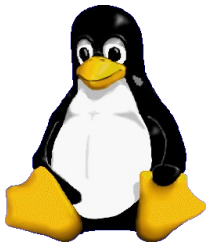


Le système GNU/Linux

By ShareVB

Outils réseaux

## Table des matières


I.Whois (port 43)....1
II.Test de DNS (port 53 UDP)....2
III.Nom de l'hôte local....3
IV.Les PING et ICMP....3
V.La route des paquets : traceroute (tracert)....3
VI.Connexion à distance/test de services : telnet....4
VII.Correspondances numéros de ports/noms de services....5
VIII.Connaître ses Interfaces :....5
IX.Les routes....6
X.Les connexions de l'ordinateur....6
XI.Le protocole FTP....7
XII.Le mail : SMTP/POP3/IMAP....7
XIII.Contenu d'un mail :....8
a)Entête :....8
b)Corps :....8
c)Enveloppe :....8
XIV.Envoi de mail :....8
a)Base ....8
b)Exemple de session SMTP....9
c)Exemple de session ESMTP....9
XV.Réception du courrier....10
a) POP : Post Office Protocol et IMAP : Internet Message Access Protocol....10
b)Le protocole POP3....10
c)Exemple de session POP....11
d)Le protocole IMAP....12
e)Exemple de session IMAP....13
f)Interaction du mail avec le DNS....13
XVI.SNMP : Simple Network Management Protocol....14
XVII.NMS : Network Management System....16
XVIII.Big Brother....17
XIX.MRTG : ....17
a) Installation....17
XX.Découverte de réseaux : psnmp, nomad....17


# I. Whois (port 43)

Les informations sur les noms de domaines sont déclarées par les prestataires de service lors de l'achat du domaine auprès des organismes de nommage (NIC France, InterNIC, etc.). Ces informations sont généralement les noms des serveurs de noms qui seront autoritaires sur la zone,

leurs adresses IP, … Si vous souscrivez un hébergement (OVH, OneAndOne..), il assure, en général, l'enregistrement auprès du bon organisme de nommage.

L'accès à cette base de données est libre, et permet donc à chacun de savoir si un nom de domaine est libre, ou par qui il est utilisé.

La syntaxe de la commande whois est la suivante (la partie commençant au @ est facultative):

```
whois <zone>@<server_whois>
```

Le serveur de base est `whois.iana.org`. Il permet de savoir qui gère un Top Level Domain, par exemple, .net, .org, .com... On peut rechercher qui gère le domaine `.org` par `whois org@whois.iana.org`.

Une fois que l'on a trouvé le gestionnaire on peut lui demander qui gère un site particulier :

```
whois mozilla.org@whois.pir.org
```

Si vous faites une requête sur un nom de domaine sans préciser le serveur Whois, le processus est le suivant:

- on demande à whois.iana.org le serveur Whois qui gère le TLD du domaine demandé.
- on demande au serveur trouvé auparavant les infos Whois du domaine...Si on n'obtient pas d'informations, on obtient à la place le nom du serveur Whois auquel il faut demander les informations...et ainsi de suite...

# II. Test de DNS (port 53 UDP)

Un serveur DNS a deux rôles fondamentaux :

- Répondre aux requêtes de toutes les autres machines du monde concernant son (ou ses) domaine(s). Par exemple si le serveur gère le domaine titi.com, la question pourrait être : "Quelle est l'adresse IP de la machine www.titi.com ?".
  - Le serveur détenant les informations de zone est appelé primaire
  - Les serveurs qui sont là en cas de panne ou surcharge du primaire sont appelés secondaires. Ils détiennent une copie des informations du domaine.
- Aider les machines clientes à obtenir les résolutions DNS dont elles ont besoin

Pour avoir l'IP d'une URL, on utilisera au choix :

```
dig adresse_internet   [@server_DNS]
host adresse_internet   [server_DNS]
nslookup adresse_internet   [server_DNS]
```

Pour obtenir le nom d'une adresse IP, on utilisera :

```
dig -x IP   [@server_DNS]
host IP   [server_DNS]
nslookup IP    [server_DNS]
```

Pour le nom du serveur de mail d'un domaine, on utilisera :

```
dig -t MX nom_domaine    [@server_DNS]
host -t MX nom_domaine    [@server_DNS]
nslookup -query=MX nom_domaine    [server_DNS]
```

# III. Nom de l'hôte local

Le nom de la machine locale peut s'obtenir par `hostname`

# IV. Les PING et ICMP

PING est le premier outil pour déterminer si une machine est accessible par le réseau. Ce programme donne également le temps d'aller-retour entre le système source et le système destination. Il envoie un message de requête ICMP `echo-request` en direction de la machine spécifiée (`echo-request : type=8, code=0`) et si la machine destination est connectée sur le réseau, elle retourne un message ICMP réponse `echo-reply` (`echo-reply : type=0, code=0`).

```
ping URL_ou_IP
```

L'option `-R` permet de connaître la route qu'a emprunté le paquet dans une limite de 9 routeurs et si les pare-feux ne bloquent pas ICMP dans cette variante. Il faut aussi noter que cette option donne les adresses IP des interfaces de sortie des routeurs mais si les routes ne dépassent pas 4 routeurs aller-retour, on obtient les interfaces par pairs « sorties/entrées ».

Par exemple, si on a l'IP 192.168.4.56, que l'on passe par un routeur 192.168.4.45/10.1.1.1 pour atteindre 10.1.1.10, on obtient par `ping -r` :

**192.168.4.56**
**10.1.1.1**
**10.1.1.10**
**10.1.1.10**
**192.168.4.45**
**192.168.4.56**

Note : si la machine cible est un routeur alors il peut arriver que les Ips du milieu ne soit pas les mêmes.

# V. La route des paquets : traceroute (tracert)

Traceroute est un programme permettant de suivre la route suivi par les datagrammes IP. Il n'est pas limité en nombre de routeurs traversés et utilise le protocole UDP, ICMP et le champ TTL de l'en-tête IP.

Il faut noter que les temps donnés par traceroute peuvent être erronés (les datagrammes UDP et ICMP n'utilisent pas forcément la même route et n'ont pas la même priorité) et que **les adresses IP relevées sont les interfaces entrantes des routeurs**.

Enfin, si la route est modifiée pendant l'exécution du programme, la nouvelle est indiquée.

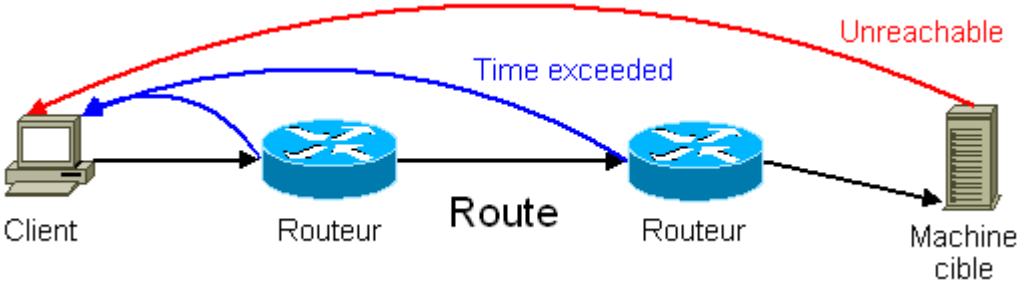
Un traceroute repose sur le champs TTL de l'entête IP d'un datagramme UDP sans importance. Le

paquet UDP à un port destination très élevé afin d'avoir une probabilité très faible d'atteindre un port ouvert.

Le fonctionnement du champs TTL est le suivant :

- Il est décrémenté par chaque routeur traversé jusqu'à 0
- Lorsque le TTL = 0 et si la destination n'est pas atteinte, un message ICMP "time exceeded" est renvoyé à la machine source (avec l'adresse IP de l'interface d'entrée du paquet dans le routeur ayant généré ce message) et le paquet est détruit.

L'intérêt du champs TTL est que si on ne trouve pas la machine destination assez rapidement alors le paquet ne voyage pas indéfiniment sur le réseau ce qui causerait une surcharge des routeurs.




On notera que l'on parle de « temps » alors qu'il s'agit en faite, de nombre de routeurs traversés, ce qui revient légèrement au même.

Le principe est donc le suivant :

- on envoie un datagramme IP/UDP avec un TTL de 1 puis de 2 puis de 3...le numéro de port est très élevé afin d'avoir une probabilité très faible d'atteindre un port ouvert sur la machine cible.
- on attend les paquets ICMP « `time exceeded` » qui informent sur les interfaces d'entrées des routeurs qui ont rejetté les paquets
- lorsque le datagramme UDP atteint la machine cible, on espère que le port n'est pas ouverts et que l'on va donc recevoir un paquet « `unreachable port` » (`port inaccessible, type=3, code=3`) indiquant la fin de la route.

Sous Linux :

```
traceroute URL_ou_IP
```

Sous Windows :

```
tracert URL_ou_IP
```

# VI. Connexion à distance/test de services : telnet

TELNET est l'outil le plus utilisé pour déterminer les paramètres sur une machine via le réseau. Il permet de se connecter à distance et de simuler un écran de terminal. Datant de 1969, il est "implémenté" avec TCP/IP sur la plupart des systèmes et utilise une négociation d'options entre le client et le serveur pour déterminer les fonctionnalités fournies à chaque extrémité. Une fois connecté, il est possible de taper des commandes et de visualiser le résultat comme sur un écran-terminal.

Ce programme permet également de tester des services ouverts en TCP, par exemple pour tester le bon fonctionnement d'un serveur de mail. **Et c'est d'ailleurs la seule utilisation raisonnable de cette outils de nos jours.**

On l'utilise ainsi :
`telnet IP_ou_nom_DNS port`

# VII. Correspondances numéros de ports/noms de services

On trouve la liste des services avec les ports correspondants dans `/etc/services :`
Gestion :

- 53 : DNS (résolution de nom vers IP et inversement)
- 43 : whois (informations sur les noms de domaines)
- 161 : snmp (gestion et statistiques des appareils réseaux)

Transfert de données :

- 20/21 : FTP (transfert de fichiers)
- 80 : HTTP (transfert de pages Web)
- 443 : HTTPS (transfert sécurisé de pages Web)
- 445 : Samba (partage de fichiers Windows)

Remote Control :

- 22 : ssh (terminal distant sécurisé)
- 23 : telnet (terminal distant non sécurisé)

Mail :

- 25 : smtp (envoi de mails)
- 110 : pop3 (récupération de mails)
- 143 : imapv4 (gestion/lecture de boîtes aux lettres)
- 220 : imapv3 (gestion/lecture de boîtes aux lettres)

Autres :

- 123 : ntp (réglage de l'heure/serveur de temps)
- 137/138/139 : netbios (nommage NetBIOS)
- 7 : echo (renvoi les paquets reçu)

# VIII. Connaître ses Interfaces :

Il peut être utile à certains moments de vérifier la validité de ses interfaces réseaux : actives ou pas, adresse IP, configuration IP …
Il existe, pour cela, sous Unix la commande ifconfig qui énumère les interfaces actives ainsi que diverses autres informations. Cette commande permet également de configurer une nouvelle interface réseau.
Son équivalent sous Windows est ipconfig qui permet également de récupérer l'ensemble des paramètres Ips.

Pour obtenir des informations sur toutes les interfaces d'une machine :
`ifconfig`

Pour obtenir des informations sur l'interface spécifiée :

`ifconfig` *`interface`*

Pour changer la configuration d'une interface :
`ifconfig` *`interface IP`* `netmask` *`masque_sous_réseau`*

# IX. Les routes

La commande *route* permet de voir, d'ajouter ou d'enlever les routes se trouvant déclarées sur votre machine. Ainsi pour indiquer à votre machine où aller trouver les adresses qui ne sont pas les adresses de votre réseau local, vous devez lui indiquer la passerelle (ou gateway) vers laquelle elle doit envoyer tous les paquets.
Pour voir les routes indiquer `route -n` (on peut aussi utiliser `netstat -nr`) L'option `-n` permet de ne pas avoir la résolution des noms donc seulement les adresses IP (c'est plus rapide).

Pour ajouter une route :

```
route add [default] [-net | -host] IP_réseau_ou_hôte [netmask masque] [gw IP_passerelle] [[dev] Interface]
```

Pour supprimer une route :

```
route del [default] [-net | -host] IP_réseau_ou_hôte [netmask masque] [gw IP_passerelle] [[dev] Interface]
```

- `default :` indique que l'on ajoute la route par défaut
- *`IP_réseau_ou_hote`* `:` adresse IP du réseau ou de la machine cible de la route
- *`masque`* `:` masque de sous-réseau de l'IP précédente (si différente d'un masque de classe)
- *`IP_passerelle`* `:` IP de la machine qui relaie les paquets depuis et vers le réseau cible
- *`Interface`* `:` interface à utiliser pour transmettre les paquets

Pour garder les réglages de routes des interfaces sous Red Hat, il faut modifier les fichiers `/etc/sysconfig/network` et `/etc/sysconfig/network-scripts/ifcfg-`*`interface`*. Sous Debian, il s'agit de `/etc/interfaces`. Voir tuto sur la configuration.

# X. Les connexions de l'ordinateur

Elle permet en effet de connaître les ports en écoute sur votre machine, sur quelles interfaces, avec quels protocoles de transport (TCP ou UDP), les connexions actives et de connaître les routes.

Pour voir les connexions actives en tcp en format numérique (IP au lieu de nom DNS) :

```
netstat -nt.
```

L'option `-t` liste les connexions TCP, `-u` UDP, `-ip` INET, `-x` UNIX.
L'option `-a` énumère les ports en cours d'utilisation ou ceux qui sont écoutés par le serveur.
L'option `-i` donne des informations sur les interfaces réseau.

# XI. Le protocole FTP

Le protocole FTP (File Transfert Protocol) est, comme son nom l'indique, un protocole optimisé pour le transfert de fichiers. Les premiers clients a être apparus sont évidemment en ligne de commande sous Windows comme sous Unix (commande `ftp`).

Il sera détaillé dans le tuto sur le FTP et vsftpd.

Exemple de session :

```
[utilisateur@machine ~]$ ftp
ftp> open ftp.domaine.fr
Connected to ftp.domaine.fr (x.y.z.w).
220 Bienvenue sur le serveur ftp...
Name (ftp.domaine.fr:utilisateur): ftp
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (x,y,z,w,251,146)
150 Here comes the directory listing.
drwxr-xr-x    3 200      50           4096 Nov 10 2004 pub
-rw-r--r--    1 200      50            345 Apr 18 2002 welcome.msg
226 Directory send OK.
ftp> quit
```

# XII. Le mail : SMTP/POP3/IMAP

Le système de courrier électronique s'appuie sur plusieurs composantes formant ainsi des modules, qui coopèrent les uns avec les autres. Cette architecture permet une grande souplesse, par exemple, le remplacement d'un seul module par un élément plus récent suffit à faire évoluer le système de courrier électronique. Les composantes sont au nombre de quatre :

- Mail Transfer Agent (MTA) : il s'agit de la gestion de SMTP (sendmail/postfix)
- Mail Delivery Agent (MDA) : il s'agit de la gestion de la récupération depuis le stockage : POP/IMAP (dovecot/procmail/sendmail)
- Boîte aux lettres : il s'agit de la gestion du stockage à l'arrivée des mails
- Mail User Agent (MUA) : il s'agit de l'outils qui vous permet de lire vos mails : mail, fetchmail, mutt

# XIII. Contenu d'un mail :

## a) Entête :

Cette partie contient toutes les informations qui vont permettre au courrier électronique d'être acheminé jusqu'à son destinataire. Chaque ligne de l'en-tête est composée de

*mot-clé : arguments*

- Reply-to: adresse email de réponse
- Received: MTA intermédiaires
- Date: date d'envoi
- From: email émetteur
- To: email destinataire
- Cc: Copie(s) à
- Bcc: Copie(s) cachée(s) à
- Subject: sujet du mail

## b) Corps :

Cette partie contient les données du courrier électronique. Il s'agit de texte, codé soit sur 7 bits (protocole SMTP : Simple Mail Transfer Protocol) ce qui interdit l'utilisation des caractères accentués, soit sur 8 bits (protocole ESMTP : Extended Simple Mail Protocol) et dans ce dernier cas, le problème précédent ne se pose plus.

## c) Enveloppe :

Il convient donc de véhiculer une nouvelle information avec un message. Cette information par analogie avec la poste, s'appelle l'enveloppe. Elle sert à router les messages.

# XIV. Envoi de mail :

## a) Base

SMTP signifie Simple Mail Transfert Protocol. Il s'agit donc du protocole de transfert de courrier électronique le plus basique, aussi bien dans ses spécificités que dans son utilisation. Tout envoi de courrier électronique commence par l'établissement d'un canal de communication entre les deux correspondants que sont l'expéditeur et le destinataire. SMTP est un protocole application, au dessus de la couche Transport TCP.
Après l'ouverture du canal, des commandes spécifiques permettent de traiter le courrier à envoyer, mais il existe aussi un certain nombre de commandes qui effectuent des actions propres au suivi de la communication entre les deux entités.
Chaque commande invoquée est suivie d'une réponse, composée d'un code numérique, et d'un message. Par exemple, le code numérique « 250 » est utilisé pour signifier « OK », tandis que « 500 » signifiera « commande inconnue ».

## b) Exemple de session SMTP

Voici un exemple de session SMTP :

```
[machine]$ telnet serveur 25
Trying x.y.z.w...
Connected to serveur.
Escape character is '^]'.
220 serveur ESMTP Sendmail 8.11.2/8.11.2; Thu, 3 Jan 2004 13:52:55
+0100
HELO serveur
250 serveur Hello machine [x.y.z.w], pleased to meet you
HELP
214-2.0.0 This is sendmail version 8.11.2
214-2.0.0 Topics:
214-2.0.0         HELO       EHLO  MAIL   RCPT  DATA
214-2.0.0         RSET       NOOP  QUIT   HELP  VRFY
214-2.0.0         EXPN       VERB  ETRN   DSN   AUTH
214-2.0.0         STARTTLS
214-2.0.0 For more info use "HELP <topic>".
214-2.0.0 To report bugs in the implementation send email to
214-2.0.0         sendmail-bugs@sendmail.org.
214-2.0.0 For local information send email to Postmaster at your
site.
214 2.0.0 End of HELP info
MAIL FROM: emetteur@domaine.fr
250 2.1.0 emetteur@domaine.fr... Sender ok
RCPT TO: <destinataire>
250 2.1.5 <destinataire>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Subject: Essai de mail

Ceci est le corps du message ...
.
250 2.0.0 g03CsNl07973 Message accepted for delivery
NOOP
250 2.0.0 OK
QUIT
221 2.0.0 serveur closing connection
Connection closed by foreign host.
[machine]$
```

## c) Exemple de session ESMTP

Voici un exemple de session ESMTP :

```
[machine]$ telnet serveur 25
Trying x.y.z.w...
Connected to serveur.
Escape character is '^]'.
```

```
220 serveur ESMTP Sendmail 8.11.2/8.11.2; Thu, 3 Jan 2004 14:00:03
+0100
ehlo serveur
250-serveur Hello machine [x.y.z.w], pleased to meet you
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-SIZE 5000000
250-DSN
250-ONEX
250-ETRN
250-XUSR
250 HELP
MAIL FROM: emetteur@domaine.fr
250 2.1.0 emetteur@domaine.fr... Sender ok
RCPT TO: <destinataire>
250 2.1.5 <destinataire>... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
message ...
.
250 2.0.0 g03D1tl10892 Message accepted for delivery
QUIT
221 2.0.0 serveur closing connection
Connection closed by foreign host.
[machine]$
```

# XV. Réception du courrier

## a) POP : Post Office Protocol et IMAP : Internet Message Access Protocol

Le protocole POP est le plus ancien et le plus largement répandu. Très simple, il possède environ une dizaine de commandes alors que le protocole IMAP est plus récent et propose plus de fonctionnalités que POP.

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 (serveur 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, 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.

## b) Le protocole POP3

Ce protocole est défini par la RFC1939. Généralement, ce protocole est utilisé en mode offline, c'est à dire que le client ramène dans sa boite aux lettres locale son courrier stocké sur le serveur depuis sa dernière relève. Le courrier est alors lu après l'avoir rapatrié, hors connection, en local. Lorsque

le client souhaite effacer son courrier, il doit rendre un "flag" de suppression actif pour le message ç supprimer. C'est seulement à la fin de la connexion, que le courrier ainsi marqué sera effacé du disque du serveur. Par contre, il n'y a qu'une seule boite aux lettres sur le serveur.

En général, le serveur POP écoute le port 110 de la machine.

La syntaxe des commandes du protocole POP est la suivante:
*[ commande ] [ argument ] [ argument ] ( retour chariot )*

Les commandes peuvent être :

- **USER** : prend comme argument le nom de l'utilisateur qui veut se connecter à sa boite aux lettres.
- **PASS** : prend comme argument le mot de passe du client qui veut se connecter.
- **STAT** : pas d'argument, affiche le nombre de messages de la boite aux lettres et sa taille.
- **LIST** : si il n'y a pas d'argument, affiche pour chaque message son numéro et sa taille, sinon ne le fait que pour le message passé en argument.
- **RETR** : prend en argument le numéro du message que l'on veut rapatrier.
- **DELE** : prend en argument le numéro du message que l'on souhaite effacer.
- **NOOP** : ne rien faire...mais prouve que l'on n'est pas planté
- **RSET** : permet de retirer tous les drapeaux "effacés".
- **QUIT** : pas d'argument, permet de quitter la connexion en cours.

Les réponses retournent +OK en début de ligne si tout ce passe bien sinon retournent - ERR en cas de problème.

# c) Exemple de session POP

```
[utilisateur]$ telnet serveur 110
Trying x.y.z.w...
Connected to serveur.
Escape character is '^]'.
+OK POP3 Welcome to GNU POP3 Server Version 0.9.8
<11965.1010143243@serveur>
USER utilisateur
+OK
PASS motdepasse
+OK opened mailbox for utilisateur
STAT
+OK 1 369
LIST 1
+OK 1 369
RETR 1
+OK
Return-Path: <utilisateur@domaine.fr>
Received: (from nom@machine)
            by serveur (8.11.2/8.11.2) id g04BKZJ10755
            for utilisateur; Fri, 4 Jan 2002 12:20:35 +0100
Date: Fri, 4 Jan 2002 12:20:35 +0100
From:  utilisateur@domaine.fr
Message-Id: <200201041120.g04BKZJ10755@morglum.iut-amiens.fr>
To:  utilisateur2@domaine.fr
Subject: Essai de mail
```

```
Bonjour utilisateur
.
DELE 1
+OK Message 1 marked
QUIT
+OK
Connection closed by foreign host.
[utilisateur]$
```

# d) Le protocole IMAP

Ce protocole est définie par la RFC2060. Il permet la gestion de plusieurs boites aux lettres sur plusieurs serveurs à partir de plusieurs clients. Il permet également la consultation de messages MIME. Le mode online supporté veut dire que le client et le serveur sont connecté pendant toute la durée de la manipulation du courrier électronique. Le client peut donc consulter son courrier en restant connecté.

De ce fait, on peut créer toute une arborescence de boites aux lettres à partir de la principale (inbox). Par exemple, l'utilisateur *toto* peut avoir des sous boites aux lettres, une pour les messages personnels, une pour les message du travail, une pour la cuisine, ....

La syntaxe des commandes du protocole IMAP est la suivante:

*[ **tag** ] [ commande ] [ argument ] [ argument ] ( retour chariot )*

**Le tag est un code alphanumérique** que vous pouvez choisir à votre guise. En générale, on utilise un code croissant pour identifier temporellement les commandes.
Les commandes principales sont:

- **LOGIN** : prend comme argument le nom et le mot de passe du client.
- **SELECT** : prend comme argument le nom de la boite que l'on veut sélectionner.
- **CREATE** : le nom de la nouvelle boite est passée en argument.
- **DELETE** : efface la boite aux lettres qui est passée en argument.
- **RENAME** : prend comme argument le nom de l'ancienne boite et le nouveau.
- **LIST** : prend comme argument la référence (ex : etc/mail/....) et le nom de la boite.
- **STATUS** : donne les informations sur une boite.
- **SEARCH** : commande qui permet de rechercher des messages selon des critères spécifiques.
- **CLOSE** : pas d'argument, ferme la boite mais attend de nouveau un login.
- **FETCH** : prend plusieurs arguments, permet de consulter un message

Le serveur peut retourner plusieurs types de réponses:

- *tag* * OK : la commande c'est bien déroulée.
- *tag* * NO : échec de la commande.
- BAD : erreur de protocole.
- PREAUTH : message d'accueil, indique des fois qu'il n'est pas nécessaire de se loger.
- BYE : le serveur va fermer sa session.

# e) Exemple de session IMAP

```
[utilisateur]$ telnet serveur 143
Trying x.y.z.w...
Connected to serveur.
Escape character is '^]'.
```

```
* OK gip Cyrus IMAP4 v2.0.16 server ready
a001 LOGIN utilisateur motdepasse
a001 OK User logged in
a002 select inbox
* FLAGS (\Answered \Flagged \Draft \Deleted \Seen)
* OK [PERMANENTFLAGS (\Answered \Flagged \Draft \Deleted \Seen
\*)]
* 7 EXISTS
* 1 RECENT
* OK [UNSEEN 1]
* OK [UIDVALIDITY 1004181443]
* OK [UIDNEXT 3652]
a002 OK [READ-WRITE] Completed
a003 FETCH 7 FULL
* 7 FETCH (FLAGS (\Recent) INTERNALDATE " 4-Jan-2004 12:43:40
+0100" RFC822.SIZE
636 ENVELOPE ("Fri, 4 Jan 2004 12:43:40 +0100" "essai de mail
IMAP" ((NIL NIL
"utilisateur" "domaine.fr")) ((NIL NIL "utilisateur"
"domaine.fr")) ((NIL NIL "utilisateur"
"domaine .fr")) ((NIL NIL "destinataire" "domaine2.fr")) NIL NIL
NIL
"<200201041143.g04Bhe018099@domaine.fr>")                  BODY
("TEXT"       "PLAIN"
("CHARSET" "us-ascii") NIL NIL "7BIT" 11 1))
a003 OK Completed
a004 FETCH 7 body
* 7 FETCH (BODY ("TEXT" "PLAIN" ("CHARSET" "us-ascii") NIL NIL
"7BIT" 11 1))
a004 OK Completed
a005 FETCH 7 BODY[TEXT]
* 7 FETCH (FLAGS (\Recent \Seen) BODY[TEXT] {11}
Voila un message...
)
a005 OK Completed
a006 logout
* BYE LOGOUT received
a006 OK Completed
Connection closed by foreign host.
[utilisateur]$
```

# f) Interaction du mail avec le DNS

L'envoie de courrier à user@site peut se faire de deux façons:

- la première méthode consiste à contacter site et de lui remettre le mail,
- la seconde consiste à délivrer le message à une machine qui assurerait un service de poste centrale pour un groupe de machine.

Si la première méthode est simple à l'esprit à mettre en oeuvre, il n'en est pas de même dans la réalité, car elle nécessite la maintenance d'un MTA sur chaque machine du réseau. C'est pour cela que dans la pratique, on utilise généralement la deuxième méthode.

C’est à ce niveau que le DNS intervient, car il permet de connaître ces machines postales. Il est en effet capable de stocker dans sa base de données ce type d’informations? Il s’agit en l’occurrence des RR de type MX.

Au lieu de contacter le DNS pour obtenir l’adresse IP de la machine destinatrice, on va le contacter pour lui demander le MX de la station destinatrice, puis l’adresse IP de cette dernière et c’est à lui que sendmail se connectera pour transmettre le message.

Dans la pratique, l’utilisation d’un MX présente plusieurs avantages :

- Il permet de joindre des machines en bordure de l’Internet (serveurs mails) en spécifiant la passerelle vers ces machines, par exemple une passerelle capable de communiquer en UUCP.
- Mettre le même MX pour toutes les stations d’un même domaine permet de centraliser la distribution du courrier. On peut ainsi fiabiliser la distribution en passant par une machine maintenue, administrée et à jour qui distribue les courriers à l’intérieur d'une entreprise. Le DNS, pour résumer grossièrement, ne contient que des enregistrements de données de type (clé, valeur). C’est ce qui permet ainsi d’envoyer du courrier à un domaine en mettant simplement dans la base le RR suivant :

  *domaine.fr IN MX 10 serveur_courrier_entrant*

Le domaine *domaine.fr* a donc un MX et on peut donc utiliser l’adresse (*username*)*@domaine.fr* qui dirigera les courriers vers *machine.domaine.fr*.

- On peut indiquer plusieurs MXs dans le DNS, avec des priorités différentes, ce qui permet d’assurer la continuité du service du courrier électronique en cas de panne ( envoi des mails vers les MXs secondaires ) et aussi d’éviter l’inondation du MX principal lors de sa remise en marche. Ceci est dû au fait que les mails en attente ne se présenteront pas tous successivement à lui mais au contraire feront l’objet d’une seule connexion SMTP de la part des MXs de secours qui, selon le principe des MXs, retransmettent les mails au MX le plus prioritaire.

  *domaine.fr IN MX 10 machine1*
  *domaine.fr IN MX 20 machine2*

- Même si un MX présente énormément d’intérêts, on peut néanmoins trouver quelques limitations. En effet, comme l’envoie de courrier passe automatiquement par le MX, cela peut être coûteux en termes de performances ( passages inutiles par des routeurs internes ) et le surcharger inutilement.

# XVI. SNMP : Simple Network Management Protocol

Les activités d'administration ont suivies les évolutions en taille des réseaux et sont devenus de plus en plus complexes. Il a ainsi fallu mettre en oeuvre très rapidement des outils informatiques pour prendre en compte ces activités. SNMP (Simple Network Management Protocol) est le premier outil à avoir été développé. Initialement, il avait été conçu comme une solution provisoire et rapidement disponible en attendant des solutions plus complètes en cours d‘étude. D'autres protocoles sont déjà définis comme CMIP/CMIS (Common Management Information Protocol / CMIS: Common Management Information Service ) mais ne sont pas encore utilisés par les constructeurs.

SNMP fonctionne à partir de quatre éléments :

- les MIBs (Management Information Base) :

Chaque noeud administrable du réseau (équipements et stations) possède une base de données contenant des informations locales au noeud, appelée Management Information Base.
Une MIB définit les informations spécifiques d'administration réseaux et leur signification. Elle est composée d'un ensemble de valeurs et paramètres manipulables par le système d'administration.

Les informations disponibles sont de type:

- statiques: nom, constructeur, version d'un équipement;
- dynamiques: état à un instant donné;
- statistiques: compteurs, trafic depuis la mise en route de l'équipement.

Pour désigner de façon unique et sans ambiguïté un élément d'information, il est nécessaire de posséder une règle de nommage. Le terme d'identifiant d'objet est utilisé pour modéliser cette règle : elle est de type hiérarchique. Un identifiant d'objet est une séquence de nombres entiers qui traverse un arbre global constitué d'une racine rattachée par des arcs à un certain nombre de noeuds étiquetés. L'étiquette d'un noeud est l'association d'une courte description textuelle et d'un entier: ISO 1, CCITT 2...

La désignation peut ainsi se faire de deux façons :

- Par un unique mnémonique : ISO.Org.DoD.Internet.
- Par une suite d'entiers : 1.3.6.1.

Chaque sous-élément possède un numéro d'identification, le premier étant 0, un type (entier ou chaîne d'octets) et une valeur.

- les agents :

Un agent sur le noeud se charge de collecter les informations en réponse aux interrogations d'un manager. L'agent peut prendre l'initiative de la communication avec un manager en lui envoyant des Traps pour signaler des événements anormaux (alarmes).

- le manager :

Les managers sont utilisés en deux modes complémentaires :

- Ils sont chargés de questionner les différents agents et de fournir à l'administrateur les informations récupérées sous forme de mails, de graphe, …
- Ils doivent également gérer les Traps, c'est à dire les alertes générées par un agent SNMP du réseau et prévenir l'administrateur.

- le protocole SNMP :

L'agent SNMP collecte les informations de la MIB de l'équipement et répond aux requêtes du manager. On trouve maintenant des agents SNMP sur tout équipement dit administrable. Les constructeurs fournissent également des agents pour les stations du réseau. Toutefois, certains anciens équipements administrables ne sont pas conformes à SNMP. Dans ce cas, il peut être possible d'utiliser un proxy-agent sur un équipement SNMP, qui va servir d'intermédiaire avec l'équipement non SNMP

Le SNMP fonctionne avec des requêtes, des réponses et des alertes. Simplement dit, le manager envoie des requêtes à l'agent sur chaque élément du réseau et celui-ci doit exécuter la requête et envoyer sa réponse.
Il peut aussi y avoir des alertes asynchrones venant des agents lorsqu'il veulent avertir le manager d'un problème. L'agent écoute sur le port 161 et envoie ses réponses sur le port 162.

Sur les équipements réseaux, le protocole est de plus en plus implémenté (switchs, routeurs, …). Sur des serveurs Unix il peut également être utilisé via un démon spécifique : snmpd.

Dans tous les exemples suivants, `public` désigne la communauté (par défaut), sorte de mot de passe pour accéder à SNMP. Si vous l'avez changé, remplacer `public` par le nom de votre communauté.

Sous unix, il existe des clients snmp en mode console :

- `snmpget` : permet d'interroger un équipement et de récupérer la valeur d'une variable en précisant les numeros de la MIB ou les noms, exemple :

```
$ snmpget -v 2c -c public IP_équipement 1.3.6.1.2.1.4.7.0
IP-MIB::ipInUnknownProtos.0 = Counter32: 0
$ snmpget -v 2c -c public IP_équipement ipInUnknownProtos.0
IP-MIB::ipInUnknownProtos.0 = Counter32: 0
```

- `snmpgetnext` : permet de récuperer la variable suivante :

```
$ snmpgetnext -v 2c -c public IP_équipement ipInUnknownProtos.0
IP-MIB::ipInDiscards.0 = Counter32: 0
$ snmpgetnext -v 2c -c public IP_équipement ipInDiscards.0
```

- `snmpstatus` : permet d'obtenir certaines informations sur un appareil SNMP

```
$ snmpstatus -c public 172.20.0.10
```

- `snmpwalk` : permet de récupèrer toute une branche de la MIB :

```
$ snmpwalk -Os -c public -v 1 IP_équipement system
```

# XVII. NMS : Network Management System

Les outils N.M.S. permettent la gestion à différents niveaux (visualisation, contrôle, modification, ...) des éléments (ou sous éléments) d'un réseau. Il faut toutefois faire attention lorsqu'on souhaite déployer un N.M.S. au ratio suivant : `coût_temps + coût_monétaire /Utilité + trafic_généré`.
Ces outils peuvent être :

- Complètement intégré à un périphérique comme par exemple la gestion d'un switch via une interface web,
- Utilisant des composants et/ou protocoles réseaux propriétaires ouverts ou non.
- Utilisant des connexions TCP/IP classiques pour tester des connexions et/ou services, ex : test icmp de l'accessibilité d'un serveur.
- Utilisant des protocoles réseaux standards, ex : Utilisation de SNMP avec la M.I.B. classique et/ou privée.

# XVIII. Big Brother

Big Brother permet de surveiller le réseau via une interface Web et fonctionne sur le principe du "clients-serveur". Il enregistre les statistiques de fonctionnement de chaque machine et de chaque service configurés et prévient l'administrateur par "mail" si un problème survient.
Chaque service de chaque machine surveillée, aura une page d'information accessible de puis votre navigateur Web : Big Brother utilise des couleurs de fond de page pour prévenir l'administrateur sur l'état du réseau, vert=Tout va bien, rouge=vous avez un pb !

# XIX. MRTG :

MRTG (Multi Router traffic Grapher) est un outil qui utilise le protocole SNMP, et permet de récupérer des informations. Il est essentiellement utilisé pour récupérer les données concernant le trafic sortant et entrant. Il va ensuite générer des graphiques d'états concernant le service à observer. Il a deux modes de fonctionnement : soit lancé à intervalles réguliers grâce à cron par exemple, soit en mode daemon. MRTG permet donc d'interroger n'importe quel type de matériel (switch, routeur, PC, …) à partir du moment où ce dernier supporte le protocole SNMP.

## a) Installation

- Installer MRTG en RPM (http://oss.oetiker.ch/mrtg/)
- Exécuter :

```
# cfgmaker public@IP_équipement > $HOME/mrtg.cfg
```

- Dans le fichier `$HOME/mrtg.cfg` :
  - Changer : `WorkDir: /dossier/page/web/pour/mrtg`
  - Ajouter : `RunAsDaemon: Yes`
- Créer un index

```
# indexmaker $HOME/mrtg.cfg >
/dossier/page/web/pour/mrtg/index.htm
```

# XX. Découverte de réseaux : psnmp, nomad

Ces logiciels se basent sur le protocole SNMP pour découvrir un réseau. Cependant, si un appareil ne gère pas SNMP dans sa globalité, la découverte du réseau peut être fortement réduite.