LP R&T RSF IR TD3
SMTP et POSTFIX

Rappels et compléments sur SMTP :

Le **S**imple **M**ail **T**ransfert **P**rotocol est un protocole d'échange de messages électroniques indépendant du protocole de transport sou-jacent. Pour les réseaux IP, SMTP est implanté au dessus de TCP (port 25), il est directement accessible via telnet (ex: telnet <nom serveur> 25).

Un serveur SMTP est une machine « cible » qui se présente comme un « bureau de poste » vis à vis de clients SMTP.

Les commandes de base de SMTP :
HELO <domaine> : Initialisation de la session SMTP
MAIL FROM : <route-inverse> : déclaration de l'émetteur du mail
RCPT TO : <route-directe> : déclaration du destinataire du mail
DATA : initialisation de la séquence de saisie des données
RSET : initialisation de la séquence de saisie des données
SEND FROM : <route-inverse> : message direct plutôt que postage.
SOML FROM : <route-inverse> : message direct OU postage.
SAML FROM : <route-inverse> : message direct ET postage.
VRFY <chaîne> : vérification de l'existence d'un destinataire
EXPN <chaîne> : extraction des destinataires inscrits dans une liste de diffusion
HELP [<chaîne>] : demande d'aide (éventuellement sur une commande)
NOOP : aucune opération
QUIT : clôture de la session SMTP
TURN : demande d'inversion des rôles d'émetteur et de récepteur

Exercice 1 :

1.1 A quoi correspond le dialogue suivant (commentez) entre le client et le serveur ?

```
>telnet monserveursmtp 25
                        220 lifl.lifl.fr ESMTP Sendmail
                           8.9.3/jtpda-5.3.3 ready at Mon
                           22 Apr 2002 11:36:22 +0200<EOL>
HELO pater.lifl.fr<EOL>
                        250 lifl.lifl.fr Hello pater
                           [134.206.10.153], pleased to
                           meet you<EOL>
VRFY gigrimaud<EOL>
                        550 gigrimaud... User unknown<EOL>
VRFY grimaud<EOL>
                        250 Gilles.Grimaud
                           <grimaud@lifl.lifl.fr><EOL>
QUIT<EOL>
                        221 lifl.lifl.fr closing
                           connection<EOL>
```

CLIENT ***SERVEUR***

=> Il s'agit de la validation d'une adresse mail

1.2 A quoi correspond le dialogue suivant (commentez) entre le client et le serveur ?

```
                     220 lifl.lifl.fr ESMTP Sendmail
                        8.9.3/jtpda-5.3.3 ready at Mon
                        22 Apr 2002 11:36:22 +0200 <EOL>
```

HELO pater.lifl.fr<EOL>
250 lifl.lifl.fr Hello pater
[134.206.10.153], pleased to
meet you <EOL>
MAIL FROM:grimaud@lifl.fr<EOL>
250 grimaud@lifl.fr... Sender ok <EOL>
RCPT TO: grimaud@lifl.fr<EOL>
250 grimaud@lifl.fr... Recipient ok <EOL>
RCPT TO:jean@lifl.fr<EOL>
250 jean@lifl.fr... Recipient ok <EOL>
DATA<EOL>
354 Enter mail, end with "." on a
line by itself <EOL>
Bla bla bla bla<EOL>
Bla bla bla bla<EOL>
.<EOL>
250 LAA21250 Message accepted for
delivery <EOL>
QUIT<EOL>
221 lifl.lifl.fr closing connection <EOL>

CLIENT ***SERVEUR***

=> Il s'agit de l'envoi d'un message.

Exercice 2 :

On suppose pour la suite que le programme client de mail utilise pour ses émissions un serveur SMTP installeé sur la machine dont l'adresse est 134.206.10.18. L'envoi de mail qui nous intéresse est simplement un corps de message << HELLO WORLD >> émis par grimaud@lifl.fr et reçu par jean@lifl.fr. De plus, l'émetteur utilise la machine 134.206.10.193.

2.1 Donner la liste minimale des commandes/réponses qui transitent sur le réseau pour la transmission de ce mail.

< 220 134.206.10.18 ESMTP ready
> HELO 134.206.10.193
< 250 Hello 134.206.10.193, pleased to meet you
> MAIL FROM: grimaud@lifl.fr
< 250 jean@lifl.fr ... Sender ok
> RCPT TO: jean@lifl.fr
> 250 grimaud@lifl.fr ... Recipient ok
> DATA
< 354 Enter mail, end with "." on a line by itself.
> HELLO WORLD
> .
< 250 LAA21250 Message accepted for delivery
> QUIT
< 211 134.206.10.18 closing connection

2.2 Couche de transport en mode connecté :

Tous les échanges SMTP sont effectués au dessus de TCP/IP. On suppose pour la suite que chaque ligne de texte correspond à un paquet émis par les machines (client et serveur). De plus chaque ligne se termine par un caractère spécial EOL. Pour les questions suivantes nous supposons qu'il n'y a pas de problème réseau entre le client et le serveur. De plus, les acquittements du serveur peuvent être transmis en même temps que les lignes de réponses (encapsulés dans l'entête TCP des paquets << réponses SMTP >>).

2.2.1 Combien de paquets seront échangés entre le client et le serveur pour la

transmission de ce mail?

=> Ouverture TCP
1 paquet pour la demande d'ouverture TCP du client vers le serveur
+ 1 pour acceptation du serveur
+ 1 pour SYN-ACK du client

=> Transmission data en SMTP
+ 7 pour les 7 lignes de texte du serveur vers le client
+ 7 pour les 7 commande du client vers le serveur

=> Fermeture TCP
+ 1 pour la confirmation de rupture de connexion envoyée par le client.

Notez que toutes les informations de fenêtres sont encapsulées dans des paquets contenant des données utiles, la demande de fin de connexion envoyée par le serveur vers le client aussi. Seule la confirmation de fin de transmission par le client et les trois paquets d'initialisation de la connexion ne sont pas fusionnée à des données utiles.

2.2.2 En considérant que chaque ligne de commande ou de réponse se termine par un caractère EOL et que chaque paquet TCP n'est émis qu'une fois, calculez le nombre de caractères échangés entre les deux machines.

- Combien d'octets la couche TCP ajoute-t-elle à chaque échange (en supposant que le champs option n'est pas utilisé) ?
  Nombre d'octets échangés : 367 (caractères)
  Nombre d'octets TCP par paquet : 20
- Combien d'octets ajoute-t-elle au total ?
  Nombre d'octets TCP en tout : 20 x 18 = 360
  Conclusion : il faut 367+360 octets pour transférer 367 caractères !

Compléments sur POSTFIX :

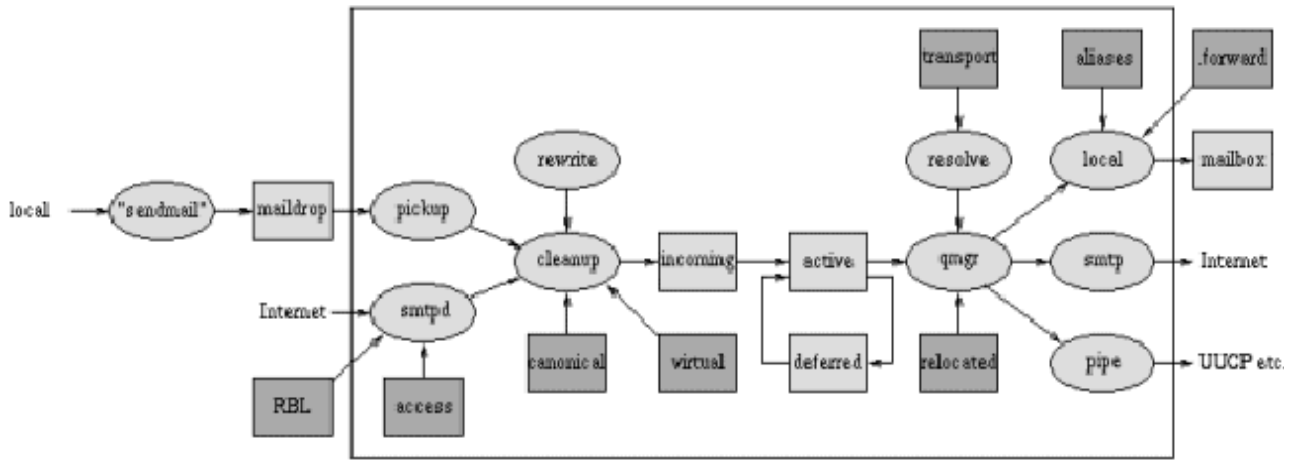
De nombreux MTA existent dans le monde Unix. Un des plus connus est sans doute sendmail. Il est installé par défaut dans nombre de distributions. Mais, on lui reproche généralement d'être complexe à configurer et d'être trop "monolithique". C'est pourquoi, nous avons choisi postfix. Cette partie s'inspire très fortement d'un document de Cyril Jovet (http://cjovet.free.fr/cours/postfix.htm).

1. Généralités sur postfix

Nous travaillerons ici avec postfix. Ce logiciel est reconnu pour les qualités suivantes :

- Performance : il est optimisé et cherche à minimiser l'utilisation du système ;
- Compatibilité : il est compatible avec Sendmail afin de faciliter la migration ;
- Sûreté et robustessse : il se comporte de façon rationnelle face au nombre de tâches demandées. Si le système n'a plus de mémoire ou d'espace disque, il n'agravera pas la situation. Il a été conçu pour rester sous contrôle ;
- Flexibilité : il est conçu de façon modulaire, une douzaine de petits programmes effectent des tâches bien précises. Il est possible de remplacer ces programmes par des produits maison, voire de supprimer certains programmes inutiles dans certains cas (un firewall ou une station de travail n'a pas besoin de livrer localement des e-mails) ;
- Sécurité : il utilise plusieurs niveaux de défense afin de protéger le système de toute intrusion. Chaque programme est enfermé dans sa cage (chrooté), il n'y a aucun lien direct entre le réseau et les programmes sensibles comme la livraison du courrier local.

L'architecture globale de postfix est la suivante :



La figure montre les composants principaux du système Postfix et le cheminement de l'information :

- Les ellipses claires sont des programmes de traitement du courrier ;
- Les rectangles clairs sont des files d'attente ou des dossiers de courrier.
- Les rectangles foncés sont des tables de consultation.
- L'ensemble des programmes situés dans le cadre noir fonctionnent sous le controle du démon master.
- Les données situées dans le cadre noir appartiennent au système de courrier Postfix.

Note : afin de conserver une certaine lisibilité, certains éléments ont été omis.

Toutes les opérations de postfix sont bien compartimentées. Cela garantit donc une sécurité optimale.

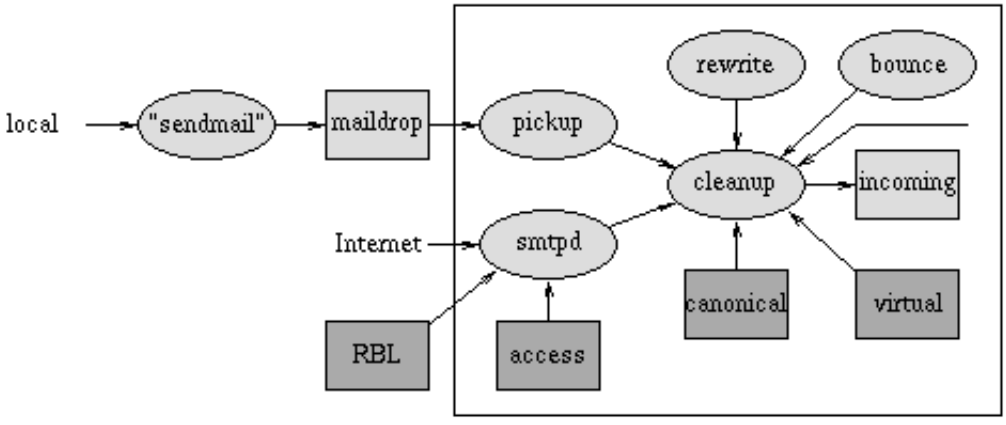
Postfix utilise quatre files d'attentes :

- maildrop : contient les messages locaux ;
- incoming : contient les messages qui ont été prélevés dans maildrop par le démon pickup, puis qui ont été traités par le démon cleanup. Cette file contient aussi les messages venant de l'extérieur ;
- active est une file contenant les messages en cours de délivrance par le démon qmgr ;
- deferred : contient les messages qui n'ont pas pu être délivrés car ils contiennent des erreurs.

2. Réception du courrier

Lorsqu'un message arrive dans le système de courrier Postfix, qu'elle que soit son origine, son premier arrêt se fait dans la file d'attente Incoming. La figure ci-dessous montre les composants principaux qui sont impliqués lors de l'arrivée d'un nouveau courrier :

Arrivée d'un nouveau courrier :



Soit le courrier est posté par des utilisateurs directement sur la machine locale. Le programme sendmail (ne pas confondre avec le MTA concurrent) de postfix appelle le programme privilégié postdrop qui dépose le message dans la file maildrop, où le message est pris par le démon de collecte pickup. Ce démon fait quelques contrôles, afin d'éviter de polluer

le reste du système de mail avec du courrier invalide. Pour éviter des accidents, les permissions du répertoire contenant maildrop sont telles que tout le monde peut y écrire, mais aucun utilisateur n'a le droit d'effacer le contenu (sticky bit positionné).

Soit le courrier provient du réseau. Le serveur SMTP Postfix (smtpd) reçoit le message et fait quelques contrôles afin de protéger le reste du système Postfix. Le serveur SMTP peut être configuré pour mettre lutter contre le spam (ECU ou Unsollicited Commercial Mail) sur la base de listes noires locales ou issues du réseau, de requêtes DNS (domaine de l'expéditeur) ou de toute autre information concernant l'émetteur.

D'autres origines sont possibles, car le courrier peut être généré par le système Postfix lui-même, dans le but de renvoyer un message non-délivrable à son expéditeur. C'est le démon bounce ou defer qui se charge de ce travail. Le courrier peut aussi être généré par le système Postfix afin d'avertir le postmaster d'un problème.

3. Traitement du courrier

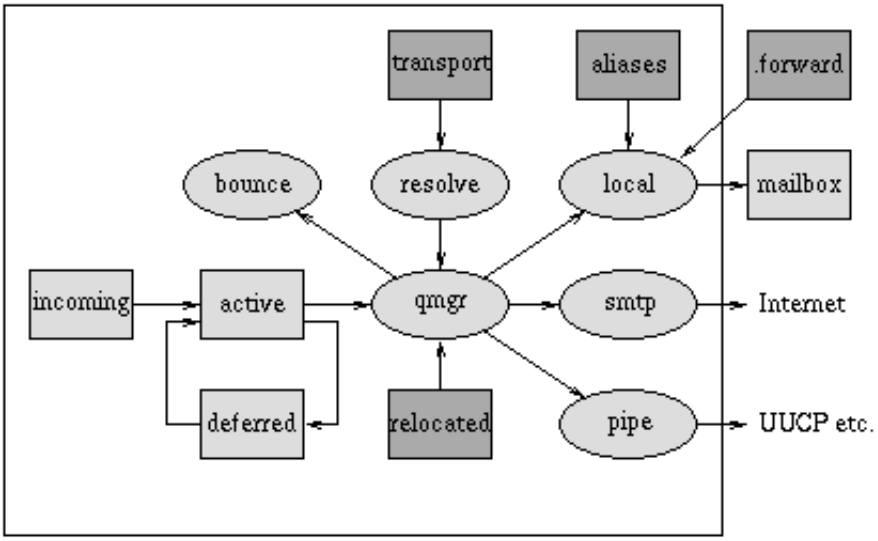
Si le message est livrable : le démon de nettoyage cleanup ajoute le champ From: et d'autres en-têtes manquantes dans le message. Il demande éventuellement au demon trivial-rewrite de réécrire l'adresse de réponse dans le format standard user@fully.qualified.domain. Le démon cleanup insère le résultat sous la forme d'un seul fichier dans la file d'attente incoming et informe le gestionnaire de files d'attente queue manager (qmgr) de l'arrivée du nouveau courrier.

Si le courrier n'est pas livrable, un e-mail est généré automatiquement afin de renvoyer le courrier non livrable à l'expéditeur. Ce sont les démons bounce ou defer qui annoncent la mauvaise nouvelle. Un autre e-mail est également généré pour informer le responsable de la messagerie (postmaster) du problème.

4. Livraison du courrier

Une fois qu'un message a atteint la file d'attente incoming, l'étape suivante consiste à le livrer. La figure montre les principaux composants du système de distribution du courrier de Postfix.

Livraison du courrier :



Qmgr est le coeur du système de messagerie Postfix. Le gestionnaire maintient une file d'attente deferred séparée pour chaque message qui n'a pas pu être livré. Le gestionnaire maintient une petite file d'attente active avec juste quelques messages prêts pour la livraison. Sur la demande de qmgr, les demons bounce et defer génèrent des rapports de non-livraison lorsqu'un message ne peut être acheminé.

5. Les commandes disponibles

- La commande **postfix** contrôle le système de courrier. C'est la commande pour démarrer et arrêter le système ou le relancer (paramètre **reload**), et pour quelques autres opérations administratives.
- La commande **postalias** sert à convertir le fichier aliases en format bases de données (*.db).

- La commande **postcat** montre le contenu de la file d'attente de Postfix.
- La commande **postconf** montre les paramètres donnés dans le fichier main.cf de Postfix : les valeurs réelles, les valeurs par défaut, ou les paramètres qui n'ont pas de valeur par défaut.

Exemples :
#postconf -n affiche les paramètres modifiés par notre configuration.
#postconf -d affiche les paramètres par défaut.

- La commande **postqueue** est l'utilitaire lancée par la commande de sendmail pour vider ou lister la file d'attente du courrier.
- La commande **postmap** sert à créer des bases de données à partir de fichiers passés en paramètre.

6. Les fichiers de configuration

Sous Linux, les fichiers de configuration sont placés dans /etc/postfix.

Le principal fichier de configuration est **main.cf**. Il définit les paramètres généraux de postfix. Son contenu pourrait ressembler à :

srv001:~# cat /etc/postfix/main.cf

```
# see /usr/share/postfix/main.cf.dist for a commented, fuller
# version of this file.
# Do not change these directory settings - they are critical to Postfix
# operation.
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
program_directory = /usr/lib/postfix
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
setgid_group = postdrop
biff = no
# Nom du fichier d'alias
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
# Nom du fichier de correspondance pour les adresses virtuelles
#virtual_maps = hash:/etc/postfix/virtual
# Nom de domaine
# Ce paramètre ne sert pas directement, mais peut être utilisé par la suite.
mydomain=massol.com
# Nom d'hôte
# Ce paramètre ne sert pas directement, mais peut être utilisé par la suite.
myhostname=mail.massol.com
# Extension pour les mails envoyés depuis la machine
myorigin=$myhostname
# Liste des domaines pour lesquels le serveur accepte le mail
# ET délivre le mail en local
mydestination = $myhostname, localhost.$mydomain, localhost
# Liste des domaines pour lesquels le serveur accepte le mail
# ET le relaie à d'autres serveurs de mail
#relay_domains =
#Dans le cas où on a besoin d'un serveur pour relayer les mails sortants:
#relayhost =
# Réseaux en lesquels j'ai confiance
# i.e. pour lequel mon serveur mail accepte de relayer du mail
# ATTENTION : il ne faut pas mettre n'importe quoi pour que le serveur
# mail ne devienne pas un relai pour le spam !
mynetworks = 127.0.0.0/8
# Commande à exécuter pour délivrer les mails en local
mailbox_command = procmail -a "$EXTENSION"
# Taille maximale pour les mailbox (0 = pas de limite)
mailbox_size_limit = 0
# appending .domain is the MUA's job.
append_dot_mydomain = no
```

default_transport=smtp
#transport_maps =

smtpd_banner indique le message de bienvenue affiché par le démon. Les mots qui commencent par un $ sont des variables.

setgid_group indique le groupe propriétaire du démon postfix.

append_dot_mydomain indique si postfix doit compléter les noms d'utilisateurs avec le nom du domaine. En général, on laisse ce travail au client.

myhostname indique le nom FQDN du serveur.

mydomain indique le nom de domaine.

myorigin indique le nom du serveur qui apparaîtra dans l'en-tête du courrier.

mydestination indique les domaines terminaux. Ce paramètre est fondamental pour éviter les boucles de routage du courrier.

Un autre fichier important s'appelle **master.cf**. Son apparence est la suivante (extraits) :

#
==========
 # or port may be giv
 # service type private unpriv chroot wakeup maxproc command + args
 # for the SMTP server: # (yes) (yes) (yes) (never) (50)
#
==========
#
 # Transport type: "inet" for Intern
smtp inet n - n - - smtpd
 # sockets, "fifo" f
#628 inet n - - - - qmqpd: whether or not access is restricted to the mail system. pickup fifo n - - 60
flush unix n - - 1000? 0 flush
smtp unix - - n - - smtp
showq unix n - - - showq

Ce fichier sert à configurer le démon master responsable de la gestion des différents services qui composent postfix.

7. Installation du MUA et journal

Logiquement, votre système a été installé avec la commande mail. Celle-ci est pratique pour tester le fonctionnement du serveur. D'autres clients de messagerie existent (en mode texte : pine, mutt...) et en mode graphique (kmail, mozilla...). Les messages de l'utilisateur sont stockés dans son répertoire personnel, dans le fichier mbox (parfois, ils sont stockés dans un répertoire Maildir).

Le fichier log à utiliser sous Linux est : /var/log/mail.log

Exercice 3 :

3.1 Donnez les commandes linux permettant de créer un lien symbolique pointant vers la commande postfix dans le répertoire init.d puis de permettre une activation de postfix au démarrage.

 # ln -s /usr/sbin/postfix /etc/init.d/postfix
 # update-rc.d postfix defaults

Adding system startup for /etc/init.d/postfix ...
/etc/rc0.d/K20postfix -> ../init.d/postfix
/etc/rc1.d/K20postfix -> ../init.d/postfix
/etc/rc6.d/K20postfix -> ../init.d/postfix
/etc/rc2.d/S20postfix -> ../init.d/postfix
/etc/rc3.d/S20postfix -> ../init.d/postfix
/etc/rc4.d/S20postfix -> ../init.d/postfix

/etc/rc5.d/S20postfix -> ../init.d/postfix

3.2 Donnez la commande linux permettant de copier le modèle de fichier main.cf dans le répertoire des fichiers de configuration de postfix.
 # cp /usr/share/postfix/main.cf.debian /etc/postfix/main.cf

3.3 Donnez la commande linux permettant de rendre accessible à tout le monde, le dossier de file d'attente des messages envoyés en local.
 # chmod 1733 /var/spool/postfix/maildrop

3.4 Donnez la commande linux permettant de créer la base des alias.
 # postalias hash:/etc/aliases

Exercice 4 :

Il est possible de contourner les interdits de certains FAI dont les administrateurs imposent de plus en plus de restrictions sur l'origine des messages qu'ils reçoivent : vérification du domaine de l'expéditeur, Blacklists...

4.1 Tout faire passer par le MTA de son propre FAI :
Donnez les modifications à faire dans le fichier de configuration pour faire transiter (relayer) tous les mails par votre FAI dont le serveur SMTP est : smtp.free.fr
relayhost = smtp.free.fr
En cas de lenteur du smtp de votre FAI, les clients ne seront pas perturbés, c'est votre MTA qui gérera au mieux les faiblesses du MTA du FAI.

4.2 Faire un tri sélectif des adresses des destinataires :
Il est possible de travailler plus "finement" en délivrant directement les messages aux destinataires qui les acceptent et en passant par le smtp du FAI pour les autres.
Il faut tout d'abord créer un fichier texte /etc/postfix/**transport_maps**, dans lequel seront indiqués le nom des domaines qui doivent être redirigés vers le smtp de notre FAI avec la syntaxe suivante :
<domaine à relayer> smtp:<nom du serveur smtp du FAI>

4.2.1 Donnez le fichier permettant de relayer les domaines aol.com, .aol.com, aol.fr et .aol.fr.

aol.com smtp:smtp.free.fr
.aol.com smtp:smtp.free.fr
aol.fr smtp:smtp.free.fr
.aol.fr smtp:smtp.free.fr

4.2.2 Donnez la commande permettant de créer la base de donnée correspondante

postmap /etc/postfix/transport_maps

4.2.3 Donnez la modification à apporter à main.cf pour que postfix prenne en compte cette base de donnée

transport_maps = hash:/etc/postfix/transport_maps

postfix reload

De cette manière, seuls les domaines qui posent problèmes seront acheminés via le MTA de votre FAI, tous les autres étant joints directement.