



SMTP (*Simple Mail Transfer Protocol*)

Anne Wei
CNAM

1



Plan

- Introduction
- Architecture de SMTP
- Protocoles SMTP et ESMTP
- Protocoles POP et IMAP
- Implantations

2

Références



- J. Klensin, «Simple Mail Transfer Protocol », RFC 5321, IETF oct. 2008
- **Gérard Florin**, support de cours, « La messagerie électronique – Exemple de la messagerie Internet », 2008
- P. Resnick, “Internet Message Format”, RFC 5322, IETF, oct. 2008
- N. Freed, “Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies », RFC 2045, IETF, novembre 1996
- J. Meyers, « Post Office Protocol - Version 3 », RFC 1939, IETF, mai 1996
- M. Crispin, « Internet Message Access Protocol version 4 rev 1», RFC 3501, mars 2003

3

Introduction



- Proposé en 1982, **SMTP** (*Simple Mail Transfer Protocol*) est un protocole largement déployé dans les réseaux Internet.
- SMTP *transporte* les messages sur les différents réseaux (TCP/IP ou l'autres réseaux)
- SMTP gère principalement **le serveur** de courrier électronique (*email*).
- **Le client** utilise SMTP pour émettre les messages au serveur de courrier électronique. Quant à la réception de messages électronique, le client utilise les autres protocoles tels que POP (*Post Office Protocol*), IMAP (*Internet Message Access Protocol*) et les systèmes propriétaires (*Microsoft Exchange* et *Lotus Note*)

4

Historique



- Dans les années 1960, le message électronique fonctionne en mode *point à point*. L'arrivée de l'Internet permet la connexion entre les machines sous systèmes différents dans les années 1970
- En 1982, Jon Postel a publié l'idée de SMTP (le standard RFC 821) qui permet d'échanges de message en mode *point à multipoint* en appuyant sur le mécanisme «*store and forward*»
- L'arrivée de WWW (*World Wide Web*) pousse SMTP d'offrir aux utilisateur l'authentification et la sécurité de relayer du courrier électronique. Aussi les standards RFC 2554 «SMTP Service Extension for Authentication » et RFC 2476 «Message Submission» sont publiés dans les années 1990
- Largement implémenté, SMTP a eu un nouveau standard RFC 5321 en 2008 (courrier sur mobiles, gateway...)

5

Plan



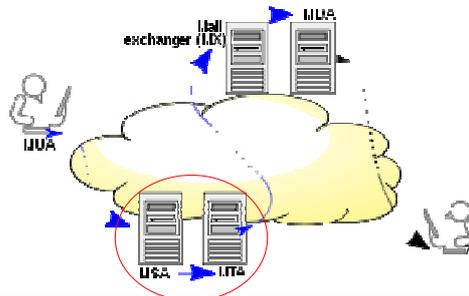
- Introduction
- Architecture de SMTP
- Protocoles SMTP et ESMTP
- Protocoles POP et IMAP
- Implantations

6

Modèle SMTP (1)



- Le modèle principal SMTP consiste en cinq parties suivantes :
- **MUA** (*Mail User Agent*) est un **client de messagerie** ; **Agent utilisateur de messagerie** soumet le courrier électronique au serveur MSA
- **MSA** (*Mail Submission Agent*) est un **serveur de messagerie et relai** qui transfère le courrier au MTA
- **MTA** (*Mail Transfer Agent*) est un **serveur de messagerie et commutateur de courriers** ; **Agent de transfert de courriers** connecte au serveur du destinataire **MX**
- **MSA et MTA** sont souvent intégrés dans un seul serveur.

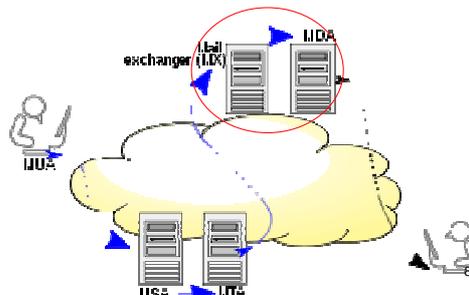


source: wikipedia

Modèle SMTP (2)



- Le modèle principal SMTP consiste en cinq parties suivantes :
- **MX** (*Mail Exchanger*) est un **serveur de messagerie du destinataire** qui accepte le courrier et le transfère au MDA
- **MDA** (*Mail Delivery Agent*) est un **serveur de messagerie du destinataire** qui délivre des messages locaux
- **MX et MDA** sont souvent intégrés dans un seul serveur.



source: wikipedia en anglais

Fonctionnement principal (1)



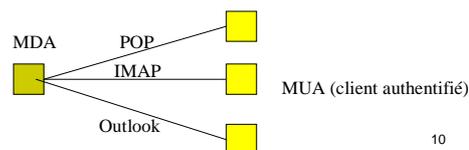
- **MUA** (*Mail User Agent*) soumet le courrier électronique au serveur MSA en passant par SMTP/TCP port 587 (ou port 25 avec l'ancienne version)
- **MSA** (*Mail Soumission Agent*) transfère le courrier au MTA
- **MTA** (*Mail Transfer Agent*) recherche d'abord la localisation du destinataire du courrier par la technique DNS (*type MX*). A l'aide du retour de RR (*Registre Record*) – nom de hôte, MTA recherche l'adresse IP (*type A*). Ensuite, MTA connecte au serveur du destinataire étant un client SMTP
- **MX** (*Mail Exchanger*) accepte le courrier et le transmet à MDA
- **MDA** (*Mail Delivery Agent*) est le responsable de courriers locaux
 - MDA enregistre les courriers en format *mailbox*
 - MDA stocke les courriers *ou* les transmet au réseau local par le protocole LMTP (*Local Mail Transfer Protocol*)

9

Fonctionnement principal (2)



- Une fois un courrier est arrivé au serveur de messagerie du destinataire (serveur local), comment retirer le courrier?
- Les protocoles de client :
- **POP** (*Post Office Protocol*)
- **IMAP** (*Internet Message Access Protocol*)
- **Systèmes propriétaires** tels que Microsoft Exchange -Outlook et Lotus Notes-Domino
- Le client (MUA) doit *s'authentifier* pour retirer ses courriers stockés dans le serveur local. Le relevé de courriers a besoin de primitives spéciales.



10



Plan

- Introduction
- Architecture de SMTP
- Protocoles SMTP et ESMTP
- Protocoles POP et IMAP
- Implantations

11



Protocole SMTP

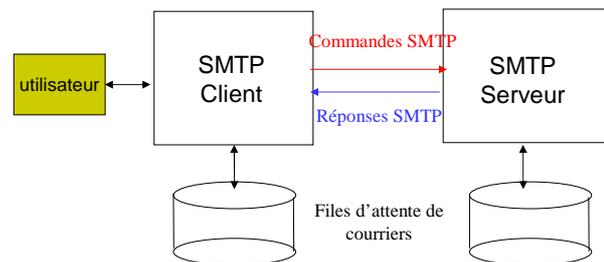
- Structure et Syntaxe d'adressage
- MIME: format des messages et des en-têtes
- Commandes (primitives)

12

Structure client-serveur



- Un utilisateur souhaite émettre un courrier. Le client SMTP se charge de trouver le destinataire en échangeant les commandes/réponses avec le serveur SMTP. Le serveur SMTP local se charge de transférer le courrier



13

Adresses de courrier



- Les adresses globales de courrier sont définies par les standards RFC 5321 et 5322
- une adresse se compose en deux parties : le **nom de boîte à lettre** et le **nom de domaine DNS**

exemple: `anne.wei@cnam.fr`

- La taille du nom de boîte à lettre ne doit pas dépasser 64 octets ; le point « . » est autorisé
- En théorie, le nom de boîte à lettre n'attache pas au login. En pratique, on utilise le login (anne.wei, par exemple) comme le nom de boîte à lettre.
- Le système de nom de domaines (*DNS*) permet de déterminer le serveur de courrier (RR de type MX).

14

Syntaxe des adresses



- La case de nom de boîte aux lettres est significative (anne.wei et Anne.WEI sont différents)
- anne.wei@cnam.fr - la forme la plus simple
- “[anne.wei](mailto:anne.wei@cnam.fr)”@cnam.fr – est égale à anne.wei@cnam.fr
- Anne Wei <anne.wei@cnam.fr> - la partie entre < et > est validée. Le reste est ignoré
- anne.wei@136.173.24.11 – avec l’adresse IP du serveur. C’est une adresse « littérale »
- [anne.wei \(département info\)@cnam.fr](mailto:anne.wei(département info)@cnam.fr) – une chaîne entre parenthèses est un commentaire donc ignorée
- Les caractères ASCII codés entre 33 et 90 (en décimal) et entre 94 et 126 sont utilisés

15

Syntaxe des adresses explicites



- Adresse avec **routage** par la source :
- <ibp.fr, @uvsq.fr: anne.wei@cnam.fr> - cette adresse indique que le courrier doit être d’abord envoyé à *ibp.fr*, ensuite à *uvsq.fr* qui enfin envoie le courrier à *cnam.fr*
- Il est fortement déconseillé d’utiliser une adresse explicite

* L’autres adresses existent dans l’autres réseaux que l’Internet

16

Format des messages



- Le format des messages/courriers est constitué de deux parties distinctes



- L'en-tête (*return-path* + *received*) : la partie *return-path* concerne les informations ajoutées à travers le routage ; la partie *received* concerne la traçabilité du chemin parcouru.
- Le corps est séparé de l'en-tête par au moins une ligne vide. Il est défini par **MIME** (*Multipurpose Internet Mail Extensions*)

MIME définit le format d'en-têtes non-US ASCII

17

Format des messages - entête



- Au moins trois lignes *obligatoires* :
 - **From** : adresse émetteur
 - **To** : adresse destinataire
 - **Date** : date de création du message
- Nombreuses autres possibilités normalisées (*In-Reply-To* : une adresse pour la réponse ; *Subject* : un sujet du message ; *Message-ID* : un identifiant du message ; *Reference* : identifiant de messages précédemment envoyés ; *cc* ; *bcc*...)
- Possibilité de créer des **en-têtes propriétaires** à condition de les faire précéder de *X-*

18

Exemple d'un entête



Delivered-To: gerard.florin@cnam.fr
Date: Thu, 21 Mar 2002 15:15:39 +0100
From: Bruno Traverson <bruno.traverson@der.edf.fr>
Organization: EDF-DER
X-Accept-Language: fr
MIME-Version: 1.0
To: th-rntl-accord@rd.francetelecom.com
Cc: TRAVERSON Bruno <Bruno.Traverson@der.edfgdf.fr>
Subject: Un premier retour sur le modèle

↕ Corps du courrier électronique constitué de lignes (de longue au plus égale à 1000 caractères)

19

Protocole SMTP



- Structure et Syntaxe d'adressage
- MIME: format des messages et des entêtes
- Commandes (primitives)

20

Objectifs de MIME



- Première version de SMTP RFC 822 ne définit que l'en-tête du message (US ASCII), alors, le corps de message doit être défini.
- Besoin élémentaire : transmettre des messages textuels qui utilisent des **jeux de caractères autres que l'US-ASCII** (par exemple ISO-Latin)
- De manière beaucoup plus générale : MIME permet la définition d'un **système très général de typage pour des documents multimédia** : textes, images, sons, tableaux, ...
- Permettre de transmettre des corps de message **comportant plusieurs parties** (message avec plusieurs attachements)
- Permettre de transmettre **des en-têtes qui n'utilisent pas l'US-ASCII**

**MIME consiste en plusieurs standards du rfc2014 au rfc 2049*

21

Types principaux de données MIME



Cinq types de données 'discrets' (avec **sous types**) par RFC 2046

- **Type texte** : données lisibles.
`text/plain [RFC2646]` ; `text/html [RFC2854]`
- **Type image** : différents codages image
`image/jpeg` ; `image/gif`
- **Type son** : différents codages 'audio'
`audio/basic (MIC mu 8000 Hz 8 bits)`
- **Type vidéo** : images animées.
`video/mpeg`
- **Type application** : les données qui restent.
`application/octet-stream` (une application à ailleurs) ; ou
`application/PostScript`

22

Les types principaux de données MIME : données *message*



- A part les cinq types cités, le type *message* est défini pour transporter dans un corps de courrier électronique un autre courrier électronique.
- *RFC822* : lorsqu'un routeur de courrier renvoie un message d'erreur
- *external-body* pour spécifier une donnée qui se trouve *ailleurs*, Dans ce cas, il faut indiquer comment faire pour obtenir la donnée
- *message partiel* : lorsqu'un message MIME est gros, il faut le découper en plusieurs morceaux

23

Les types principaux de données MIME : données composites



- **Données composites** ou assemblées (*multipart*) : plusieurs types de données sont combinés en un seul corps.
- **Cinq principaux types** : de **syntaxe identique** mais de **sémantique différente**.
 - **multipart/mixed** : les données assemblées sont indépendantes.
 - **multipart/alternative** : les données sont des alternatives d'une même information (pour l'affichage le meilleur sur un destinataire).
 - **multipart/digest** : la forme par défaut text/plain est la forme textuelle la plus simple d'un message soit text/rfc822 => permet de transférer une suite de messages ou le contenu d'une boîte aux lettres.
 - **multipart/parallel** : les données sont présentées en parallèle (texts, image).
 - **multipart/related** : les données sont reliées (comme un document HTML qui comprendrait des images incluses).

24

MIME : la représentation des contenus et l'encodage



- Pour transférer des données **quelconques** (des suites d'octets), MIME définit **cinq formats** de codage :
 - Format texte 7 bits (codé en 7 bits, SMTP)
 - Format Quoted-Printable (codé en 7 bits, SMTP, à transformer les rares caractères accentués en une séquence équivalente)
 - Format Base 64 (codé en 7 bits, SMTP)
 - Format 8 bits (codé en 8 bits, ESMTP)
 - Format binaire (codé en 8 bits, ESMTP, à transporter le son, l'image et etc.)
- Existence d'autres formats ayant été définis mais non normalisés MIME : binhex (apple), xxencode, (unix) ...

25

Complément : codage des champs d'entête



- Problème : Existence de données d'autres alphabets que l'US ASCII dans les entêtes.

- Règle d'encodage =?charset?encodage?valeur?=
Charset : le jeu de caractère,
Encodage : Q pour quoted et B pour base64,
valeur: résultat d'encodage

- Directives
MIME-version
Content-type
Content-Transfer-Encoding
Content-ID

```
From: paul@uvsq.fr
To: jean@jussieu.fr
Subject: essai de MIME
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="toto-tata-titi" ←
Ceci est un message mime
--toto-tata-titi
Content-Type: text/plain; charset=iso-8859-1 ←
Content-Transfer-Encoding: 8bit ←
Bonjour Jean, je t'envoie ci-après le texte de la
RFC sur MIME.
--toto-tata-titi
Content-Type: message/external-body; access-type=anon-ftp;
site="ftp.jussieu.fr"; ←
name="/pub/rfc/rfc/rfc1341.txt"
```

*message composite; source: Pierre David

MIME: Conclusion



- Le format **par excellence** de transmission de fichiers de **données multimédia**.
- L'utilisation pour de très **nombreux formats** de données (des centaines de types différents).
- Un mécanisme **d'extension** permettant d'intégrer en permanence de **nouveaux formats**.
- MIME est utilisé avec des modifications très mineures par le protocole **HTTP** du WEB pour ses propres échanges de données multimédia.
- Définition de **mécanismes de sécurité** pour les échanges de courrier : SMIME ('Secure MIME').

Protocole SMTP



- Structure et Syntaxe d'adressage
- MIME: format des messages et des entêtes
- Commandes (primitives)

Commandes de requêtes de client SMTP (1)



- Chaque requête (un message du protocole SMTP) correspond à une ligne de texte terminée par CRLF (‘ carriage return ’ code 13 et ‘ line feed code ’ 10).
- HELO** <SP> <domaine> <CRLF> : L’ouverture de session entre le client et le serveur (le message contient le nom de domaine FQDN du client).
- MAIL** <SP> FROM: <route-retour> <CRLF> : Définit l’adresse mail de l’émetteur (utilisé pour le retour éventuel d’erreurs).
- RCPT** <SP> TO: <route-aller> <CRLF> : Définit l’adresse d’un destinataire (le routage du courrier est possible en donnant une liste de MTA à visiter : routage par la source @Hote_1,@ Hote_2,usager@ Hote_3)
- DATA** <CRLF>: Définit l’enveloppe (l’entête) et le corps (le texte) du message.
- QUIT** <CRLF>: Termine un courrier.

29

Commandes de requêtes de client SMTP (2)



- Chaque requête (un message du protocole SMTP) correspond à une ligne de texte terminée par CRLF (‘ carriage return ’ code 13 et ‘ line feed code ’ 10).
- SEND** <SP> FROM: <route-retour> <CRLF> : L’envoi d’un message sur un terminal.
- * *SEND or MAIL* (ou « *SEND and MAIL* ») : l’envoi sur un terminal ou une boîte à lettre
- **Quelques commandes de requêtes annexes**
- RSET** : Commande pour abandonner le courrier en cours de transmission et restaurer la connexion
- VRFY** : Commande pour vérifier une adresse de destinataire sans lui transmettre de courrier (utilisable pour déterminer la cause d’un problème).
- NOOP** : Commande vide qui oblige simplement le serveur à répondre 200 OK.
- EXPN** : Expansion d’une liste de diffusion (‘mailing list’).
- TURN** : Inversion des rôles client et serveur pour envoyer du courrier dans l’autre sens sans ouvrir une nouvelle connexion TCP.

Liste des principales réponses



211 System status, or system help reply
214 Help message [Information on how to use]
220 <domain> Service ready ←
221 <domain> Service closing transmission channel ←
250 Requested mail action okay, completed ←
251 User not local ; will forward to <forward-path> ←
354 Start mail input ; end with <CRLF>.<CRLF> ←
421 <domain> Service not available, closing channel
451 Requested action aborted: local error in processing
452 Requested action not taken: insufficient storage
500 Syntax error, command unrecognized
501 Syntax error in parameters or arguments
502 Command not implemented
503 Bad sequence of commands
504 Command parameter not implemented
550 Requested action not taken: mailbox unavailable [E.g., mailbox not found, no access]
551 User not local; please try <forward-path>
552 Requested mail action aborted: exceeded storage allocation
553 Requested action not taken: mailbox name not allowed [E.g., mailbox syntax incorrect]
554 Transaction failed

31

Exemple type de transmission d'un courrier



```
sun% mail -v lo@artemis.ibp.fr
lo@artemis... Connecting to artemis.ibp.fr (TCP)... ← le récepteur signale qu'il est en ligne
220 artemis.ibp.fr Sendmail 3.2/SML-3.2 ready at Sun, 4 Mar 99 17:24:07 PST
>>> HELO liti.ibp.fr l'émetteur Machine, programme, version, configuration, date
250 artemis.ibp.fr HELO liti.ibp.fr, pleased to meet you le récepteur, 250 suivi l'@ du récepteur
>>> MAIL From:<im@liti.ibp.fr> l'enveloppe
250 <im@liti.ibp.fr>... Sender ok
>>> RCPT To:<lo@artemis.ibp.fr>
250 <lo@artemis.ibp.fr>... Recipient ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>>
Ceci est un essai
.
250 Mail accepted
>>> QUIT l'émetteur
221 artemis.ibp.fr delivering mail
eh@artemis... Sent
sun%
```

↑
↓
l'entête et le corps

32

Interactions avec DNS



- Le protocole SMTP est dépendant du DNS comme tout protocole utilisant des noms de sites

- un exemple:

```
jussieu.fr      IN      MX 10  shiva.jussieu.fr
jussieu.fr      IN      MX 20  soleil.uvsq.fr
```

Tous les mails adressés à *jeussieu.fr* vont être enregistrés dans le serveur *shiva.jussieu.fr* (en priorité). Si *shiva.jussieu.fr* est en panne, *soleil.uvsq.fr* reçoivent les mails.

- afin d'éviter faire une enquête DNS chaque fois, chaque machine pourra être configuré par

```
shiva.jussieu.fr IN      MX 0  shiva.jussieu.fr
```

33

Plan



- Introduction
- Architecture de SMTP
- Protocoles SMTP et ESMTP
- Protocoles POP et IMAP
- Implantations

34

Protocole Extended SMTP



- Extended SMTP (**ESMTP**) est créée pour le but de résoudre les limitations de SMTP
- messages sur 7 bits (tronqués explicitement à 7 bits); nom d'utilisateur < 64 caractères; nom de domaine < 64 caractères; nombre de destinataires < 100 et lignes < 1000 caractères
- ESMTP, défini par le standard RFC 1869 qui remplace RFC 1651 en 1995, supporte la compatibilité avec les implémentations de SMTP

35

Primitives de ESMTP (1)



- **EHLO** <SP> <domaine> <CRLF> (RFC 1869) Ouverture de session entre un client et un serveur en mode ESMTP. Réponse : liste des extensions supportées par le serveur si l'émetteur et le récepteur connaissent ESMTP.
- **8BITMIME** (RFC 1652) Définit un encodage en caractères 8 bits au lieu de 7: MAIL FROM: <adresse> **BODY=8BITMIME**.
- **CHUNKING** : (RFC 1830) remplace la commande DATA. Elle permet de transmettre par morceaux de BDAT.
- **BINARYMIME** (RFC 1830) Permet de transmettre des données binaires. Exemple : **BODY=BINARYMIME** le courrier est en binaire et est introduit par BDAT au lieu de DATA.
- **ETRN** (RFC 1985) Permet d'envoyer un message sur plusieurs connexions TCP successives. De plus, cette commande supporte l'émission des messages en attente pour terminaux qui ne connectent pas à l'Internet via un relai (un serveur par exemple)

36

* Voir le transparent p.25

Primitives de ESMTP (2)



- **DSN** (RFC 1891, 1892, 1894) (*Delivery Service Notification*) définit un système d'accusé de réception pour les courriers électroniques
MAIL FROM: xxx RET=portion_retournée ENVID=ident RCPT
TO: yyy NOTIFY=motif ORCPT=adresse retour
- **SIZE** (RFC 1870) définit une taille pour un courrier
MAIL FROM: xxx SIZE=nnnnn
- **PIPELINING** (RFC 2197) pour enchaîner plusieurs commandes SMTP sans attendre la réponse.
- **AUTH** (RFC 2554) introduit une authentification de l'émetteur d'un message.
- **CHECKPOINT** (RFC 1845) permet de créer des transactions de transmission de courrier repreneables en cas de panne
MAIL FROM: xxx TRANSID=yyyyy

37

Un exemple - ESMTP



```
220 soleil.uvsq.fr ESMTP Sendmail 8.12.1/jtpda-5.4 ready at Fri, 23 Nov 2001 09:30:15 (GMT)
EHL0 shiva.jussieu.fr ←
250-soleil.uvsq.fr Hello shiva.jussieu.fr, pleased to meet you
250-DSN ←
250 HELP
MAIL FROM: <jean@jussieu.fr> RET=HDRS ENVID=toto
250 <jean@jussieu.fr>... Sender ok
RCPT TO: <paul@uvsq.fr> NOTIFY=SUCCESS ORCPT=rfe822;paul@uvsq.fr
250 <paul@uvsq.fr>... Recipient ok ↑
DATA
354 Enter mail. end with "." on a line by itself
From: jean@jussieu.fr (Jean Breille)
To: paul@uvsq.fr (Paul Ochon)
Subject: essai

ceci est un essai
.
250 LAA00602 Message accepted for delivery
QUIT
```

source: Pierre David

38



Plan

- Introduction
- Architecture de SMTP
- Protocoles SMTP et ESMTP
- Protocoles POP et IMAP
- Implantations

39



Introduction

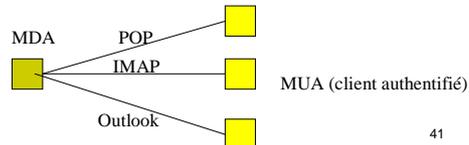
- Les protocoles POP (*Post Office Protocol*) et IMAP (*Internet Message Access Protocol*) sont les protocoles dérivés du protocole SMTP.
- Ils sont utilisé pour *relever* du courrier dans une boîte aux lettres
- Fonctions de transfert de courrier d'un serveur de messagerie vers un client de messagerie
- Fonctions de gestion des archives de courrier (liste de messages en attente, destruction de message...)

40

POP (*Post Office Protocol*)



- Le protocole POP3 (*Post Office Protocol version 3*), le standard RFC 1939, est largement implémenté. C'est un protocole le plus simple.
- **Le principe** : le client connecte au serveur pour *relever définitivement* les messages en attente via TCP (port 110) ; les messages sont transmis vers la boîte aux lettres du client.
- Afin d'éviter tous problèmes de sécurité, le client doit *s'authentifier*. Les mesures de TLS (Transport Layer Security) et SSL (Secure Socket Layer) pourront être utilisées.



41

Commandes POP3 principales



- **USER** Fourniture du nom de la boîte aux lettres
- **PASS** Fourniture du mot de passe en clair
- **APOP** Fourniture cryptée du mot de passe
- **STAT** Nombre de messages dans la boîte
- **LIST** Liste des messages présents
- **RETR** Transfert du message *n*
- **DELE** Marquage message pour la suppression
- **LAST** Numéro du dernier message consulté
- **RSET** Annulation des actions d'une session
- **QUIT** Fin de session.

42

Un exemple - POP3



```
+OK QPOP (version 2.2) at soleil.uvsq.fr starting. <4145.845637275@soleil.uvsq.fr> ←
USER pda
+OK Password required for pda.
PASS monmotdepasse
+OK pda has 73 messages (146800 octets).
STAT
+OK 73 146800
LIST 1
+OK 1 008
RETR 1
+OK 998 octets
suit le message, avec ses en-têtes, terminé par un point seul sur sa ligne
DELE 1
+OK Message 1 has been deleted.
QUIT
+OK Pop server at soleil.uvsq.fr signing off.
```

source: Pierre David

43

IMAP4 (*Internet Message Access Protocol*)



- Le protocole IMAP4 (*Internet Message Access Protocol*), le standard RFC 3501 qui remplace le RFC 2060, est largement implémenté par *client*. C'est un protocole le plus complet.
- Le principe
- deux modes (*connected and disconnected modes*) permettent à un client de connecter (ou déconnecter, les courriers récupérés durant une connexion sont dans un cache) au serveur pour *relever* les messages en attente via TCP (port 143)
- les messages sont souvent restés dans la boîte aux lettres du serveur de messagerie; ce mode permet à un utilisateur de consulter ses courriers via plusieurs machines différentes
- La boîte aux lettres du serveur peut être consultée par plusieurs clients

44

Commandes IMAP4 principales



- **AUTHENTICATE** : Mécanisme d'authentification choisi.
- **LOGIN** : Usager mot de passe.
- **LOGOUT** : Fin de session IMAP.
- **CREATE/DELETE/RENAME** : Nom de boîte aux lettres.
- **SELECT/EXAMINE** : Nom de boîte aux lettres.
- **LIST/LSUB/STATUS** : Etat de boîte aux lettres.
- **EXPUNGE/CLOSE** : Détruit les messages marqués (et ferme).
- **SEARCH** : Recherche de message sur différents critères.
- **FETCH** : Récupération des données concernant un courrier.
- **COPY** : Recopie d'un message d'une boîte aux lettres dans une autre.
- **CAPABILITY** : Liste des fonctions implantées d'un serveur.
- **NOOP** : Opération vide.

45

Plan



- Introduction
- Architecture de SMTP
- Protocoles SMTP et ESMTP
- Protocoles POP et IMAP
- **Implantations**

46

Serveurs de messagerie libre (MTA)



- **sendmail**, crée par Eric Allman en 1980 est un logiciel *Open Source*. La version récente est la 8.14.14 en 2009; 27% de marche. La technique « séparation de tâches » est implémenté afin d'éviter le problème de sécurité. (voir le fichier *sendmail.cf*)
- **Postfix**, crée principalement par Vietse Venema à IBM en 2001. 10% de marche. (voir *main.cf* et *master.cf*)
- **Qmail**, crée par Daniel. J. Bernstein en 1997. La sécurité de messages a été un des objectifs. Qmail utilise *maildir* permettant de stocker les messages client par client
- **Exim**, crée principalement par Philippe Hazel en 1997

47

Serveurs de messagerie propriétaire (MTA)



- Les messageries d'entreprise sont souvent intégrés dans des sites bureautiques ou serveurs WEB
- **Microsoft Exchange** / Internet Information Service
- **Lotus Notes/Domino** (IBM)
- **IMAIL**

48

Serveurs de délivrance de messages (MDA)



- **Gestion de boîte aux lettres**
- Stockage: plusieurs utilisateurs utilisent le même fichier dans lequel plusieurs messages sont stockés ou chaque utilisateur possède un propre répertoire
- Filtrage des messages (SPAM, par exemple) et l'envoi de message de réponse automatique
- **Logiciels différents** : *procmail*, *maildrop*, *deliver* et *mailfilter*. Des MDA sont aussi intégrés aux grands logiciels de messagerie intégrés (Exim, Exchange, par exemple)
- **Standards de boîte aux lettres** : mbox et maildir
- Les MDA incorporent les outils de protection contre les virus et le SPAM (très grande variété de produits)

49

Client de messagerie (MUA)



- **Clients lourds** : Nécessite l'installation d'une application particulière sur le poste client.
- Quelques logiciels: Outlook Express, Mozilla Thenderbird, Eudora, foxmail ...
- **Clients légers** (clients WEB WEBMAIL, hotmail, Gmail et Yahoo par exemple) : Pour consulter du courrier avec une application client serveur WEB.
- Quelques logiciels : sqWebMail, IMP, SquirrelMail, ...

50

Conclusion



- ✉ Internet est devenue la seule messagerie *fédératrice*. **Un moyen de communication majeur** dans l'entreprise, entre entreprises ou entre particuliers. Une des applications les plus **importante** de l'Internet.
- ✉ Une attente **de plus en plus importante** vis à vis de la messagerie : intégration de la circulation d'information, outils de travail coopératifs.
- ✉ Une **évolution permanente** des messageries standards ou propriétaires vers les standards Internet.
- ✉ Problème par le courrier non sollicité (spam)
- ✉ Problèmes de **sécurité de la messagerie**
 - Sécurisation en **confidentialité, intégrité, authentification** (S-MIME)
 - Vulnérabilité de ce média aux **différentes attaques (virus)**.