

# SIMPLE MAIL TRANSFER PROTOCOL

Network Working Group  
 Request for Comments: DRAFT  
 Remplace: RFC 788, 780, 772  
 Traduction :

J. Postel / ISI  
 August 1982  
 V. Fremaux / EISTI

## 1. INTRODUCTION

Le but du protocole de courrier Simple Mail Transfer Protocol (SMTP) est de transférer du courrier électronique selon un procédé efficace et fiable.

SMTP est indépendant de tout sous-système de transmission et ne nécessite qu'un canal de transmission fiable et ordonné. Les Appendices A, B, C, et D décrivent l'utilisation de SMTP sur diverses couches transport.

Une fonctionnalité importante de SMTP est sa capacité à relayer des courriers dans des environnements de service de transport. Un service de transport dispose d'un environnement de communication inter-processus (IPCE). Un IPCE peut avoir une portée limitée à un seul réseau, étendue à plusieurs réseaux, ou réduite à un sous ensemble d'un réseau. Il est fondamental de comprendre que les environnements de communication inter-processus (ou IPCE) ne sont pas nécessairement corrélés à un réseau. Un processus peut communiquer directement avec un autre processus à travers toute paire d'IPCE se reconnaissant l'une l'autre. Le courrier électronique est une application, ou simplement une utilisation des mécanismes de communication inter-processus. Le courrier peut être transmis entre deux processus par différents IPCE et éventuellement en étant relayé par un autre processus connecté à au moins deux (ou plus) IPCE. Plus généralement, le courrier peut être relayé entre hôtes sur des systèmes de transport différents.

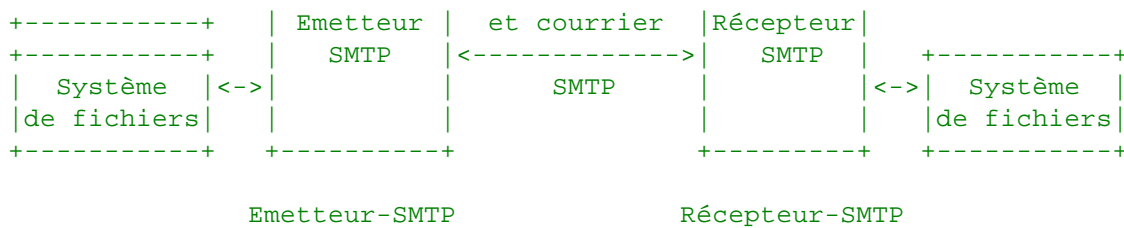
## 2. LE MODELE SMTP

Le design de SMTP est basé sur le modèle suivant de communication : suite à une requête de l'utilisateur du courrier user, L'émetteur SMTP établit une communication bidirectionnelle vers un récepteur-SMTP. Celui-ci peut être soit la destination finale, soit seulement un intermédiaire. Les commandes SMTP sont générées par l'émetteur SMTP et sont émises vers le récepteur SMTP. Les réponses SMTP sont envoyées par le récepteur-SMTP à l'émetteur-SMTP en réponse aux commandes.

Une fois le canal de transmission établi, l'émetteur SMTP envoie une commande MAIL mentionnant l'émetteur d'un courrier. Si le récepteur SMTP peut accepter le courrier, il répondra par un message OK. L'émetteur-SMTP envoie alors une commande RCPT identifiant un récipiendaire pour ce courrier. Si le récepteur-SMTP peut accepter un courrier pour ce récipiendaire, alors il répondra par un message OK ; sinon, il répond par un message refusant le courrier pour ce récipiendaire (mais n'annulant totalement pas la transaction de courrier). L'émetteur SMTP et le récepteur SMTP pourront négocier plusieurs récipiendaires. Une fois cette négociation effectuée, l'émetteur SMTP envoie le contenu du courrier, en le terminant par une séquence spéciale. Si le récepteur SMTP traite avec succès la réception du contenu du courrier, il répondra par un message OK. Le dialogue est volontairement un dialogue pas à pas, à étapes verrouillées.

### *Illustration 1 / Modèle d'utilisation de SMTP*





SMTP procure un mécanisme de transmission des courriers, directement à partir de l'hôte de l'émetteur du message jusqu'à l'hôte du récipiendaire pour autant que les deux hôtes soient raccordée au même service de transport, ou à travers une chaîne de relais SMTP (serveurs) lorsque les hôtes source et destination ne sont pas raccordés au même service de transport.

Pour pouvoir officier en temps que relais, le serveur SMTP devra connaître le nom de l'hôte du destinataire ainsi que le nom de la boîte aux lettres du récipiendaire. L'argument associé à la commande MAIL est le chemin inverse, qui spécifie qui est l'émetteur du courrier. L'argument associé à la commande RCPT est un chemin direct, qui spécifie à qui est destiné le courrier. Le chemin direct sert à l'acheminement, tandis que le chemin inverse est utilisé pour renvoyer un message d'erreur à l'émetteur (par exemple émis par un relais).

Lorsque le même message est émis à destination de plusieurs récipiendaires, SMTP encourage la transmission d'une seule copie du contenu du message pour tous les récipiendaires hébergés sur le même hôte destinataire.

Les commandes et réponses du système de courrier suivent une syntaxe rigide. Les réponses sont aussi exprimées sous forme de code numérique. Dans ce qui suit, les exemples utilisent des commandes et réponses usuelles. La liste complète des commandes et réponses est donnée en section 4 traitant des "spécifications".

Les commandes et réponses ne tiennent pas compte de la casse. C'est-à-dire, qu'une commande ou réponse peut être écrite en majuscules, minuscules, ou tout mélange des deux. Notez que ceci n'est pas vrai pour les noms des utilisateurs des boîtes aux lettres. En effets, cette faculté dépend du système d'exploitation sur lequel sont implantées ces boîtes aux lettres, et les implémentations de SMTP devront donc veiller à conserver la casse des noms d'utilisateurs telle qu'ils ont été écrits dans l'argument définissant la boîte aux lettres. Les noms d'hôtes, eux, ne dépendent pas de la casse.

Les commandes et réponses sont composées de caractères ASCII [1]. Lorsque le service de transport utilisé permet la transmission sur une base de 8-bits (octet), tous les caractères 7-bits sont transmis justifiés à droite (dans l'octet de transmission), le huitième bit étant toujours forcé à 0.

Lorsque nous spécifierons la forme syntaxique générale des commandes et des réponses, un argument (ou un symbole "spécial") sera exprimé sous forme d'une variable (ou constante) "métalinguistique", par exemple, "<chaîne>" ou "<route-inverse>". Ici, les signes "inférieur à" et "supérieur à" indiquent le caractère métalinguistique de la variable. Cependant, certains arguments utilisent ces signes dans leur expression littérale. Par exemple, un chemin inverse est actuellement encapsulé par ces signes, c'est-à-dire, "<John.Smith@USC-ISI.ARPA>" est une instance de la variable générale <route-inverse> (les signes "inférieur à" et "supérieur à" sont effectivement transmis dans la commande ou la réponse).

### 3. LES PROCEDURES SMTP

Cette section présente en plusieurs étapes les procédures utilisées par SMTP. Tout d'abord est analysée la procédure de base définie comme une transaction de transfert de courrier. Suite à ceci seront décrits la façon de rémettre un courrier, vérifier des noms de boîtes aux lettres et expander des listes de diffusion, d'émettre vers de terminaux plutôt que (ou conjointement) vers des boîtes aux lettres, ainsi que les procédures d'ouverture et de fermeture des échanges. Vous trouverez à la fin de cette section des commentaires sur le principe de relais, une note sur les domaines de courrier, et une discussion sur l'échange des rôles. Tout au long de cette section seront donnés des exemples de séquences partielles de commandes et réponses, des scénarios complets de transactions sont proposés dans l'Appendice F.

### 3.1. Emission de courrier

Une transaction de courrier SMTP se déroule en trois étapes. La transaction est initiée par une commande MAIL donnant l'identification de l'émetteur. Une série contenant une ou plusieurs commandes RCPT suit, donnant les informations sur les destinataires. Puis une commande DATA passe le contenu du message. Dans cette troisième phase, la marque de fin de données à transmettre marque la fin de la transaction.

La première étape de la procédure est donc la commande MAIL. L'argument <route-inverse> contient le nom de la boîte aux lettres de l'émetteur.

```
MAIL <SP> FROM:<reverse-path> <CRLF>
```

Cette commande indique au récepteur SMTP qu'une nouvelle transaction de courrier débute et lui demande de réinitialiser ses tables d'états et ses tampons, y compris ses boîtes de réception et toute donnée de courrier latente. Elle lui donne le chemin inverse qui pourra être utilisé en cas de rapport d'erreur à transmettre. Si l'émetteur accepte la commande, un code 250 OK est renvoyé à l'émetteur.

L'argument <route-inverse> peut contenir plus d'une boîte aux lettres. On peut y inscrire une liste de plusieurs boîtes aux lettres dans des hôtes différents. La première boîte inscrite dans l'argument <route-inverse> doit néanmoins être celle désignant l'émetteur du message.

La deuxième étape de la procédure est l'émission des commandes RCPT.

```
RCPT <SP> TO:<cheminDirect> <CRLF>
```

Cette commande transmet une adresse de courrier désignant un récipiendaire. Si elle est acceptée, le récepteur SMTP renvoie une réponse de code "250 OK", et mémorise le chemin d'acheminement. Si le récipiendaire est inconnu, le récepteur SMTP renvoie un code d'erreur "550 Failure". Cette seconde étape de la procédure peut être répétée autant de fois que nécessaire.

L'argument <route-directe> peut contenir plus d'une adresse de boîte aux lettres. On peut y inscrire une liste de boîtes aux lettres dans des hôtes différents. Le premier hôte à être mentionné dans l'argument <route-directe> sera l'hôte à qui est envoyé cette commande.

La troisième étape consiste en l'émission de la commande DATA.

```
DATA <CRLF>
```

Si elle est acceptée, le récepteur SMTP renvoie une réponse de code "354 Intermediate" et prend en compte toutes les lignes de texte suivantes comme étant le texte du message. Lorsque la marque indiquant la fin du texte est reçue et enregistrée, le récepteur SMTP envoie une réponse de code "250 OK".

Comme les données du courrier sont émises sur le même canal de transmission, il faudra donner un indicateur de fin de données pour que le dialogue requête-réponse puisse reprendre. SMTP indique la fin des données du message en envoyant une ligne contenant un point unique. Une procédure de "transparence" est utilisée pour éviter que celle-ci n'interfère avec le texte de l'utilisateur (voir Section 4.5.2). Notez que les données du message incluent un résumé d'en-tête comprenant les champs tels que Date, Subject, To, Cc, From [2].

L'indicateur de fin de message confirme en outre la transaction de courrier et signale au récepteur-SMTP qu'il peut désormais traiter les récipiendaires enregistrés pour ce message. Si les données sont acceptées, le récepteur-SMTP renvoie une réponse de code "250 OK". La commande DATA ne doit échouer que si la transaction de courrier s'avère incomplète (par exemple, pas de récipiendaires), ou si les ressources suffisantes pour traiter le courrier ne sont pas disponibles.

La procédure qui suit est un exemple de transaction de courrier. Ces commandes ne doivent être employées que dans l'ordre précisé ci-avant. L'exemple 1 (ci-dessous) illustre l'utilisation de ces commandes dans une transaction.

#### **Exemple 1 / Exemple de procédure SMTP**

Cet exemple de séquence SMTP montre un courrier envoyé par Smith, utilisateur de l'hôte ARPA Alpha, à Jones, Green, et Brown utilisateurs de l'hôte ARPA Beta. Nous supposons que l'hôte Alpha contacte l'hôte Beta directement.

```
S: MAIL FROM:<Smith@Alpha.ARPA>
R: 250 OK

S: RCPT TO:<Jones@Beta.ARPA>
R: 250 OK

S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: <CRLF>.<CRLF>
R: 250 OK
```

Le courrier a été accepté pour Jones et Brown. Green n'avait pas de boîte aux lettres sur Beta.

### **3.2. Réémissions (ou redirections)**

Il existe certains cas où l'information concernant un destinataire donnée dans la <route-directe> est incorrecte, mais le récepteur-SMTP connaît la destination exacte. Dans un tel cas, l'une des réponses suivantes pourra être émise pour permettre à l'émetteur de contacter la bonne destination.

#### **251 User not local; will forward to <route-directe>**

Cette réponse indique que le récepteur-SMTP sait que la boîte-aux-lettres du destinataire est hébergée sur un autre hôte et indique le chemin d'accès correct de cette boîte aux lettres pour des messages futurs. Notez que l'hôte, ou l'utilisateur ou les deux peuvent être différents de ceux de la requête originale. Le récepteur prend dans ce cas la responsabilité de délivrer le message actuel à la bonne destination.

#### **551 User not local; please try <route-directe>**

Cette réponse indique que le récepteur-SMTP sait que la boîte-aux-lettres du destinataire est sur un autre hôte et signale le chemin d'accès correct à utiliser pour joindre cette boîte-aux-lettres. Notez que l'hôte, ou l'utilisateur ou les deux peuvent être différents de ceux de la requête originale. Dans ce cas, le récepteur refuse d'accepter des messages pour cet utilisateur, et l'émetteur devra soit rediriger explicitement son courrier conformément à l'information fournie ou arrêter la transaction et retourner un message d'erreur à son utilisateur.

L'exemple 2 illustre l'utilisation de ces réponses.

#### **Exemple 2 / Exemple de réémission**

Soit

```
S: RCPT TO:<Postel@USC-ISI.ARPA>
R: 251 User not local; will forward to <Postel@USC-ISIF.ARPA>
```

Ou

```
S: RCPT TO:<Paul@USC-ISIB.ARPA>
R: 551 User not local; please try <Mockapetris@USC-ISIF.ARPA>
```

### **3.3. VERIFICATION DE BOITES ET EXPANSION DE LISTE DE DIFFUSION**

SMTP propose au titre de fonctionnalités additionnelles, des commandes pour vérifier un nom de destinataire ou pour élargir une liste de diffusion. Ces deux opérations peuvent être menées respectivement avec les commandes VRFY et EXPN, qui acceptent toutes deux un argument sous forme de chaîne de caractères. Pour la commande VRFY, la chaîne en argument est un nom d'utilisateur, la réponse à cette commande pouvant inclure le nom complet de cet utilisateur et devant fournir une adresse complètement qualifiée de boîte-aux-lettres pour cet utilisateur. Pour la commande EXPN, la chaîne en argument identifie une liste de diffusion, la réponse multilignes à cette commande pouvant inclure le nom complet des utilisateurs et DEVANT inclure l'ensemble des boîtes-aux-lettres contenues dans la liste.

La sémantique de l'expression "nom d'utilisateur" est assez floue, cette expression étant employée en parfaite connaissance de cause. Si un hôte implémente la commande VRFY ou EXPN, alors on suppose que l'hôte reconnaît au moins les boîtes-aux-lettres locales en tant que "noms d'utilisateur". Mais un hôte peut utiliser une définition toute autre pour les "noms" de ses utilisateurs.

Sur certains hôtes la distinction entre une liste de diffusion et une série d'alias pour une boîte-aux-lettres unique n'est pas très claire, dans la mesure où les structures de données standard peuvent accueillir ces deux types d'entrées, et qu'il est possible de constituer des listes de diffusion réduites à une seule boîte-aux-lettres. Lorsqu'une requête d'expansion d'une liste de diffusion est lancée, une réponse positive peut être renvoyée si, lorsqu'un message est reçu pour cette liste, ce dernier est transmis simultanément à tous les participants inscrits dans cette liste. Dans tout autre cas, un message d'erreur devra être renvoyé (ex., "550 Ceci est un utilisateur, pas une liste de diffusion"). Lorsqu'une requête de vérification d'un utilisateur est lancée, une réponse positive pourra être donnée si la réponse peut être formée d'une liste ne contenant qu'un seul nom. Dans tout autre cas une erreur devra être renvoyée (ex., "550 Ceci est une liste de diffusion, pas une boîte aux lettres").

Dans le cas d'une réponse multilignes (réponse courante à une commande EXPN), chaque ligne ne doit spécifier qu'une boîte-aux-lettres et une seule. Dans le cas d'une requête ambiguë, par exemple, "VRFY Dupont", sur un hôte où deux personnes nommées Dupont sont hébergées, la réponse devra être de type "553 utilisateur ambigu". L'exemple 3 illustre l'utilisation de la commande VRFY.

### **Exemple 3 / Exemple de Vérification d'un Nom d'Utilisateur**

Soit

```
S: VRFY Smith  
R: 250 Fred Smith <Smith@USC-ISIF.ARPA>
```

Ou

```
S: VRFY Smith  
R: 251 User not local; will forward to <Smith@USC-ISIQ.ARPA>
```

Ou

```
S: VRFY Jones  
R: 550 String does not match anything.
```

Ou

```
S: VRFY Jones  
R: 551 User not local; please try <Jones@USC-ISIQ.ARPA>
```

Ou

```
S: VRFY Gourzenkyinplatz  
R: 553 User ambiguous.
```

L'exemple 4 illustre quant à lui l'expansion de liste de diffusion.

### **Exemple 4 / Exemple d'expansion d'une liste de messagerie**

Soit

```
S: EXPN Example-People  
R: 250-Jon Postel <Postel@USC-ISIF.ARPA>  
R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>  
R: 250-Sam Q. Smith <SQSmith@USC-ISIQ.ARPA>  
R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>  
R: 250-<joe@foo-unix.ARPA>  
R: 250 <xyz@bar-unix.ARPA>
```

Ou

```
S: EXPN Executive-Washroom-List  
R: 550 Access Denied to You.
```

L'argument en chaîne de caractères des commandes VRFY et EXPN ne peut ici être plus "cadré" du fait de la différence d'implémentation des concepts de noms d'utilisateurs et de listes de boîtes-aux-lettres suivant les plates-formes. Sur certains systèmes, l'argument d'une commande EXPN pourrait être de façon tout à fait appropriée le nom d'un fichier contenant la liste de diffusion, mais encore à ce titre, il existe de multiples conventions de nommage pour les fichiers dans le monde Internet.

Les commandes VRFY et EXPN ne font pas partie de l'implémentation minimale de SMTP (voir Section 4.5.1). En outre, lorsqu'elles sont implémentées, aucune obligation n'est faite qu'elles sachent fonctionner au travers des différents relais de courrier.

### 3.4. EMISSION ET POSTAGE

Le but principal de SMTP est de délivrer des messages dans une boîte-aux-lettres d'un utilisateur. Un service assez similaire existant sur certains hôtes consiste à délivrer les messages directement sur le terminal de l'utilisateur (pourvu que cet utilisateur soit actuellement "loggé" sur l'hôte en question). Le routage d'un message vers une boîte-aux-lettres d'un utilisateur est appelé "postage", l'envoi du même message sur le terminal de l'utilisateur est appelé "transfert". Du fait que, dans de nombreux hôtes, l'implémentation du transfert est quasiment identique à celle du postage, ces deux fonctions ont été combinées dans SMTP. Toutefois, les commandes de transfert n'ont pas été reportées dans la définition de l'implémentation minimale (voir Section 4.5.1). Les utilisateurs devront avoir dans tous les cas la possibilité de contrôler l'inscription des messages provenant de transferts sur leur écran. La plupart des hôtes ont une fonction qui permettent de désactiver l'écriture de ces messages.

Les trois commandes qui suivent sont définies dans le cadre de ce fonctionnement optionnel en mode transfert. Elles sont utilisées dans la transaction en lieu et place de la commande MAIL et informent le récepteur-SMTP qu'une interprétation particulière doit être fait pour cette transaction :

```
SEND <SP> FROM:<reverse-path> <CRLF>
```

La commande SEND demande que le message soit affiché sur le terminal de l'utilisateur. Si l'utilisateur n'est pas connecté au système (ou n'accepte pas l'affichage interactif des messages sur son terminal), une réponse de code 450 sera renvoyé en réponse à la commande RCPT. La transaction est considérée comme aboutie lorsque le message est affiché sur le terminal.

```
SOML <SP> FROM:<reverse-path> <CRLF>
```

La commande Send Or Mail demande que le message soit affiché sur le terminal utilisateur si ce dernier est connecté au système (et accepte l'affichage interactif de messages sur ce terminal). Dans le cas contraire, le message doit être redirigé vers sa boîte-aux-lettres. La transaction est considérée comme aboutie si le message est soit affiché sur le terminal, soit déposé dans la boîte-aux-lettres.

```
SAML <SP> FROM:<reverse-path> <CRLF>
```

La commande Send And Mail demande que le message soit affiché sur le terminal utilisateur si ce dernier est connecté au système (et accepte l'affichage interactif de messages sur ce terminal). Dans tous les cas le message doit être copié dans sa boîte-aux-lettres. La transaction est considérée comme aboutie si le message a au moins pu être copié dans la boîte-aux-lettres. Les réponses à ces commandes sont les mêmes que pour la commande MAIL.

### 3.5. OUVERTURE ET FERMETURE DE LIAISON

Dès que le canal de transmission est établi, un échange de protocole s'assure que l'hôte demandeur communique bien avec l'hôte attendu. Les deux commandes suivantes sont utilisées dans les phases d'établissement et de fermeture de canal :

```
HELO <SP> <domain> <CRLF>  
QUIT <CRLF>
```

Par la commande HELO, l'hôte émetteur s'identifie ; cette commande peut être assimilée à l'expression "Bonjour, je suis le domaine <domaine>".

#### Exemple 5 / Exemple d'ouverture de connexion

```
R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready  
S: HELO USC-ISIF.ARPA  
R: 250 BBN-UNIX.ARPA
```

### Exemple 6 / Exemple de fermeture de connexion

```
S: QUIT
R: 221 BBN-UNIX.ARPA Service closing transmission channel
```

## 3.6. RELAI DE MESSAGES

Le chemin d'acheminement des messages peut être une "route" de la forme "@UN,@DEUX:JOE@TROIS", dans laquelle UN, DEUX, et TROIS sont des noms d'hôtes. Cette forme est employée dans le but d'accentuer la différence formelle entre une adresse et une route. La boîte-aux-lettres est une adresse absolue, la route est une information permettant d'y accéder. Ces deux concepts doivent toujours être dissociés.

Dans le concept, les éléments de la route sont déplacés vers la route inverse au fur et à mesure que le message circule à travers les serveurs-SMTP intermédiaires. La route inverse est donc construite comme la route prise en sens inverse, (c-à-d., une route partant de la position courante et aboutissant à l'émetteur du message). Lorsqu'un serveur-SMTP supprime son identifiant de la route et l'insère dans la route inverse, il doit utiliser le nom sous lequel il est connu du destinataire du message, et non de l'émetteur, dans le cas où ces deux noms diffèrent.

Lorsque le message arrive sur un serveur-SMTP, et que le premier élément de la route (la première destination intermédiaire) ne correspond pas à la machine courante, cet élément ne doit pas être effacé de la route et est utilisé par le serveur-SMTP comme nouvelle destination de réémission. Dans tous les cas, le serveur-SMTP courant ajoute son propre identifiant à la route inverse.

Par cette technique des routes, les récepteurs-SMTP reçoivent des messages qu'ils doivent relayer vers d'autres serveurs-SMTP. Le récepteur-SMTP peut accepter ou refuser de jouer ce rôle de relais de la même manière qu'il accepte ou refuse de traiter les messages de tel ou tel utilisateur local. Le récepteur-SMTP transforme l'argument de commande en déplaçant son propre identifiant de la route vers la première position de la route inverse. Le récepteur-SMTP devient à ce moment un émetteur-SMTP, établit un canal de transmission vers le prochain agent SMTP mentionné dans la route, et lui envoie le message.

Le premier hôte mentionné dans la route inverse doit être l'agent SMTP émettant les commandes, le premier hôte d'une route doit être l'agent SMTP recevant ces commandes.

Notez que les routes et routes inverses apparaissent à la fois dans les commandes et les réponses SMTP, mais pas nécessairement dans le corps de message. C'est-à-dire, La mention de ces chemins sous cette syntaxe précise n'est pas obligatoire dans les champs "To:", "From:", "CC:" de l'en-tête du corps de message.

Si un serveur-SMTP a accepté de relayer un message, puis s'aperçoit que la route est incorrecte ou que le message ne peut pas poursuivre son chemin quelle qu'en soit la raison, alors il devra composer un message "undeliverable mail" et le renvoyer à l'origine (dont il connaît la route par la route inverse).

Ce message de notification doit être émis au nom du serveur-SMTP sur cet hôte. Bien sûr, les serveurs-SMTP ne devront pas provoquer l'émission de tels messages de notification si le message en faute est lui-même un message de notification d'erreur. Une manière d'éviter l'établissement de boucles de rapports d'erreur est de forcer une route inverse vide dans la commande MAIL émettant un tel message. De même, lorsqu'un tel message est relayé, il est alors permis de laisser la route inverse vide. Une commande MAIL affichant une route inverse vide s'exprime ainsi :

```
MAIL FROM:<>
```

Un message de notification d'un courrier "undeliverable" est montré dans l'exemple 7. Cette notification est la réponse à un message émis par JOE sur l'hôte HOSTW et envoyé via l'hôte HOSTX à l'hôte HOSTY avec des instructions pour un relais vers l'hôte HOSTZ. L'exemple montre la transaction entre l'hôte HOSTY et l'hôte HOSTX, qui constitue la première étape de la notification d'erreur.



### **Exemple 7 / Exemple de notification d'impossibilité de livraison de courrier**

```
S: MAIL FROM:<>
R: 250 ok
S: RCPT TO:<@HOSTX.ARPA:JOE@HOSTW.ARPA>
R: 250 ok
S: DATA
R: 354 send the mail data, end with .
S: Date: 23 Oct 81 11:22:33
S: From: SMTP@HOSTY.ARPA
S: To: JOE@HOSTW.ARPA
S: Subject: Mail System Problem
S:
S:   Sorry JOE, your message to SAM@HOSTZ.ARPA lost.
S:   HOSTZ.ARPA said this:
S:     "550 No Such User"
S: .
R: 250 ok
```

## **3.7. DOMAINES**

Les domaines sont un concept récent dans le système de courrier de l'ARPA Internet. L'utilisation du système de domaines transforme l'espace d'adressage depuis un espace plat global de chaînes de caractères nommant des hôtes vers une structure hiérarchisée arborescente d'adresses globales. Le nom d'hôte est remplacé par un "domaine" et un identifiant d'hôte, couple représenté par une séquence d'éléments de domaines sous forme de chaînes de caractères, séparés par des points, et dans un ordre conventionnellement établi du plus spécifique au plus général.

Par exemple, "USC-ISIF.ARPA", "vf.eisti.FR", et "PC7.LCS.MIT.ARPA" sont des domaines à part entière.

Lorsque des noms de domaines figurent dans des messages SMTP seuls les noms officiels sont autorisés, et l'usage des pseudo ou alias n'est pas permis.

## **3.8. ECHANGE DES RÔLES**

La commande TURN peut être utilisée pour échanger les rôles des deux programmes en communication via le canal de transmission établi.

Si un programme-A est actuellement l'émetteur-SMTP, envoie la commande TURN et reçoit une réponse OK (250), alors le programme-A devient le récepteur-SMTP. Si un programme-B est actuellement en position de récepteur-SMTP, reçoit une commande TURN à laquelle il répond par une réponse OK (250), alors le programme-B devient l'émetteur-SMTP.

L'émission d'une réponse 502 indique que le récepteur refuse l'échange de rôles.

Notez toutefois que l'implémentation de cette commande est optionnelle. Elle ne devra d'ailleurs pas être utilisée en temps normal lorsque le canal de transmission est établi sous TCP. Cependant, lorsque le "coût" d'établissement d'un canal de transmission est élevé, cette commande peut s'avérer utile. Par exemple, lorsque le canal de transmission s'appuie sur le réseau commuté public (téléphone), surtout lorsque certains hôtes font "tourner" une liaison vers un certain nombre d'autres hôtes pour mettre à jour les échanges de messages.

# **4. LES SPECIFICATIONS SMTP**

## **4.1. COMMANDES SMTP**

### **4.1.1. SEMANTIQUE DES COMMANDES**

Les commandes SMTP englobent les fonctions de transfert de message et les fonctions système appelées par l'utilisateur. Les commandes SMTP sont des chaînes de caractères terminées par une séquence <CRLF>. Les codes de commande eux-mêmes sont constitués d'une séquence de digits en mode texte terminée par un <SP> lorsque des

paramètres suivent, sinon par <CRLF>. La syntaxe des boîtes-aux-lettres doit se conformer aux exigences conventionnelles du destinataire. Les commandes SMTP sont décrites ci-après. Les réponses SMTP à ces commandes sont décrites en Section 4.2.

Une transaction de courrier implique un certain nombre d'objets de données, transmis sous forme d'arguments dans les diverses commandes générées. La route inverse est l'argument de la commande MAIL, la route directe est l'argument de la commande RCPT, le contenu du message est l'argument de la commande DATA. Ces arguments ou objets de données doivent être transmis et conservés jusqu'à obtention de l'indication de "fin de données de courrier" qui achève la transaction. Le modèle préconisé pour cette tâche est de disposer de tampons de données distincts pour chaque type d'objet de données, c'est-à-dire, un tampon de route inverse, un tampon de route (directe), et un tampon de données de courrier. Certaines commandes spécifiques provoqueront l'ajout d'informations à tel ou tel tampon, ou provoquera l'effacement de tels ou tels tampons.

## **HELLO (HELO)**

Cette commande sert à identifier l'émetteur-SMTP vis à vis du récepteur-SMTP. L'argument est formé du nom de l'hôte où réside l'émetteur-SMTP.

Le récepteur-SMTP s'identifie à son tour vis-à-vis de l'émetteur-SMTP en répondant aux "civilités", dans un message de réponse à cette commande.

Cette commande, ainsi que la réponse OK qui y sera apportée confirme que les deux agents-SMTP se faisant face sont tous deux dans leur état initial, c'est-à-dire, qu'aucune transaction n'est en cours de traitement et que toutes les tables d'état et tampons sont vides.

## **MAIL (MAIL)**

Cette commande initialise une transaction de courrier par laquelle le message sera délivré dans une ou plusieurs boîtes-aux-lettres. L'argument mentionne une route inverse.

La route inverse consiste en une liste optionnelle d'hôtes ainsi que la boîte-aux-lettres de l'émetteur. Lorsque la liste d'hôtes est mentionnée, il s'agit d'une route "inversée" qui indique que le courrier a été relayé via chacun des hôtes de la liste (le premier hôte mentionné est le dernier relais par lequel est passé le message). Cette liste est utilisée comme route directe lorsqu'un message d'erreur doit être retourné à l'émetteur. Lorsque chaque relais utilisé ajoute son propre identifiant au début de la liste, le nom ajouté doit être celui qui identifie le relais vis-à-vis du relais suivant plutôt que celui qui l'identifie vis-à-vis de l'émetteur du message (si ces noms sont différents). Pour certains types de messages (par exemple, une notification d'erreur) la route inverse peut être vide (voir Exemple 7).

Cette commande provoque l'effacement des trois tampons (route inverse, route directe, et message) et initialise le tampon de route inverse avec l'argument fourni.

## **RECIPIENT (RCPT)**

Cette commande permet l'identification d'un récipiendaire unique du message ; lorsque le message doit être délivré à plusieurs personnes, on multipliera d'autant l'usage de cette commande.

La route consiste en une liste optionnelle d'hôtes, ainsi que la mention de la boîte-aux-lettres du récipiendaire. Lorsque la liste d'hôtes est mentionnée, il s'agit d'une route d'acheminement qui indique que le message doit être relayé vers le premier relais mentionné dans la liste. Si le récepteur-SMTP n'implémente pas la fonction de relais de courrier, il répondra par le même message qu'il aurait généré pour un utilisateur local inconnu (550).

Lorsque le message est relayé, l'hôte relais doit supprimer son identifiant de la route (normalement en première position de celle-ci) et rajoute ce dernier en début de la route inverse. Lorsque le courrier atteint sa destination finale, (la route directe ne contient en effet qu'une seule boîte-aux-lettres cible), il est ajouté par le récepteur-SMTP dans cette boîte-aux-lettres selon les conventions locales de l'hôte support.

Par exemple, un courrier reçu par un hôte relais A avec les arguments

```
FROM : <USERX@HOSTY . ARPA>  
TO : <@HOSTA . ARPA , @HOSTB . ARPA : USERC@HOSTD . ARPA>
```

sera retransmis vers l'hôte B avec les arguments :

```
FROM: <@HOSTA.ARPA:USERX@HOSTY.ARPA>  
TO: <@HOSTB.ARPA:USERC@HOSTD.ARPA> .
```

Cette commande ajoute l'argument de route au tampon de route.

## **DATA ( DATA )**

Le récepteur traite les lignes suivant cette commande comme le corps du message émis par l'émetteur. Cette commande provoque l'ajout de ce texte au tampon de message. Le texte du corps de message peut contenir tout caractère parmi les 128 premiers codes ASCII.

Le corps de message est terminé lorsqu'une ligne ne contenant qu'un seul point est rencontrée, équivalent à la séquence "<CRLF>.<CRLF>" (voir la Section 4.5.2 à propos de la notion de Transparence). Cette séquence marque la fin de message.

La marque de fin de message signal au récepteur qu'il doit maintenant traiter l'ensemble des informations concernant la transaction. Ce traitement consomme l'information inscrite dans le tampon de route inverse, le tampon de route directe, et le tampon de message, lesquels sont vidés à la fin du traitement. Si le traitement s'effectue correctement, le récepteur doit envoyer une réponse OK. Si le traitement échoue, il envoie une réponse d'erreur.

Lorsque le récepteur-SMTP accepte un message, soit pour le relayer, soit pour livraison "finale", il insère au début des données de courrier une ligne de marquage temporel. Cette ligne indique l'identité de l'hôte origine du message, l'identité de l'hôte ayant reçu le message (celui qui écrit ce marqueur), ainsi que la date et l'heure à laquelle ce message a été reçu. Des messages relayés afficheront de multiples marqueurs temporels.

Lorsqu'un récepteur-SMTP effectue la livraison "finale" d'un courrier, il insère au début des données de courrier une ligne de route inverse. Cette ligne préserve l'information de l'argument <route-inverse> associé à la commande MAIL reçue. A partir de ce moment, la livraison "finale" signifie que le message quitte le "monde" SMTP. Normalement, ceci indique que le message a été correctement délivré à son récipiendaire (pas qu'il l'ait lu !) ; dans d'autres cas, le message pourra subir d'autres traitements par d'autres systèmes de courrier locaux.

La boîte-aux-lettres mentionnée dans la route inverse peut être différente de celle de l'émetteur effectif du message, par exemple, lorsque des réponses d'erreur sont routées vers une boîte-aux-lettres commune, spécialement réservée par le système à cet usage.

Les deux paragraphes précédants impliquent que les données de courrier "finales" commencent effectivement par une ligne de route inverse, suivi par une ou plusieurs lignes de marquage temporel. Ces lignes seront immédiatement suivies par l'en-tête de corps de message et le corps de message lui-même [2]. Voir l'exemple 8.

Un cas particulier de réponse doit être développé et des actions supplémentaires envisagées lorsque le traitement suivant la réception de l'indicateur de fin de message n'a pu se dérouler que partiellement. Ceci peut intervenir si, après avoir accepté des récipiendaires et le message, le récepteur-SMTP s'aperçoit qu'il est impossible de délivrer le courrier à certains destinataires, alors que d'autres peuvent être servis sans problème (par exemple, à cause d'un manque de ressources mémoires temporaire). Dans de telles situations, la réponse à une commande DATA doit néanmoins être une réponse OK. Par contre, le récepteur-SMTP doit composer et émettre un message de notification d'erreur à l'émetteur. Celui-ci peut être unique et lister tous les récipiendaires qui n'ont pu recevoir le message, ou peut être multiplié par le nombre de récipiendaires en faute, chaque message constituant une notification d'erreur simple (voir exemple 7). Toutes les notifications d'erreur sont émises via la commande MAIL (même si elles résultent du traitement de commandes SEND, SOML, ou SAML).

### **Exemple 8 / Exemple de réception de chemin de retour et de marquage temporel**

```
Return-Path: <@GHI.ARPA,@DEF.ARPA,@ABC.ARPA:JOE@ABC.ARPA>  
Received: from GHI.ARPA by JKL.ARPA ; 27 Oct 81 15:27:39 PST  
Received: from DEF.ARPA by GHI.ARPA ; 27 Oct 81 15:15:13 PST  
Received: from ABC.ARPA by DEF.ARPA ; 27 Oct 81 15:01:59 PST  
Date: 27 Oct 81 15:01:01 PST  
From: JOE@ABC.ARPA  
Subject: Improved Mailing System Installed
```

To: SAM@JKL.ARPA

This is to inform you that ...

### **SEND (SEND)**

Cette commande initialise une transaction par laquelle le message doit être envoyé interactivement sur un ou plusieurs terminaux. L'argument mentionne une route inverse. La commande est considérée accomplie avec succès si le message a pu être inscrit sur le terminal.

La route inverse consiste en une liste optionnelle d'hôtes ainsi que la boîte-aux-lettres de l'émetteur. Lorsque la liste d'hôtes est mentionnée, il s'agit d'une route "inversée" qui indique que le courrier a été relayé via chacun des hôtes de la liste (le premier hôte mentionné est le dernier relais par lequel est passé le message). Cette liste est utilisée comme route directe lorsqu'un message d'erreur doit être retourné à l'émetteur. Lorsque chaque relais utilisé ajoute son propre identifiant au début de la liste, le nom ajouté doit être celui qui identifie le relais vis-à-vis du relais suivant plutôt que celui qui l'identifie vis-à-vis de l'émetteur du message (si ces noms sont différents).

Cette commande provoque l'effacement des trois tampons (route inverse, route directe, et message) et initialise le tampon de route inverse avec l'argument fourni.

### **SEND OR MAIL (SOML)**

Cette commande initialise une transaction par laquelle le message doit être envoyé interactivement sur un ou plusieurs terminaux ou boîtes-aux-lettres. Pour chaque récipiendaire, le message est délivré sur le terminal si une session y est ouverte par le destinataire (et celui-ci accepte l'affichage interactif de messages), et par défaut, dans sa boîte-aux-lettres. L'argument mentionne une route inverse. Cette commande est considérée accomplie avec succès si le message est correctement délivré soit sur le terminal, soit dans la boîte-aux-lettres.

La route inverse consiste en une liste optionnelle d'hôtes ainsi que la boîte-aux-lettres de l'émetteur. Lorsque la liste d'hôtes est mentionnée, il s'agit d'une route "inversée" qui indique que le courrier a été relayé via chacun des hôtes de la liste (le premier hôte mentionné est le dernier relais par lequel est passé le message). Cette liste est utilisée comme route directe lorsqu'un message d'erreur doit être retourné à l'émetteur. Lorsque chaque relais utilisé ajoute son propre identifiant au début de la liste, le nom ajouté doit être celui qui identifie le relais vis-à-vis du relais suivant plutôt que celui qui l'identifie vis-à-vis de l'émetteur du message (si ces noms sont différents).

Cette commande provoque l'effacement des trois tampons (route inverse, route directe, et message) et initialise le tampon de route inverse avec l'argument fourni.

### **SEND AND MAIL (SAML)**

Cette commande initialise une transaction par laquelle le message doit être envoyé interactivement sur un ou plusieurs terminaux et simultanément dans les boîtes-aux-lettres associées aux utilisateurs concernés. Pour chaque récipiendaire, le message est délivré sur le terminal si une session y est ouverte par le destinataire (et celui-ci accepte l'affichage interactif de messages), et quelque soit le résultat de l'action précédente, dans sa boîte-aux-lettres. L'argument mentionne une route inverse. Cette commande est considérée accomplie avec succès si le message est au moins délivré dans la boîte-aux-lettres.

La route inverse consiste en une liste optionnelle d'hôtes ainsi que la boîte-aux-lettres de l'émetteur. Lorsque la liste d'hôtes est mentionnée, il s'agit d'une route "inversée" qui indique que le courrier a été relayé via chacun des hôtes de la liste (le premier hôte mentionné est le dernier relais par lequel est passé le message). Cette liste est utilisée comme route directe lorsqu'un message d'erreur doit être retourné à l'émetteur. Lorsque chaque relais utilisé ajoute son propre identifiant au début de la liste, le nom ajouté doit être celui qui identifie le relais vis-à-vis du relais suivant plutôt que celui qui l'identifie vis-à-vis de l'émetteur du message (si ces noms sont différents).

Cette commande provoque l'effacement des trois tampons (route inverse, route directe, et message) et initialise le tampon de route inverse avec l'argument fourni.

## **RESET (RSET)**

Cette commande demande à avorter le traitement de la transaction en cours. Toute donnée d'émetteur, de récipiendaires, et/ou de message, tous les tampons, et tables d'états doivent être effacés. Le récepteur doit émettre une réponse OK.

## **VERIFY (VRFY)**

Cette commande demande au récepteur de confirmer que les arguments fournis désignent bien un utilisateur. S'il s'agit d'un nom d'utilisateur, le nom complet de ce dernier (s'il est connu du récepteur) ainsi que la boîte-aux-lettres totalement qualifiée doivent être renvoyés au requérant.

Cette commande n'affecte aucun des tampons courants de la transaction de courrier.

## **EXPAND (EXPN)**

Cette commande demande au récepteur de confirmer si l'argument associé identifie une liste de diffusion, et, si c'est le cas, de renvoyer la liste des membres de cette liste. Le nom complet des utilisateurs (s'il est connu) et les adresses de boîtes-aux-lettres entièrement qualifiées seront renvoyées via une réponse multilignes.

Cette commande n'affecte aucun des tampons courants de la transaction de courrier.

## **HELP (HELP)**

Cette commande demande au récepteur d'envoyer des informations d'aide à l'émetteur de la commande. Cette commande peut accepter un argument (ex., tout nom de commande). La réponse contient des informations plus détaillées sur cette commande.

Cette commande n'affecte aucun des tampons courants de la transaction de courrier.

## **NOOP (NOOP)**

Cette commande n'affecte aucune donnée ni l'état de la transaction en cours. Il ne demande de la part du récepteur aucune action particulière autre que le renvoi d'une réponse OK.

Cette commande n'affecte aucun des tampons courants de la transaction de courrier.

## **QUIT (QUIT)**

Cette commande demande au récepteur de répondre par OK, puis de fermer le canal de transmission .

Le récepteur ne devra pas lui-même fermer son canal de transmission avant qu'il n'ait reçu la réponse à la commande QUIT qu'il a envoyée (même si une erreur est détectée). L'émetteur ne devra pas plus fermer son canal de transmission sans avoir émis une commande QUIT et reçu une réponse (même s'il a reçu une notification d'erreur à une commande précédente). Si la connexion est fermée prématurément, le récepteur devra agir comme s'il avait reçu une commande RSET (annulant toute transaction en cours, mais sans annuler toute commande précédemment conclue jusqu'à son terme), l'émetteur devra quant à lui réagir comme si la commande active s'était conclue par une erreur (4xx).

## **TURN (TURN)**

Cette commande demande au récepteur de soit, (1) répondre par OK puis prendre le rôle d'un émetteur-SMTP, soit (2) envoyer une réponse de refus et conserver son rôle de récepteur-SMTP.

Si un programme-A est actuellement l'émetteur-SMTP, envoie une commande TURN et reçoit OK en réponse (code 250), alors le programme-A prend le rôle du récepteur-SMTP. Le programme-A se retrouve alors dans le même état initial que lorsque le canal de transmission vient de s'établir, et envoie la réponse 220 indiquant qu'il est prêt à servir.

Si le programme-B est actuellement en position de récepteur-SMTP, reçoit une commande TURN et répond par OK (250), alors le programme-B devient l'émetteur-SMTP. Le programme-B se retrouve alors dans le même état initial que lorsque le canal de transmission vient de s'établir, et s'attend à recevoir le message de code 220.

Le récepteur envoie la réponse 502 pour indiquer qu'il refuse le changement de rôle.

#### **Restrictions sur l'ordre d'apparition des commandes**

Il existe des restrictions sur l'ordre dans lequel ces commandes doivent être utilisées.

La première commande d'une session doit toujours être la commande HELO. Cette commande peut quant à elle être réutilisée plus tard dans la même session. Si l'argument associé à la commande HELO ne peut être accepté par le récepteur, une réponse d'erreur 501 doit être renvoyée et le récepteur-SMTP reste dans le même état.

Les commandes NOOP, HELP, EXPN, et VRFY peuvent être utilisées à tout moment dans une session.

Les commandes MAIL, SEND, SOML, ou SAML initialisent une transaction. Une transaction consiste en l'une des commandes d'initialisation de transaction, une ou plusieurs commandes RCPT, et une commande DATA, le tout dans cet ordre précis. Une transaction en cours peut être annulée par la commande RSET. Une session peut traiter zéro ou plus transactions.

Si l'argument de commande d'initialisation de transaction ne peut être accepté par le récepteur, celui-ci répondra par une notification d'erreur de code 501 et restera dans le même état que précédemment. Si une commande est reçue, et que celle-ci déroge à l'ordre précisé ci-avant, une notification d'erreur de code 503 doit être renvoyée, le récepteur-SMTP restant dans le même état.

La dernière commande d'une session doit être la commande QUIT. Cette commande ne peut être utilisée qu'à ce moment d'une session.

### **4.1.2. SYNTAXE DE COMMANDES**

Les commandes consistent en un code de commande suivi d'un champ d'argument. Les codes de commande sont constitués de quatre caractères alphabétiques. Il n'est pas tenu compte de la casse pour le code de commande. Ainsi, les écritures suivantes sont acceptées pour la commande MAIL :

```
MAIL    Mail    mail    MaIl    mAIl
```

Ceci s'applique également à tout symbole représentant des valeurs d'arguments, telles que "TO" ou "to" pour l'expression de la route. Les codes de commande doivent être séparés du champ d'argument par un ou plus espaces. Cependant, à l'intérieur des expressions de route et de route inverse, la casse devient importante. En particulier, sur certains hôtes, l'utilisateur "smith" est différencié de l'utilisateur "Smith".

Le champ d'argument est constitué d'une chaîne de caractères de longueur variable terminée par la séquence <CRLF>. Le récepteur ne doit pas exécuter d'action tant que cette séquence n'est pas reçue.

Des crochets signalent la présence d'un argument optionnel. Si l'option n'est pas utilisée, une valeur par défaut est prise pour ce paramètre.

Les commandes SMTP sont les suivantes :

```
HELO <SP> <domaine> <CRLF>
MAIL <SP> FROM:<route-inverse> <CRLF>
RCPT <SP> TO:<route-directe> <CRLF>
DATA <CRLF>
RSET <CRLF>
SEND <SP> FROM:<route-inverse> <CRLF>
SOML <SP> FROM:<route-inverse> <CRLF>
SAML <SP> FROM:<route-inverse> <CRLF>
VRFY <SP> <chaîne> <CRLF>
```

```

EXPN <SP> <chaîne> <CRLF>
HELP [<SP> <chaîne>] <CRLF>
NOOP <CRLF>
QUIT <CRLF>
TURN <CRLF>

```

La syntaxe des champs d'argument ci-dessus (en utilisant la notation BNF lorsqu'elle est applicable) est précisée ci-après. La notation "..." indique que le champ peut être répété plusieurs fois.

```

<route-inverse> ::= <chemin>
<route-directe> ::= <chemin>
<chemin> ::= "<" [ <a-d-l> ":" ] <bal> ">"
<a-d-l> ::= <at-domaine> | <at-domaine> "," <a-d-l>
<at-domaine> ::= "@" <domaine>
<domaine> ::= <élément> | <élément> "." <domaine>
<élément> ::= <nom> | "#" <nombre> | "[" <adresseIP> "]"
<bal> ::= <partie-locale> "@" <domaine>
<partie-locale> ::= <chaîne-pointée> | <chaîne-quotée>
<nom> ::= <a> <ldh-ch> <alphanum>
<chaîne-hyp> ::= <alphanum-hyp> | <alphanum-hyp> <chaîne-hyp>
<alphanum> ::= <alpha> | <digit>
<alphanum-hyp> ::= <alpha> | <digit> | "-"
<chaîne-pointée> ::= <chaîne> | <chaîne> "." <dot-string>
<chaîne> ::= <char> | <char> <chaîne>
<chaîne-quotée> ::= "\"" <qtext> "\""
<texte> ::= "\" <x> | "\" <x> <qtext> | <uq> | <uq> <qtext>
<xcar> ::= <char> | "\" <x>
<adresseIP> ::= <short> "." <short> "." <short> "." <short>
<nombre> ::= <digit> | <digit> <nombre>
<CRLF> ::= <CR> <LF>
<CR> ::= Le caractère Retour Chariot (ASCII code 13)
<LF> ::= Le caractère Nouvelle Ligne (ASCII code 10)
<SP> ::= Le caractère Espace (ASCII code 32)
<short> ::= un, deux, ou trois digits représentant un entier décimal
de 0 à 255
<alpha> ::= tout caractère parmi les 52 caractères alphabétiques A à
Z (majuscules) et a à z (minuscules)
<char> ::= any one of the 128 ASCII characters, but not any
<special> or <SP>
<digit> ::= un digit de 0 à 9
<uq> ::= tout caractère parmi les 128 caractères ASCII sauf <CR>,
<LF>, quote ("), ou antislash (\)
<x> ::= tout caractère de l'ASCII (pas d'exceptions)
<special> ::= "<" | ">" | "(" | ")" | "[" | "]" | "\" | "." | "," | ";"
| ":" | "@" | "\"" | les caractères de contrôle de l'ASCII
(codes 0 à 31 inclus et 127)

```

Notez que l'antislash, "\", est un caractère d'échappement, utilisé pour indiquer que le caractère qui suit est à interpréter littéralement (au lieu de son interprétation normale). Par exemple, "Joe\,Smith" pourrait être écrit pour indiquer un champ unique de neufs caractères avec la virgule comme quatrième caractère.

Les hôtes sont généralement connus par des noms traduits en adresses réseau dans chaque machine. Notez que les noms issus du système de domaines sont les noms officiels – n'est pas autorisé l'usage de pseudo ou d'alias.

Parfois, un hôte n'est pas connu de la fonction de translation et la communication est bloquée. Pour éviter ce blocage, deux formes numériques sont admises pour l'expression des hôtes. La première forme est un entier décimal préfixé par le symbole dièse "#", qui indique que ce nombre est l'adresse de l'hôte. L'autre forme est une adresse IP d'hôte constituée par quatre entiers compris entre 0 et 255, séparés par des points et "encapsulés" dans des crochets,

ex., "[123.255.37.2]", et qui donne une adresse Internet ARPA sur 32 bits. Les lignes de route inverse et de marquage temporel sont formellement définies comme suit :

```
<ln-route-inverse> ::= "Return-Path:" <SP><route-inverse><CRLF>
<ln-marqueur-temporel> ::= "Received:" <SP> <marqueur> <CRLF>
<marqueur> ::= <de-domaine> <par-domaine> <info-opt> ";" <datation>
<de-domaine> ::= "FROM" <SP> <domaine> <SP>
<par-domaine> ::= "BY" <SP> <domaine> <SP>
<info-opt> ::= [<via>] [<avec>] [<id>] [<pour>]
<via> ::= "VIA" <SP> <lien> <SP>
<avec> ::= "WITH" <SP> <protocole> <SP>
<id> ::= "ID" <SP> <chaîne> <SP>
<pour> ::= "FOR" <SP> <chemin> <SP>
<lien> ::= Les noms de liens standard tels qu'ils sont
           enregistrés par le Network Information Center.
<protocole> ::= les noms des protocoles standard tels que définis par
           le Network Information Center.
<datation> ::= <SP> <date> <SP> <heure>
<date> ::= <dd> <SP> <mois> <SP> <yy>
<heure> ::= <hh> ":" <mm> ":" <ss> <SP> <zone>
<dd> ::= le nombre à un ou deux digits représentant le
           quantième du mois entre 1 et 31.
<mois> ::= "JAN" | "FEB" | "MAR" | "APR" | "MAY" | "JUN" | "JUL"
           | "AUG" | "SEP" | "OCT" | "NOV" | "DEC"
<yy> ::= les deux derniers chiffres de l'année entre 00 et 99.
<hh> ::= le nombre à deux chiffres indiquant l'heure du jour
           entre 00 et 24.
<mm> ::= le nombre à deux chiffres désignant les minutes entre
           00 et 59.
<ss> ::= le nombre à deux chiffres indiquant les secondes
           entre 00 et 59.
<zone> ::= "UT" pour Universal Time (par défaut) ou tout autre
           identificateur de zone horaire (comme spécifié dans
           [2]).
```

#### **Exemple 9 / Exemple de route inverse**

```
Return-Path: <@CHARLIE.ARPA,@BAKER.ARPA:JOE@ABLE.ARPA>
```

#### **Exemple 10 / Exemple de marquage temporel**

```
Received: FROM ABC.ARPA BY XYZ.ARPA ; 22 OCT 81 09:23:59 PDT
Received: from ABC.ARPA by XYZ.ARPA via TELENET with X25
        id M12345 for Smith@PDQ.ARPA ; 22 OCT 81 09:23:59 PDT
```

## **4.2. REPONSES SMTP**

Les réponses aux commandes SMTP sont instaurées pour la synchronisation des requêtes et actions relatives au processus de transfert de courrier, et pour garantir à tout moment à l'émetteur-SMTP de connaître l'état du récepteur-SMTP. Chaque commande doit générer une réponse et une seule.

Les détails du séquençement des commandes et réponses sont explicités en Sections 4.3 "Séquencement" et 4.4 "Diagrammes d'état".

Une réponse SMTP consiste en un nombre à trois digits (transmis sous forme de trois caractères numériques) suivi d'un texte. Le code numérique est à destination d'automates pour déterminer l'état suivant à atteindre ; la réponse textuelle est plutôt destinée à un utilisateur humain. Il est établi que le code à trois digits contient suffisamment d'information pour que l'émetteur-SMTP n'ait pas à examiner le texte, lequel peut être ignoré ou



répercuté au niveau de l'interface utilisateur, selon le cas. En particulier, le texte peut différer selon le récepteur et le contexte, pour un code de réponse donné. Une discussion sur la théorie des codes de réponse est exposée en Appendice E. Au niveau formel, une réponse est définie comme la séquence : un code à trois chiffres, <SP>, une ligne de texte, et <CRLF>, ou éventuellement une réponse multilignes (comme défini dans l'Appendice E). Seules les commandes EXPN et HELP peuvent résulter en des réponses multilignes dans des circonstances normales, bien que par principe, des réponses multilignes soient autorisées pour toutes les commandes.

#### 4.2.1. CODES DE REPONSE PAR GROUPES DE FONCTION

500 Erreur de syntaxe, commande non reconnue [y compris des erreurs de type "ligne de commande trop longue"]

501 Erreur de syntaxe dans des paramètres ou arguments

502 Commande non implémentée

503 Mauvaise séquence de commandes

504 Paramètre de commande non implémenté

211 Etat système, ou réponse d'aide système

214 Message d'aide [Informations sur l'utilisation d'un récepteur ou signification d'une commande non standard particulière; en clair pour un opérateur humain]

220 <domaine> Service disponible

221 <domaine> Canal de transmission en cours de fermeture

421 <domaine> Service non disponible, canal en fermeture [Réponse à émettre sur tous les canaux lorsque le système exécute une séquence d'arrêt]

250 Action de messagerie effectuée, succès

251 Utilisateur non local ; réémission vers <route-directe> (avec relais automatique)

450 Action non effectuée : boîte-aux-lettres non disponible [Ex., bloquée par un autre utilisateur]

550 Action non effectuée : boîte-aux-lettres non disponible [Ex., boîte-aux-lettres non trouvée, pas d'accès]

451 Action arrêtée : erreur de traitement

551 Utilisateur non local ; essayer <route> (sans relais automatique)

452 Action non effectuée : manque de ressources système

552 Action arrêtée : manque de ressources de stockage

553 Action non effectuée : nom de boîte-aux-lettres non autorisé [Ex., erreur de syntaxe dans le nom de boîte]

354 Début de message ; arrêt par <CRLF>.<CRLF>

554 Transaction échouée

#### 4.2.2. CODES REPONSE PAR ORDRE NUMERIQUE

211 Etat système, ou réponse d'aide système

214 Message d'aide [Informations sur l'utilisation d'un récepteur ou signification d'une commande non standard particulière; en clair pour un opérateur humain]

220 <domaine> Service disponible

221 <domaine> Canal de transmission en cours de fermeture

250 Action de messagerie effectuée, succès

251 Utilisateur non local ; réémission vers <route-directe> (avec relais automatique)

354 Début de message ; arrêt par <CRLF>.<CRLF>

421 <domaine> Service non disponible, canal en fermeture [Réponse à émettre sur tous les canaux lorsque le système exécute une séquence d'arrêt]

450 Action non effectuée : boîte-aux-lettres non disponible [Ex., bloquée par un autre utilisateur]

451 Action arrêtée : erreur de traitement

452 Action non effectuée : manque de ressources système

500 Erreur de syntaxe, commande non reconnue [y compris des erreurs de type "ligne de commande trop longue"]

501 Erreur de syntaxe dans des paramètres ou arguments

502 Commande non implémentée

503 Mauvaise séquence de commandes

504 Paramètre de commande non implémenté  
550 Action non effectuée : boîte-aux-lettres non disponible [Ex., boîte-aux-lettres non trouvée, pas d'accès]  
551 Utilisateur non local ; essayer <route> (sans relais automatique)  
552 Action arrêtée : manque de ressources de stockage  
553 Action non effectuée : nom de boîte-aux-lettres non autorisé [Ex., erreur de syntaxe dans le nom de boîte]  
554 Transaction échouée

### 4.3. SEQUENCEMENT DES COMMANDES ET REPONSES

Les communications entre l'émetteur et le récepteur sont prévues pour suivre un dialogue alterné, sous contrôle de l'émetteur. En temps que tel, l'émetteur envoie une commande à laquelle le récepteur répond. L'émetteur doit attendre cette réponse avant d'envoyer d'autres commandes.

Une réponse importante est l'acquiescement des "civilités". Normalement, un récepteur renvoie une réponse 220 "Service disponible" lorsque la connexion est établie. L'émetteur doit attendre ce message avant d'émettre toute autre commande.

Note : tous ces messages de réponse à "civilités" doivent faire figurer le nom officiel de l'hôte hébergeant le récepteur en première place après le code numérique de réponse.

Par exemple,

```
220 <SP> USC-ISIF.ARPA <SP> Service ready <CRLF>
```

La table ci-après liste les réponses soit positives, soit négatives qui peuvent être attendues après chacune des commandes. Ces règles de séquençement doivent être strictement adoptées ; un récepteur peut changer le texte qu'il ajoute pour expliciter la réponse, mais la séquence de réponse, ainsi que sa signification générale doit être préservée.

### SEQUENCES COMANDES-REPONSES

Chaque commande est listée avec toutes ses réponses possibles. Les préfixes utilisés avant les réponses possibles sont "P" pour préliminaire (contexte non utilisé par SMTP), "I" pour intermédiaire, "S" pour succès, "F" pour faute (échec), et "E" pour erreur. La réponse 421 (service non disponible, refermant le canal de transmission) peut être renvoyée suite à toute commande si le récepteur-SMTP sait qu'il doit arrêter son système. La liste suivante sert de base à l'établissement des diagrammes d'état en Section 4.4.

#### Etablissement de connexion (Implicite)

```
S: 220  
F: 421
```

#### HELO

```
S: 250  
E: 500, 501, 504, 421
```

#### MAIL

```
S: 250  
F: 552, 451, 452  
E: 500, 501, 421
```

#### RCPT

```
S: 250, 251  
F: 550, 551, 552, 553, 450, 451, 452  
E: 500, 501, 503, 421
```

#### DATA

```
I: 354 -> data -> S: 250  
F: 552, 554, 451, 452
```

F: 451, 554  
E: 500, 501, 503, 421

## RSET

S: 250  
E: 500, 501, 504, 421

## SEND

S: 250  
F: 552, 451, 452  
E: 500, 501, 502, 421

## SOML

S: 250  
F: 552, 451, 452  
E: 500, 501, 502, 421

## SAML

S: 250  
F: 552, 451, 452  
E: 500, 501, 502, 421

## VERFY

S: 250, 251  
F: 550, 551, 553  
E: 500, 501, 502, 504, 421

## EXPN

S: 250  
F: 550  
E: 500, 501, 502, 504, 421

## HELP

S: 211, 214  
E: 500, 501, 502, 504, 421

## NOOP

S: 250  
E: 500, 421

## QUIT

S: 221  
E: 500

## TURN

S: 250  
F: 502  
E: 500, 503

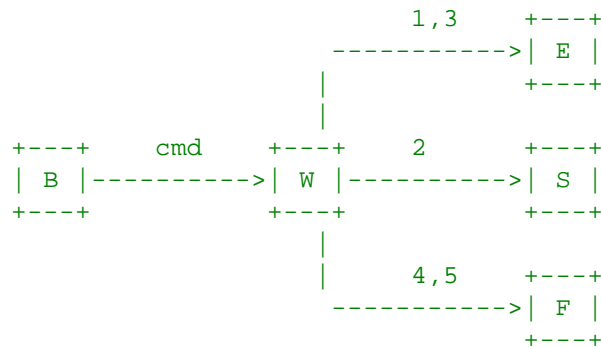
## 4.4. DIAGRAMMES D'ETATS

Ci-après sont exposés les diagrammes d'état pour ce qui constituerait une implémentation SMTP "légère". Seul le premier digit du code réponse est exploité. Un diagramme d'état correspond à un groupe de commandes SMTP. Le

regroupement des commandes a été fait en construisant un modèle pour chaque commande, puis en réunissant les commandes de même modèle structurel.

Pour chacune des commandes, on définit trois issues possibles : "succès" (S), "faute (échec)" (F), et "erreur" (E). Les diagrammes d'états utilisent le symbole B pour "début" (begin), et le symbole W pour "attendre la réponse" (wait for answer).

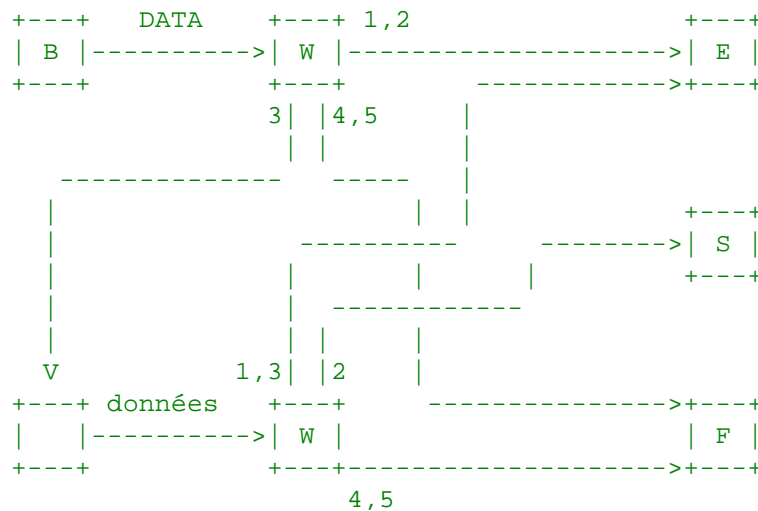
Est exposé tout d'abord le diagramme représentant la plupart des commandes SMTP :



Ce diagramme correspond aux commandes :

HELO, MAIL, RCPT, RSET, SEND, SOML, SAML, VRFY, EXPN, HELP, NOOP, QUIT, TURN.

Un diagramme plus complexe modélise la commande DATA :



Notez que les "données" sont ici une série de lignes envoyées par l'émetteur au récepteur sans qu'une réponse ne soit attendue avant la fin de la dernière ligne.

## 4.5. DETAILS

### 4.5.1. IMPLEMENTATION MINIMALE

Pour qu'une implémentation de SMTP puisse fonctionner, les récepteurs devront au minimum reconnaître les commandes suivantes :

COMMANDS -- HELO

MAIL  
RCPT  
DATA  
RSET  
NOOP  
QUIT

#### 4.5.2. PRINCIPE DE TRANSPARENCE

Sans aucun mécanisme destiné à préserver la transparence des données, la séquence de caractères "<CRLF>.<CRLF>" termine le texte du message et ne peut donc être envoyée littéralement par l'utilisateur. En général, les utilisateurs ne sont pas informés de l'existence de telles séquences "interdites". Pour permettre que tout texte d'un utilisateur quelconque puisse être transmis de manière transparente, on suivra les procédures suivantes.

1. Avant d'envoyer une ligne du texte du message, l'émetteur-SMTP analyse le premier caractère de la ligne. S'il s'agit d'un point, ce point est doublé.
2. Lorsqu'une ligne de texte est reçue par le récepteur-SMTP, ce dernier analyse la ligne. Si la ligne ne contient qu'un point unique, alors il s'agit de l'identificateur de fin de message. Si le premier caractère est un point, et que d'autres caractères suivent sur la même ligne, alors ce premier point est supprimé.

Le texte du message peut être composé avec les 128 premiers caractères du code ASCII. Tous les caractères sans exception doivent être copiés dans la boîte-aux-lettres y compris des caractères de contrôle de format ou autres contrôles. Si le canal de transmission propose un mode de transmission étendu à 8 bits (canal binaire), les codes en 7 bits de l'ASCII sont transmis justifiés à droite (dans l'octet) avec le premier bit (poids forts) forcé à zéro.

Sur certains systèmes, il peut être nécessaire d'opérer une transformation sur les données au moment où elles sont reçues et enregistrées. Ceci peut être nécessaire pour des hôtes qui utilisent un jeu de caractères différent de l'ASCII pour la représentation locale des données, ou qui enregistrent les données sous forme d'enregistrements plutôt que sous forme de chaînes de caractères. Si de telles transformations sont nécessaires, alors elles doivent être réversibles – et surtout si les courriers ainsi traités doivent être relayés vers d'autres destinations.

#### 4.5.3. TAILLES

Un certain nombre d'objets de données ont des tailles minimales et maximales définies. C'est-à-dire, toute implémentation doit s'attendre à recevoir des objets d'au moins la taille minimale, mais ne doit elle-même pas envoyer d'objets d'une taille supérieure à la taille maximale.

```
*****  
*  
*   POUR UNE ADAPTABILITE OPTIMALE, LES TECHNIQUES   *  
* D'IMPLEMENTATION QUI N'IMPOSENT PAS DE LIMITES   *  
* SUR LA TAILLE DE CES OBJETS SONT PREFEREES.      *  
*                                                    *  
*****
```

#### Nom d'utilisateur

La longueur totale maximale d'un nom d'utilisateur est de 64 caractères.

#### Nom de domaine

La longueur totale maximale d'un nom de domaine est de 64 caractères.

#### Routes

La longueur maximale des routes et routes inverses est de 256 caractères chacune (y compris la ponctuation et les séparateurs d'éléments).

## Ligne de commande

La longueur totale maximale de la ligne de commande, incluant le code de commande et le <CRLF> final est de 512 caractères.

## Ligne de réponse

La longueur totale maximale de la ligne de réponse, incluant le code de réponse et le <CRLF> final est de 512 caractères.

## Ligne de texte (corps de message)

La longueur maximale d'une ligne de texte y compris le <CRLF> final est de 1000 caractères (le point dupliqué en tête de ligne pour assurer le principe de transparence n'est pas compté).

## Tampon des récipiendaires

Le nombre maximal de récipiendaires à enregistrer pour une transaction est fixé à 100.

```
*****
*
*   POUR UNE ADAPTABILITE OPTIMALE, LES TECHNIQUES
*   D'IMPLEMENTATION QUI N'IMPOSENT PAS DE LIMITES
*   SUR LA TAILLE DE CES OBJETS SONT PREFEREES.
*
*****
```

Des erreurs dues au dépassement des limites ci-dessus peuvent être reportées en utilisant les extensions de réponses suivantes :

500 Ligne trop longue.

501 Chemin trop long

552 Trop de récipiendaires.

552 Message trop long.

## APPENDICE A

### ***Service de transport de données TCP***

Le Transmission Control Protocol [3] est le service de transport standard de l'ARPA Internet, ainsi que dans tout réseau supportant les standards de protocoles inter-réseaux du "Department of Defense" américain.

### ***Etablissement de la connexion***

Le canal de transmission SMTP est une connexion TCP établie entre le port U du processus émetteur et le port L du processus récepteur. Cette connexion unique bidirectionnelle est utilisée à titre de canal de transmission. Ce protocole est assigné au port de service 25 (31 en octal), soit L=25.

## ***Transport des données***

Les connexions TCP supportent la transmission de données en 8 bits. Les données SMTP sont limitées à des caractères ASCII encodés sur 7 bits. Chaque caractère est envoyé calé à droite de l'octet de transmission, avec le bit de poids fort forcé à 0.

## **APPENDICE B**

### ***Service de transport de données NCP***

Le protocole de Host-to-Host Protocol [4] (implémenté par le Network Control Program) peut être utilisé dans ARPANET.

*(NdT : ne concerne en fait que le réseau militaire américain)*

### ***Etablissement de la connexion***

Le canal de transmission SMTP est établi sous NCP entre le socket U du processus émetteur et le socket L du processus récepteur. Le protocole Initial Connection Protocol [5] est suivi, aboutissant à l'établissement d'une paire de connexions simples. Cette paire de connexions est utilisée à titre de canal de transmission. Ce protocole est assigné au socket contact 25 (31 en octal), soit L=25.

### ***Transport des données***

Les connexions NCP supportent la transmission de données en 8 bits. Les données SMTP sont limitées à des caractères ASCII encodés sur 7 bits. Chaque caractère est envoyé calé à droite de l'octet de transmission, avec le bit de poids fort forcé à 0.

## **APPENDICE C**

### ***NITS***

Le service de transport Network Independent Transport Service [6] peut être utilisé pour supporter SMTP.

### ***Etablissement de connexion***

Le canal de transmission SMTP est établi sous NITS entre le processus émetteur et le processus récepteur. Le processus émetteur exécute la primitive CONNECT, et le processus récepteur en attente de connexion exécute la primitive ACCEPT.

### ***Transport des données***

Les connexions NITS supportent la transmission de données en 8 bits. Les données SMTP sont limitées à des caractères ASCII encodés sur 7 bits. Chaque caractère est envoyé calé à droite de l'octet de transmission, avec le bit de poids fort forcé à 0.

## **APPENDICE D**

### ***Service de transport X.25***

On pourra dans certains cas utiliser le service de transport X.25 [7] fourni par les opérateurs de télécommunications publics. Cependant, il est conseillé qu'un protocole sécurisé de bout en bout tel que TCP soit intercalé entre SMTP et la base protocolaire X.25.

# APPENDICE E

## ***Théorie des codes de réponse***

Les trois digits du code de réponse ont chacun leur signification particulière. Le premier digit qualifie globalement le sens de la réponse, en indiquant si elle est positive, négative ou incomplète. Un émetteur-SMTP simpliste pourra déterminer la prochaine action à faire (exécuter comme prévu, refaire, se replier, etc.) par la simple analyse de ce premier chiffre. Un émetteur-SMTP qui souhaite connaître la cause approximative de d'une erreur (ex., erreur du système de messagerie, erreur de syntaxe de commande) pourra examiner le second digit, le troisième étant réservé pour une analyse très fine.

Le premier digit peut prendre cinq valeurs :

### **1yz Réponse positive préliminaire**

La commande a été acceptée, mais l'action à effectuer a été mise en attente, par exemple de confirmation de l'information fournie dans cette réponse. Un émetteur-SMTP devrait dans ce cas émettre une nouvelle fois la commande pour confirmer son exécution.

[Note : SMTP ne connaît aucune commande qui admette ce type de réponse, et de ce fait ne dispose d'aucune commande Continue ou Abort.]

### **2yz Réponse positive définitive**

L'action demandée a été entièrement exécutée avec succès. Une nouvelle commande peut être reçue.

### **3yz Réponse positive intermédiaire**

La commande a été acceptée, mais le traitement est mis en attente de réception d'information complémentaire. Un émetteur-SMTP devrait émettre une nouvelle commande pour apporter le complément d'information attendu. Cette réponse est utilisée pour de groupes de séquences de commandes.

### **4yz Réponse négative temporaire**

La commande n'a pas été acceptée et l'action n'a pas eu lieu. Cependant, la condition d'erreur est temporaire, indiquant que la requête peut être présentée à nouveau. L'émetteur-SMTP devra, dans le cas de commandes groupées en séquence recommencer l'intégralité de la séquence de commandes. Il est difficile de donner une définition précise de ce que signifie "temporaire" dans la mesure où les deux sites (récepteur et émetteur) doivent s'accorder sur cette interprétation. Chacune des réponses de cette catégorie peuvent jouer avec des notions de temps assez différentes. Néanmoins, l'émetteur est encouragé à essayer de nouveau la commande. Une règle pratique pour déterminer si une réponse doit être associée à la classe 4yz ou 5yz (vois ci-dessous) est la suivante : une réponse sera de catégorie 4yz si la commande, répétée à l'identique, et sans aucun changement de configuration ou de paramètres de l'émetteur ou du récepteur, peut aboutir sous cette forme.

### **5yz Réponse négative définitive**

La commande n'a pas été acceptée et l'action n'a pas eu lieu. Cette classe de réponses décourage l'émetteur-SMTP de répéter la commande sous sa forme actuelle (dans la même séquence). Certaines conditions dites permanentes d'erreur peuvent néanmoins être corrigées, et l'utilisateur humain pourra alors redéclencher la séquence de commandes par action manuelle, lorsque la cause d'erreur aura disparu (ex., après avoir corrigé une dysorthographe, ou avoir manipulé les paramètres de configuration).

Le second digit précise la réponse dans chacune des catégories :

### **x0z Syntaxe –**

Cette série de réponse s'intéresse plus particulièrement aux erreurs de syntaxes, aux commandes syntaxiquement correctes mais qui n'ont pas de signification sémantique, ou commandes superflues ou non implémentées.



## x1z Information –

Qualifie des réponses à des demandes d'information, telles que l'aide ou un état.

## x2z Connexions –

Qualifie les réponses afférentes au canal de transmission.

## x3z Non spécifié.

## x4z Non spécifié.

## x5z Système de messagerie –

Ces réponses qualifient l'état du service de messagerie côté récepteur pour le transfert demandé ou pour toute autre action système.

Le troisième digit permet d'effectuer une distinction encore plus fine dans chacune des catégories définies par l'usage des deux premiers digits. La liste des réponses prédéfinies illustre ce fait.

Les textes associés aux réponses sont des textes préconisés, mais pas imposés, et peuvent éventuellement changer suivant la commande à l'origine de la réponse. A l'opposé, les codes de réponse doivent strictement suivre les spécifications. Les implémentations de récepteurs devront se garder d'inventer des nouveaux codes pour répondre à des situations légèrement différentes que celles présentées ici, et devront se limiter à adapter les codes prédéfinis.

Par exemple, une commande telles que NOOP dont l'exécution complète ne fournit à l'émetteur-SMTP aucune information nouvelle, renverra une réponse de code 250. La réponse sera de code 502 lorsque la commande demande d'exécuter une action non implémentée, et indépendante du site. Une réponse 504 sera donnée pour une commande implémentée, mais associée à un paramètre non implémenté.

Le texte de réponse peut être plus long qu'une simple ligne ; dans ce cas, le texte complet doit être formaté pour que l'émetteur-SMTP sache jusqu'à quel point il doit prendre en compte la réponse. Ceci nécessite l'usage d'un format particulier pour les réponses multilignes.

Le format des réponses multilignes demande que chaque ligne, sauf la dernière, commence par le code de réponse, immédiatement suivi du caractère Hyphénation, "-" (moins), le texte sur la ligne étant placé à la suite. La dernière ligne commencera aussi par le code de réponse, mais immédiatement suivi d'un espace <SP>, éventuellement la fin du texte de réponse, et enfin <CRLF>.

Par exemple :

```
123-Première ligne
123-Deuxième ligne
123-234 texte commençant par des nombres
123 La dernière ligne
```

Dans de nombreux cas, l'émetteur-SMTP n'aura qu'à rechercher le code de réponse suivi d'un <SP> au début d'une ligne, et pourra ignorer toutes les lignes précédentes. Dans quelques uns des autres cas, des informations d'importance pour l'émetteur peuvent figurer dans le "texte". L'émetteur connaît ces cas suivant le contexte.

## APPENDICE F

### Scenarios

Cette section présente des exemples de scénarios de plusieurs types de sessions SMTP.

## **Un scénario de transaction typique sous SMTP**

Ce exemple de scénario SMTP montre le cas d'un courrier envoyé par Smith, utilisateur de l'hôte USC-ISIF, à Jones, Green, et Brown, utilisateurs de l'hôte BBN-UNIX. Nous supposons ici que USC-ISIF contacte BBN-UNIX directement. Le courrier est accepté par Jones et Brown. Green ne dispose pas de boîte-aux-lettres sur BBN-UNIX.

```
R: 220 BBN-UNIX.ARPA Simple Mail Transfer Service Ready
S: HELO USC-ISIF.ARPA
R: 250 BBN-UNIX.ARPA

S: MAIL FROM:<Smith@USC-ISIF.ARPA>
R: 250 OK

S: RCPT TO:<Jones@BBN-UNIX.ARPA>
R: 250 OK

S: RCPT TO:<Green@BBN-UNIX.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@BBN-UNIX.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 BBN-UNIX.ARPA Service closing transmission channel
```

## **Scénario de transaction SMTP avortée**

```
R: 220 MIT-Multics.ARPA Simple Mail Transfer Service Ready
S: HELO ISI-VAXA.ARPA
R: 250 MIT-Multics.ARPA

S: MAIL FROM:<Smith@ISI-VAXA.ARPA>
R: 250 OK

S: RCPT TO:<Jones@MIT-Multics.ARPA>
R: 250 OK

S: RCPT TO:<Green@MIT-Multics.ARPA>
R: 550 No such user here

S: RSET
R: 250 OK

S: QUIT
R: 221 MIT-Multics.ARPA Service closing transmission channel
```

## **Scénario de relais de courrier**

Etape 1 -- Hôte source vers hôte relais

```
R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready
```

S: HELO MIT-AI.ARPA  
R: 250 USC-ISIE.ARPA

S: MAIL FROM:<JQP@MIT-AI.ARPA>  
R: 250 OK

S: RCPT TO:<@USC-ISIE.ARPA:Jones@BBN-VAX.ARPA>  
R: 250 OK

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Date: 2 Nov 81 22:33:44  
S: From: John Q. Public <JQP@MIT-AI.ARPA>  
S: Subject: The Next Meeting of the Board  
S: To: Jones@BBN-Vax.ARPA  
S:  
S: Bill:  
S: The next meeting of the board of directors will be  
S: on Tuesday.  
S: John.  
S: .  
R: 250 OK

S: QUIT  
R: 221 USC-ISIE.ARPA Service closing transmission channel

Etape 2 -- Hôte relais vers destination finale

R: 220 BBN-VAX.ARPA Simple Mail Transfer Service Ready  
S: HELO USC-ISIE.ARPA  
R: 250 BBN-VAX.ARPA

S: MAIL FROM:<@USC-ISIE.ARPA:JQP@MIT-AI.ARPA>  
R: 250 OK

S: RCPT TO:<Jones@BBN-VAX.ARPA>  
R: 250 OK

S: DATA  
R: 354 Start mail input; end with <CRLF>.<CRLF>  
S: Received: from MIT-AI.ARPA by USC-ISIE.ARPA ;  
2 Nov 81 22:40:10 UT  
S: Date: 2 Nov 81 22:33:44  
S: From: John Q. Public <JQP@MIT-AI.ARPA>  
S: Subject: The Next Meeting of the Board  
S: To: Jones@BBN-Vax.ARPA  
S:  
S: Bill:  
S: The next meeting of the board of directors will be  
S: on Tuesday.  
S: John.  
S: .  
R: 250 OK

S: QUIT  
R: 221 USC-ISIE.ARPA Service closing transmission channel

## **Scénario de vérification et de transfert**

```
R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready
S: HELO MIT-MC.ARPA
R: 250 SU-SCORE.ARPA

S: VRFY Crispin
R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

S: SEND FROM:<EAK@MIT-MC.ARPA>
R: 250 OK

S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 SU-SCORE.ARPA Service closing transmission channel
```

## **Scénario de postage et de transfert**

Dans un premier temps, le nom d'utilisateur est vérifié, puis est exécutée une tentative de transfert du message sur le terminal du destinataire. Suite à l'échec de celle-ci, le message est posté vers sa boîte-aux-lettres.

```
R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready
S: HELO MIT-MC.ARPA
R: 250 SU-SCORE.ARPA

S: VRFY Crispin
R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>

S: SEND FROM:<EAK@MIT-MC.ARPA>
R: 250 OK

S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
R: 450 User not active now

S: RSET
R: 250 OK

S: MAIL FROM:<EAK@MIT-MC.ARPA>
R: 250 OK

S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK
```

```
S: QUIT
R: 221 SU-SCORE.ARPA Service closing transmission channel
```

### ***Le même scénario exécuté de manière plus efficace***

```
R: 220 SU-SCORE.ARPA Simple Mail Transfer Service Ready
S: HELO MIT-MC.ARPA
R: 250 SU-SCORE.ARPA
```

```
S: VRFY Crispin
R: 250 Mark Crispin <Admin.MRC@SU-SCORE.ARPA>
```

```
S: SOML FROM:<EAK@MIT-MC.ARPA>
R: 250 OK
```

```
S: RCPT TO:<Admin.MRC@SU-SCORE.ARPA>
R: 250 User not active now, so will do mail.
```

```
S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK
```

```
S: QUIT
R: 221 SU-SCORE.ARPA Service closing transmission channel
```

### ***Scénario de liste de diffusion***

Tout d'abord, chacune des deux listes de diffusion est expansée sous deux sessions distinctes vers des hôtes distincts. Le message est ensuite envoyé à chacun des destinataires de chaque liste (sans duplication) via un hôte relais.

Etape 1 -- Expansion de la première liste

```
R: 220 MIT-AI.ARPA Simple Mail Transfer Service Ready
S: HELO SU-SCORE.ARPA
R: 250 MIT-AI.ARPA

S: EXPN Example-People
R: 250-<ABC@MIT-MC.ARPA>
R: 250-Fred Fonebone <Fonebone@USC-ISIQ.ARPA>
R: 250-Xenon Y. Zither <XYZ@MIT-AI.ARPA>
R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
R: 250-<joe@foo-unix.ARPA>
R: 250 <xyz@bar-unix.ARPA>

S: QUIT
R: 221 MIT-AI.ARPA Service closing transmission channel
```

Etape 2 -- Expansion de la seconde liste

```
R: 220 MIT-MC.ARPA Simple Mail Transfer Service Ready
S: HELO SU-SCORE.ARPA
R: 250 MIT-MC.ARPA
```

```
S: EXPN Interested-Parties
R: 250-Al Calico <ABC@MIT-MC.ARPA>
R: 250-<XYZ@MIT-AI.ARPA>
R: 250-Quincy Smith <@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
R: 250-<fred@BBN-UNIX.ARPA>
R: 250 <xyz@bar-unix.ARPA>

S: QUIT
R: 221 MIT-MC.ARPA Service closing transmission channel
```

### Etape 3 -- Postage vers tous les destinataires via l'hôte relais

```
R: 220 USC-ISIE.ARPA Simple Mail Transfer Service Ready
S: HELO SU-SCORE.ARPA
R: 250 USC-ISIE.ARPA

S: MAIL FROM:<Account.Person@SU-SCORE.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:ABC@MIT-MC.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:Fonebone@USC-ISIQA.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:XYZ@MIT-AI.ARPA>
R: 250 OK
S: RCPT
  TO:<@USC-ISIE.ARPA,@USC-ISIF.ARPA:Q-Smith@ISI-VAXA.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:joe@FOO-UNIX.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:xyz@BAR-UNIX.ARPA>
R: 250 OK
S: RCPT TO:<@USC-ISIE.ARPA:fred@BBN-UNIX.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 USC-ISIE.ARPA Service closing transmission channel
```

### **Scénarios de redirection**

```
R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready
S: HELO LBL-UNIX.ARPA
R: 250 USC-ISIF.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>
R: 250 OK

S: RCPT TO:<fred@USC-ISIF.ARPA>
R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>
```

```
S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 USC-ISIF.ARPA Service closing transmission channel
```

---

Etape 1 -- Essai de la première boîte-aux-lettres sur le premier hôte

```
R: 220 USC-ISIF.ARPA Simple Mail Transfer Service Ready
S: HELO LBL-UNIX.ARPA
R: 250 USC-ISIF.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>
R: 250 OK

S: RCPT TO:<fred@USC-ISIF.ARPA>
R: 251 User not local; will forward to <Jones@USC-ISI.ARPA>

S: RSET
R: 250 OK

S: QUIT
R: 221 USC-ISIF.ARPA Service closing transmission channel
```

Etape 2 -- Postage des messages sur le deuxième hôte

```
R: 220 USC-ISI.ARPA Simple Mail Transfer Service Ready
S: HELO LBL-UNIX.ARPA
R: 250 USC-ISI.ARPA

S: MAIL FROM:<mo@LBL-UNIX.ARPA>
R: 250 OK

S: RCPT TO:<Jones@USC-ISI.ARPA>
R: OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 USC-ISI.ARPA Service closing transmission channel
```

### **Scénario avec trop de récipiendaires**

```
R: 220 BERKELEY.ARPA Simple Mail Transfer Service Ready
S: HELO USC-ISIF.ARPA
```

```
R: 250 BERKELEY.ARPA

S: MAIL FROM:<Postel@USC-ISIF.ARPA>
R: 250 OK

S: RCPT TO:<fabry@BERKELEY.ARPA>
R: 250 OK

S: RCPT TO:<eric@BERKELEY.ARPA>
R: 552 Recipient storage full, try again in another transaction

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: MAIL FROM:<Postel@USC-ISIF.ARPA>
R: 250 OK

S: RCPT TO:<eric@BERKELEY.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>
S: Blah blah blah...
S: ...etc. etc. etc.
S: .
R: 250 OK

S: QUIT
R: 221 BERKELEY.ARPA Service closing transmission channel
```

Notez qu'une implémentation réelle doit accepter autant de récipiendaires que défini en section 4.5.3.

## GLOSSAIRE

### ASCII

American Standard Code for Information Interchange [1].

### Boîte-aux-lettres

Une chaîne de caractères (adresse) qui identifie un "container" de messages appartenant à l'utilisateur d'un hôte donné. La boîte-aux-lettres consiste généralement en une définition d'hôte et une définition d'un utilisateur de cet hôte. La convention de nommage pour les boîtes-aux-lettres est généralement "utilisateur@domaine".

### Canal de transmission

Un chemin de communication bidirectionnel entre un émetteur-SMTP et un récepteur-SMTP en vue de l'échange de commandes, de réponses et de messages.

### Commande

Une requête émise par un émetteur-SMTP pour déclencher l'exécution d'une action par un récepteur-SMTP.



## Emetteur-SMTP

Un processus qui transfère des messages en coopération avec un récepteur-SMTP. Un langage local peut être utilisé pour le dialogue au niveau de l'interface utilisateur. L'émetteur-SMTP prend l'initiative de l'ouverture de la connexion sur le service de transport. Il génère les commandes SMTP, reçoit les réponses, et pilote le transfert des messages.

## Hôte

Une machine située sur un réseau interconnecté hébergeant une boîte-aux-lettres et/ou exécutant un processus SMTP.

## Indicateur de fin de message

Une séquence ASCII particulière indiquant la fin des données d'un message. Dans le cas particulier de SMTP, la séquence <CRLF>.<CRLF>.

## Ligne

Une séquence de caractères ASCII terminée par un <CRLF>.

## Message

Une séquence de caractères ASCII de longueur arbitraire, conforme aux spécifications du document "Standard for the Format of ARPA Internet Text Messages" (RFC 822 [2]).

## Mot

Une séquence de caractères imprimables.

## Nom de domaine

L'adresse hiérarchisée d'un hôte sous forme de chaînes de texte.

## Récepteur-SMTP

Un processus qui reçoit les messages et travaille en coopération avec un émetteur-SMTP. Ce processus attend qu'une connexion soit ouverte via le service de transport de données. Il reçoit les commandes émanant de l'émetteur-SMTP, renvoie des réponses, et exécute les actions demandées.

## Réponse

Une réponse est un acquittement (positif ou négatif) envoyé du récepteur à l'émetteur à travers le canal de transmission, et en réponse à une commande. La forme générale d'une réponse est un code numérique (indiquant la nature de la réponse) suivi par une chaîne de caractères. Les codes numériques sont destinés à un traitement automatique, les textes sont à l'usage d'utilisateurs humains.

## Service de transport de données

Un service de communication de données fiable et basé sur un concept de flux. Par exemple, NCP, TCP, NITS.

## Session

L'ensemble des échanges effectués à partir de l'ouverture du canal de transmission.

## Transaction

L'ensemble des échanges nécessaires à la transmission d'un message vers un ou plusieurs récipiendaires.

## Utilisateur

Un être humain (ou un programme sous contrôle d'un être humain) désirant utiliser le service de messagerie. Par extension, un récipiendaire de message sur un hôte.

## <CRLF>

La séquence constituée du caractère Retour Chariot (code 13) et du caractère Nouvelle ligne (code 10).

<SP>

le caractère espace (code 32 de l'ASCII).

## REFERENCES

[1] ASCII

ASCII, "USA Code for Information Interchange", United States of America Standards Institute, X3.4, 1968. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

[2] RFC 822

Crocker, D., "Standard for the Format of ARPA Internet Text Messages," RFC 822, Department of Electrical Engineering, University of Delaware, August 1982.

[3] TCP

Postel, J., ed., "Transmission Control Protocol - DARPA Internet Program Protocol Specification", RFC 793, USC/Information Sciences Institute, NTIS AD Number A111091, September 1981. Also in: Feinler, E. and J. Postel, eds., "Internet Protocol Transition Workbook", SRI International, Menlo Park, California, March 1982.

[4] NCP

McKenzie, A., "Host/Host Protocol for the ARPA Network", NIC 8246, January 1972. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

[5] Initial Connection Protocol

Postel, J., "Official Initial Connection Protocol", NIC 7101, 11 June 1971. Also in: Feinler, E. and J. Postel, eds., "ARPANET Protocol Handbook", NIC 7104, for the Defense Communications Agency by SRI International, Menlo Park, California, Revised January 1978.

[6] NITS

PSS/SG3, "A Network Independent Transport Service", Study Group 3, The Post Office PSS Users Group, February 1980. Available from the DCPU, National Physical Laboratory, Teddington, UK.

[7] X.25

CCITT, "Recommendation X.25 - Interface Between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for Terminals Operating in the Packet Mode on Public Data Networks," CCITT Orange Book, Vol. VIII.2, International Telephone and Telegraph Consultative Committee, Geneva, 1976.