Chapitre IV Partie 5 Protocoles d'echange Mail (SMTP, POP et IMAP)

Ilyas Bambrik

Table des matières

Protocoles d'échange de mail

I

- Si le web est l'application qui domine Internet actuellement, l'échange de mails est classé en deuxième position ;
- Le système d'échange de courriers électroniques est inspiré du modèle de courrier traditionnel (*sauf que la transmission de courrier électronique est instantanée*). Deux types de protocoles composent ce système : Protocoles de transmission de mail (SMTP), et les Protocoles d'accès aux mails (POP et IMAP) ;

  L'envoie d'un mail passe par deux phases :

  1. La première phase consiste à transmettre le mail au serveur mail du destinataire (cette étape est assurée par le protocole *SMTP*) ;
  2. La deuxième phase consiste à que l'utilisateur récupère le mail depuis le serveur (cette étape est assurée par le protocole *POP* ou *IMAP*) ;

Format de mail électronique

II

- Comme d'autres protocoles (TCP, IP, FTP, HTTP, etc), les protocoles d'échange de mail fonctionnent avec un format mail. En plus du message qu'un utilisateur souhaite transmettre, *le mail aura un ensemble des entêtes qui décrivent comment le mail sera traité ainsi que son contenu et à qui il est destiné*;
- Deux formats de mail ont été conçus pour cet objectif :
    - Standard *RFC 822* (conçu seulement pour le contenu texte);
    - Format *Multipurpose Internet Mail Extensions (MIME)* (conçu pour transférer d'autres types de contenus comme les fichiers images etc);

Remarque : Standard RFC 822

- Même si le standard *RFC 822* a été conçu pour des messages textuels seulement, celui-ci permet la définition des entêtes personnalisés par l l'utilisateur. La seule contrainte est que le récepteur du message puisse interpréter l'entête personnalisé;
- Ainsi, lors du développement du format MIME, le format *RFC 822* initial a été étendu par d'autres entête spéciaux pour encoder chaque partie du message séparément ;

Multipurpose Internet Mail Extensions (MIME)

Le format MIME (Multipurpose Internet Mail Extensions) est basé sur le standard RFC 822 et ajoute des entêtes pour encoder le contenu multimedia sous forme ASCII avec des entêtes supplémentaires indiquant comment reconstruire le format multimédia à partir du contenu ASCII.

Deux types de structures de contenu MIME existent :

- *Structure Simple (Media Discret) :* le contenu du media est d'un seul type (image ou texte, etc);
- *Structure Complexe :* le message contient plusieurs types de media. Dans ce cas, le message est composé d plusieurs *body parts (dit aussi des entités MIME)* . Chaque entité MIME dans un message complexe est une *Structure Simple* ;

Avec la structure MIME, *le message entier ainsi que les parties* (pièces jointes, images, etc) qui le compose sont décrites par des entêtes MIME ;

Les entêtes MIME basiques sont les suivants :

- *MIME-Version :* Cet entête indique que le message est structuré avec le format MIME et avec quel versio de celui-ci. Cet entête est présent seulement pour le message entier (les parties qui composent le message ne contiennent pas cet entête) ;
- *Content-Type :* Indique le type du contenu du message ou d'une partie du message (exemple : *Content-Type : text/html* pour le contenu texte html ou bien *Content-Type : image/jpeg* pour le contenu image de type jpeg) ;
- *Content-Transfer-Encoding* : Pour un message d'une structure simple indique l'encodage du message. Pou un message composé, indique l'encodage de chaque partie du message ("*Content-Transfer-Encoding* `: 7bit"` pour encoder un contenu ASCII);
- *Content-ID* : Identifiant d'une partie du message (similairement au Message-ID, celui ci identifie d'une manière unique chaque partie composant un mail électronique) ;
- *Content-Description* : Entête optionnel permettant de décrire le contenu d'un message ou d'une partie d'un message ;

D'autres entêtes MIME ont été définies qui ne sont pas seulement utilisés pour les courriers électroniques (HTTP utilise ces entêtes aussi) :

- *Content-Disposition :* Indique comment le contenu est affiché chez le récepteur (`"Content-Disposition : inline"` pour indiquer que le contenu d'une entité MIME doit être affiché dans le message, `"Content-Disposition: attachment"` indique que le contenu de l'entité MIME est une pièce jointe du mail) ;
- *Content-Length* : Indique la taille du contenu de l'entité en octets ;

Complément

L'entête *Content-Type* Indique au récepteur la nature de l'entité MIME pour que celle-ci soit décodée d'une manière appropriée. La syntaxe générale de cet entête :

Content-Type: <type du contenu>/<Sous Type du contenu> [; parameter1=Valueur1 ; parameter2=Valueur2, ...; parameterN,=ValueurN]

<type de contenu> = peut prendre l'une des valeurs suivantes : text, image, audio, video ou application

<Sous Type du contenu> = le sous type spécifique de ce contenu (par exemple Content-Type: Image/PNG désigne le sous type *PNG* du type *Image*)

Par exemple :

- Content-Type: text/html ; charset=ascii
- Content-type: image/jpeg; name="ryanpicture.jpg"

Fondamental

Comme les entêtes HTTP, la ligne contenant un entête MIME ou RFC822 *se termine par un CRLF*

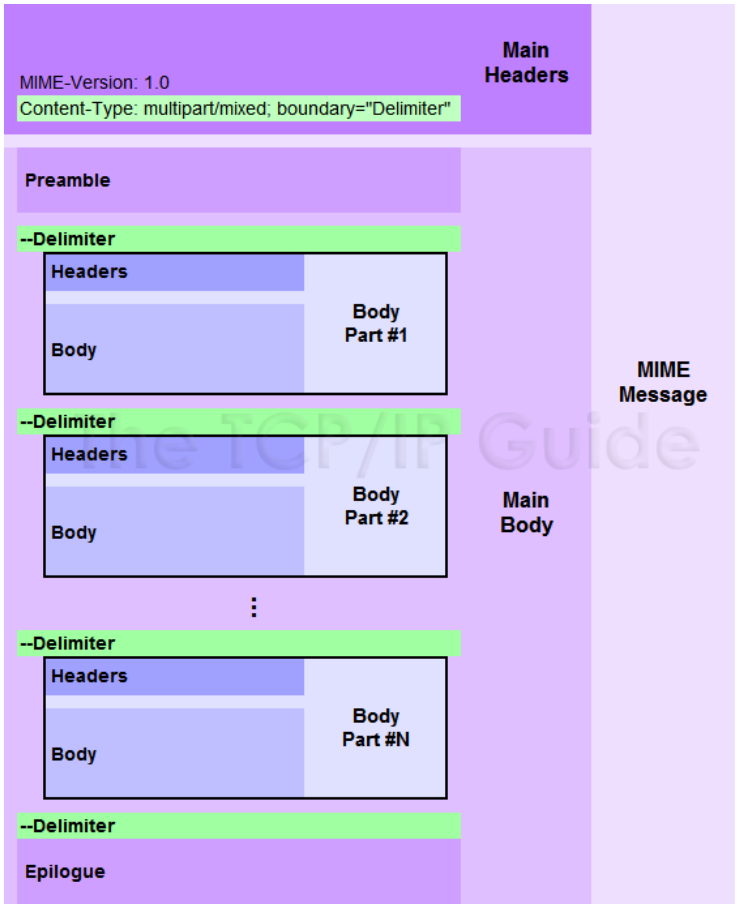
Contenu MIME composé

IV

Pour indiquer que le message contient une structure MIME complexe, l'entête Content-type prend la valeur *multipart* ou *message* (dans le cas où le message contient un seul type de media, la valeur de cet entête prend simplement texte, image, audio, vidéo ou application) ;

multipart : Indique que le message est une entité MIME contenant plusieurs entités MIME de type simple ;

message : Indique que le message incorpore un autre message (un autre mail entier / partiel par exemple);



Syntaxe : Exemple multipart mail

Un contenu de type *multipart* doit contenir la spécification d'un séparateur qui délimite la fin d'une entité MIME imbriquée et le début d'une autre. Un message contient aussi un *preamable* au début et épilogue à la fin du message qui ne seront pas traités par le serveur.

Le séparateur (*boundry delimiter*) est choisi d'une manière aléatoire pour éviter qu'il soit confondu avec le contenu du message. Pour marquer la fin d'une entité, le séparateur est précédé par *"--" ;*

Dans l'exemple suivant, boundary="exampledelimtext123" (*ligne 6*) indique que les parties du message sont séparées par "--exampledelimtext123" (lignes 8, 15, 21)

```
From: Joe Sender <joe@someplace.org>
To: Jane Receiver <jane@somewhereelse.com>
Date: Sun, 1 Jun 2003 13:28:19 –0800
Subject: Photo and discussion
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="exampledelimtext123"
This is a multipart message in MIME format
–exampledelimtext123
Content-Type: text/plain
Jane, here is the photo you wanted me for the new client.
Here are some notes on how it was processed.
(Blah blah blah...)
Talk to you soon,
Joe.
–exampledelimtext123
Content-Type: image/jpeg; name="clientphoto.jpg"
Content-Transfer-Encoding: base64
SDc9Pjv/2wBDAQoLCw4NDhwQEBw7KCIoOzs7Ozs7Ozs
...
zv/wAARCADIARoDASIAAhEBAxEB/8QAHAAAAQUBA
–exampledelimtext123
(Epilogue)
```

SMTP (Simple Mail Transfer Protocol)

V

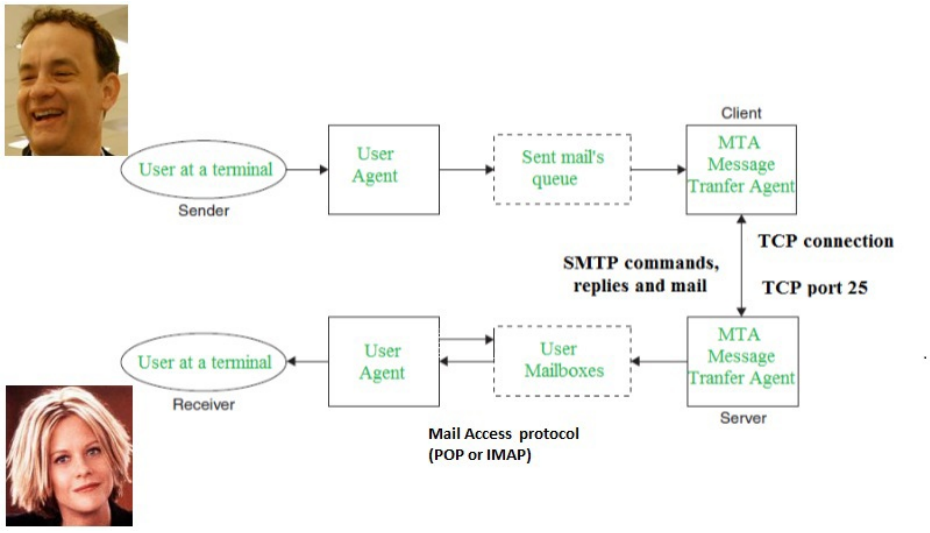
- Pour qu'une machine soit capable de recevoir un mail par SMTP, *celle-ci doit être fonctionnelle 24/7* (pour recevoir le mail au moment de son envoi). Ce genre de machine est appelée *Serveur Mail* ;
- SMTP utilise généralement le port TCP 25 ;
- SMTP utilise les enregistrements *DNS MX* (voir le cours DNS) pour récupérer le nom / adresse IP du serveur mail destinataire ;

Exemple : Utilisation DNS pour repérer le serveur mail du destinataire

Supposant qu'une personne possédant l'adresse mail *P1@gmail.com* souhaite transmettre un mail à *P2@hotmail.com*. (*P1 est P2 possèdent des boites mails dans deux différents serveur mails : gmail.com et hotmail.com*)

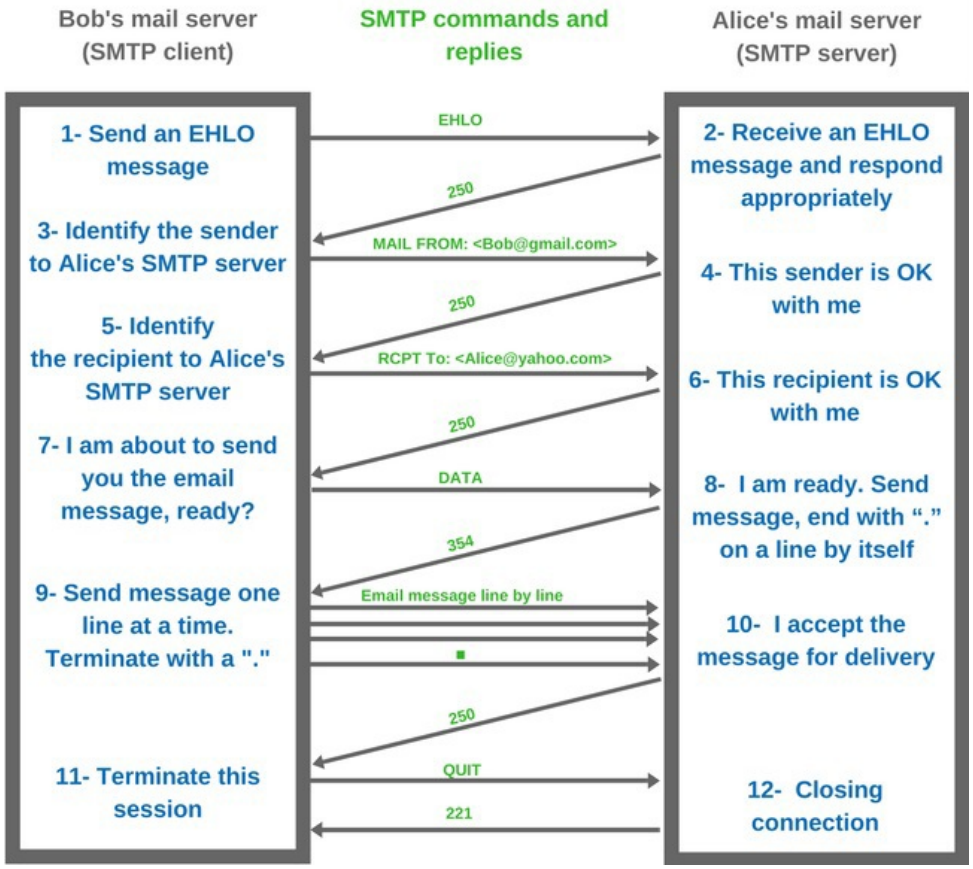
1. Ce mail passera au début par le serveur mail du domaine *gmail.com* ;
2. Par la suite, le serveur *gmail.com* relayera le mail au serveur mail du destinataire. Pour faire ceci, le serveur mail de *gmail.com* envoie une r*equête DNS pour enregistrements MX dans le domaine hotmail.com afin de repérer le serveur mail dans ce domaine* ;
3. Cette recherche retourne le nom de la machine suivant : *hotmail-com.olc.protection.outlook.com* ; Après la récupération du nom du serveur mail, le serveur mail gmail.com initie une session SMTP pour transférer le mail au serveur mail *hotmail-com.olc.protection.outlook.com* (connecte au port TCP 25) ;

Cette étape est omise si le nom de domaine du serveur mail destinataire est le même que celui de la source, par exemple P1@gmail.com envoie un mail à P2@gmail.com ;



- Un serveur transmetteur initie une session SMTP par une connexion au port 25 TCP du serveur récepteur

- Le serveur récepteur envoie automatiquement un Code 220 avec un message indiquant qu'il est prêt à recevoir des commandes SMTP;
- Le serveur transmetteur par la suite commence par envoyer une des deux commandes suivantes : *HELO* ou *EHLO* avec son nom du domaine; Par exemple, si *med@Master1.edu* souhaite transmettre un mail à *amine@Liscence3.edu* alors le serveur mail *mail.Master1.edu (le nom du serveur mail Master1.edu)* établie une connexion *mail.Liscence3.edu (le nom du serveur mail Liscence3.edu*) au port TCP/25 et ensuite une commande "*HELO mail.Master1.edu*" est envoyée de *mail.Master1.edu* vers *mail.Liscence3.edu* ;
- Le récepteur répondra avec le code 250 et son nom de domaine ;
- Si le client transmet *EHLO* au lieu de *HELO*, ceci indique que le transmetteur supporte les extensions de SMTP. En réponse, le serveur récepteur envoie les extensions supportées (par exemple : *PIPELINING* : Commande Pipe Lining pour permettre au serveur transmetteur d'envoyer plusieurs commandes sans attendre les réponses correspondantes aux commandes précédentes ; Message Size Declaration "*SIZE*" pour déclarer la taille maximal du message que le récepteur peut accepter, *DSN* : Delivery Status Notification permet de recevoir des notifications à propos de la livraison / échec du message)
- Pour fermer la session SMTP après la transmission du mail, le transmetteur envoie une commande *QUIT*. Le récepteur répond par un code *221* et un message *"good bye"* et clôture la connexion TCP;



Transmission du mail avec SMTP

VI

- Après l'établissement de la session SMTP, la transmission du mail avec SMTP passe par trois étapes :
    1. Envoie de l'adresse mail du transmetteur ;
    2. Envoie des adresses mail des récepteurs (ceux qui possèdent des boites mails dans le serveur récepteur) ;
    3. Envoie du contenu du mail ;
- L' adresse mail du transmetteur et la liste des récepteurs représente une enveloppe du contenu du mail ;
- Même si le mail lui même contient les informations nécessaires (*Standard RFC 822 les entêtes To :, From :, Cc :, Bcc :, etc*) pour livrer le mail au destinataire, la transmission des adresses des récepteurs séparément est plus efficace du coté du serveur récepteur (pour éviter le traitement supplémentaire du message car les récepteurs du mail peuvent ne pas être dans le même serveur mail);
- Le transmetteur peut vérifier la validité d'une adresse mail par la commande *VRFY* (Par exemple : transmettre "*VRFY* ilyas9111" vers yahoo.fr afin de vérifier si ilyas9111@yahoo.fr est une adresse valide dans yahoo.fr) ;
- Pour commencer la transmission du courrier, le serveur transmetteur commence par envoyer la commande *MAIL* suivie par un entête *From :* qui indique l'expéditeur du mail ;
- Par la suite, la commande *RCPT* est transmise suivie par l'entête *To :* et la liste des destinataires gérés par le serveur récepteur ;
- La commande *DATA* indique le début de la transmission du mail lui même. La fin de la transmission du mail est indiquée par la séquence "*CRLFCRLF.CRLFCRLF*";
- *La commande RSET* sert à arrêter la transmission du mail (suite à une erreur) et *NOOP* pour vérifier que l connexion avec le serveur est toujours établie ;

De l'autre coté le serveur récepteur répond avec des codes de réponse similaires à ceux générés par FTP (trois chiffres XYZ):

- X prend une valeur de 1 jusqu'à 5 (1 == Réponse positif préliminaire, 2== Commande exécutée avec succès, 3 == Réponse positif intermédiaire, 4== Échec de l'exécution qui peut être temporaire, 5 ==Échec de l'exécution permanent) ;
- Y prend une valeur de 0 jusqu'à 5 (0 == réponse relative à la syntaxe, 1== réponse à une demande d'information, 2== réponse relative à la connexion entre serveur transmetteur et récepteur, la signification de 3 et 4 n'est pas définie, 5== réponse relative au service Mail)
- Z prend une valeur de 0 à 9 et donne 10 codes par catégorie ;

Remarque : Content-encoding : base64

Les parties binaires d'un mail sont encodées avec base64 pour éliminer les caractères spéciaux (*CRLF*). Ceci empêche que le contenu transmis par l'utilisateur contient des commandes SMTP valide lors du transport mail.

Fonctionnalités de sécurité de SMTP

VII

Plusieurs fonctionnalités on été implémentées par les serveurs mail pour éliminer les failles de SMTP (cependant chaque serveur implémente ses mécanismes différemment) :

- Limite de la taille / nombre des messages acceptées ;
- Accepter les connexions seulement des appareils autorisées ;
- Vérification de la validité des mails des récepteurs avant d'accepter la réception du mail ;
- Restriction de commandes aux utilisateurs authentifiés ;
- Logging des accès au serveur ;
- Authentification avec *POP before SMTP* ;
- Plusieurs commandes ont été désactivées (par exemple *VRFY* a été désactivée car cette commande a été abusée pour le spam)