

Introduction to SMTP

RES, Lecture 4

Olivier Liechti



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD

www.heig-vd.ch

13



Labo SMTP, part 1

Olivier Liechti

14



Labo SMTP, part 2

Olivier Liechti

15



Labo SMTP, part 3

Olivier Liechti

16



Labo SMTP, part 4

Olivier Liechti

- SMTP demo & hints
- SMTP protocol
- Mock server
- Implementation walk-through



What happens when Bob wants
to **send an e-mail** to Alice?



Bob uses **Thunderbird** to write his mail.



Alice uses **MS Outlook** to check and read her mails.



In the technical specs (RFCs), these programs are called **Mail User Agents (MUA)**

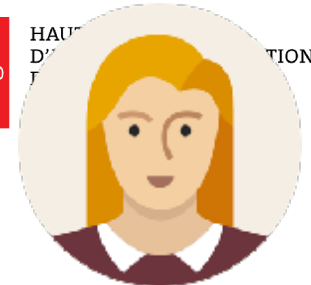




Bob uses his professional e-mail address. His company runs a **MS Exchange Server**.



heig-vd



Alice uses her private address. She has an account (and a **mailbox**) on the **Google gmail** infrastructure.



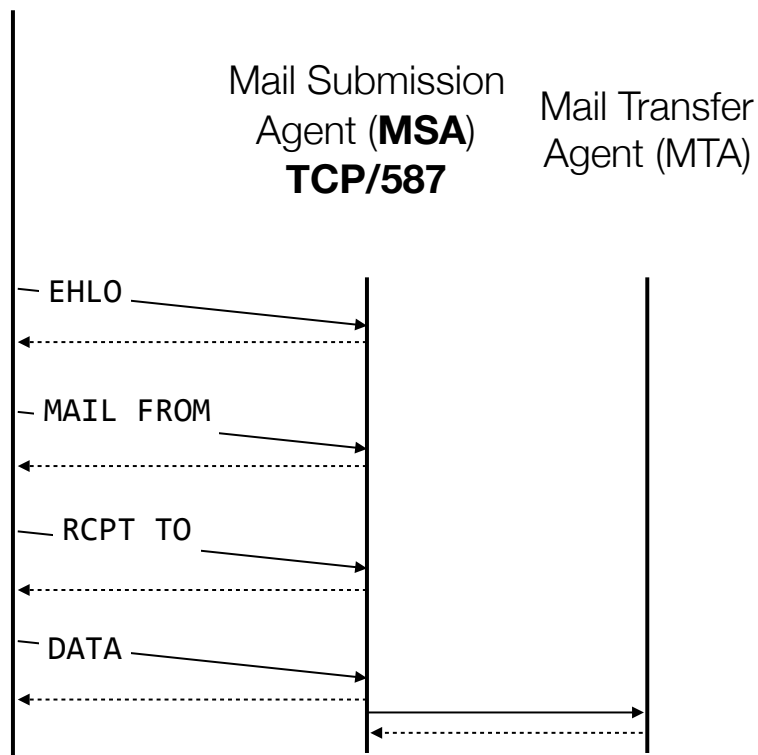


Bob writes a message to “**alice.res@gmail.com**”. He pushes on the “Send” button.

The Exchange Server is made of **2 logical components**: the **MSA** and the **MTA**.

Bob's MUA asks Bob's MSA to deliver the mail. It uses the **SMTP** protocol for that purpose and (should) use TCP port 587.

After enforcing **usage policies**, the MSA delegates the work to the MTA. We don't know how.





Mail Transfer
Agent (MTA)

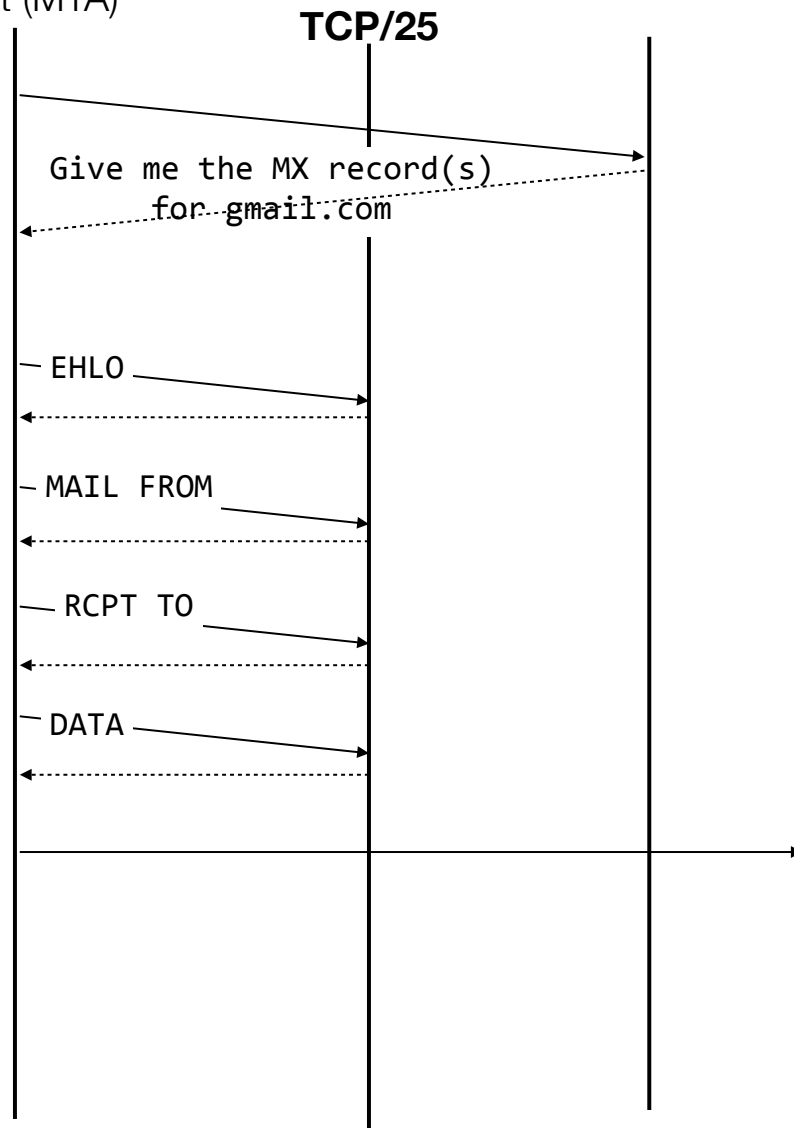


Mail Transfer
Agent (MTA)

DNS



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch



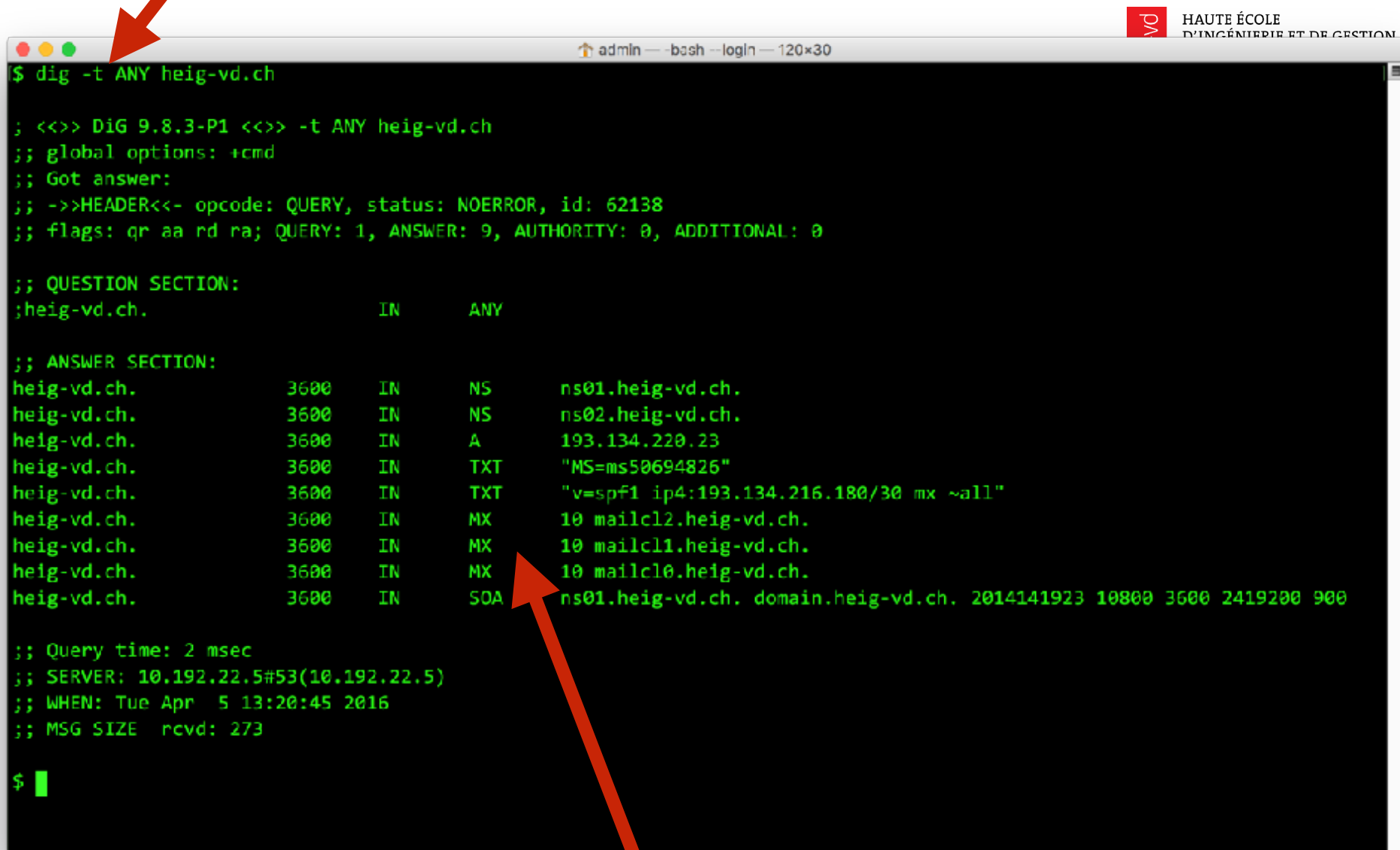
Bob's MTA initially does not know where to forward the mail...

It issues a **DNS** query to get a list of **MX records** for Alice's domain (gmail.com).

When Bob's MTA knows the IP address of Alice's MTA, it uses the **SMTP** protocol once more to forward the message. **TCP port 25** is used in this case.

When Alice's MTA receives the mail, it stores it in Alice's **mailbox** (for later retrieval).

dig -t ANY heig-vd.ch



```
$ dig -t ANY heig-vd.ch

; <<>> DiG 9.8.3-P1 <<>> -t ANY heig-vd.ch
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 62138
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;heig-vd.ch.                IN      ANY

;; ANSWER SECTION:
heig-vd.ch.                3600    IN      NS      ns01.heig-vd.ch.
heig-vd.ch.                3600    IN      NS      ns02.heig-vd.ch.
heig-vd.ch.                3600    IN      A       193.134.220.23
heig-vd.ch.                3600    IN      TXT     "MS=ms50694826"
heig-vd.ch.                3600    IN      TXT     "v=spf1 ip4:193.134.216.180/30 mx ~all"
heig-vd.ch.                3600    IN      MX      10 mailcl2.heig-vd.ch.
heig-vd.ch.                3600    IN      MX      10 mailcl1.heig-vd.ch.
heig-vd.ch.                3600    IN      MX      10 mailcl0.heig-vd.ch.
heig-vd.ch.                3600    IN      SOA     ns01.heig-vd.ch. domain.heig-vd.ch. 2014141923 10800 3600 2419200 900

;; Query time: 2 msec
;; SERVER: 10.192.22.5#53(10.192.22.5)
;; WHEN: Tue Apr  5 13:20:45 2016
;; MSG SIZE  rcvd: 273

$
```

MX records point to the SMTP servers for the domain



SMTP
587

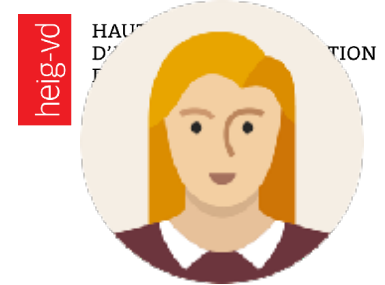


In the last step, Alice's MUA uses another protocol (e.g. IMAP, POP3) to fetch mails from the mailbox.

SMTP
25



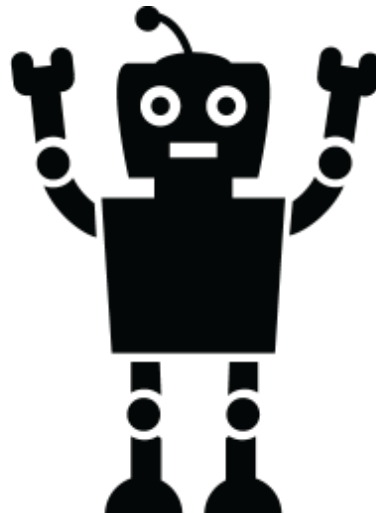
IMAP/POP3





Let's be human Exchange Servers
(and play the role of Bob's MTA).

But instead of forwarding the mail
to gmail, let's forward the mail via
the **HEIG-VD's SMTP** server.



```
dig -t MX heig-vd.ch  
heig-vd.ch. 600 IN MX 10 mail01.heig-vd.ch.
```

```
telnet mail01.heig-vd.ch 25
```

```
openssl s_client -starttls smtp -crlf -connect  
mail01.heig-vd.ch:25 (or use 465)
```

```
EHLO mycompany.com
```

```
$ telnet mailcl10.heig-vd.ch 25
mailcl10.heig-vd.ch: nodename nor servname provided, or not known
$ telnet mailcl0.heig-vd.ch 25
Trying 193.134.216.181...
Connected to mailcl0.heig-vd.ch.
Escape character is '^]'.
220 heig-vd.ch ESMTP MailCleaner (Enterprise Edition 2016.01) Tue, 05 Apr 2016 14:18:24
+0200
EHLO mycompany.com
250-heig-vd.ch Hello mbp-de-admin.einet.ad.eivd.ch [10.192.116.92]
250-SIZE 20480000
250-8BITMIME
250-PIPELINING
250-AUTH PLAIN LOGIN
250-STARTTLS
250 HELP
MAIL FROM: bob@bob.com
250 OK
RCPT TO: olivier.liechti@wasabi-tech.com
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
From: bob@areyousure.com
To: olivier.liechti@wasabi-tech.com
Subject: demo

Ok. Cool. Bye.
.
250 OK id=1anPx9-0003KC-BC
quit
221 heig-vd.ch closing connection
Connection closed by foreign host.
```

SMTP command
!=
Message header

<https://tools.ietf.org/html/rfc5321>

Table of Contents

RFC 5321

SMTP

October 2008

1. Introduction	5	3.9.2. List	31
1.1. Transport of Electronic Mail	5	4. The SMTP Specifications	32
1.2. History and Context for This Document	5	4.1. SMTP Commands	32
1.3. Document Conventions	6	4.1.1. Command Semantics and Syntax	32
2. The SMTP Model	7	4.1.2. Command Argument Syntax	41
2.1. Basic Structure	7	4.1.3. Address Literals	43
2.2. The Extension Model	9	4.1.4. Order of Commands	44
2.2.1. Background	9	4.1.5. Private-Use Commands	46
2.2.2. Definition and Registration of Extensions	10	4.2. SMTP Replies	46
2.2.3. Special Issues with Extensions	11	4.2.1. Reply Code Severities and Theory	40
2.3. SMTP Terminology	11	4.2.2. Reply Codes by Function Groups	50
2.3.1. Mail Objects	11	4.2.3. Reply Codes in Numeric Order	52
2.3.2. Senders and Receivers	12	4.2.4. Reply Code 502	53
2.3.3. Mail Agents and Message Stores	12	4.2.5. Reply Codes after DATA and the Subsequent <CRLF>,<CRLF>	53
2.3.4. Host	13	4.3. Sequencing of Commands and Replies	54
2.3.5. Domain Names	13	4.3.1. Sequencing Overview	54
2.3.6. Buffer and State Table	14	4.3.2. Command-Reply Sequences	55
2.3.7. Commands and Replies	14	4.4. Trace Information	57
2.3.8. Lines	14	4.5. Additional Implementation Issues	61
2.3.9. Message Content and Mail Data	15	4.5.1. Minimum Implementation	61
2.3.10. Originator, Delivery, Relay, and Gateway Systems	15	4.5.2. Transparency	62
2.3.11. Mailbox and Address	15	4.5.3. Sizes and Timeouts	62
2.4. General Syntax Principles and Transaction Model	16	4.5.3.1. Size Limits and Minimums	62
3. The SMTP Procedures: An Overview	17	4.5.3.1.1. Local-part	63
3.1. Session Initiation	18	4.5.3.1.2. Domain	63
3.2. Client Initiation	18	4.5.3.1.3. Path	63
3.3. Mail Transactions	19	4.5.3.1.4. Command Line	63
3.4. Forwarding for Address Correction or Updating	21	4.5.3.1.5. Reply Line	63
3.5. Commands for Debugging Addresses	22	4.5.3.1.6. Text Line	63
3.5.1. Overview	22	4.5.3.1.7. Message Content	63
3.5.2. VRFY Normal Response	24	4.5.3.1.8. Recipients Buffer	64
3.5.3. Meaning of VRFY or EXPN Success Response	25	4.5.3.1.9. Treatment When Limits Exceeded	64
3.5.4. Semantics and Applications of EXPN	26	4.5.3.1.10. Too Many Recipients Code	64
3.6. Relaying and Mail Routing	26	4.5.3.2. Timeouts	65
3.6.1. Source Routes and Relaying	26	4.5.3.2.1. Initial 220 Message: 5 Minutes	65
3.6.2. Mail eXchange Records and Relaying	26	4.5.3.2.2. MAIL Command: 5 Minutes	65
3.6.3. Message Submission Servers as Relays	27	4.5.3.2.3. RCPT Command: 5 Minutes	65
3.7. Mail Gatewaying	28	4.5.3.2.4. DATA Initiation: 2 Minutes	66
3.7.1. Header Fields in Gatewaying	28	4.5.3.2.5. Data Block: 3 Minutes	66
3.7.2. Received Lines in Gatewaying	29	4.5.3.2.6. DATA Termination: 10 Minutes	66
3.7.3. Addresses in Gatewaying	29	4.5.3.2.7. Server Timeout: 5 Minutes	66
3.7.4. Other Header Fields in Gatewaying	29	4.5.4. Retry Strategies	66
3.7.5. Envelopes in Gatewaying	30	4.5.5. Messages with a Null Reverse-Path	60
3.8. Terminating Sessions and Connections	30	5. Address Resolution and Mail Handling	69
3.9. Mailing Lists and Aliases	31	5.1. Locating the Target Host	69
3.9.1. Alias	31	5.2. IPv6 and MX Records	71
		6. Problem Detection and Handling	71