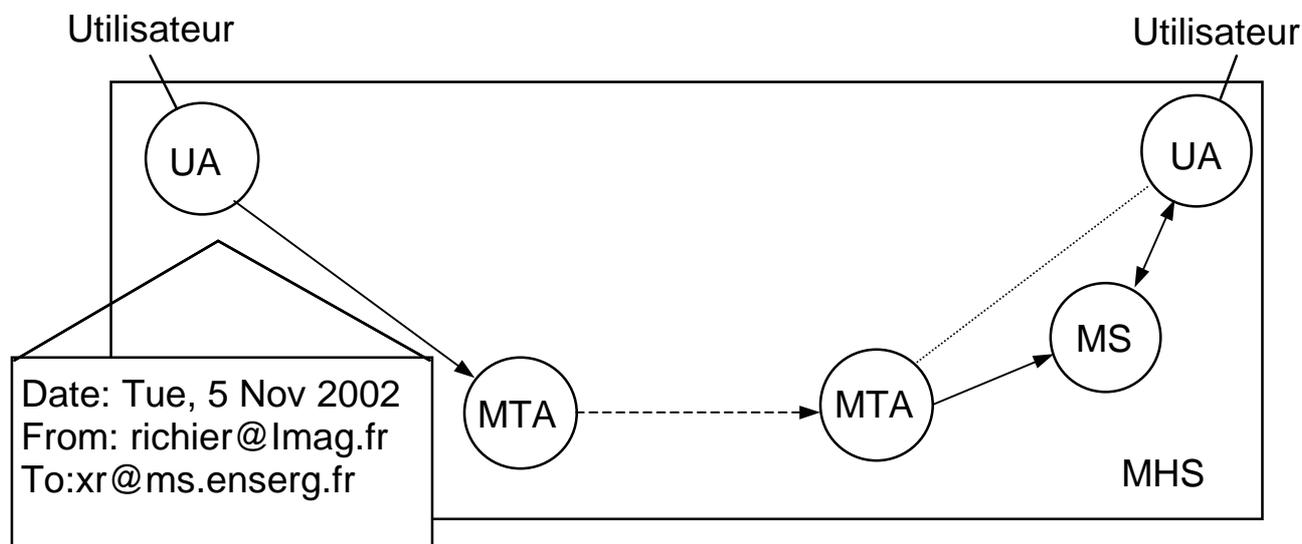


# Messagerie

## Principe

Transmission non sollicitée, transmission asynchrone

## Modèle



Le système de messagerie (MHS, **mail handling system**) envoie des **messages** à une ou plusieurs adresses

L'**user agent** (UA) est l'interface utilisateur

Les MTA (**mail transfert agent**) acheminent les messages

Remise (immédiate ou) différée – utilisation de MS, **mail store**

Origine : personne ou programme (accusé de réception, message erreur système ...)

Destinataire(s) : personne, fax, programme (e.g. mail server)

Différents formats de messages et différents protocoles de transports — possibilité de **passerelles**

- Propriétaires : Bitnet/Earn (IBM), MS Mail, Dec, ...
- Non propriétaires : X400 (ISO/ITU-T), SMTP (Internet)

# Messages

Structure : en-tête + corps

+ enveloppe extérieure pour le transport

**X400** — non décrit ici. Il existe 2 versions X400/83 (limité, pas de support international et structures) et X400/88

**RFC822** — C'est la base, toujours supportée et utilisable, mais obsolète en pratique – On utilise des extensions

Format des données : fichier texte, en ASCII NVT (7 bits, pas d'accents, caractères imprimables), suite de lignes terminées par RC/LF

La structure du texte doit être :

En-tête            -- peut être absent, sans ligne vide  
Ligne vide        -- obligatoire, même si pas d'en-tête  
Corps du message -- le reste, jusqu'au EOF

## • En-tête

- contient des descriptions et des directives
- Format : mot-clé (casse ignorée), ":" information
- possibilité de lignes longues : replier, et commencer les lignes suite par un espace
- analysée et (peut-être) modifiée par le transport
- liste non limitée – définis dans RFC822 :

FROM:            -- adresse expéditeur, mis par l'UA  
TO:                -- destinataire « pour action », mis par l'UA  
CC:                -- destinataire « pour info », mis par l'UA  
BCC:              -- destinataire caché « pour info », mis par l'UA  
REPLY-TO:        -- si ≠from, mis par l'UA  
ERROR-TO:        -- pour retour d'erreur si ≠from, mis par l'UA  
DATE:             -- format analysable par les UA, mis par l'UA  
SUBJECT:         -- mis par l'UA  
RECEIVED:        -- info de trace, mis par les MTA  
MESSAGE-ID:      -- info de trace, mis par le premier MTA

- Possibilité d’extensions : n’importe quel mot-clé (mais risque de conflit) – Les RFC conseillent le format :  
X-... :                      -- extension : serveur liste, anti-spam ...
- Les UA ignorent les lignes inconnues
- Selon le type une ligne peut être répétée ou non, et de format défini ou non :
  - Subject : format libre, unique (0 ou 1)
  - Received : format libre, répétée (>=0)
  - To : format fixe : liste d’adresses séparées par “,”

## • Corps

- Suite de lignes terminées par RC/LF
- Taille maximum de lignes non précisée (mais elle existe, et effet du dépassement non défini)
- Taille maximum du corps non précisée (mais elle existe, rejet en erreur si trop grand)
- ASCII NVT (7 bits, pas d’accents)
- Pas de structure - certains UA ont des extensions (e.g. sun-attachements), et il existe des programmes d’extension classique (shar, binhex, uuencode). Certains UA les déterminent en étudiant le texte du corps e.g. uuencode commence par begin 0644 nom

## Extensions SMTP – Multimédia et caractères nationaux

- **Pour le corps : MIME (RFC 1521)**

**But** Standardiser le format de données “8 bits”

Structurer le corps des messages en contenus (body-parts)

Standardiser les différents contenus possibles

Le message est en général codé comme un corps RFC 822

⇒ il faut préciser les codages

### En-tête Mime

Un en-tête est ajouté à ceux du RFC 822 pour préciser que le corps est un message au format MIME :

#### **Mime-version: 1.0**

Seule la version 1.0 est définie à ce jour.

Le message est structuré en partie séparée par des délimiteurs, chaque partie ayant un en-tête et un corps (séparé par ligne vide). L'en-tête définit le contenu, son format et son codage.

- **Content-type**

Ces contenus sont standardisés en 7 types (extensible).

Chaque type est qualifié par un sous-type. Des paramètres peuvent être présents (syntaxe « ; mot-clé=valeur »)

- **Multipart**

Permet d'inclure plusieurs parties dans un message

- Multipart/mixed            -- par défaut : suite de parties
- Multipart/parallel        -- à afficher en parallèle
- Multipart/digest          -- résumé + contenu
- Multipart/alternative    -- afficher l'une des parties
- Multipart/signed         -- contenu + signature

- **Message** – Permet d’inclure un message
  - Message/rfc822 -- contenu == message
  - Message/partial -- contenu == message tronqué
  - Message/external-body -- le contenu est ailleurs
- **Text** – il faut préciser l’alphabet : « ; charset=ISO-8859-15 »
  - Text/plain -- le défaut
  - Text/richtext -- avec formatage
  - Text/html -- avec formatage
- **Image**
  - Image/gif
  - Image/jpeg
- **Audio**
  - Audio/basic
- **Video**
  - Video/mpeg
- **Application**
  - Application/octet-stream
  - Application/postscript

– “**Attachement**”

On met le Content-type (image, Application ...) et un en-tête :

`Content-Disposition: attachment; filename=...`

• **Format de codage (Content-Transfer-Encoding)**

5 formats de codage :

– **Text** format défaut : 7 bits, US-ASCII ==RFC822

– **Quoted-Printable** pour textes nationaux

Remplacer tout ce qui n'est pas US-ASCII par une

séquence d'échappement =XY où XY est le code ASCII

hexadécimal du caractère. **Perte:** environ 20 %, selon les

données.

Les lignes «longues» sont repliées (marqué par =<NL>).

- **8Bit** pour textes nationaux  
Le message reste constitué de lignes terminées par CR/LF  
Les lignes sont composées de caractères 8 bits  
Codage non compatible avec SMTP «de base»
- **Base64** texte, données  
Codage sur un jeu de caractère à un alphabet 6 bits : 24 bits explosés en 4 caractères US-ASCII. **Perte:** 30% (fixe)  
Avantage sur uuencode :  
Pas d'interférence connue à ce jour avec un MTA, pas de conflits avec EBCDIC, pas de versions différentes
- **Binary** texte, données  
Le message est du binaire pur, pas de notion de lignes

### Remarque :

- 8bit et Binary posent un problème de transmission :  
nécessite ESMTP
- Un texte nécessite de préciser le codage :  
charset=ISO-8859-1, charset=utf8, ...

## Exemple de message Mime

**MIME-Version:** 1.0  
**From:** Nathaniel Borenstein <nsb@bellcore.com>  
**To:** Ned Freed <ned@innosoft.com>  
**Subject:** A multipart example  
**Content-Type:** multipart/mixed;  
boundary=unique-boundary-1

This is the preamble area of a multipart message.

Mail readers that understand multipart format should ignore this preamble.

If you are reading this text, you might want to consider changing to a mail reader that understands how to properly display multipart messages.

--unique-boundary-1

...Some text appears here...

[Note that the preceding blank line means no header fields were given and this is text,

with charset US ASCII. It could have been done with explicit typing as in the next part.]

--unique-boundary-1

**Content-type:** text/plain; charset=US-ASCII

This could have been part of the previous part,  
but illustrates explicit versus implicit typing of body parts.

--unique-boundary-1

**Content-type:** message/external-body; access-type="anon-ftp" ;  
site="ftp.imag.fr"; directory="pub"; name="msg"

--unique-boundary-1

**Content-Type:** multipart/parallel;  
boundary=unique-boundary-2

--unique-boundary-2

**Content-Type:** audio/basic  
**Content-Transfer-Encoding:** base64

... base64-encoded 8000 Hz single-channel  
mu-law-format audio data goes here....

--unique-boundary-2

**Content-Type:** image/gif  
**Content-Transfer-Encoding:** base64

... base64-encoded image data goes here....

--unique-boundary-2--

--unique-boundary-1

**Content-type:** text/richtext

This is <bold><italic>richtext.</italic></bold>  
<smaller>as defined in RFC 1341</smaller>  
<nl><nl>Isn't it  
<bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1

**Content-Type:** message/rfc822

**From:** (mailbox in US-ASCII)

**To:** (address in US-ASCII)

**Subject:** (subject in US-ASCII)

**Content-Type:** Text/plain; charset=ISO-8859-1

**Content-Transfer-Encoding:** Quoted-printable

... Additional text in ISO-8859-1 goes here ...

--unique-boundary-1--

## Conditions à remplir par un UA MIME

- Accepter et afficher du texte US-ASCII
- Accepter les autres jeux de caractères
- Accepter et afficher les caractères ISO-8859-\* commun à ceux présent dans US-ASCII
- Offrir de sauver les contenus non reconnus dans un fichier pour traitement ultérieur
- Reconnaître et afficher les contenus de type Message/rfc822
- Reconnaître le type Multipart/mixed
- Reconnaître le type Multipart/alternative
- Traiter les Multipart/\* non reconnus comme Multipart/mixed
- Décoder les contenus de Application/\* si l'encodage quoted-printable ou base64 est utilisé, puis au minimum être capable de sauver le résultat dans un fichier.

### • Extension des en-têtes (RFC1342)

But : supporter les **textes** non ASCII NVT dans les en-têtes

N.B.: ne fait pas partie de MIME et est assez controversé (pas assez puissant, pas compatible avec adresses dans les champs To:, From:, ...)

Principe : ajouter le nom du jeu de caractères et coder en “qp” ou “base64” (cf. plus bas)

Exemple :

**From:** =?ISO-8859-1?Q?Andr=E9\_?= Pirard <Pirard@site.fr>

**Subject:** =?ISO-8859-2?B?dSB1bmRlc nNOYW5kIHROZSBleGFtcGxlLg==?=

### • Extension des adresses – pas encore : en attente de l'internationalisation du DNS

# Transport

Les messages sont transférés de l'UA au MTA, puis de MTA en MTA, et enfin vers l'UA destinataire

Principe "store and forward" : un agent reçoit un message, puis le transmet.

- S'il y a un problème temporaire de transmission, le message est mis en attente.
- Si erreur irrécupérable, envoi d'un NACK
- Sur arrivée à l'UA (ou sur lecture par le destinataire), possibilité envoi d'un accusé de réception / de remise

## Protocoles UA – MTA

- X400 : protocoles normalisés P3 (UA-MTA) et P2 (UA-UA)
- IP – en émission : SMTP
  - en réception : boîte locale, boîte partagée par NFS, POP (RFC1225), IMAP (RFC1176)

## Protocoles MTA – MTA

- X400 : utilise le protocole normalisé P1
- UUCP (Unix to Unix Copy) – historique, plus très utilisé

Protocole de copie de fichiers sur ligne asynchrone, X25, ... sert aussi au transfert de fichiers et à l'exécution distante

Caractéristiques : réseau commuté (un modem ou X25 suffit), possibilité de transfert par lots

## • SMTP (Simple Mail Transfert Protocol, RFC821)

Protocole sur TCP – commande/réponse

Texte 7 bits NVT-ASCII, ligne à ligne (CR/LF) -

```
220-mailhost.mondomaine.fr Sendmail 8.6.12/8.6.12 ready at Mon,  
30 Oct 1995 15:38:46 +0100
```

```
220 ESMTP spoken here
```

```
HELO client.autredomaine.fr
```

```
250-mailhost.mondomaine.fr Hello client.autredomaine.fr  
[193.55.0.1], pleased to meet you
```

```
MAIL FROM: <alain.durand@autredomaine.fr>
```

```
250 alain.durand@autredomaine.fr... Sender ok
```

```
RCPT TO: <jean.dupont@mondomaine.fr>
```

```
250 jean.dupont@mondomaine.fr... Recipient ok
```

```
DATA
```

```
354 Enter mail, end with "." on a line by itself
```

```
from: <alain.durand@autredomaine.fr>
```

```
to: <jean.dupont@mondomaine.fr>
```

```
subject: test
```

```
sdfsdfs
```

```
...
```

```
.
```

```
250 PAA02281 Message accepted for delivery
```

```
QUIT
```

```
221 mailhost.mondomaine.fr closing connection
```

## • ESMTP (Extended SMTP, RFC1651)

Extension de SMTP (RFC 1651) supportant le 8 bit.

Le client utilise le mot clef EHLO à la place de HELO. Si le serveur est OK, il répond la liste des services supportés.

```
250-mailhost.mondomaine.fr Hello client.autredomaine.fr [193.55.0.1], pleased to meet you
```

```
250-EXPN
```

```
250-SIZE
```

```
250-8BITMIME
```

```
250 HELP
```

Si l'option 8BITMIME est supportée par le serveur, le client peut l'utiliser dans le champ RCPT-TO: pour indiquer que la suite est du 8 bits. Le message transmis doit avoir des en-têtes MIME valides.

```
RCPT TO: <jean.dupont@mondomaine.fr> BODY=8BITMIME
```

```
250 <jean.dupont@mondomaine.fr>... Sender and 8BITMIME ok
```

# Protocoles Internet UA-MTA

- En émission : En général utilisation de SMTP  
Mais problèmes sécurité / identification  
→ Proposition de protocoles spécifiques - ou d'extensions de POP/IMAP - ou ajout de sécurité (tls et certificats expéditeur) dans sendmail
- En réception : deux protocoles sur TCP
  - POP : récupérer des messages sur un serveur (transférés sur le client et détruits sur le serveur)  
Protocole de type ftp : get message, list headers, peek  
Problème : synchronisation de lecteurs multiples
  - IMAP (IMAP2, maintenant IMAP4) : consulter/gérer une boîte aux lettres sur un serveur. Gestion de dossiers, ...  
Intérêt/défaut : serveur centralisé (gestion plus sûre, permet des sauvegardes de site ...)

## Sécurité en réception

- identification : mot de passe (à chaque session ou stocké dans client), ou par défi/réponse ou certificat (TLS)
- connexion chiffrée (certificat serveur) : pop/s et imap/s
- Utilisation de VPN (virtual channel) : tunnels codés - permet à un poste distant d'être traité comme si il est à l'intérieur du domaine

# Sécurisation

- UA → MTA : besoin d'identifier l'émetteur ~ login  
Utilisation de protocole de sécurisation session (TLS) - pas encore dans tous les UA  
ou via intégration à pop/imap (expérimental)  
+ vérification nom/adresse machine source / émetteur
- MTA → MTA : contrôler les ressources domaine  
→ règles de filtrage : vérifier id appelant, pas de relais (le MTA imag n'accepte que les courriers pour ou depuis des adresses imag – ou 'voisines : inrialpes, ...), liste de sites blacklistés, greylisting  
⇒ interdiction des relais de MTA (suppose Internet de bout en bout), blacklistage des « relais ouverts » (peut poser des problèmes avec le forward automatique)
- MTA/MS → UA : comme ci-dessus + besoin d'identifier le lecteur : TLS (certificat utilisateur) ou mot de passe sur SSL (imap/s ou pop/s avec certificat serveur)
- + analyse des messages : anti-virus, spamassassin ...

## Sécurité de bout en bout : PEM et PGP

Les messages contiennent un champ signature (lié au From:) et/ou sont chiffrés (lié au To:) Permet de sécuriser d'UA à UA.

Deux formats : S/MIME et PGP (extension ad hoc de MIME)

Distribution des clés :

- PGP : "groupes de confiance" sans autorité globale
- S/MIME : via certificats délivrés par une autorité de certification - validation par connaissance du certificat de l'autorité

# Adresses

Doit permettre d'identifier l'expéditeur et les destinataires  
Devrait être géré par un annuaire (X500, LDAP)

Utilisées dans les en-têtes et en général aussi dans une **enveloppe** extérieure.

La syntaxe dépend du protocole de transport utilisé

- Adressage UUCP - relatif : donne le chemin d'accès par rapport au site courant (modifié à chaque transfert)

user

host1!host2!host3!host4!user

- Adressage X400

Structure hiérarchisée – pays, administration, domaine ...  
syntaxe avec attributs

/C=fr; ADMD=atlas400; PRMD=inria; S=Dupont; .../

On peut aussi utiliser une adresse téléphonique

X121=93399383800

- Adressage Internet - absolu : nom et site (RFC822)

– Au début : site = machine, nom = nom login

richier@imag (imag : nom mondialement connu)

puis richier@vercors.imag.fr (machine & domaine)

– Evolution : site = domaine e.g. richier@Imag.Fr

– Aujourd'hui : nom = identificateur normalisé

e.g. Pierre.Laforgue@Imag.Fr

– pseudo-domaines : nom@frulm97.bitnet

– adressage dirigé : nom%vax1.decnet@imag.fr

(→ imag, puis nom@vax1.decnet)

(n'est plus très utilisé, car gêne les filtrages)

# Routage applicatif

Le problème : trouver le MTA auquel transmettre un message

– UA : Chaque UA connaît “son” MTA : par configuration, ou dérivé du nom de domaine DNS (prendre le MX)

– MTA : Le choix du MTA “suivant” dépend du type de messagerie, de la structure de l’adresse (route explicite ou non), et de la configuration réseau

- **UUCP** (un des premiers protocoles, obsolète)

Routage explicite dans l’adresse (et recalcul).

Extensions : le chemin peut être incomplet : les machines “importantes” se connaissent (carte mondiale), et savent router entre elles (e.g. uunet!imag!vercors!richier)

→ architecture maillée avec concentrateurs (les “backbones”)

- **X400**

Routage statique chez l’opérateur.

On utilise la structure hiérarchique ADMD-PRMD : chaque niveau possède des “WEP” (Well-known Entry Point).

Routage : PRMD-WEP→ADMD-WEP→ADMD-WEP→PRMD-WEP

En fait pas de solution universelle (suppose que tous les ADMD se connaissent, et pas encore de réseau mondial unique et intégré)

Utilisation des annuaires X500 nécessaire, prévue (?) – et vivement attendue

## • SMTP

Plusieurs possibilités :

– routage limité : le MTA connaît un relais défaut

Intérêt : configuration simplifiée

En pratique, ne sert que qu'à l'intérieur d'un domaine, pour traitement centralisé (e.g. réécriture d'adresses), politique de firewall ...

– routage par table : le MTA calcule un relais, final ou intermédiaire.

En pratique déconseillé : entraine des transferts multi-hop qui sont de facto interdits

– direct par DNS : adresser directement le MTA final

– utilisation des enregistrements A (déconseillé/obsolète)

site → nom machine → adresse IP via DNS

– utilisation des enregistrements MX (solution actuelle)

site → nom → (nom internet et préférence)\*

envoi à une des machines *accessible* de plus *petite*

“préférence” (priorité plus grande)

On cherche dans la liste des noms, en commençant par la plus petite préférence (et en testant *toutes* les adresses associées). Si plusieurs machines ont la même préférence, on les teste dans un ordre *arbitraire*.

Note : il y a un risque de bouclage ⇒ suppression des MX de préférence  $\geq$  celle de l'émetteur s'il est dans la liste.

Par exemple, dans l'exemple ci-dessous, pour un courrier à xx@imag.fr on teste ms.imag.fr ou ms2.imag.fr, puis l'autre, et enfin ms-ext.imag.fr, et pas 135.1.2.1.

Si on est sur ms-ext.imag.fr on n'essaie que ms ou ms2, et sur ms on ne peut utiliser les MX (donc message en erreur)

- Mélange de MX et A. Possible mais déconseillé en pratique. En fait les RFC disent que s'il n'y a pas de MX et s'il y a un A, il faut envoyer à l'adresse définie par le A. C'est à déconseiller : ne pas activer de MTA accessible de l'extérieur sur les machines non serveur mail, et contrôler les noms de domaines des mails reçus.
- Configuration conseillée de la zone DNS
  - Mettre des MX sur les noms de zone mail
  - Ne pas mettre des MX sur les noms de machines
  - Ne pas utiliser de wilcard '\*'

### Exemple

```
imag.fr.      IN  MX    40 ms.imag.fr.
              IN  MX    40 ms2.imag.fr.
              IN  MX    80 ms-ext.imag.fr.
              IN  A     135.1.2.1 ; inutile pour mail
dir.imag.fr.  IN  MX    40 ms.imag.fr.
serv.imag.fr. IN  A     135.1.2.5
host.imag.fr. IN  A     135.1.2.3
```

; déconseillé car suppose une adresse mail non normalisée

```
host.imag.fr. IN  MX    40 ms1.imag.fr.
              IN  MX    40 ms2.imag.fr.
              IN  MX    80 ms-ext.imag.fr.
```

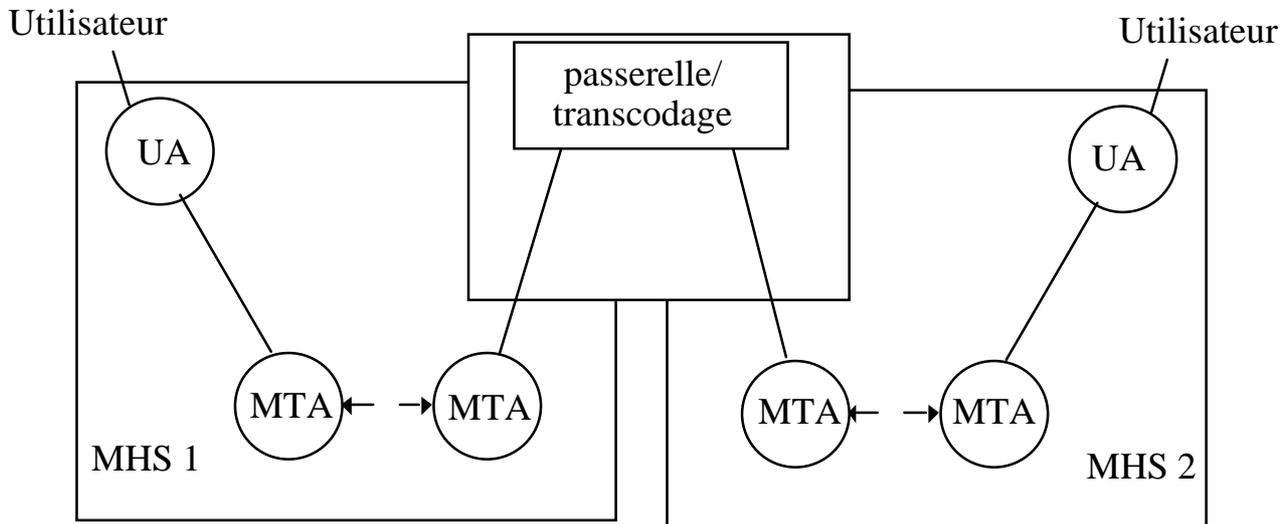
; déconseillé car ne s'applique que pas si autre enregistrement

```
*.imag.fr.   IN  MX    40 ms.imag.fr. ; déconseillé
```

- la ligne IN A 135.1.2.1 est déconseillée car tous les MTA distants devraient être en « MX »
- les lignes MX pour host.imag.fr sont déconseillées car elles ne servent que pour un mail @host.imag.fr non normalisé
- la ligne \*.imag.fr est déconseillée car elle répond pour tous les xx.imag.fr *sauf* host, dir et serv

# Passerelles inter-protocoles

Entre deux systèmes de messagerie, utilisation d'une passerelle



- Gestion des passerelles. Peut être :

- locale (e.g. point d'entrée SMTP et UUCP du site imag)
- globale au domaine (e.g. passerelle BITNET<-> SMTP)
- régionale (e.g. WEP entre X400 et SMTP)

NB : En pratique 2 et 3 ne s'utilisent plus.

- Accès à la passerelle

- soit explicitement par adressage dirigé du courrier :

nom%site@bitnet-relay.edu

- soit implicitement par le routage

- tables de routages explicites,
- réécriture d'adresse dans sendmail.cf,
- pseudo-domaines : MX record pour \*.bitnet – ou pour *chaque* machine nom.bitnet

- Actions réalisées par la passerelle

- Transcodage du corps (par exemple ASCII<->EBCDIC)

Pas encore de normes —la structuration (Mime) peut aider

- Transcodage des en-têtes – plus ou moins normalisé (e.g. RFC1327 entre X400 et SMTP) :

- Champs semblables (From:, Subject:...) : copiés
- Champs sans équivalent (Keywords: Encrypted:) : ajoutés au corps du message

- Transcodage des adresses – plus ou moins normalisé

- Pour SMTP utilisation des pseudo-domaines

nom RFC822 du site bitnet FRCIME51 : frcime51.bitnet

- Pour UUCP déclaration de machines internet dans les tables e.g. déclaration de imag ≡ imag.imag.fr

- Pour X400 : domaines <-> ADMD et PRMD :

foo@bar.edu <=> C=us;ADMD=rfc822:PRMD=edu;O=bar;S=foo

foo@sunir.reunir.fr <=> C=fr;ADMD=atlas:PRMD=reunir;O=sunir;S=foo

(ou notation “dirigée” : foo/O=sunir/@reunir.fr)

Cas de Unix : le routage peut être fait par le démon sendmail, piloté par des tables (fichier sendmail.cf) – pour tous les protocoles de messagerie connus

Plusieurs configurations passerelle possibles :

- tables définissant les noms connus + relais vers l’extérieur

- utilisation du serveur de noms — MX records

# Gestion du courrier sur un domaine

## Choix de la messagerie

- SMTP — MIME
- Autres solutions obsolètes : X400, messagerie privée + une passerelle vers l'extérieur

## Choix d'un produit

- logiciel commercial MS-mail, Novell, Outlook ...
- logiciels « gratuit » (e.g. MTA sendmail, UA thunderbird, ...)

## Architecture générale du domaine

- **Nommage** nom@... (critère : pas d'annuaire vs risque spam)
  - login
  - P.I.N (code 4 lettres, numéro,...)
  - Prenom.Nom
- **Adressage** @domaine (critère : noms stables et gérables)
  - un seul domaine
  - plusieurs domaines
  - un domaine et des sous-domaines
- **Routage** Solution la plus classique
  - une passerelle de messagerie pour l'extérieur et routage direct dans le domaine – les autres machines relaient vers la passerelle
  - un serveur/passerelle de messagerie pour TOUS les courriers
  - prévoir de la redondance
- DNS, choix des MX
  - pour le domaine et les sous-domaines
  - déclarer des MX pour chaque station ? (à éviter)
  - wild-card MX pour tout un domaine ? (à éviter)
- Liens vers l'extérieur
  - connectivité IP globale
  - sinon : lien vers sous-traitance du courrier à l'extérieur

- Passerelles – si nécessaire, au cas par cas, en interne
- Livraison du courrier
  - Quelles machines doivent recevoir du courrier ?
  - toutes ?
  - les machines administrées correctement ?
  - quelques serveurs ?
  - Comment traiter celles qui n'en reçoivent pas ?
  - Comment traiter les Mac, PC, ... ?
  - Quelles machines peuvent envoyer du courrier ?
  - Utiliser une interface Webmail ? Évite les UA, facile d'accès à l'extérieur, mais lourd. Conseillé, mais en secours
- Partage du courrier
  - Lecture sur une machine unique
  - Montage NFS ? (déconseillé)
  - Serveur POP/IMAP
- MTA/ cas du plus classique Sendmail
  - Choix d'un "sendmail" : distribution constructeur ?
  - critères: récent, hooks pour anti-spam, anti-virus, LDAP...
  - Besoin d'une conf (sendmail.cf – génération à partir de prototypes .mc (par m4)

Exemple: sendmail.mc d'une passerelle

```
divert(-1)
include(`../m4/cf.m4')
VERSIONID(`@(#)tcpproto.mc 8.2 (Berkeley) 8/21/93')
define(`confLOG_LEVEL',10)dnl
define(`confAUTO_REBUILD',`True')dnl
define(`confCOPY_ERRORS_TO',`postmaster')dnl
define(`confSMTP_MAILER',esmtpdnl
define(`UUCP_RELAY',`chenas.inria.fr')dnl
define(`BITNET_RELAY',`frmop11.cnusc.fr')dnl
define(`SMTP_MAILER_MAX',`2000000')dnl
OSTYPE(sunos4.1)dnl
FEATURE(nouucp)
FEATURE(notsticky)
FEATURE(always_add_domain)
MAILER(local)
MAILER(smtp)
MASQUERADE_AS(inria.fr)
LOCAL_CONFIG
```

```
Cwinria.fr
Kgenerics hash /usr/lib/mail/generics
LOCAL_NET_CONFIG
R$*<@$.m.>$* $#esmtplib $@ [$2.m] $: $1<@$2.m.>$3
LOCAL_RULE_1
R$*<@$.>$* @$$(generics $1@$2 $:$1<@$2.>$3 $)
R$+ @$$(generics $1 $:$1 $)
```

## Exemple: sendmail.mc d'un client

```
divert(-1)
include(`../m4/cf.m4')
VERSIONID(`@(#)tcpproto.mc 8.2 (Berkeley) 8/21/93')
define(`confLOG_LEVEL',10)dnl
define(`confAUTO_REBUILD',`True')dnl
define(`confCOPY_ERRORS_TO',`postmaster')dnl
define(`SMART_HOST',esmtplib:concorde.inria.fr)dnl
define(`confSMTP_MAILER',esmtplib)dnl
define(`SMTP_MAILER_MAX', `2000000')dnl
OSTYPE(hpux)dnl
FEATURE(nouucp)
FEATURE(notsticky)
FEATURE(always_add_domain)
MAILER(local)
MAILER(smtp)
```

## Exemples de solutions

- UA RFC822: Mail (Unix BSD), mailx (Unix SV), xmh (X11), mush, Mailtool (Sun) ...
- UA Mime: exmh, Zmail, Eudora (Macintosh/PC), mutt, outlook, thunderbird, seamonkey ...
- ou alors hébergement chez fournisseurs : du MTA ou de tour le courrier – avec interface WWW : gmail, CompuServe, AOL, La poste, Microsoft network, ...

# Autres systèmes de communication

## News

Système de conférence distribuée, organisé en groupes de discussion hiérarchisés. Au total 3000 “newsgroups” à ouverture totale ou “modérée”.

- Distribution modulable : laboratoire, France, Europe, monde
- Classement thématique — E.g. comp.ai, comp.arch, comp.lang.pascal, comp.sys.ibm.pc, comp.windows.x, fnet.culture, rec.music.synth, sci.astro, ...
- Structure d'accès à 2 niveaux
  - des serveurs qui stockent les messages et les redistribuent entre eux ( $\geq 2$  GØ si conservation de 5 jours à un mois)  
Protocoles utilisés : UUCP ou NNTP
  - des clients qui consultent et créent les messages  
Protocole utilisé : NNTP, ou accès par NFS à une base locale

## Listes de diffusion

Fournit : filtrages messages, tests anti-bouclage, archive, gestion inscriptions automatisée ...

Logiciels : majordomo, tulp, listserv, sympa

## Mail servers

## WWW

Agents WEB : WEBmail (sécurité par http/s)

Flux RSS

Sites d'échange, WiKi, ...