Messaging protocols (SMTP,POP3 and IMAP4)

Introduction to e-mail

E-mail is considered as being the most widely used service on the Internet. So the TCP/IP protocol suite offers a range of protocols allowing the easy management of email routing over the network.

The SMTP protocol

The **SMTP protocol** (*Simple Mail Transfer Protocol*) is the standard protocol enabling mail to be transferred from one server to another by point to point connection.

This is a protocol operating in online mode, encapsulated in a TCP/IP frame. The mail is sent directly to the recipient's mail server. SMTP protocol works using text commands sent to the SMTP server (on port 25 by default). Each command sent by the client (validated by the ASCII character string *CR/LF*, equivalent to a press on the enter key) is followed by a response from the SMTP server comprising of a number and a descriptive message.

Here is a scenario of a request for sending mail to an SMTP server

- When opening the SMTP session, the first command to be sent is the *HELO* command followed by a space (written *<SP>*) and the domain name of your machine (in order to say "hello, I am this machine"), then validated by enter (written *<CRLF>*). Since April 2001, the specifications for the SMTP protocol, defined in RFC 2821, mean that the HELO command is replaced by the *EHLO* command.
- The second command is "*MAIL FROM:*" followed by the email address of the originator. If the command is accepted the server sends back the message "*250 OK*"
- The next command is "*RCPT TO:*" followed by the email address of the recipient. If the command is accepted the server sends back the message "*250 OK*"
- The *DATA* command is the third stage for sending email. It announces the start of the message body. If the command is accepted the server sends back an intermediary message numbered *354* indicating that the sending of the email body can begin and considers the collection of following lines until the end of the message located by a line containing only a dot. The email body possibly contains some of the following headers:
- Date
- Subject
- Cc
- Bcc

- From

If the command is accepted the server sends back the message "*250 OK*"

Here is an example of a transaction between a client (C) and an SMTP server (S)

S: 220 smtp.commentcamarche.net SMTP Ready

C: EHLO machine1.commentcamarche.net

S: 250 smtp.commentcamarche.net

C: MAIL FROM:<webmaster@kioskea.net>

S: 250 OK

C: RCPT TO:<meandus@meandus.net>

S: 250 OK

C: RCPT TO:<tittom@tittom.fr>

S: 550 No such user here

C: DATA

S: 354 Start mail input; end with <CRLF>.<CRLF>

C: Subject: Hello

C: Hello Meandus,

C: How are things?

C:

C: See you soon!

C: <CRLF>.<CRLF>

S: 250 OK

C: QUIT

R: 221 smtp.commentcamarche.net closing transmission

The basic specifications of the SMTP protocol mean that all the characters sent are coded in ASCII code over 7 bits and that the 8th bit is explicitly put at zero. So to send accented characters it is necessary to resort to algorithms integrating MIME specifications:

- **base64** for attached files
- **quoted-printable** (abbreviated to *QP*) for special characters contained within the message body

It is therefore possible to send an email using a simple telnet on port 25 of the SMTP server:

telnet smtp.commentcamarche.net 25

(the server indicated above is deliberately nonexistent, you can try by replacing*commentcamarche.net* by the domain name of your internet service provider)

Here is a summary of the principal SMTP commands

Command	Example	Description
HELO (now EHLO)	EHLO 193.56.47.125	Identification using the IP address or domain name of the originator computer
MAIL FROM:	MAIL FROM: originator@domain.com	Identification of the originator's address
RCPT TO:	RCPT TO: recipient@domain.com	Identification of the recipient's address
DATA	DATA message	Email body
QUIT	QUIT	Exit the SMTP server
HELP	HELP	List of SMTP commands supported by the server

All the specifications for the SMTP protocol are defined in <u>RFC 821</u> (since April 2001, the SMTP protocol specifications are defined in <u>RFC 2821</u>).

The POP3 protocol

The **POP protocol** (*Post Office Protocol*) as its name indicates makes it possible to go and collect the email on a remote server (POP server). It is necessary for people not permanently connected to the Internet so that they can consult emails received offline. There are two main versions of this protocol, POP2 and POP3, to which <u>ports</u>109 and 110 are allocated respectively and which operate using radically different text commands.

Just like with the SMTP protocol, the POP protocol (POP2 and POP3) works using text commands sent to the POP server. Each of these commands sent by the client (validated by the *CR/LF* string) comprises a key word, possibly accompanied by one or several arguments and is followed by a response from the POP server comprising of a number and a descriptive message.

Here is a summary of the principal POP2 commands:

POP2 Commands	
Command	Description

HELLO	Identification using the IP address of the originator computer
FOLDER	Name of the inbox to be consulted
READ	Number of the message to be read
RETRIEVE	Number of the message to be picked up
SAVE	Number of the message to be saved
DELETE	Number of the message to be deleted
QUIT	Exit the POP2 server

Here is a summary of the principal POP3 commands

POP3 Commands	
Command	Description
USER identification	This command makes it possible to be authenticated. It must be followed by the user name, i.e. a character string identifying the user on the server. The USER command must precede the *PASS* command.
PASS password	The *PASS* command makes it possible to specify the user's password where the name has been specified by a prior *USER* command.
STAT	Information on the messages contained on the server
RETR	Number of the message to be picked up
DELE	Number of the message to be deleted
LIST [msg]	Number of the message to be displayed
NOOP	Allows the connection to be kept open in the event of inactivity
TOP <messageID> <n>	Command displaying *n* lines of the message, where the number is given in the argument. In the event of a positive response from the server, it will send back the message headers, then a blank line and finally the first *n* lines of the message.
UIDL [msg]	Request to the server to send back a line containing information about the message possibly given in the argument. This line contains a character string called

	a *unique identifier listing*, making it possible to uniquely identify the message on the server, independently of the session. The optional argument is a number relating to a message existing on the POP server, i.e. an undeleted message).
QUIT	The *QUIT* command requests exit from the POP3 server. It leads to the deletion of all messages marked as deleted and sends back the status of this action.

The POP3 protocol thus manages authentication using the user name and password; however, it is not secure because the passwords, like the emails circulate in plain text (in an <u>unencrypted</u> way) over the network. In reality, according to <u>RFC 1939</u>, it is possible to encrypt the password using the <u>MD5</u>algorithm and thus benefit from secure authentication. However, since this command is optional, few servers implement it. Furthermore, POP3 protocol blocks inboxes during access which means that simultaneous access of the same inbox by two users is impossible.

In the same way that it is possible to send an email using telnet, it is also possible to access your incoming mail using a simple telnet over the port for the POP server (110 by default):

telnet mail.commentcamarche.net 110
(the server indicated above is deliberately nonexistent, you can try by replacing*commentcamarche.net* by the domain name of your internet service provider)
S: +OK mail.commentcamarche.net POP3 service
S: (Netscape Messaging Server 4.15 Patch 6 (built Mar 31 2001))
C: USER jeff
S: +OK Name is a valid mailbox
C: PASS password
S: +OK Maildrop ready
C: STAT
S: +OK 2 0
C: TOP 1 5
S: Subject: Hello
S: Hello Meandus,
S: How are things?
S:
S: See you soon!
C: QUIT
S: +OK

⚠	The data display that you capture depends on the Telnet client that you are using. Depending on your Telnet client, you may need to activate the *echo local* option.

The IMAP protocol

The **IMAP** protocol (*Internet Message Access Protocol*) is an alternative protocol to that of POP3 but offering many more possibilities:

- IMAP allows several simultaneous accesses to be managed
- IMAP makes it possible to manage several inboxes
- IMAP provides more criteria which can be used to sort emails

Source: http://en.kioskea.net/contents/289-messaging-protocols-smtp-pop3-and-imap4