

## SERVEUR DE MESSAGERIE

Le service messagerie SENDMAIL s'appuie sur le service DNS. En particulier la configuration des serveurs est précisée dans les champs MX des fichiers du DNS. Nous nous intéressons donc ici uniquement à des courriers comportant des noms qualifiés. Le service de messagerie sous UNIX est un routeur multiprotocole. Quelle que soit l'adresse, éventuellement non INTERNET, le courrier parvient à son destinataire si cette adresse est correcte.

### 1 - FONCTIONNEMENT GENERAL

Chaque instance de messagerie est appelée MTA (Mail Transmission Agent). Le protocole utilisé entre les MTA est SMTP (Simple Mail Transfer Protocol). Les MTA s'exécutent sur des hôtes afin de permettre un transfert sur d'un point à l'autre du réseau.

En particulier certains MTA ne font que le relais de mail c'est à dire le transfert d'un point à un autre avec stockage temporaire.

Un MTA détermine pour chaque message un triplet (protocole, prochain MTA à contacter, adresse-qualifiée du destinataire). La figure 1 illustre les principaux agents MTA pour un utilisateur se situant sur Aquarel.

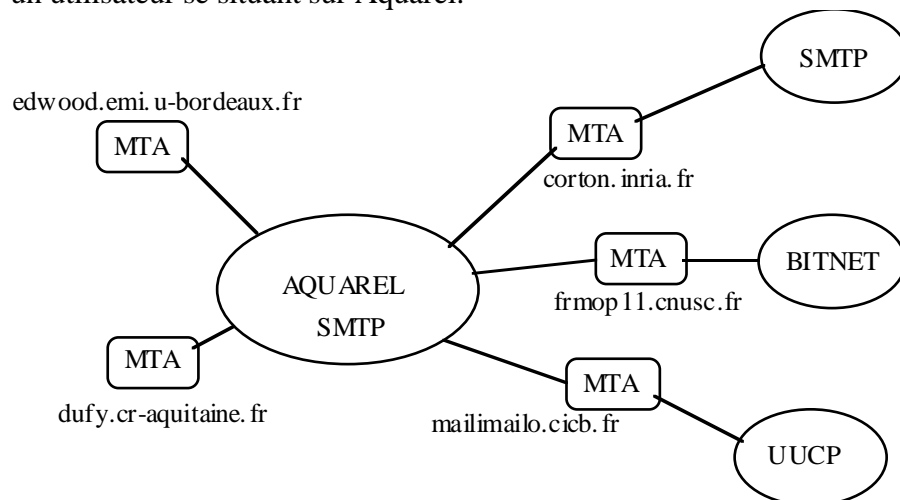


Figure 1. Configuration de MTA.

Le sendmail exploite plusieurs fichiers de manière à acheminer le courrier :

- le fichier sendmail.cf contient un ensemble de règles permettant de réécrire toutes les adresses sous une forme qualifiée,
- le fichier aliases permet de définir proprement les utilisateurs et les listes d'utilisateurs
- le fichier generics permet de définir des adresses équivalentes lors de l'envoi du mail.

C'est un démon attaché au port 25. Il s'appuie sur le service TCP. La commande de lancement peut-être par exemple :

```
/usr/lib/sendmail -bd -q1h
```

Lors de son lancement les divers fichiers seront traduits en base de donnée.

Pour un site donné, il y a un MTA relai (sur emi c'est aliboron) et des MTA locaux sur chaque machine (les stations autres qu'aliboron du domaine emi).

Toute requête d'un utilisateur est envoyée dans la file d'attente du MTA local. Le MTA local envoie le courrier au MTA relai qui résoudra le problème d'acheminement sur le réseau non local. Le MTA récepteur dépose les courriers reçus dans la boîte à lettre des utilisateurs

exemple sur emi

/var/spool/mail/nom\_de\_login

Ces messages seront exploités par les requêtes de l'utilisateur lors de sa connexion.

## 2 - PROTOCOLE SMTP

Le protocole SMTP s'effectue en caractère alphabétique comme la plupart des protocoles de la couche 5. Après connexion au MTA destinataire, la liste des messages émis par le client est:

**helo machine**

**mail from : <adresse1>** #Personne qui a créée le message

**rcpt to : <adresse2 >** # Adresse destinataire primaire

data

From : <adresse1>

To : <adresse2>

Cc : <adresse3> # Adresse destinataire secondaire

Bcc : <adress4> # Adresse destinataire cachés des précédents

Received : <adresse5 > # Adresse émetteur

Received : # traces des MTA successifs

Return-Path : # utilisable pour trouver un chemin de retour

Subject : .....

.....

.

**quit**

La première partie en caractère gras est appelée enveloppe et est non modifiable. C'est elle qui authentifie le courrier. La seconde partie est le message délivré et est non vérifiée. Le serveur répond après chaque requête par un acquittement. Une telle séquence peut être testée par la commande "telnet machine smtp".

Ne sont listées ici que les commandes les plus banales. L'apparition d'images et de sons ont nécessité leur extension. L'envoi de documents encodés est maintenant soumis à l'utilisation de commandes afin que l'utilisateur puisse par simple lecture de son courrier décoder l'information sans intervention. Par exemple, on peut attacher un document mpeg (images

animées), lors de la lecture du courrier l'utilitaire associé sera déclenché afin que l'utilisateur puisse effectivement voir le film. Cette norme est appelée norme MIME (Voir paragraphe 6).

Le MTA relais s'appuie sur le DNS pour résoudre les problèmes d'adressage. L'algorithme est grossièrement le suivant :

- 1 - interroger son DNS pour connaître le numéro du serveur du domaine de destination,
- 2 - interroger le serveur DNS distant pour connaître le numéro du serveur correspondant à la machine et/ou au domaine destinataire
  - \* s'il y a un champ MX correspondant, le numéro du champ MX est le destinataire,
  - \* dans le cas où un champ MX n'existe pas, numéro associé à un champ CNAME pour le domaine évoqué (cas du mailhost par exemple) s'il existe,
  - \* sinon numéro du serveur DNS.
- 3 - Le numéro ainsi récupéré est utilisé pour exécuter le protocole.

### **3 - FICHIER ALIASES ET GENERICS**

Le fichier alias permet de définir des

- Synonymes, par exemple remplacer l'adresse root par l'adresse de la personne qui est root,
- des listes de diffusion.

Il est utilisé par le sendmail quand il reçoit un courrier.

La syntaxe est la suivante :

*nom\_alias* : {*receveur* , *liste\_de\_receveurs* , :include:*fichier*}

et dans le cas d'une liste le gestionnaire de la liste doit être précisé par la commande

*owner-liste* : *adresse\_gestionnaire* .

Exemple. Postmaster: dupont

utilisateurs:include:/usr/local/lib/mail/includes/utilisateurs

owner-utilisateurs: durand

comite-technique: destre, sinestre, utilisateurs

owner-comite-technique: dupont

Ce fichier est utilisé à la réception du courrier par le MTA relais pour le forwarder vers les utilisateurs concernés.

Remarque . Il existe un autre mécanisme de forward qui se fait après l'exploitation du fichier alias c'est l'exploitation du fichier .forward chez l'utilisateur. Ce fichier permet par exemple lors de l'arrivée d'un message dans la boîte à lettre de l'utilisateur d'avoir une réponse automatique pour l'expéditeur.

Le fichier generics permet de qualifier une adresse lors de l'envoi d'un courrier. On s'assure ainsi que les courriers sortant d'un serveur donné ont des adresses lisibles de l'extérieur.

Exemple. Il peut permettre de remplacer un login par un nom

maylis@claudia.labri.u-bordeaux.fr

devient après traitement

Maylis.Delest@labri.u-bordeaux.fr

#### 4 - QUALIFICATION DE L'ADRESSE

Tout courrier quittant un MTA doit avoir une adresse expéditeur et une adresse receveur qualifiée afin de ne pas émettre de requête non valide auprès du DNS. A cette fin un mécanisme de réécriture est implémenté dont le travail est de s'assurer que l'adresse est correcte en la réécrivant si nécessaire.

En toute généralité, une règle de réécriture est une règle qui va permettre de réécrire un mot écrit sur un alphabet X en un mot sur un alphabet Y. Elle est définie par une paire (d,r) où d est la description du motif recherché et r est la manière de le réécrire.

Exemple.  $X=Y=\{a, b\}$ ,  $r_1=(w_1aa^*w_2, w_1aw_2)$ .

Une règle peut être appliquée une fois sur la "première forme reconnue" ou appliquée récursivement suivant sa caractérisation ou appliquée parallèlement sur toutes les décompositions qui l'acceptent.

Exemple . Soit  $w = abaaabbaabbba$ , appliquer une fois dans un parcours gauche droite

$r_1(w) = ababbaabbba$

appliquer récursivement

$r_1(w) = ababbabbba$

appliquer parallèlement

$r_1(w) = \{ababbaabbba, abaaabbabbba\}$

Un système de règles est une suite de règles qui sont exploitées les unes après les autres sauf ordre de terminaison du système.

Le MTA exploite un ensemble de système de réécriture qui s'enchaînent suivant les noms à qualifier : expéditeur, destinataire ou nom du MTA à contacter. Ces règles sont regroupées dans un fichier /etc/sendmail.cf. Lors de la mise en route du sendmail il y a relecture du fichier pour créer une base de donnée qui sera exploitées lors des requêtes.

La syntaxe du sendmail.cf est aride. Il est conseillé de récupérer un fichier sendmail.cf existant et éventuellement le kit de génération qui l'accompagne (sur [ftp.aquarel.fr](http://ftp.aquarel.fr) par exemple) plutôt que d'essayer d'en concevoir un.

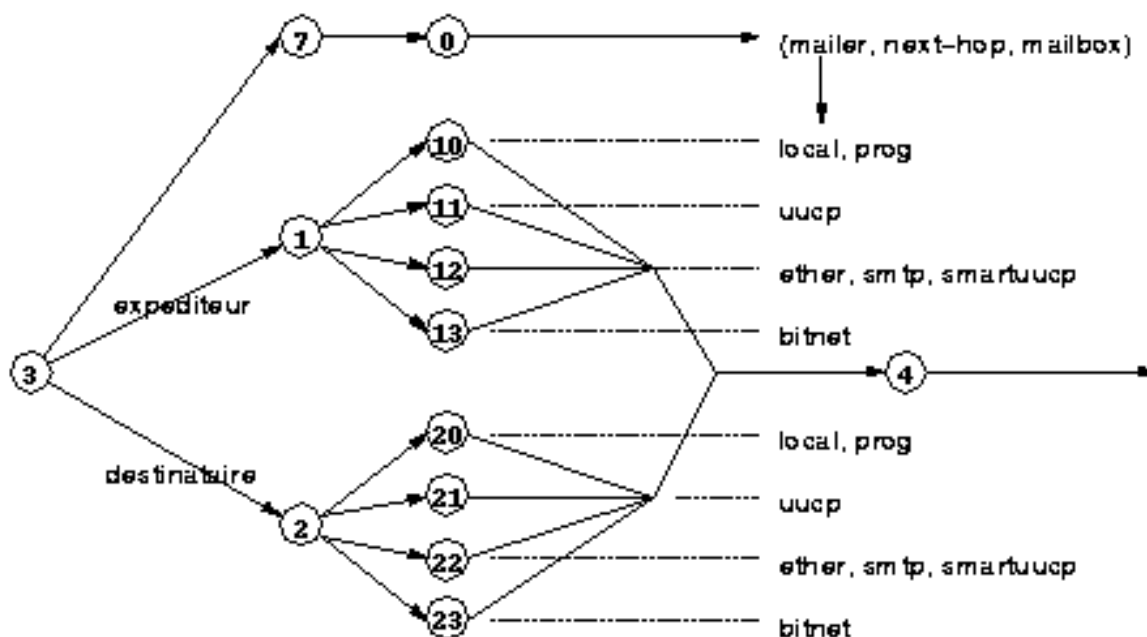


Figure 2. Les systèmes de réécriture (extrait Kit-Sendmail, R. Dirlwanger).

## 5 - LE FICHIER SENDMAIL.CF

Le fichier comporte une zone d'initialisation de variables locales appelées macros puis 9 systèmes de réécriture. L'enchaînement de ces systèmes est montré figure 2.

Chaque enregistrement du sendmail commence par une lettre qui précise le type de définition R pour règle, Si pour système numéro i, autres pour les macros. La plupart des zones de macros restent inchangées d'un site à l'autre. Les kits de génération permettent d'obtenir un sendmail.cf personnalisé sur le site en utilisant une syntaxe simple et sans rentrer dans la description complète des macros.

Une macro est définie par

*lettre {variable}valeur .*

Par exemple, la lettre T (Trusted) désigne les utilisateurs ayant le droit de modifier une enveloppe, on a ainsi : Troot; la lettre D désigne une macro de définition de chaîne de caractère, DDaquarel.fr indique que le domaine (seconde lettre D) est aquarel.fr.

Symbole	Signification
\$*	≥0 jetons
\$+	≥1 jetons
\$-	=1 jeton
\$=y	n'importe quel jeton défini dans la classe y
\$~y	pas un jeton de la classe y
\$x	tous les jetons de la classe x

Figure 3. Les principaux motifs de reconnaissance.

La syntaxe des règles est guidée par une reconnaissance de motif dans les chaînes de caractères. Elle a la forme :

*Rmotif            transformation            commentaire*

Une chaîne de caractères entre deux séparateurs s'appelle un jeton. La figure 3 montre les reconnaissances de jeton possibles.

Exemple .

L'adresse Maylis.Delest@labri.u-bordeaux.fr

peut-être reconnue par une quelconque des expressions suivantes :

\$+@\$+, \$-.\$\*@\$+, \$+.\$-

Les expressions qui sont associés aux jetons sont appelés jetons indéfinis et sont numérotés au fur et à mesure de leur reconnaissance dans l'ordre gauche droite.

Exemple . \$-.\$\*@\$+ conduira à la définition des jetons suivants :

\$1=Maylis, \$2=Delest, \$3=labri.u-bordeaux.fr.

De la même manière, la partie transformation utilise des metasymboles pour désigner les champs. Ils sont décrits figure 4.

Exemple L'adresse maylis@dufy sera requalifiée par la conjonction des deux enregistrements suivants :

DDaquarel.fr

R\$+@\$-            \$:\$1@\$2.\$D            adresses locales

On obtiendra maylis@aquarel.fr. La réécriture s'arrêtera.

Symbole	Signification
\$n	inscrire le jeton indéfini n
\$(name\$)	inscrire un nom prédéfini par name
\$>n	appeler la règle de réécriture n
\$@	sortir du système de réécriture
\$:	arrêter après exécution de la règle

Figure 4. Les principaux motifs de transformation.

## 6 - NORME MIME

Le sigle MIME veut dire Multipurpose Internet Mail Extensions. Sa description complète est celle du RFC1521. Son but est de permettre des échanges de données autres que celle de texte non accentué : alphabets particuliers, son, image.

Il s'agit essentiellement d'une structuration du corps du message et non de l'enveloppe afin que les utilitaires de lecture de courrier puissent interpréter correctement les traitements à activer.

Tout message ne comportant pas dans son en-tête un champ MIME-version est considéré comme du texte anglais.

Le champ Content-Type permet de définir chaque entité différente dans le message. Sa valeur est <type>/<sous-type>. On trouvera figure 5, les divers type actuellement définis.

Exemple. Dans l'en-tête du message,

Content-Type=multipart/mixed permet de préciser que le message est constitué de plusieurs item (multipart) et de natures différentes (mixed). Il est suivi de l'option boundary qui précise le délimiteur entre les item.

Dans les cas du sous-type external, des champs précisent l'adresse et le port ou se connecter ainsi que l'adresse sur ce site.

Le champ Content-Description indique au destinataire le sujet de l'item. Le champ Content-Transfer-Encoding précise l'encodage utilisé. Il y a actuellement cinq encodages possibles et un échappement. Les encodages les plus utilisés sont base64 et QP (quoted-printable).

Type	Sous-Type	Description
Text	Plain	non formaté
	Richtext	formatage simple
Image	Gif	format GIF
	Jpeg	format JPEG
Audio	Basic	son audible
Video	Mpeg	film format MPEG
Application	Octet-stream	Suite d'octet non interprété
	Postscript	document imprimable
Message	Rfc822	Message MIME
	Partial	message coupé pour la transmission
	External-body	message à prendre sur le réseau
Multipart	Mixed	Parties indépendantes
	Alternative	Même message mais formats différents
	Parrallel	Parties à voir simultanément
	Digest	chaque morceau est en RFC822

## 7 - ACCES DISTANT AU COURRIER

Dans un grand nombre d'endroits, les utilisateurs utilisent le courrier électronique sans être reliés en permanence à un réseau. C'est le cas notamment lorsqu'on accède à Internet par RTC.

La RFC 1225 définit un protocole simple (Post Office Protocol) qui permet à un utilisateur de se connecter uniquement pour lire, supprimer et envoyer son courrier sur une boîte à lettre distante. Le protocole plus récent POP3 permet d'ajouter la fonctionnalité de copie du courrier sur l'ordinateur local.

La RFC 1064 définit le protocole IMAP (Interactive Mail Access Protocol) qui permet de gérer un courrier lorsqu'un utilisateur souhaite y accéder depuis plusieurs machines, il gère le référentiel de l'utilisateur et ne copie pas le courrier en local.

La RFC 1056 décrit le protocole DMSP (Distributed Mail System Protocol) qui correspond au PCMAIL.

L'ensemble de ces protocoles permettent :

- le filtrage des courriers entrants,
- les réponses automatiques notamment en cas d'absence du destinataire,
- le renvoi vers une autre adresse y compris la redirection du champ subject vers une radiomessagerie.

### EXEMPLE DE COURRIER

**Mail-from:**

**From** Maylis.Delest@labri.u-bordeaux.fr Sat Oct 11 09:10 MET 1997  
**Received:** from [147.210.8.135] (gruissan [147.210.8.135]) by  
batman.labri.u-bordeaux.fr (8.7.5/8.7.3) with SMTP id JAA06562 for  
<maylis>; Sat, 11 Oct 1997 09:10:31 +0200 (MET DST)  
**X-Sender:** maylis@batman.labri.u-bordeaux.fr (Unverified)  
**Message-Id:** <v01520d00ae8261c9d108@[147.210.8.135]>

**Mime-Version:** 1.0

**X-Mailer:** Eudora F1.5.2

**Date:** Thu, 10 Oct 1996 09:19:24 +0100

**To:** maylis

**From:** Maylis.Delest@labri.u-bordeaux.fr (Maylis Delest)

**Subject:** essai

**Content-Type:** multipart/mixed; **boundary**="=====-  
1367186932==\_====="

**Content-Length:** 181183

\*\*\* EOOH \*\*\*

X-Sender: maylis@batman.labri.u-bordeaux.fr (Unverified)

Mime-Version: 1.0

X-Mailer: Eudora F1.5.2

Date: Thu, 10 Oct 1996 09:19:24 +0100

To: maylis

From: Maylis.Delest@labri.u-bordeaux.fr (Maylis Delest)

Subject: essai

Content-Type: multipart/mixed; **boundary**="=====-  
1367186932==\_====="

Content-Length: 181183

--=====-1367186932==\_=====  
Content-Type: text/plain; charset="us-ascii"

et voila

--=====-1367186932==\_=====  
Content-Type: application/mac-binhex40; name="navarre.doc"  
Content-Disposition: attachment; filename="navarre.doc"

(This file must be converted with BinHex 4.0)

:"fjKGQ&bFQ8!9cC#6Ne69d3"!!!!+!!!!!!'\$A3ca(JSE%Di3!!!!!!!!!!!!!!

.....



rrrrra\$4!!!:

--===== \_-1367186932==\_=====

Content-Type: image/jpeg; name="plage.jpg"

; x-mac-type="4A504547"

; x-mac-creator="4A565752"

Content-Disposition: attachment; filename="plage.jpg"

Content-Transfer-Encoding: base64

/9j/4AAQSkZJRgABAQAgEwAAAEgAAArgCuAAD/7QEeUGhvdG9zaG9wIDMuMAA4QklNA+k  
AAAAAAHgA

DSAAAAADKAIw//b/7wM+AkEsAgV4A94BAAAAAWgBaAAAAAPyArwAAEBAABk

--===== \_-1367186932==\_=====