

The Application Layer: e-mail

COMPUTER NETWORKS A.A. 24/25



Leonardo Maccari, DAIS: Ca' Foscari University of Venice,
leonardo.maccari@unive.it

Venice, fall 2024

Sect. 1 Electronic Mail: e-mail

Let's invert the our roles: you talk.



- Now you tell me, if you had to plan a service like e-mail, how would you do it?
- Take two extremes as reference:
 - A phone call: no intermediary needed
 - The DNS service we analyzed yesterday
- Questions you have to consider:
 - Who are the entities that are involved in the service?



Let's invert the our roles: you talk.



- Now you tell me, if you had to plan a service like e-mail, how would you do it?
- Take two extremes as reference:
 - A phone call: no intermediary needed
 - The DNS service we analyzed yesterday
- Questions you have to consider:
 - Who are the entities that are involved in the service?
 - What is the format of the data exchanged?



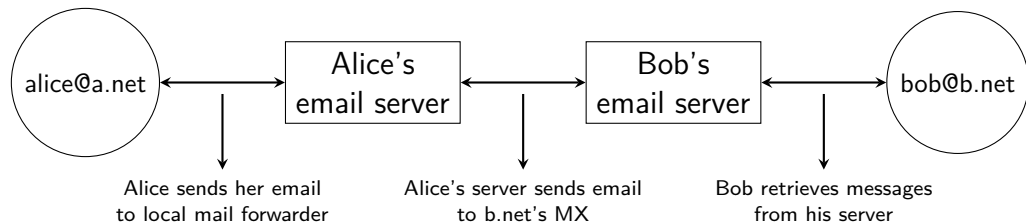
Let's invert the our roles: you talk.



- Now you tell me, if you had to plan a service like e-mail, how would you do it?
- Take two extremes as reference:
 - A phone call: no intermediary needed
 - The DNS service we analyzed yesterday
- Questions you have to consider:
 - Who are the entities that are involved in the service?
 - What is the format of the data exchanged?
 - What kind of security service I want to have on this application?



e-mail architecture



e-mail obviously don't go straight from computer to computer, there are several components involved.

- An e-mail client, that formats the message in the correct way
- Alice's e-mail server that Alice e-mail client sends the e-mail to
- Bob's e-mail server that receives the e-mail from Alice's e-mail server
- Bob's e-mail client that receives the e-mail from his server.
- When you use a web e-mail service, the web page running on a web server replaces the e-mail client
- The chain of servers can be longer than just two.

Several protocols are involved:

RFC5322: Internet Message Format

<https://datatracker.ietf.org/doc/html/rfc5322>

RFC2045: Multipurpose Internet Mail Extensions (e altre RFC)

<https://datatracker.ietf.org/doc/html/rfc2045>

RFC5321: Simple Mail Transfer Protocol

<https://datatracker.ietf.org/doc/html/rfc5321>

RFC1939,9051: Post Office Protocol, Internet Message Access Protocol,

<https://datatracker.ietf.org/doc/html/rfc1939>

<https://datatracker.ietf.org/doc/html/rfc9051>

- The first and the second protocols specify how the e-mail messages are formatted. These formats are extremely rich and complicate, way more than DNS messages
- The third one specifies the protocol needed to deliver the message from Alice's e-mail client to Bob's server
- The fourth one(s) specify how Bob can retrieve e-mails from his server.



Electronic Mail: e-mail

↳ 1.1 e-mail Format

- E-mails are textual messages, initially they were formatted in ASCII text, later on the permitted encoding was extended to other, more modern encodings.
- E-mails are made of multiple lines, each one ends with CRLF, two special symbols often used to specify end of line in many operating systems.
- e-mails have some initial lines that are *headers* and then more lines that make the body of the message. Valid headers are From, To, Date, CC, BCC, Subject....

e-mail Example



```
1 From: Bob Smith <bob@machine.example>
2 To: Alice Doe <alice@example.net>, Alice Smith <alice@machine.example>
3 Subject: Hello
4 Date: Mon, 8 Mar 2010 19:55:06 -0600
5 This is the "Hello world" of email messages.
6 This is the second line of the body
```



Other Relevant Header Fields



- **Message-Id:** This is a unique ID that is added to the email, and it is used by other e-mails that refer to this message
- **In-reply-to:** This is used when replying to a previous message. This field contains the message Id of the one that is replying to
- **Received:** Since the message will pass through a sequence of servers before being delivered, each of them can add a Received line. This is used for debugging purposes.
- **X-:** The client or the server can add any custom field that starts with X-.

Message Headers Format



```
1 Received: from smtp3.sgsi.ucl.ac.be (Unknown [10.1.5.3]) by mmp.sipr-dc.ucl.ac.be
2 (Sun Java(tm) System Messaging Server 7u3-15.01 64bit (built Feb 12 2010)) with ESMTPT id <0
   KYY00L85LI5JLE0@mmp.sipr-dc.ucl.ac.be>; Mon, 08 Mar 2010 11:37:17 +0100 (CET)
3 Received: from mail.ietf.org (mail.ietf.org [64.170.98.32]) by smtp3.sgsi.ucl.ac.be (Postfix) with ESMTPT
   id B92351C60D7; Mon, 08 Mar 2010 11:36:51 +0100 (CET)
4 Received: from [127.0.0.1] (localhost [127.0.0.1]) by core3.amsl.com (Postfix) with ESMTPT id F066A3A68B9
   ; Mon, 08 Mar 2010 02:36:38 -0800 (PST)
5 Received: from localhost (localhost [127.0.0.1]) by core3.amsl.com (Postfix) with ESMTPT id A1E6C3A681B
   for <rrg@core3.amsl.com>; Mon, 08 Mar 2010 02:36:37 -0800 (PST)
6 Received: from mail.ietf.org ([64.170.98.32]) by localhost (core3.amsl.com [127.0.0.1]) (amavisd-new,
   port 10024) with ESMTPT id erw8ih2v8VQa for <rrg@core3.amsl.com>; Mon, 08 Mar 2010 02:36:36 -0800 (
   PST)
7 Received: from gair.firstpr.com.au (gair.firstpr.com.au [150.101.162.123]) by core3.amsl.com (Postfix)
   with ESMTPT id 03E893A67ED for <rrg@irtf.org>; Mon, 08 Mar 2010 02:36:35 -0800 (PST)
8 Received: from [10.0.0.6] (wira.firstpr.com.au [10.0.0.6]) by gair.firstpr.com.au (Postfix) with ESMTPT
   id DOA49175B63; Mon, 08 Mar 2010 21:36:37 +1100 (EST)
9 Date: Mon, 08 Mar 2010 21:36:38 +1100
10 From: Robin Whittle <rw@firstpr.com.au>
11 Subject: Re: [rrg] Recommendation and what happens next
12 In-reply-to: <C7B9C21A.4FAB%tony.li@tony.li>
13 To: RRG <rrg@irtf.org>
14 Message-id: <4B94D336.7030504@firstpr.com.au>
```


You can see that the sequence of SMTP servers can be longer than 2

- The simplest message body is just a text with ASCII text lines
- However we also need to send binary files and attachments
- RFC2045 introduced Multipurpose Internet Mail Extensions (MIME), that is, how to break down the body of the email into several pieces that can be used to send binary files.

- MIME-Version: The version number of the protocol (1.0 is the newest)
- Content-Type:
 - text/plain: simple text messages
 - multipart/mixed: simple text messages + other non text content
 - multipart/alternative: same message in text and in some other format
 - image, audio, video...: binary formats.

When the content type is multipart, the header also specifies a boundary to divide the parts.
- Content-Transfer-Encoding: ASCII, UTF-8, Base64... This can be repeated in every part.
- Multiparts can be nested in a tree structure


```
1 Date: Mon, 20 Sep 1999 16:33:16 +0200
2 From: Nathaniel Borenstein <nsb@bellcore.com>
3 To: Ned Freed <ned@innosoft.com>
4 Subject: Test
5 MIME-Version: 1.0
6 Content-Type: multipart/mixed; boundary="simple boundary"
7 --simple boundary
8 Content-Type: text/plain; charset=us-ascii
9 First part
10 --simple boundary
11 Content-Type: text/plain; charset=us-ascii
12 Second part
13 --simple boundary
```

image attached 



Leonardo Maccari <leonardo.maccari@unive.it>


a posta ▾

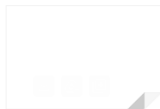
1x1 pixel image attached

--

Leonardo Maccari, Associate Professor @DAIS, Ca' Foscari University of Venice.

<http://www.dais.unive.it/~maccari/> GPG ID: AABE2BD7

Un allegato • Scansione eseguita da Gmail 



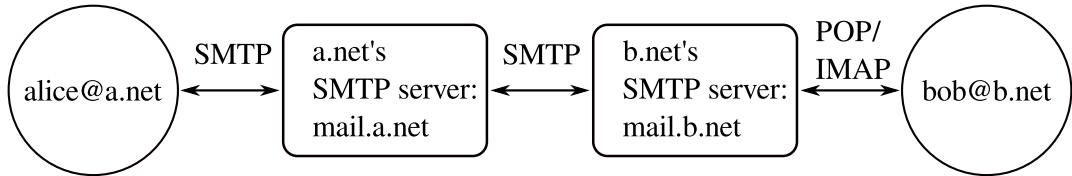
```
1 From: Leonardo Maccari <leonardo.maccari@unive.it>
2 Date: Wed, 18 Oct 2023 16:51:15 +0200
3 Message-ID: <CAMJfarHWLkY1hBNiPuxCP9_kV+XJNg-addi3kCCnMrw1-TU6sg@mail.gmail.com>
4 Subject: image attached
5 To: leonardo.maccari@unive.it
6 Content-Type: multipart/mixed; boundary="00000000000061c37d0607fec679"
7
8 --00000000000061c37d0607fec679
9 Content-Type: text/plain; charset="UTF-8"
10
11 1x1 pixel image attached
12
13 --
14 Leonardo Maccari, Associate Professor @DAIS, Ca Foscari University of Venice.
15 http://www.dais.unive.it/~maccari/      GPG ID: AABE2BD7
16
17 --00000000000061c37d0607fec679
18 Content-Type: image/gif; name="1pixel.gif"
19 Content-Disposition: attachment; filename="1pixel.gif"
20 Content-Transfer-Encoding: base64
21 Content-ID: <f_lnvvfaz20>
22 X-Attachment-Id: f_lnvvfaz20
23
24 R0lGODlhAAQABAIAAAP////////yH+EUNyZWFOZWQgd2l0aCBHSU1QACwAAAAAAQABAAACakQBADs=
25 --00000000000061c37d0607fec679--
```

Electronic Mail: e-mail

↳ 1.2 Simple Mail Transfer Protocol - SMTP

- The format of an e-mail is mostly relevant for the end-points of the e-mail transfer, called mail user agent (MUA).
- The Mail Transfer Agents (MTA) are instead the servers that take care of the delivery
- We still need to know how the communication between them takes place.
- For that we need two protocols, SMTP and POP/IMAP

e-mail Protocols



Text-Based Protocol



- SMTP is a text-based protocol like many other application-layer protocols on the Internet.
- It relies on the byte-stream service (so connection-oriented TCP).
- Servers listen on port 25, clients send commands that are each composed of one line of ASCII text terminated by CRLF.
- Servers reply by sending ASCII lines that contain a three digit numerical error/success code and optional comments.

- To send an e-mail you need to use commands. The most used ones are EHLO:
MAIL FROM:, RCPT TO:
- Then the message is started with the DATA and concluded with the QUIT command.

- The server always replies with a numeric code of 3 digits and a comment.
- Reply codes are grouped by their initial number:
 - 2XX: positive outcome.
 - 3XX: positive outcome but more input required
 - 4XX: negative outcome, but the problem is transient, try again with the same command
 - 5XX: negative outcome, the problem is permanent, do not try again

Reply Codes Examples



- 220: service ready. Used when the connection is opened.
- 250: Requested mail action okay, completed
- 450: Requested mail action not taken: mailbox unavailable (e.g., mailbox busy or temporarily blocked for policy reasons)
- 550: Requested action not taken: mailbox unavailable (e.g., mailbox not found, no access, or command rejected for policy reasons)

Example Exchange



```
1 S: 220 smtp.example.com ESMTP MTA information
2 C: EHLO mta.example.org
3 S: 250 Hello mta.example.org, glad to meet you
4 C: MAIL FROM:<alice@example.org>
5 S: 250 Ok
6 C: RCPT TO:<bob@example.com>
7 S: 250 Ok
8 C: DATA
9 S: 354 End data with <CR><LF>.<CR><LF>
10 C: From: "Alice Doe" <alice@example.org>
11 C: To: Bob Smith <bob@example.com>
12 C: Date: Mon, 9 Mar 2010 18:22:32 +0100
13 C: Subject: Hello
14 C:
15 C: Hello Bob
16 C: This is a small message containing 4 lines of text.
17 C: Best regards,
18 C: Alice
19 C: .
20 S: 250 Ok: queued as 12345
21 C: QUIT
22 S: 221 Bye
```

Let's Try

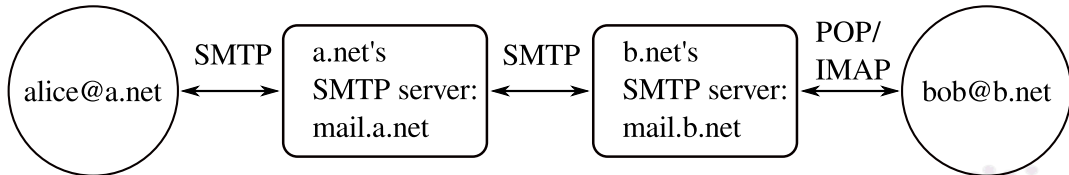


- `freesmtpservers.com` is a website that offers you a simple SMTP server you can connect to, and deliver e-mail
- e-mail will not be delivered to the real destination, they will be delivered to a fake local inbox that you can actually see at:
`https://www.wpoven.com/tools/free-smtp-server-for-testing`
- `telnet smtp.freesmtpservers.com 25`

SMTP Operation

Some practical notes on SMTP. These are not rules mandated by the protocol, they are commonly used configurations.

- The MUA must be configured with a domain name of an SMTP server, in the image, Alice will configure its MUA to use `mail.a.net` as the SMTP server.
- The MUA will connect and use some special commands to perform an authentication.
- In this case, the client can send e-mail to any domain. We say the SMTP server allows *relaying* of e-mails to other domains, for instance `bob@b.net`



- Initially, there was no authentication for SMTP, then extensions to the protocol were added to support authentication

```
1  S: 220-smtp.example.com ESMTP Server
2  C: EHLO client.example.com
3  S: 250-smtp.example.com Hello client.example.com
4  S: 250-AUTH DIGEST-MD5 CRAM-MD5
5  S: 250-ENHANCEDSTATUSCODES
6  S: 250 STARTTLS
7  C: AUTH CRAM-MD5
8  S: 334 PDQxOTI5NDIzNDEuMTI4Mjg0NzJAc291cmNlZm91ci5hbmRyZXcuY211LmVkdT4=
9  C: cmpzMyB1YzNhNTlmZWQzOTVhYmExZWZWM2MzY3YzRmNGIOMWFjMA==
10 S: 235 2.7.0 Authentication successful
```

- A server that supports authentication will add more 250- lines as a EHLO reply, with a list of supported authentication methods (line 4-6)
- The client can choose one (line 7)

Challenge-Response Authentication Mechanism



- CRAM-MD5 stands for Challenge-Response Authentication Mechanism, based on MD5
- It is one of many authentication methods we have, in this case line 8 contains a time stamp (encoded in base64)
- Line 9 contains the username, concatenated with the MD5 hash of the user password and the timestamp¹
- Using MD5 makes it possible to show to the server that the client knows the password, without sending it in clear.
- This is of course a bit rudimentary, nowadays we use the TLS protocol that adds a layer of cryptography on top of this (more in future lessons)

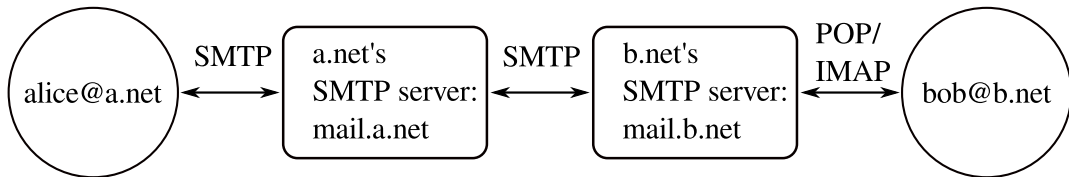
¹See RFC 2196 <https://datatracker.ietf.org/doc/html/rfc2195>



SMTP Operation (2)



- When the SMTP server at a.net has received the e-mail from the MUA, it has to deliver it to the SMTP server of the destination domain: b.net
- However, the server does not know what is the IP of the mail server that manages emails with destination domain b.com.



SMTP Operation (3)



- So it will make a DNS resolution using MX type to its DNS server, and receive a non-terminal response, possibly with more than one option, ranked by their score (lowest, better)
- Let's imagine the receiver is @gmail.com

```
1 # example with gmail.com
2 $ dig gmail.com MX
3 ;; ANSWER SECTION:
4 gmail.com.      2696  IN  MX  5  gmail-smtp-in.1.google.com.
5 gmail.com.      2696  IN  MX  20 alt2.gmail-smtp-in.1.google.com.
6 gmail.com.      2696  IN  MX  30 alt3.gmail-smtp-in.1.google.com.
7 gmail.com.      2696  IN  MX  40 alt4.gmail-smtp-in.1.google.com.
8 gmail.com.      2696  IN  MX  10 alt1.gmail-smtp-in.1.google.com.
```

- It will then query again and receive the IP address of mail.b.com

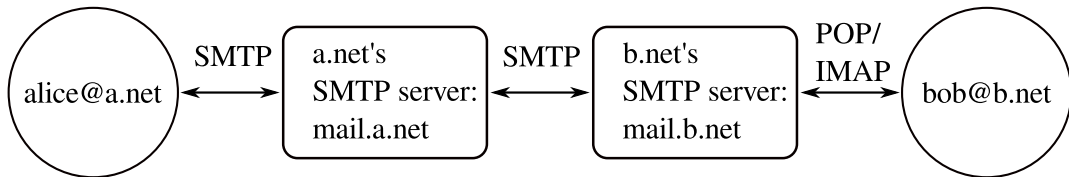
```
1 $ dig gmail-smtp-in.1.google.com
2 ;; ANSWER SECTION:
3 gmail-smtp-in.1.google.com. 300 IN  A 108.177.127.27
```



SMTP Operation (4)



- The server at a.com will now connect (as a client) to the SMTP server mail.b.net. This is the same process the MUA did towards mail.a.net
- However, the MTA does not perform any authentication, so the server at b.net will not allow relaying: Only e-mail destined to b.com will be allowed.



SMTP Operation (5)

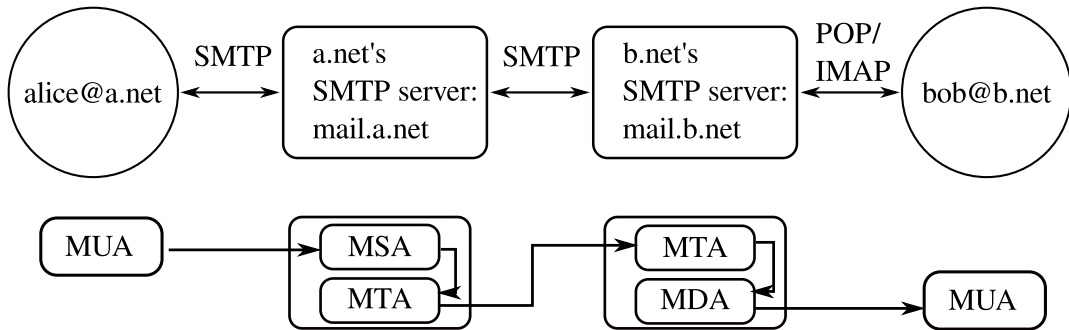


- When connecting to an MTA, a MUA or another MTA will make the same actions, send an e-mail
- However, the MUA connects to the MTA that shares valid credentials with
- The MUA will then make an authentication.
- The MTA will then allow **relaying**: sending e-mails to any domain in the world.
- Without the authentication, the MTA will not allow **relaying**: the client can send e-mail only if the destination domain is the one configured on the MTA to be its own domain.

SMTP Operation (6)



- This is an important step that was made explicit in RFC 5068.
- The RFC makes this explicit using two names for the two MTAs, introducing the **Mail Submission Agent**: an MTA that performs authentication



- *Submission Port Availability: the domain's MSAs MUST support the SUBMISSION port 587*
- *Submission Authentication: MSAs MUST perform authentication on the identity asserted during all mail transactions on the SUBMISSION port, even for a message having a RCPT TO address that would not cause the message to be relayed outside of the local administrative domain.*
- *Submission Authorization: An operator of an MSA MUST ensure that the authenticated identity is authorized to submit email, based on an existing relationship between the submitting entity and the operator.*
- *Submission Accountability after Submission: For a reasonable period of time after submission, the message SHOULD be traceable by the MSA operator to the authenticated identity of the user who sent the message*

- In the end, the e-mail was delivered to the server at b.com
- The e-mail will be saved in a mailbox for the user, that will be able to retrieve it with another protocol: POP (or IMAP)
- The last image introduced a last component, the Mail Delivery Agent (MDA)

Electronic Mail: e-mail

↳ 1.3 Post Office Protocol (POP)

- Once the e-mail arrived at the last MTA, it is delivered to another entity called a Mail Delivery Agent (MDA)
- The MDA makes it available to the MUA of Bob with a dedicated protocol.
- The POP protocol (currently at version 3) is one of the protocols used to deliver the e-mail from the MDA to the MUA
- POP was superseded by IMAP, that is a more feature rich protocol, but its basic principles are the same.

- POP is another textual protocol, with its own commands.
- Relevant commands are USER (followed by the username), PASS (followed by the password), STAT (asks for the mailbox status), LIST (list new messages), RETR (followed by a number, retrieves the n-th message)
- The server answers with +OK or error codes similarly to SMTP

POP3 Example



```
1  S:    +OK POP3 server ready
2  C:    USER alice
3  S:    +OK
4  C:    PASS 12345pass
5  S:    +OK alice maildrop has 2 messages (620 octets)
6  C:    STAT
7  S:    +OK 2 620
8  C:    LIST
9  S:    +OK 2 messages (620 octets)
10 S:    1 120
11 S:    2 500
12 S:    .
13 C:    RETR 1
14 S:    +OK 120 octets
15 S:    <the POP3 server sends message 1>
16 S:    .
17 C:    DELE 1
18 S:    +OK message 1 deleted
19 C:    QUIT
20 S:    +OK POP3 server signing off (1 message left)
```

- Note that in the original protocol the authentication was performed in the clear
- Nowadays also POP supports authentication extensions (including also CRAM-MD5, see <https://datatracker.ietf.org/doc/html/rfc2195>)



Example Gmail Configuration



Passaggio 2: modifica le impostazioni di SMTP e altre impostazioni nel client di posta

Utilizza la tabella di seguito per aggiornare il tuo client con le informazioni corrette. Per assistenza, cerca le istruzioni per l'impostazione di IMAP nel Centro assistenza del tuo client di posta.

Server posta in arrivo (IMAP)	imap.gmail.com Richiede SSL: Sì Porta: 993
Server posta in uscita (SMTP)	smtp.gmail.com Richiede SSL: Sì Richiede TLS: Sì (se disponibile) Richiede autenticazione: Sì Porta per SSL: 465 Porta per TLS/STARTTLS: 587
Nome completo o Nome visualizzato	Il tuo nome
Nome account, Nome utente o Indirizzo email	Il tuo indirizzo email completo
Password	La tua password Gmail

