mail relays for the destination
  - the administrator of the remote site decides of this information
  - allows for very flexible administration, and avoids unmanageable and costly routing table maintenance

- Example:

```
x0.dk.      MX 0   mail.x0.dk.
            MX 10  otherserver.somedomain.
```

  - It can be useful to have multiple MX (the smaller the number, the higher the priority)

- Fault tolerance or load balancing:

```
my.net. IN MX 10 mailrelay.my.net.
mailrelay.my.net.     A 10.10.0.1
                      A 10.10.0.2
```

Mail routing algorithm (1/2)

- Look in local configuration files
  - If nothing is found, ask the DNS
  - If no MX, use the IP address. If no IP address, reject the mail: no information allow its transport
  - If we are in the list of MXes, remove all those which have a factor higher or equal to ours
  - Deliver to the remaining MXes, starting with the one with the smallest number (highest priority)
  - In case of failure, keep the message and try again later at regular intervals

Mail routing algorithm (2/2)

- If we are the preferred MX and there are no local rules to handle it, the message is rejected (loop avoidance)
- Multiple messages can be sent during a single SMTP session

ESMTP: Extended SMTP

- Defined in RFC 1651 and following
  - Adds new functionality
    - Transport of 8bit MIME messages
    - Maximum message size limit
    - Function limitation (EXPN, VRFY, ...)
    - Other extensions (pipelining, private extensions)
  - The welcome message for ESMTP is EHLO (instead of HELO). In case of a negative answer, the client must revert to the old protocol.

EHLO: Extended HELO

The server indicates the extensions it supports

- Example

```
220 server.bluepipe.net ESMTP Postfix
EHLO macbook.catpipe.net
250-server.bluepipe.net
250-PIPELINING
250-SIZE 104857600
250-VRFY
250-ETRN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-BINARYMIME
250 CHUNKING
MAIL From: <regnauld@x0.dk>
...
```

A few options

- SIZE
  - specifies, before sending, the maximum size of a message (it's the client's job to find another path if the message is larger)
- 8BITMIME
  - the server accepts 8bit messages in MIME format (therefore with other character sets than ASCII) and carry out the necessary transformations if it needs to relay to another non-8BITMIME server
- EXPN
  - the EXPN function (alias expansion) is authorized
- DSN
  - Delivery Status Notification is supported

Security: the risks (1/2)

- Losing a mail because due to a Mail Transport Agent
  - buggy software
  - badly written software (does not manage limit cases well)
- Losing a mail because due to a user
  - message was deleted
  - locking problems during mailbox access
- Losing a message due to a hardware failure
  - disk crash
  - MX down for a long time

Security: the risks (2/2)

- Message eavesdropping
  - eavesdropping on the network by legal authorities or unauthorized persons
  - message disclosure due to a system misconfiguration
- False messages
  - sending mails impersonating other people
  - content modification

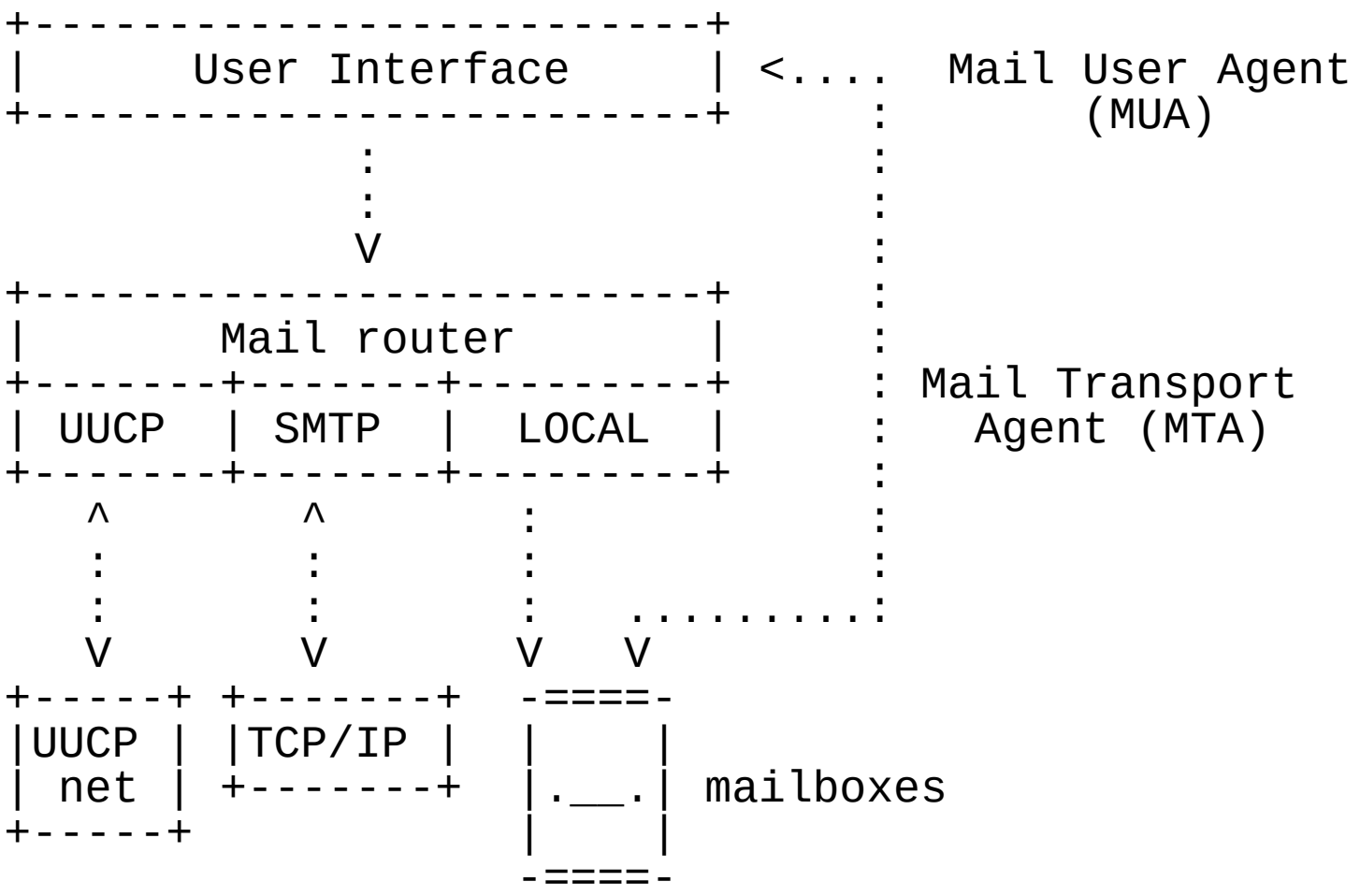
Passive security

- DSN extensions provide the sender with greater feedback on as to what happens to his message (in SMTP, it is the MAILER-DAEMON bounce which comes to mind)
- Each relay knows how to notify whether a message has been sent correctly or not to the next relay
- The philosophy of SMTP is to minimize the number of relays (no relays between the main gateway sites of the source and destination sites)

- As in many network applications, security can only be attained by using cryptography between applications
- No security office worthy of the name can accept to outsource his security to a third party provider. The solutions are therefore at home.
- PGP is very well adapted to this task: it allows for the electronic signature of a message or its encryption, using public-key cryptography
- Today, is also possible to use SMPT TLS (Trusted Layer Sockets), which use X.509 certificates and cryptography at the transport level.

Architecture of mail software

Three layers



Communication between layers

- The user interface sends its message to the transport agent
  - UNIX pipe-like communication channel
  - SMTP (the interface does not know how to route), POP/IMAP
- The interface accesses the user's mailbox
  - UNIX file
  - POP/IMAP
- Transport agent
  - receives messages on a communication channel
  - carries out routing and forwarding

Mail programs (MUA)

- Many available commercially or freely
- Their quality, more than the transport protocols, are responsible for the success or failure of an mail system
- Should be able to support access to local mailboxes but also remote (POP, IMAP)

- Routing and transport are often done in the same program
  - Carries out routing of mail using local rules or standard algorithms (MX)
  - Routing policy defined via configuration files
  - Built-in transport agents (SMTP) or spawning of external programs (IMAP, ...)

A few transport agents

-
  - UNIX
    - Sendmail
    - Postfix
    - Qmail
    - Exim
  - Windows
    - Exchange
    - Lotus Notes
    - ...

Bibliography: RFCs

- RFC 2821, 2822,
- RFC 1122, 1123: prerequisites for machines connected to the Internet
- RFC 1651: extensions to the SMTP protocol
- RFC 1653: SIZE extension
- RFC 1830: transporting large messages containing binaries
- MIME RFCs...