Le courrier électronique et le protocole SMTP

14 mars 2001

Table des matières

1 Le courrier électronique et le protocole SMTP

Le courrier électronique est l'un des plus anciens services implémentés par les réseaux informatiques. Le besoin de communiquer, d'échanger des informations de façon simple et rapide a poussé ces premiers réseaux à être équipés d'un tel service. Au départ, ce service se contentait de copier un fichier d'une machine à une autre en le mettant dans le fichier *boîte aux lettres* du correspondant. Au fil des développements et des besoins (complexité du réseau, augmentation constante du flux de messages, multimédia...), beaucoup d'éléments se sont greffées par-dessus.

Le courrier électronique est un service utilisé partout dans le monde, comme le montre la figure 1 : c'est un des rares services à être présent dans à peu près tous les systèmes, y compris dans les pays n'ayant pas de réseau très développé.

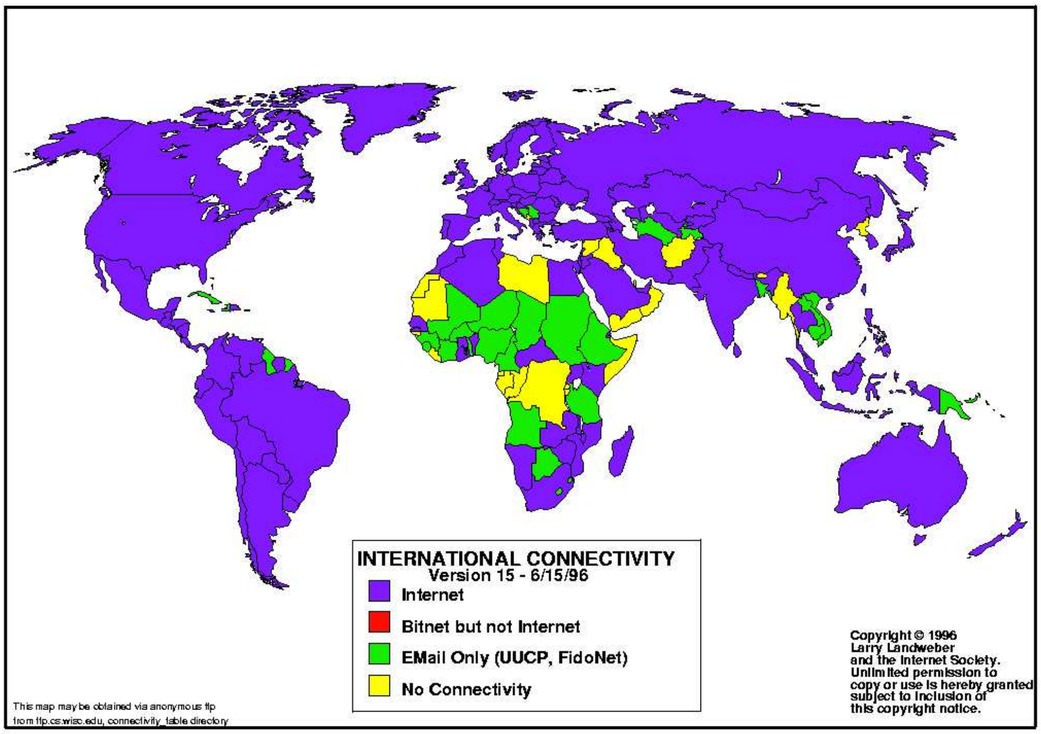

FIG. 1 – Connectivité des pays dans le monde.

Il existe beaucoup de programmes se chargant du transport du courrier. `sendmail` est sans aucun doutes le plus utilisé, toute plateformes confondues. Il a été écrit par Eric Allman au début des années 80, et s'est imposé comme un standard. La documentation du kit jussieu [1] décrit bien ce protocole, en français. Sinon, l'éternel guide de l'administration réseau sous Linux [2] explique tout cela très bien.

1.1 Comment le courrier est-il délivré ?

Il y a deux façons courantes d'envoyer ou reçevoir son courrier :

– Utiliser un serveur de méls local (comme `sendmail` ou `qmail`, par exemple) qui va accepter des connexions de la part d'autre système de courrier électronique distants.

– Se connecter directement sur un serveur POP ou IMAP (comme celui de son fournisseur d'accès, par exemple) pour rappatrier ses courriers en local ou les consulter à distance, et les envoyer.

Les besoins d'un réseau ou des utilisateurs déterminent la façon dont le courrier sera délivré. Si on est un simple utilisateur relié par une seule machine à son fournisseur d'accès, en utilisera plutôt la connexion sur un serveur POP ou IMAP, plus facile à mettre en oeuvre. Si on a plusieurs machines à configurer, on préférera la configuration d'un serveur de méls.

1.1.1 Utiliser un serveur de méls local

Un système de courrier électronique est composé de plusieurs modules spécialisés dans des tâches précises. Diviser les tâches permet de rendre plus facile l'écriture des logiciels.

Ces modules sont :

– Le ***Mail Transfer Agent (MTA)*** qui agit comme le centre de tri postal. Il reçoit donc les courriers du réseau local ou d'autre centres de tris, et décide en fonction de l'adresse du destinataire si ce courrier doit être envoyé à un autre centre de tri ou si il doit être donné au facteur pour distribution locale. Les MTA les plus connus sont `sendmail`, `qmail`, `postfix`, `smail`, `exim` et `mmdf`.

– Le ***Mail Delivery Agent (MDA)*** qui joue le rôle de ce facteur. Il distribue donc les courriers dans le réseau local, ce qui revient à copier dans la boîte aux lettres de l'utilisateur les courriers arrivant. Il ne prend aucune décision quant au routage, mais se contente simplement d'acheminer le courrier suivant la route précisée par le MTA. Comme MDA, on peut citer `mail` ou `procmail`.

– Le ***Mail User Agent (MUA)*** qui permet à l'utilisateur de consulter sa boîte aux lettres, d'écrire de nouveaux messages. Le MUA transmet donc les courriers au MTA qui va les redistribuer en fonction de l'adresse du destinataire au réseau local pour attérir dans la boîte aux lettres de l'utilisateur ou aux autres MTA. Les MUA les plus connus sont `mail`, `pine`, `mutt`.

La ***boîte aux lettres (mailbox)*** est en fait un simple fichier où tous les messages commencent par une ligne *From* indiquant l'adresse de l'expéditeur au format UUCP (parfois appellé le "From Unix"). Cette boîte aux lettres peut aussi bien se trouver dans le fichier `/var/spool/mail/login` (systèmes d'origine BSD), `/usr/mail/login` (systèmes d'origine System-V) ou encore `/var/mail/login`.

Le figure 2 montre comment ces composants interragissent entre eux :
Pour la réception des courriers :

– Les différents MTA répartits sur l'Internet interragissent au travers d'un port définit par la RFC 821. Ce port est décrit dans le fichier `/etc/services` :

```
smtp            25/tcp          mail
```

C'est donc via le port 25 que communiquent les différents MTA.

– Le MTA connaît le MDA grâce à son fichier de configuration. Pour `sendmail`, par exemple, le fichier `/etc/sendmail.cf` contient une ligne comme celle ci :

```
Mlocal,         Path=/usr/bin/procmail, Flags=lsDFMAw5:/|wA9q,
                Sender=10, Recipient=20, Argv=procmail -Y -a $h -d $u
```

lui indiquant d'utiliser `procmail` comme MDA.

– Le MDA dépose alors le courrier dans la boîte aux lettres de l'utilisateur.

Pour l'envoi de courriers :

– Le MUA consulte son fichier de configuration pour savoir à quel MTA envoyer le message. Le MUA communique avec le MTA de la façon suivante :
`cat message | MTA`
où ***message*** est le courrier saisit, grâce au MUA, qui doit être envoyé.

– Le MTA met le courrier dans une file d'attente pour le départ. Régulièrement, il fait des envois groupés de ces courriers en contactant directement les serveurs distants au travers d'une connexion SMTP.

1.1.2 Utiliser une connection POP ou IMAP

Si on utilise une connexion POP pour lire ses messages, on récupère les courriers directement sur un serveur POP qui les stocke pour l'utilisateur (ce serveur peut être celui du fournisseur d'accès, par exemple). On peut également laisser une copie des courriers sur le serveur POP, pour pouvoir être ensuite récupérés d'un autre endroit. Si on utilise une connexion IMAP,

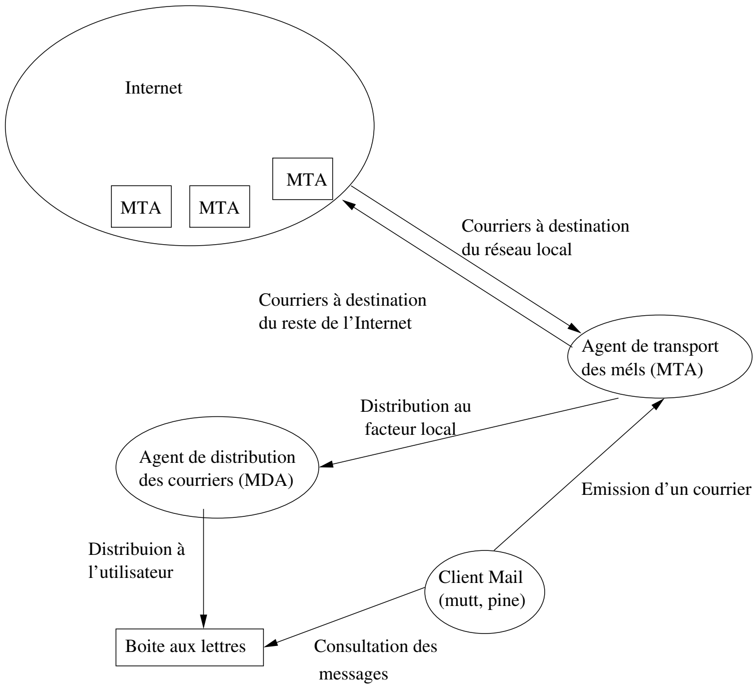

FIG. 2 – Interaction entre les divers composants d'un système de messagerie.

on consulte directement la boîte aux lettres à distance, ce qui permet d'en avoir une seule et unique copie, où que l'on soit dans le monde. De plus, il n'y a pas de transfert du courrier, donc la consultation est rapide. Pour envoyer des courriers, on se connecte à un serveur qui va relayer notre courrier vers les serveurs à contacter au travers d'une connexion SMTP. En général, ce serveur relai est celui du fournisseur d'accès (figure 3).

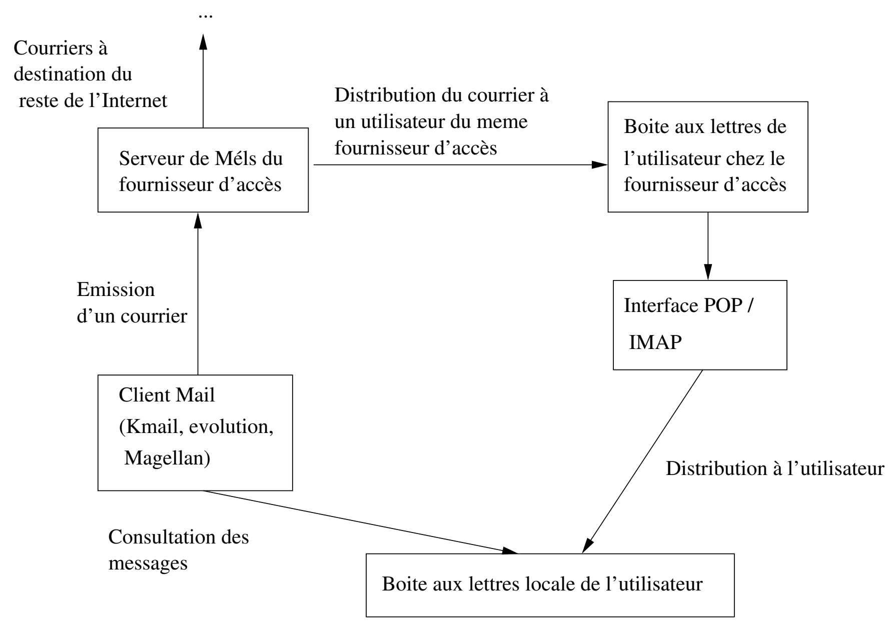

FIG. 3 – Interaction des divers éléments dans le cas d'une connexion POP / IMAP

Le protocole POP définit par la RFC 1939 permet de récupérer ses méls sur un serveur distant.

Voici un exemple de dialogue avec un serveur POP3 :

```
mdecore@tarsier:~$ telnet localhost pop3
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
+OK POP3 localhost v7.64 server ready
USER piou
+OK User name accepted, password please
PASS piou
+OK Mailbox open, 1 messages
```

```
STAT
+OK 1 545
LIST 1
+OK 1 545
RETR 1
+OK 545 octets
Return-Path: <mdecore>
Received: (from mdecore@localhost)
        by tarsier.athome.chezmoi (8.9.3+3.2W/8.9.3/Mathieu DECORE -
14/10/2000) id SAA00982
        for piou; Wed, 27 Dec 2000 18:32:04 +0100
Date: Wed, 27 Dec 2000 18:32:04 +0100
From: Mathieu DECORE <mdecor@ens.uvsq.fr>
To: "Ben c'est piou ;-)" <piou@tarsier.athome.chezmoi>
Subject: Test de pop
Message-ID: <20001227183204.A977@athome.chezmoi>
Mime-Version: 1.0
Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit
User-Agent: Mutt/1.0.1i
Status:

Voilà...
.
DELE 1
+OK Message deleted
QUIT
+OK Sayonara
Connection closed by foreign host.
mdecore@tarsier:~$
```

Après s'être authentifié (`USER piou`) et avoir donné son mot de passe (`PASS piou`), on regarde le nombre de messages ainsi que la taille de la boîte aux lettres, en octets (`STAT`). On liste les caractéristiques du premier message (`LIST 1`) que l'on lit (`RETR 1`), et que l'on efface (`DELE 1`) avant de quitter la session (`QUIT`).

1.2 Qu'est-ce qu'un message ?

Un courrier électronique est constitué de données indiquant l'adresse du destinataire, les moyens de transport utilisés, et d'un corps, qui est le texte du message. C'est un peu comme pour un courrier postal : l'enveloppe contient sur le dessus des informations, et à l'intérieur le message proprement dit.

Ces données administratives sont en fait divisées en deux catégories :

– Dans la première se trouvent toutes les informations spécifiques aux moyens de transport, comme l'adresse de l'expéditeur et du destinataire. Cette partie s'appelle l'***enveloppe***. Elle est modifiée par chaque programme relayant le message.

– Dans la deuxième se trouvent les données nécessaires à la manipulation du message, et qui n'est pas spécifique d'un mécanisme de transport : le sujet, les destinataires, la date d'expédition…Ces données sont en général placées en début de message, formant ainsi ce que l'on appelle l'***en-tête***. Une ou plusieurs ligne vide séparent cet en-tête du corps du message.

1.2.1 L'enveloppe

Puisque l'en-tête contient l'adresse du destinataire, l'utilité de l'enveloppe ne semble pas immédiate. L'en-tête contient l'adresse du destinataire d'un courrier, mais pour le router l'agent de transport ne peut s'y fier. Prenons un cas concret : si on se basait uniquement sur l'adresse contenue dans l'en-tête, un courrier adressé aux adresses **jdeuf@domaine1**, **jbon@domaine2**. Le courrier serait adressé au domaine **domaine1**, qui se chargerait de le délivrer à l'utilisateur **jdeuf**, et transmis à l'adresse **jbon@domaine2**. De son coté, le domaine **domaine2** recevrait aussi ce courrier, qui le délivrerait à l'utilisateur **jbon**, et transmettrait à l'adresse **jdeuf@domaine1**…Le système s'emballerait alors.

De même que le facteur n'ouvre pas le courrier pour connaître l'adresse du destinataire mais se contente de la regarder sur l'enveloppe, pour le courrier électronique aussi il faut une enveloppe. Cette enveloppe n'existe que lors du transport du courrier, si bien que l'utilisateur n'en n'a pas conscience. Elle n'est pas stockée dans le message, et c'est sur elle qu'est basé tout le processus du routage du courrier. Les adresses contenues dans l'en-tête ne servent qu'à titre d'information pour le destinataire, les adresses de l'enveloppe pouvant être modifiées ou réécrites lors du transport.

1.2.2 L'en-tête

L'en-tête est constituée de plusieurs lignes. Chaque ligne est constituée d'un nom de champ et de la valeur de ce champ, séparée par le caractère " :" suivi d'une espace. Un champ peut s'étendre sur plusieurs lignes, si la suivante commence par le caractère de tabulation. Les champs peuvent apparaître dans n'importe quel ordre.

Voici un exemple d'en-tête de courrier :

```
From mdecor Tue Jan  9 11:41:26 2001
Return-Path: <ptitlu@ptitlu.org>
Received: from soleil.uvsq.fr (root@soleil.uvsq.fr [193.51.24.1])
          by atlas.ens.uvsq.fr (8.11.1/jtpda-5.2) with ESMTP id f099b2h10153
          for <mdecor@ens.uvsq.fr>; Tue, 9 Jan 2001 10:37:02 +0100 (CET)
Received: from toolinux.com (www.toolinux.com [213.41.14.252])
          by soleil.uvsq.fr (8.11.1/jtpda-5.3.3) with SMTP id f099aqE91304
          for <mdecor@ens.uvsq.fr>; Tue, 9 Jan 2001 10:37:00 +0100 (CET)
Message-Id: <200101090937.f099aqE91304@soleil.uvsq.fr>
Received: (qmail 20498 invoked from network); 9 Jan 2001 10:40:12 -0000
Received: from unknown (HELO qbasic) (195.154.30.211)
  by www.toolinux.com with SMTP; 9 Jan 2001 10:40:12 -0000
Date: Tue, 9 Jan 2001 10:39:58 +0100
From: "P'tit Lu" <ptitlu@ptitlu.org>
To: Mallet laurent <malletl@python>
Cc: Mathieu DECORE <mdecor@ens.uvsq.fr>
Subject: Re: Fwd: Ma doc DNS
In-Reply-To: <01010910083900.00337@python>
References: <01010910083900.00337@python>
X-Mailer: Sylpheed version 0.4.9 (GTK+ 1.2.8; Linux 2.4.0; i686)
Mime-Version: 1.0
Content-Type: text/plain; charset=ISO-8859-1
Content-Transfer-Encoding: 8bit
X-UIDL: [m*!!T3G"!mG<!!f`Q"!
Status: RO
Content-Length: 872
Lines: 27
```

La première ligne commence par le mot-clé *From* suivi d'un espace et du chemin que le message a emprunté, la date et l'heure à laquelle il a été reçu par la dernière machine qui l'a traité, et une partie optionnelle spécifiant de quel hôte il est arrivé. Puisque ce champ est réécrit par chaque système voyant passer le message, il est parfois considéré comme faisant partie de l'enveloppe. Ce champ n'est plus utilisé, sauf pour des raisons de compatibilité.

Voici une liste des autres champs les plus courants figurant dans un en-tête et leur signification :

From : Adresse électronique de l'expéditeur, en général suivie de son nom.

To : Adresse du destinataire. Plusieurs adresses peuvent être spécifiées, séparées par une virgule.

Cc : Adresses des personnes recevant une copie du message pour information ("Carbon Copy", copie carbone, du temps où une copie sur papier carbone servait de copie d'un document). Plusieurs adresses peuvent être spécifiées, séparées par une virgule.

Bcc : Adresses des personnes recevant une copie du message sans que ces adresses n'apparaissent dans l'en-tête du message et non transmise aux destinataires spécifiés par les champs *To :* et *Bcc :* ("Blind Carbon Copy", copie carbone cachée). De cette façon, ni les destinataires des champs *To :* ou *Cc :*, ni les destinataires des champs *Bcc :* n'ont connaissance des adresses contenues dans le champ *Bcc :*. Plusieurs adresses peuvent être spécifiées, séparées par une virgule.

Subject : Contenu du courrier, en quelque mots.

Date : Date à laquelle le courrier a été expédié.

Reply-To : Adresse à laquelle l'expéditeur veut que les réponses éventuelles lui soient envoyées. Ceci est utile si on veut que les réponses arrivent à une adresse différente de l'adresse d'expédition.

Organization : Nom de l'organisme, de la société, de l'université propriétaire de la machine d'où provient le courrier.

Message-ID : Chaîne générée par l'agent de transport de courrier du système où le message a été généré. Cet identificateur est propre au message, unique dans le monde entier. C'est par exemple le nom de la machine (unique sur l'Internet) et une date ainsi qu'un identificateur de processus (unique sur cette machine).

Received : Chaque site relayant le courrier insère un champ indiquant son nom, un identificateur, la date et l'heure de réception, de quel site le message est arrivé, et quel logiciel de transport a été utilisé.

Return-Path Lors du dépot dans la boîte aux lettres, l'agent de transport y indique l'adresse de l'expéditeur dans l'enveloppe.

X-*nom-quelconque* : Informations supplémentaires concernant toute possibilité nouvelle qui n'est pas encore définie dans un document RFC, ou ne le sera jamais.

1.3 Format des adresses électroniques

Parce que les protocoles ont évolués au cours du temps, et que de nouveaux sont venus se greffer à l'Internet, différents formats d'adresse sont reconnus. Les RFC 822 et 1123 spécifient le format de ces adresses, pour lesquelles il n'y a pas de distinction majuscule / minuscule pour les caractères. Noter que la RFC 822 précise que les courriers adressés à **Postmaster** devraient être reçus et lus régulièrement, ainsi que l'adresse **abuse** pour les plaintes de spam. La RFC 2142 recense toute ces adresses qu'il est bon de savoir reconnaître.

1.3.1 Adresses Internet globales

Ces adresses sont valides en tout point de l'Internet, car elles spécifient le site sans donner de chemin pour l'atteindre.

Voici les différentes formes que peuvent prendre ces adresses (seules celles de type SMTP sont recensées) :

jdeuf@domaine C'est l'adresse la plus simple.

jdeuf @ domaine C'est la même adresse que précédement.

jdeuf (John Deuf) @ domaine Le commentaire entre parenthèses peut être inséré en n'importe quel point de l'adresse, et sera ignoré par le

MTA.

"John Deuf" @ domaine Le courrier sera distribué à l'utilisateur **John Deuf** dans le domaine **domaine**.

<jdeuf@domaine> Adresse facilement exploitable par un MTA, en raison des caractères "<" et ">".

John Deuf <jdeuf@domaine> Le courrier sera envoyé à l'adresse **jdeuf@domaine**, et tout ce qui est en dehors des caractères "<" et ">" sera considéré comme un commentaire par le MTA, donc ignoré.

"John Deuf" <jdeuf@domaine> Le commentaire est ici un mot unique.

1.3.2 Adresses Internet avec routage explicite

Une adresse avec routage explicite précise à quel site le courrier doit être envoyé et qui doit se charger de le transmettre à un autre site. Par exemple, l'adresse **@domaine1, @domaine2 :jdeuf@domaine3** précise que le courrier doit être envoyé au domaine **domaine1**, qui doit faire suivre au domaine **domaine2** qui se chargera de l'envoyer à **jdeuf@domaine3**.

Ces adresses sont déconseillées par la RFC 1123 car certains sites ne les reconnaissent pas correctement ou ne les acceptent pas à cause du spam.

1.3.3 Adresses Internet littérales

Les adresses de type **jdeuf@[192.168.42.1]** spécifient directement à quelle adresse envoyer le courrier, sans traitement préalable comme interrogation du DNS par exemple. Ce type d'adresses est déconseillé.

1.3.4 Extensions

Diverses extensions aux formats d'adresses existent :

- Routage avec % : la RFC 1123 offre cette alternative pour le routage explicite. Pour le même résultat dans l'exemple précédent, cela donne **jdeuf % domaine3 % domaine2 % domaine1**. Ce type d'adresse est également déconseillé, et en voie de disparition.

– Adresse vide : <>. Cette adresse peut être utilisée pour les messages d'erreur, car on ne peut y répondre. La tendance est plutôt de mettre comme expéditeur **Postmaster** ou **Mailer-Daemon**, bien que cela puisse provoquer des boucles de courrier.

– Extension des adresses locales : l'adresse **root+jdeuf@domaine** délivre le courrier à l'utilisateur **root**, mais l'adresse figurant dans l'en-tête reste **root+jdeuf@domaine**. Cela sert à trier les méls avec `procmail`. Par exemple, pour un réseau local ne disposant que d'un seul compte chez le fournisseur d'accès, on pourrait envoyer des courriers aux adresses **root+jdeuf@domaine** et **root+jbon@domaine**, qui arriveraient à **root**. Un filtrage adéquat permettrait alors de redistribuer le courrier à l'utilisateur destinataire du courrier (ici **jdeuf** ou **jbon**).

1.4 Le protocole SMTP

Le protocole SMTP (Simple Mail Transfer Protocol) est définit par la RFC 821. C'est grâce à ce protocol que les serveurs de mél peuvent communiquer entre eux, quelque soit leur origine.

Voici un exemple de dialogue avec un serveur de méls :

```
mdecore@tarsier:~$ telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tarsier.athome.chezmoi ESMTP Sendmail 8.9.3+3.2W/8.9.3/Mathieu DECORE -
14/10/2000; Wed, 27 Dec 2000 15:53:31 +0100
mdecore@tarsier:~$ telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tarsier.athome.chezmoi ESMTP Sendmail 8.9.3+3.2W/8.9.3/Mathieu DECORE -
14/10/2000; Wed, 27 Dec 2000 15:57:30 +0100
HELO localhost
250 tarsier.athome.chezmoi Hello mdecore@localhost [127.0.0.1], pleased to meet
you
MAIL FROM: <mdecore@localhost>
250 <mdecore@localhost>... Sender ok
```

```
RCPT TO: <piou@localhost>
250 <piou@localhost>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
From: zecastor@mondomaine.chezmoi
Subject: Petit test de dialogue SMTP
To: piou@sondomaine.chezlui
Date: Wed Dec 27 16:01:04 CET 2000

Voici le petit test...
.
250 PAA00794 Message accepted for delivery
QUIT
221 tarsier.athome.chezmoi closing connection
Connection closed by foreign host.
mdecore@tarsier:~$
```

et voici le mail reçu :

```
From: zecastor@mondomaine.chezmoi
Subject: Petit test de dialogue SMTP
To: piou@sondomaine.chezlui
Date: Wed Dec 27 16:01:04 CET 2000

Voici le petit test...
```

La première ligne annonce que la communication est établie avec l'hôte en question (**tarsier.athome.chezmoi**), qui donne le protocole utilisé (*ESMTP*), le serveur de méls utilisé (*Sendmail 8.9.3*) ainsi que la version du fichier de configuration. Le code 220 annonce que le serveur est disponible.

Ensuite, la ligne `HELO localhost` est tapée par l'utilisateur. Il s'agit d'une identification avec l'adresse de son site (c'est plus poli de dire d'où on vient). En retour, le serveur annonce qui il est (`250 tarsier.athome.chezmoi`), et nous souhaite la bienvenue (`Hello mdecore@localhost [127.0.0.1], pleased to meet you`).

Les données de l'enveloppe viennent ensuite : l'expéditeur, pour les renvois de messages d'erreur éventuels (`MAIL FROM : <mdecore@localhost>`), l'adresse du destinataire (`RCPT TO : <piou@localhost>`). A chaque fois, le serveur de méls valide la commande si elle est correcte, mais renvoie un message d'erreur si il y a une faute (par exemple pour un nom de domaine non valide, un message du style `451 <mdecore@domainebidon.truc>... Sender domain must resolve` serait envoyé).

La ligne suivante (*DATA...*) indique le début du message. Il y a d'abord l'en-tête, avec les champs habituels, et le corps du message, avec le texte. Noter que l'en-tête peut contenir des adresses mél non valides, puisque le routage s'effectue sur les adresse contenues dans l'enveloppe.

Enfin, une ligne ne contenant qu'un point termine la session. La commande `QUIT` permet de clore la session.

1.5 Le mél et le DNS

Pour router les courriers, un serveur de méls se base sur les champs de type *MX* du serveur de noms du domaine. Tout cela est définit par la RFC 974.

Par exemple, un message adressé à **machine.domaine** peut être routé vers **mailhost.domaine**. Ainsi, la machine **mailhost.domaine** reçoit tous les méls du domaine **domaine**, qu'il redistribue ensuite au hôtes concernés. Le seveur de noms doit alors contenir une ligne comme :

```
machine.domaine    IN    MX    10    mailhost.domaine
```

Si on veut que tous les méls adressés à **login@domaine** soient acceptés, il faut mettre une ligne du type :

```
domaine    IN    MX    10    mailhost.domaine
```

Les champs de type *MX* peuvent aussi servir pour indiquer des serveur de secours, au cas où la connexion serait temporairement coupée. Ainsi, pour envoyer un courrier à **jdeuf@domaine1**, si **domaine1** ne répond plus, on peut demander d'envoyer le courrier à **domaine2** (avec son accord) :

```
domaine1    IN    MX    10    mailhost.domaine
domaine2    IN    MX    20    mailhost.domaine
```

Les nombres **10** et **20** indiquent les priorités des sites à essayer : on tente d'abord d'envoyer un courrier à **domaine1**, puis à **domaine2** si **domaine1** ne répond pas.

Enfin, il peut être utile de mettre une entrée *MX* pour chaque machine. En effet, une demande d'envoi de message pour **mailhost.domaine** cherche d'abord un champ de type *MX*, puis un champ de type *A* en cas d'échec pour avoir l'adresse IP de **mailhost.domaine**. Si le champ de type *MX* existe, il n'y aura plus qu'une seule requête DNS au lieu de deux :

```
mailhost.domaine    IN    MX    0    mailhost.domaine
```

1.6 Extensions à SMTP

Le protocole SMTP est limité et figé : les messages sont codés sur 7 bits, les noms d'utilisateurs et de domaines sont de moins de 64 caractères, le nombre de destinataires est limité à 1000 et les messages ne doivent pas dépasser 1000 lignes.

Pour remédier à cela, il y a deux évolutions : ESMTP (Extended SMTP) qui permet de pouvoir faire évoluer le protocole SMTP, et MIME (Multipurpose Internet Mail Extensions) qui permet d'échanger des messages contenant des accents ou des données binaires.

1.6.1 ESMTP

ESMPT permet d'étendre SMTP tout en restant compatible avec SMTP. Pour entamer un dialogue ESMTP, il suffit de changer *HELO* par *EHLO*. Si le serveur supporte ESMTP, il annonce la liste des options qu'il supporte, sinon il répond *HELO* et la communication se poursuit avec SMTP :

```
mdecore@tarsier:~$ telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tarsier.athome.chezmoi ESMTP Sendmail 8.9.3+3.2W/8.9.3/Mathieu DECORE -
14/10/2000; Wed, 27 Dec 2000 17:42:24 +0100
EHLO world
250-tarsier.athome.chezmoi Hello mdecore@localhost [127.0.0.1], pleased to
meet you
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
QUIT
221 tarsier.athome.chezmoi closing connection
Connection closed by foreign host.
mdecore@tarsier:~$
```

On peut par exemple délivrer un accusé de réception à un message :

```
mdecore@tarsier:~$ telnet localhost smtp
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 tarsier.athome.chezmoi ESMTP Sendmail 8.9.3+3.2W/8.9.3/Mathieu DECORE -
14/10/2000; Wed, 27 Dec 2000 17:42:24 +0100
EHLO world
250-tarsier.athome.chezmoi Hello mdecore@localhost [127.0.0.1], pleased to
meet you
250-EXPN
250-VERB
250-8BITMIME
250-SIZE
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
MAIL FROM: <mdecore@localhost> RET=HDRS ENVID=mesg
250 <mdecore@localhost>... Sender ok
RCPT TO: <piou@localhost> NOTIFY=SUCCESS ORCPT=rfc822;piou@localhost
250 <piou@localhost>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Le texte...
.
250 RAA00917 Message accepted for delivery
QUIT
221 tarsier.athome.chezmoi closing connection
Connection closed by foreign host.
mdecore@tarsier:~$
```

Ici, on veut que l'accusé retourne l'en-tête (`RET=HDRS`) avec comme mot clef *mesg* (`ENVID=mesg`). Le message doit être envoyé en cas de succès (`NOTIFY=SUCCESS`), et l'adresse du destinataire au retour doit être la même que celle spécifiée pour outrepasser les réécritures (`ORCPT=rfc822 ;piou@localhost`).

Et le mél reçu pour notification par l'expéditeur contient :

```
Date: Wed, 27 Dec 2000 17:59:41 +0100
```

```
From: Mail Delivery Subsystem <MAILER-DAEMON@tarsier.athome.chezmoi>
To: <mdecore@tarsier.athome.chezmoi>
Subject: Return receipt
Auto-Submitted: auto-generated (return-receipt)

[-- Attachment #1 --]
[-- Type: text/plain, Encoding: 7bit, Size: 0.3K --]

The original message was received at Wed, 27 Dec 2000 17:57:48 +0100
from mdecore@localhost [127.0.0.1]

   ----- The following addresses had successful delivery notifications -----
<piou@localhost>  (successfully delivered to mailbox)

   ----- Transcript of session follows -----
<piou@localhost>... Successfully delivered

[-- Attachment #2 --]
[-- Type: message/delivery-status, Encoding: 7bit, Size: 0.3K --]

Original-Envelope-Id: mesg
Reporting-MTA: dns; tarsier.athome.chezmoi
Received-From-MTA: DNS; localhost
Arrival-Date: Wed, 27 Dec 2000 17:57:48 +0100

Original-Recipient: rfc822;piou@localhost
Final-Recipient: RFC822; <piou@localhost>
Action: delivered (to mailbox)
Status: 2.1.5
Last-Attempt-Date: Wed, 27 Dec 2000 17:59:41 +0100

[-- Attachment #3 --]
[-- Type: text/rfc822-headers, Encoding: 7bit, Size: 0.4K --]

Return-Path: <mdecore@tarsier.athome.chezmoi>
Received: from localhost (mdecore@localhost [127.0.0.1])
        by tarsier.athome.chezmoi (8.9.3+3.2W/8.9.3/Mathieu DECORE - 14/10/2000)
+with ESMTP id RAA00917
        for <piou@localhost>; Wed, 27 Dec 2000 17:57:48 +0100
Date: Wed, 27 Dec 2000 17:57:48 +0100
From: Mathieu DECORE <mdecore@tarsier.athome.chezmoi>
```

```
Message-Id: <200012271657.RAA00917@tarsier.athome.chezmoi>
To: undisclosed-recipients:;
```

Ceci n'est qu'un exemple des extension proposées par ESMTP, se reporter aux RFC 1651 et 1869 pour les commandes optionnelles, 1652 pour le transport des messages contenant des caractères 8 bits, 1830 pour le transport de binaires ou de gros messages, 1985 pour l'émission des messages en attente, 1891 à 1894 pour l'accusé de réception, 1870 pour la déclaration de la taille des messages, 2197 pour les commandes à la file, 2222 pour l'authentification des clients...

Tous les serveurs de méls ne supportent pas forcément ESMTP, et de nouvelles options apparaissent régulièrement. Il serait vain (et ennuyeux) de vouloir tout décrire ici.

1.6.2 MIME

Le format MIME permet de spécifier le format du corps du message, dont la RFC 822 se contente de le décrire comme un ensemble de lignes de caractères sur 7 bits. En fait, MIME permet de spécifier le type du message (texte, image, son, vidéo ou plusieurs types à la fois), et le codage du message (7 bits, 8 bits, quoted-printable ou base64). MIME est décrit par les RFC 1341 à 1345, puis 2045 à 2049.

Voici un exemple de courrier au format MIME :

```
From: jbon@domaine1.com
To: jdeuf@domaine2.com
Subject: MIME
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="mime-c-bien"

Ceci est un message mime

--mime-c-bien

Content-Type: text/plain; charset=iso-8859-1
Content-Transfer-Encoding: 8bit

Salut,

        je t'envoie le document sur le format MIME que tu m'as demandé.
```

```
        Cordialement.

                Jean BON

--mime-c-bien

Content-Type: message/external-body; access-type=anon-ftp;
                 site="ftp.domaine1.com";
                 name="/pub/Linux/Mail/MIME/mime.txt"

--mime-c-bien
```

Les nouveaux champs apportés par MIME sont :

MIME-Version Identifie le message au format MIME (1.0 est l'identification du niveau standard actuel).

Content-Type Spécifie les types, sous-types et paramètres optionnels de chaque partie du message.

Content-Transfer-Encoding Spécifie le codage de tout ou partie du message.

Content-ID attribue un identificateur unique à chaque partie du message (optionel).

Content-Description Permet de spécifier un commentaire attaché à chaque partie (optionnel).

Ce message est composé de deux parties (`multipart/mixed`) séparées par une chaîne unique (`boundary="mime-c-bien"`). La première partie est un texte normal (`text/plain`), contenant des accents, donc utilisant un jeu de caractère différent de l'ASCII classique (`charset=iso-8859-1`), tandis que la deuxième partie est une référence à un objet externe (`message/external-body`) accédé par ftp anonyme (`access-type=anon-ftp`).

Voici un autre exemple de message au format MIME :

```
From: Jean BON <Jean.Bon@domaine1.com>
```

```
To: John.Deuf@domaine2.com
Mime-Version: 1.0
Content-Type: text/enriched; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
X-UIDL: '8e!!!(`"!?4W"!5J4"!
Status: RO
Content-Length: 662
Lines: 45

<bold><color><param>0000,0000,ffff</param><bigger>
les infos de la semaine
</bigger></color></bold>


Voici une offre de job arrivée le vendredi 12 janvier :


<bold>Le Café du Palais</bold>

contacter M. Dugenoux.

Ils <bold><color><param>ffff,0000,0000</param>recherchent des
serveurs</color></bold> pour leurs différents services (midi et soir et WE).
```

La balise **bold** affichera le texte en gras. Les autres balises pour le texte enrichi MIME sont spécifiées dans la RFC 1523.

Références

[1] **Sendmail v8 Kit Jussieu**, Pierre David, Juillet 1999.
`http://www.prism.uvsq.fr/~pda/kit-jussieu/support/index.html`

[2] **Guide de l'administration réseau sous Linux**, Olaf KIRCH et Terry DAWSON, seconde édition, juin 2000.

Version française (version 1, plus à jour !) :
`ftp://ftp.lip6.fr/pub/linux/french/books/nag.french.eoit-1.0.tar.gz`.

Version originale (en anglais) :
`http://www.linuxdoc.org/LDP/nag2/index.html`.