Le système de messagerie d'Internet

Bernard Cousin

Université de Rennes I – laboratoire IRISA

http://www.univ-rennes1.fr/

Plan

- Introduction aux systèmes de messagerie
- Le protocole SMTP
  - L'architecture
  - Le protocole
  - Le format des messages
- Le format MIME
- Les protocoles POP et IMAP

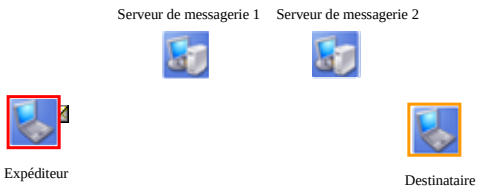
Introduction

- Système de messagerie :
  - Transmettre un message de manière efficace et fiable, entre un expéditeur et un destinataire.

  … Mais …
  - Le destinataire peut ne pas être actif lors de l'expédition ("off-line")
  - Transmettre un message de manière efficace et fiable, entre un expéditeur et la boite aux lettres d'un serveur de messagerie.

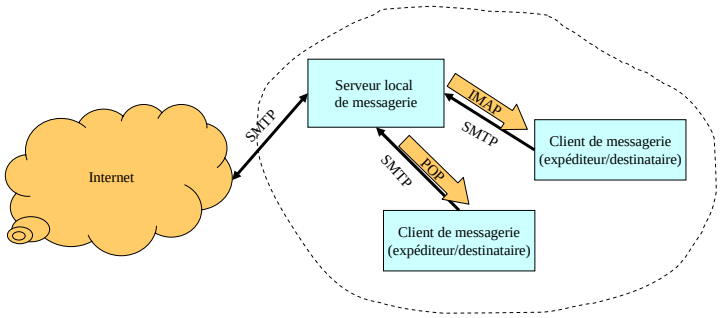
  …Mais…
  - Les serveurs de messagerie relais utilisés lors du transfert peuvent ne pas être actifs.
- Les acteurs
  - Expéditeur
  - Des serveurs de messagerie intermédiaires (obsolète)
  - Un serveur de messagerie
  - Boite aux lettres <==> destinataire du système de messagerie

Exemple de fonctionnement du système de messagerie



Environnement de messagerie typique

IRISA



- Les protocoles de récupération de messages
  - POP ou IMAP
- Le protocole de transfert de messages
  - SMTP

Quelques remarques

IRISA

- Exemple d'un protocole de niveau supérieur
  - Niveau Application + niveau Présentation
  - En mode lignes de caractères !
    - Mise au point par émulation aisé (par "telnet" : simulation de l'expéditeur ou du récepteur), interprétation aisée des messages (par "tcpdump" sans développement spécifique)
  - Autres exemples : FTP, HTTP, HTML !, "Cisco IOS command language"
- Quelques services bien connus et d'autres plus originaux
  - Liste de diffusion, client/serveur
  - Notion de système non connecté en permanence
  - Serveurs intermédiaires (relai/passerelle), couche Présentation
- SMTP : Protocole ancien (1982) mis à jour en avril 2001 (rfc 2821)
  - 5% du trafic d'Internet

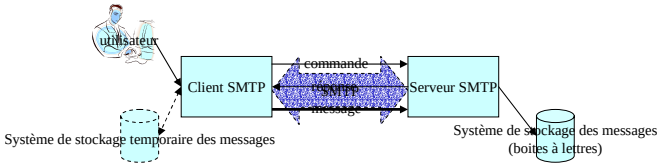
Le protocole SMTP

- "Simple message transfer protocol"
  - Rfc 821 => rfc 974 => rfc 1869 => rfc 2821 : SMTP (avril 2001)
  - Rfc 822 => rfc 2822 : "Internet message format" (avril 2001)
  - Rfc 1891 "Extended SMTP with delivery status notification"
- SMTP est indépendant du protocole de transport
  - … mais les implémentations actuelles utilisent …TCP
- SMTP peut fonctionner au-dessus d'un ensemble hétérogène de réseaux
  - Un message peut transiter par des serveurs de messagerie intermédiaires : ils effectuent un relayage/transformation des messages

Le protocole SMTP

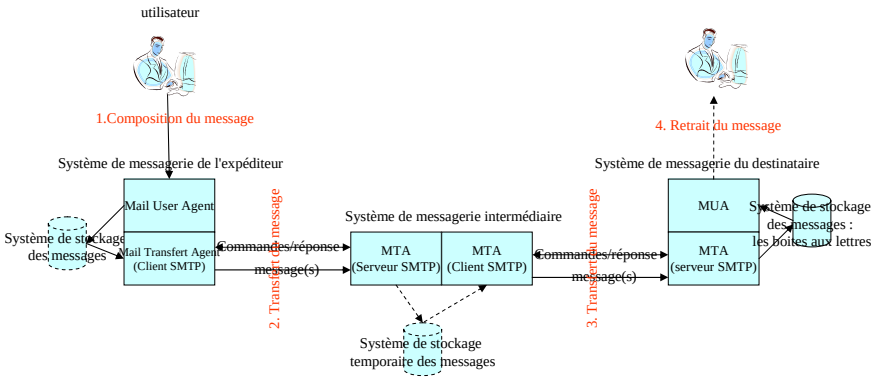
- Simple
  - Dialogue à l'alternat : une seule commande/réponse à la fois
    - Extension possible pour des commandes pipelinées
- Client-serveur
  - Protocole TCP : port 25
  - Client vers serveur : une commande
  - Serveur vers client : un code de retour (réussite, échec temporaire ou permanent).
- Efficacité :
  - SMTP peut envoyer un même message à plusieurs destinataires en transmettant une seule copie du contenu

Architecture de base



- Exemples
  - Clients de messagerie :
    - (graphiques) Eudora, Outlook, (commandes) mail, pine, elm, etc
  - Serveurs de messagerie :
    - Sendmail (le plus courant), postfix, Qmail, etc.

Autre architecture



- MTA, MUA : terminologie X.400

Client SMTP

IRISA

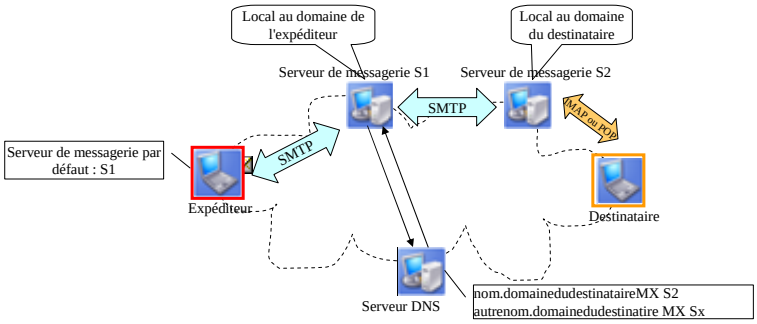
- Le client SMTP peut appartenir
  - soit à l'expéditeur initial
  - soit à un système intermédiaire (relai STMP ou passerelle vers un réseau non SMTP)
- Le client SMTP peut
  - Vérifier les adresses
  - Traiter les listes d'adresses
  - Transformer le message (dans le cas d'une passerelle)
- Le client SMTP doit
  - Assurer la conservation du message, tant qu'il n'est pas totalement délivré au serveur
- Le nom du serveur SMTP est obtenu soit
  - Par défaut au niveau du client initial : un serveur local au domaine du client
  - Grâce au DNS (enregistrement MX) en utilisant le domaine figurant dans l'adresse: un serveur probablement local au domaine du destinataire
  - Par le "source routing" (obsolète)

Serveur SMTP

IRISA

- Le serveur SMTP peut appartenir
  - soit à l'un des destinataires finaux
  - soit à un système intermédiaire (relai STMP ou passerelle vers un réseau non SMTP)
- Le serveur intermédiaire peut
  - Vérifier les adresses
- Le serveur SMTP doit
  - Notifier le résultat de l'expédition du message à l'expéditeur
  - Ajouter des informations de trace dans l'entête du message

Exemple de fonctionnement du système de messagerie



Boite aux lettres

- Endroit où le message doit être déposé :
  - Identifie en quelque sorte le destinataire du message
  - L'adresse de la boîte aux lettres est une chaîne de caractères ASCII (sauf "@", "<", ">", ",", ";", etc) :
    - local-part>@<domain
      - Ex : bcousin@ifsic.univ-rennes1.fr
  - La partie locale est traitée localement
  - Le domaine doit être de type FQDN
  - La notation entre crochets
    - Ex : Bernard Cousin <bcousin@ifsic.univ-rennes1.fr>
- Historiquement on faisait du "source routing", on indiquait la liste des serveurs de messagerie par lesquels passer pour parvenir au destinataire (il n'y avait pas de DNS).
- Liste de diffusion (nom d'une liste de boites aux lettres)
  - Permet d'envoyer un message à plusieurs destinataires sans dupliquer inutilement le contenu du message

Nom de domaines

IRISA

- FQDN : "Fully qualified domain name"
  - Une liste de labels séparées par points
    - Ex : bcousin@ifsic.univ-rennes1.fr
  - Lettres, chiffres ou tiret du jeu de caractères ASCII
  - Conforme avec DNS (MX RR, A RR, ou CNAME RR)
  - Rem : Les alias DNS sont traités avant SMTP par DNS

Les commandes de SMTP

IRISA

- Format général d'une commande
  - Une ligne de caractères ASCII (terminé par les 2 caractères <CRLF>)
  - La ligne commence par un mot clef qui définit le type de la commande
  - Certaines commandes sont optionnelles (cf. ESMTP)
  - Cela peut être négocié lors de l'établissement
- Exemple:
  - HELO <serveur> <CRLF>
- Remarque : insensible aux majuscules/minuscules
- Temporisateur :
  - Par défaut 2 à 10 minutes en fonction des commandes
  - Compatible avec la durée des temporisateurs de TCP

Les commandes de SMTP

IRISA

Commande	syntaxe	fonction
EHLO	EHLO <domain> [<SP> <add-param>]	Demande d'établissement d'une session de messagerie SMTP avec le serveur du domaine cité
MAIL FROM:	MAIL FROM: <exp-path> [<SP> <add-param>]	Demande d'un envoi d'un message Identification de l'expéditeur (le chemin vers l'expéditeur)
RCPT TO:	RCPT TO: <dest-path> [<SP> <add-param>]	Identification d'un destinataire ou "postmaster"
DATA	DATA	Délimite le début du message
.	.	Délimite la fin du message
QUIT	QUIT	Demande de libération de la session
RESET	RESET	Annulation des commandes précédentes (d'envoi d'un message, par ex.)
HELP	HELP [<string>]	Le serveur retourne des informations utiles

Les autres commandes

IRISA

Commande	syntaxe	fonction
HELO	HELO <domain>	Demande d'établissement d'une session de messagerie avec commandes (obsolète)
NOOP		"No operation"
SEND	SEND FROM: <exp-path>	Envoi d'un message au terminal (obsolète)
SOML	SOML FROM: <exp-path>	Envoi du message au terminal sinon au serveur responsable de la BAL (obsolète)
SAML	SAML FROM: <exp-path>	Envoi du message au terminal et au serveur responsable de la BAL (obsolète)
VRFY	VRFY <string>	Vérification de la chaîne de caractères
EXPN	EXPN <string>	Expansion de la chaîne de caractères
TURN		Echange les rôles entre le client et le serveur
X…		Commande étendue définie localement

Vérification

IRISA

- Expansion d'une liste de noms
  - Commande EXPN
    - Expansion d'une chaîne de caractères : "mailing list"

```
C: EXPN exemple-email-list
S: 250-John Smith <jsmith@foo.com>
S: 250 Harry Smith <hsmith@foo.com>
```

- Vérification d'un nom
  - Commande VRFY

```
C: VRFY smith
S: 553-Ambiguous: Possibilities are
S: 553-John Smith <jsmith@foo.com>
S: 553-Harry Smith <hsmith@foo.com>
```


Réponses SMTP

IRISA

- La réponse SMTP
  - Une ligne terminée par <CRLF> suivie
    - D'un code numérique de 3 chiffres (pour le client SMTP)
    - d'une chaîne de caractères (pour l'utilisateur)
  - Remarque: codes similaires à HTTP
  - Ex: `220 foo.com ready`
- Réponse multi-ligne:
  - Le code numérique est suivi d'un tiret
  - Ex:

```
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-DSN
S: 250 HELP
```

Réponses SMTP

IRISA

- Sémantique des chiffres du code
  - Premier chiffre
    - 1xy : exécution préliminaire réussie (pas utilisé par HELO)
    - 2xy : exécution réussie
    - 3xy : exécution préliminaire réussie (par ex. DATA)
    - 4xy : échec temporaire de l'exécution
    - 5xy : échec définitif de l'exécution
  - Deuxième chiffre
    - x1y : syntaxe
    - x2y : information
    - x3y : connexion
    - x5y : système de messagerie

Réponses SMTP

IRISA

500	Syntax error
501	Parameter syntax error
502	bad sequence of commands
504	Parmeter not implemented
211	System status
214	Help message
220	<domain> Service ready
221	<domain> Service closing
421	<domain> Service not available
250	Service action completed
251	User not local; will forward to <forward-path>
252	Cannot VRFY user, but will accept message and attempt delivery
450	Requested mail action not taken: mailbox busy
550	Requested action not taken: mailbox not found, no access for policy reasons
451	Requested action aborted: error in processing
551	User not local; please try <forward-path>
452	Requested action not taken: insufficient system storage
552	Requested mail action aborted: exceeded storage allocation
354	Start mail input; end with <CRLF>.<CRLF>
554	Transaction failed

Le message

IRISA

- Un message comporte :
  - Une enveloppe
    - L'adresse d'un expéditeur
      - Sert lors d'un erreur
    - Un ou plusieurs destinataires
      - Exemple : commande RCPT
  - Un contenu
      - Historiquement au format texte (cf. rfc 822)
        - Séquence de lignes de caractères utilisant l'US-ASCII
      - Au format MIME
    - Une entête
      - Une liste de paires <mot-clef> ":" <valeur>
      - Terminée par une ligne blanche
    - Un corps

Les entêtes du contenu

IRISA

- Les entêtes les plus courantes :
  - `"From: "`
  - `"To: "`
  - `"Subject: "`
  - `"Date: "`
  - `"Cc: "`
  - `"Message-ID: "`
  - `"Received: "from xxx by xxx for xxx`
  - Etc.
  - Remarque: l'ordre des entêtes n'a pas d'importance
  - Exemple : cf. la suite

L'entête "received from..."

IRISA

- Type d'entête ajouté par chaque serveur intermédiaire
  - Par exemple:

```
Received: from CGNET.COM by Arizona.EDU (PMDF V4.3-9 #2381) id <01HGUMM9OTUO9AR7DY@Arizona.EDU>; Thu, 08 Sep 1994
    00:39:13 -0700 (MST)
Received: from faop.cgnet.com by CGNET.COM (PMDF V4.3-9 #7702) id <01HGUMN7N4S000370I@CGNET.COM&>; Thu, 08 Sep 1994
    00:40:08 -0700 (PDT)
Received: from msmail.fao.org (191.0.1.130) by FAOVMS.CGNET.COM (PMDF V4.3-8 #3703) id
    <HGV4ZD1XTC8WW39N@FAOVMS.CGNET.COM>; Thu, 08 Sep 1994 09:25:10 +0200
Received: by msmail.fao.org with Microsoft Mail id <B79C6AC@msmail.fao.org>; Thu, 08 Sep 94 09:24:12 +02</H2>
```

  - Permet suivre à la trace le chemin suivi par le message :
    - cgnet.com -> faop.cgnet.com -> msmail.fao.org
  - Peut permettre de vérifier si le domaine associé à la commande "hello" est bien celui figurant dans le dernier entête "received from…" du message
    - L'adresse 191.0.1.130 devrait correspondre à la station msmail.fao.org
  - Permet de détecter les boucles de messages
  - L'ID du message permet de détecter les copies d'un même message
    - Détection de "mail bombing"

Scénario SMTP : normal

IRISA

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RCPT TO:Brown@foo.com
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Scénario : abort

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<Smith@bar.com>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: RCPT TO:<Green@foo.com>
S: 550 No such user here
C: RSET
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```


Scénario : relayage 1ère étape

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250 HELP
C: MAIL FROM:<JQP@bar.com>
S: 250 OK
C: RCPT TO:<@foo.com:Jones@XYZ.COM>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Date: Thu, 21 May 1998 05:33:29 -0700
C: From: John Q. Public <JQP@bar.com>
C: Subject:  The Next Meeting of the Board
C: To: Jones@xyz.com
C:
C: Bill:
C: The next meeting of the board of directors will be
C: on Tuesday.
C:                         John.
C: .
S: 250 OK
C: QUIT
: 221 foo.com Service closing transmission channel
```

Scénario : relayage 2ème étape

```
S: 220 xyz.com Simple Mail Transfer Service Ready
C: EHLO foo.com
S: 250 xyz.com is on the air
C: MAIL FROM:<@foo.com:JQP@bar.com>
S: 250 OK
C: RCPT TO:<Jones@XYZ.COM>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Received: from bar.com by foo.com ; Thu, 21 May 1998
C:     05:33:29 -0700
C: Date: Thu, 21 May 1998 05:33:22 -0700
C: From: John Q. Public <JQP@bar.com>
C: Subject:  The Next Meeting of the Board
C: To: Jones@xyz.com
C:
C: Bill:
C: The next meeting of the board of directors will be
C: on Tuesday.
C:                          John.
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```


Scénario : vérification

```
S: 220 foo.com Simple Mail Transfer Service Ready
C: EHLO bar.com
S: 250-foo.com greets bar.com
S: 250-8BITMIME
S: 250-SIZE
S: 250-DSN
S: 250-VRFY
S: 250 HELP
C: VRFY Crispin
S: 250 Mark Crispin <Admin.MRC@foo.com>
C: SEND FROM:<EAK@bar.com>
S: 250 OK
C: RCPT TO:<Admin.MRC@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: Blah blah blah...
C: .
S: 250 OK
C: QUIT
S: 221 foo.com Service closing transmission channel
```

Conclusion

IRISA

- SMTP est un protocole de transmission de messages :
  - Adaptation aux besoins de l'Internet
    - Rfc 821 (1982) => rfc 2821 (2001)
  - Respect de la compatibilité, même avec les versions historiques qui ne respectaient pas strictement la rfc
  - Exemple de protocole de couche Application
    - Transmission en mode caractères

Autres services de messagerie

IRISA

- Filtre de messages
  - Interception de certains messages
- Passerelle de messagerie
  - Traduction du format d'un message en un autre format
    - X400
  - Duplication d'un message
- Sécurité
  - S/MIME
- Les webmail

La rfc 822

- La rfc 822
  - Rappel : le rfc 822 date de 1982
  - Adapté aux terminaux alphanumériques et aux transmissions par caractères
    - Codage des caractères : 7bit US-ASCII
    - Lignes de 1000 caractères max. (<CRLF> inclus)

MIME

- Multipurpose Internet Mail Exchange
  - Le rfc 822 est limité
  - Extension pour transmettre des messages :
    - Jeux de caractères autres que US-ASCII (dans l'entête (rfc 2047) ou le corps du message)
    - Corps du message au format non textuel :
      - Contenu multimédia du message (rfc 2046)
    - Corps du message en plusieurs parties
  - Dernière version
    - Rfc 2045 (nov 1996), 2046, 2047, 2048, 2049
  - Le problème de MIME :
    - La compatibilité avec rfc 822 et suivant (rfc 1421)

Mécanismes de MIME

IRISA

- MIME-version pour compatibilité
  - Par ex. : `MIME-Version: 1.0`
- Le champ d'entête "Content-type"
  - Permet à l'UA de choisir la technique la plus appropriée de présentation du corps du document à l'utilisateur
- Le champ d'entête "Content-transfert-encoding"
  - Nécessaire car certains protocoles n'admettent que des messages codées sur 7 bits
- "Content-ID",
  - Permet d'identifier que le contenu de plusieurs messages forme les parties d'un tout
- "Content-description"
  - Une description des parties, notamment celles non lisibles
    - ex: audio

Les contenu de MIME

IRISA

- Le champ d'entête "Content-type"
  - Permet à l'UA de choisir la technique la plus appropriée de présentation du corps du document à l'utilisateur
  - Format:
    - `<Type> "/" <subtype> [";" <parameters>]`
    - Ex1 : `Content-type: text/plain`
    - Ex2 : `Content-type: text/plain; charset=ISO-8859-1 (commentaire)`
  - Par défaut : `Content-type: text/plain; charset="us-ascii"`
- Les types
  - Les types discrets :
    - `Video/mpeg, audio/basic, text/(plain, richtext, enriched), image/(gif, jpeg), application/(octet-stream, postscript), x../..`
  - Les types composites :
    - `Message/(822, partial, external-body), multipart/(mixed, alternative, parallel, digest), x.../..`

Multipart type

IRISA

- Le corps du message contient des parties indépendantes
  - Un paramètre obligatoire défini la ligne de séparation entre les parties
    - "boundary= "<valeur de la ligne de séparation >
  - Chaque ligne de séparation débute par 2 tirets
  - La dernière ligne de séparation se termine par 2 tirets
  - Exemple:

```
From: bernard Cousin <bcousin@ifsic.univ-rennes1.fr>
To: etudiants@univ-rennes1.fr
Subject: exemple
Mime-Version: 1.0
Content-type: multipart/mixed; boundary="une simple limite"

--une simple limite
La première partie, utilisant implicitement le format "plain ASCII text".
--une simple limite
Content-type: text/plain; charset= us-ascii

La deuxième partie, utilisant explicitement le format "plain ASCII text".
--une simple limite--
```


Les encodages MIME

IRISA

- Compromis entre
  - Compacité du code, lisibilité, complexité du traitement, compatibilité
- Format

```
"Content-Transfer-Encoding" ":" mechanism

mechanism := "7bit" / "8bit" / "binary" / "quoted-printable" /
  "base64" / ietf-token / x-token
```

  - Sans segmentation ni transformation : "binary"
  - Segmentation sans transformation : "7bit" / "8bit"
  - Segmentation et transformation : "quoted-printable" / "base64"

  - Par défaut : Content-Transfer-Encoding: 7BIT

Le codage "quoted-printable"

- Codage d'une suite d'octets en une suite de caractères qui peut être lisible par un être humain
  - Les caractères doivent être alphanumériques (imprimables)
  - Pas de codage pour les octets représentant un caractère imprimable (format US-ASCII):
    - C-à-d les octets dont la valeur est [33, 60] ou [62, 126]
  - Pour les autres valeurs d'octet : encodage par 3 octets
    - Le caractère "=" suivi des chiffres ASCII de la représentation hexadécimale de la valeur de l'octet à encoder
    - Ex : le caractère US-ASCII "form feed" (12) est codé "=0C"
  - Les octets <CRLF> de fin de ligne ne sont pas encodés, les autres le sont.
  - Les lignes au format "quoted-printable" ne font pas plus de 76 caractères (raison historique : écran de 80 caractères)
    - Un "=" précédent un <CRLF> est un "soft line break" utilisé pour ce formatage

Exemple "quoted-printable"

- Exemple
  - Contenu initial :

```
Now's the time for all folk to come to the aid of their country.
```

  - Contenu final :

```
Now's the time =
for all folk to come=
to the aid of their country.
```

Le codage "Base64"

IRISA

- Principe du codage
  - Un ensemble de 3 octets successifs (24 bits) est lu comme 4 blocs de 6 bits (24 bits)
  - Chaque bloc de 6 bits est alors transformé en un caractère (sur 8 bits): taux d'expansion = +33%
    - [0, 25] : majuscules de "A" à "Z"
    - [26, 52] : minuscules de "a" à "z"
    - [52, 61] : chiffres de "0" à "9"
    - [62, 63] : "+", "/"
    - [64] : "=" (caractère spécial)
  - On forme des lignes d'au plus 76 caractères
  - On complète avec des 0 pour avoir un dernier bloc (de 6 bits) complet.
  - On complète par 0, 1 ou 2 "=" pour transmettre un nombre total de caractères multiples de 4.

Exemple de codage "Base64"

IRISA

- Données initiales:

  `00100011 01011100`
- Codage
  - Blocs de 6 bits

    `001000 110101 110000`

    `8, 53, 48,`
  - Encodage

    `"I", "1", "w", "="`
  - Codage US-ASCII

    `Ox49 0x31 0x77 0x3D`
- Données finales (avec bit de poids fort à zéro)

  `01001001 00110001 01110111 00111101`

S/MIME

- Secure MIME
  - MIME a été sécurisé par le rfc 1847
  - Utilise le type "multipart" et crée 2 sous-types:
    - "Signed", "encrypted"
  - Le sous-type "signed":
    - Pour l'authentification ou l'intégrité
    - 3 paramètres obligatoires : boundary, protocol, micalg
  - Le sous-type "encrypted"
    - Pour la confidentialité
    - 2 paramètres obligatoires : boundary, protocol

POP

- Post Office Protocole
  - Rfc 1939 (version 3)
  - Utilise le protocole TCP (port 110)
  - Plus simple qu'IMAP : gestion de répertoires
- Sous forme de Commande/réponse
- Commandes : 4 lettres
  - `STAT` : nombre et taille des messages
  - `LIST [msg]` : donne des infos sur [le ou] tous les messages
  - `RETR msg` : récupère le message
  - `DELE msg` : detruit le message
  - `USER nom` : ouvre la session, spécifie la boite aux lettres
  - `PASS passwd` : spécifie le mot de passe associé
  - `QUIT` : quitte la session
- Reponses :
  - `+OK` = positive; `-RR` = négative

Exemple de session POP

IRISA

```
$ telnet/port=110 mail.opus1.com
Trying... Connected to MAIL.OPUS1.COM.
+OK cello.Opus1.COM MultiNet POP3 Server Process V4.0(1) at Fri 20-Sep-96 3:21PM-MST
user cousin
+OK User name (cousin) ok. Password, please.
pass monmotdepasseenclair
+OK 3 messages in folder NEWMAIL (V4.0)
list 2
+OK 2 7124
stat
+OK 3 14749
last
+OK 0
quit
+OK POP3 MultiNet cello.Opus1.COM Server exiting (3 NEWMAIL messages left)
Connection closed by Foreign Host
$
```

- ‘list’ donne la taille d'un message (en octets)
- ‘stat’ donne la taille total des messages (en octets)

IMAP

IRISA

- "Internet Message Access Protocol"
  - IMAP4 rev1 : RFC 3501(mars 2003)
  - Services :
    - Accès aux messages des boites aux lettres d'un serveur de messagerie
    - Manipulation des boites aux lettres
      - Création, destruction, changement de nom
      - Vérification de messages, destruction permanente de messages
      - Positionnement de "flags"
      - Analyse du contenu de message et recherche
    - Hiérarchie des noms de boites aux lettres
    - Autorise des accès simultanés à la même boite aux lettres
    - Les messages peuvent être identifiés
      - Par "Unique Identifier" (UID) : absolu
      - Un numéro de message : relatif à la boite aux lettres et à la session
    - Les messages sont munis d'attributs:
      - Vu, répondu, détruit, brouillon, récent, etc.
    - Les opérations sont contrôlées :
      - Plusieurs procédures d'authentification :
        - login/password, TLS, phase d'authentification spécifique, et des extensions

Commandes et réponses d'IMAP

IRISA

- Commandes et réponses
  - "Tag" : chaque commande est identifiée, ce Tag est utilisée par la réponse
  - Certaines réponses sont "untagged" : "*"
  - Commandes:
    - Listes des commandes
      - Commandes disponibles quelque soit l'état
        - CAPABILITY, NOOP, LOGOUT
      - Commandes dans l'état "non authentifié" :
        - LOGIN, STARTTLS, AUTHENTICATE
      - Commandes dans l'état "authentifié"
        - SELECT, EXAMINE, CREATE, DELETE, RENAME, SUBSCRIBE, UNSUBSCRIBE; LIST, LSUB, STATUS, APPEND
      - Commandes dans l'état "sélectionné"
        - CHECK, CLOSE, EXPUNGE, SEARCH, FETCH, STORE, COPY, UID
  - Reponses:
    - Listes des réponses
      - OK, NO, BAD, PREAUTH, BYE,
    - Réponses spécifiques de certaines commandes
      - CAPABILITY, LIST, LSUB, STATUS, SEARCH, FLAGS, EXISTS, RECENT
      - EXPUNGE

Exemple IMAP

IRISA

```
S:   * OK IMAP4rev1 Service Ready
C:   a001 login mrc secret
S:   a001 OK LOGIN completed
C:   a002 select inbox
S:   * 18 EXISTS
S:   * FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
S:   * 2 RECENT
S:   * OK [UNSEEN 17] Message 17 is the first unseen message
S:   * OK [UIDVALIDITY 3857529045] UIDs valid
S:   a002 OK [READ-WRITE] SELECT completed
C:   a003 fetch 12 full
S:   * 12 FETCH (FLAGS (\Seen) INTERNALDATE "17-Jul-1996 02:44:25 -0700"
      RFC822.SIZE 4286 ENVELOPE ("Wed, 17 Jul 1996 02:23:25 -0700 (PDT)"
      "IMAP4rev1 WG mtg summary and minutes"
      (("Terry Gray" NIL "gray" "cac.washington.edu"))
      (("Terry Gray" NIL "gray" "cac.washington.edu"))
      (("Terry Gray" NIL "gray" "cac.washington.edu"))
      ((NIL NIL "imap" "cac.washington.edu"))
      ((NIL NIL "minutes" "CNRI.Reston.VA.US")
      ("John Klensin" NIL "KLENSIN" "MIT.EDU")) NIL NIL
      "<B27397-0100000@cac.washington.edu>")
       BODY ("TEXT" "PLAIN" ("CHARSET" "US-ASCII") NIL NIL "7BIT" 3028
       92))
S:   a003 OK FETCH completed
C:   a004 fetch 12 body[header]
S:   * 12 FETCH (BODY[HEADER] {342}
S:   Date: Wed, 17 Jul 1996 02:23:25 -0700 (PDT)
S:   From: Terry Gray <gray@cac.washington.edu>
S:   Subject: IMAP4rev1 WG mtg summary and minutes
S:   To: imap@cac.washington.edu
S:   cc: minutes@CNRI.Reston.VA.US, John Klensin <KLENSIN@MIT.EDU>
S:   Message-Id: <B27397-0100000@cac.washington.edu>
S:   MIME-Version: 1.0
S:   Content-Type: TEXT/PLAIN; CHARSET=US-ASCII
S:
S:   )
S:   a004 OK FETCH completed
C:   a005 store 12 +flags \deleted
S:   * 12 FETCH (FLAGS (\Seen \Deleted))
S:   a005 OK +FLAGS completed
C:   a006 logout
S:   * BYE IMAP4rev1 server terminating connection
S:   a006 OK LOGOUT completed
```