MASTER EN SCIENCES INFORMATIQUES

Étude sur la sécurité et la confidentialité du réseau EUNET

Damien, Lucien; Maes, Bernard

Award date:
1987

Awarding institution:
Universite de Namur

Facultés Universitaires Notre-Dame de la Paix, Namur

Institut d'Informatique

Année Académique 1986 - 1987

Etude sur la sécurité et la confidentialité du réseau EUNET

Lucien DAMIEN

Bernard MAES

Promoteur : J. Ramaekers

Mémoire présenté en vue de l'obtention du grade de licencié et maître en informatique.

ABSTRACT

Cet ouvrage aborde les problèmes de la confidentialité, et de la fiabilité d'acheminement des messages sur le réseau EUNET. Ce réseau est basé sur le logiciel UUCP qui permet le transfert de fichiers d'un système à un autre. Les systèmes le constituant exploitent UNIX sous ses multiples versions. Différentes solutions sont apportées aux problèmes existants.

This work meets the problems of the privacy and the security of the EUNET network. This network is based on the UUCP package which permits file transfer between hosts. It consists of a set of systems supported by the UNIX operating system. Different solutions are considered to solve the existing problems.

Remerciements

Nous tenons ici à remercier Monsieur Jean Ramaekers, promoteur de ce mémoire, qui par ses judicieux conseils, nous a permis de le mener à bonne fin.

Nous remercions également Patrick Geurts pour les nombreux éclaircissements qu'il a eu la gentillesse de nous donner, Cécile Mabiat pour l'aide qu'elle nous a apportée tout au long de l'élaboration et de la rédaction de ce travail ainsi que les personnes qui, de près ou de loin, y ont collaboré.

Nous tenons finalement à remercier nos parents respectifs sans l'aide et le soutien desquels l'accomplissement de nos études n'aurait pas été possible.

Table des Matières

Chapitre I

Introduction générale

Si nous avions fait un sondage auprès d'informaticiens, pour obtenir la liste des systèmes d'exploitation les plus connus, UNIX y aurait sûrement figuré en bonne place. Le système d'exploitation UNIX est en effet l'un des plus répandus, et nous ne nous risquerions pas beaucoup en le qualifiant de "standard de fait".

Pour permettre l'échange d'informations entre machines utilisant ce système d'exploitation, UNIX s'est rapidement vu adjoindre un utilitaire - UUCP de son nom (Unix to Unix CoPy) - de copie de fichiers entre sites. C'était déjà un premier pas vers l'ouverture, mais cela restait loin en deçà des possibilités que laissait présager l'interconnection de machines utilisant un système d'exploitation à ce point répandu.

Les concepteurs de UNIX n'en sont donc pas restés là. L'utilitaire UUCP étant déjà à leur disposition, ils lui ont greffé les couches nécessaires pour constituer un véritable système de courrier électronique offrant la plupart des fonctionnalités qu'un utilisateur peut attendre d'un tel système.

Partant d'un logiciel point à point, soit UUCP, les concepteurs de UNIX ont donc progressivement élaboré les programmes nécessaires pour faire de machines isolées ce qui constitue actuellement un véritable réseau de machines UNIX.

Ce réseau s'est développé - et continue d'ailleurs toujours à se développer - de façon totalement anarchique et informelle, au gré des besoins. Pour contrôler quelque peu cette évolution, des organismes nationaux, et même un organisme européen ont du être créés. Le fruit de ce développement en Europe, c'est EUNET.

C'est dans le contexte de cette évolution incontrôlée du réseau, que s'inscrit notre mémoire. Tant que les machines n'étaient mises en relation que par un logiciel point à point, les problèmes de sécurité et de confidentialité pouvaient être négligés, ou réglés localement. Dans le cadre d'un réseau, non seulement les problèmes sont beaucoup plus nombreux, mais les solutions doivent être reconsidérées. C'est ce que nous nous étions assignés comme but en rédigeant cet ouvrage.

Pour trouver des réponses aux nombreux problèmes de sécurité et de confidentialité que pose l'interconnection de machines en réseau, nous avons procédé comme suit :

dans un premier temps, nous décrivons les notions du système UNIX que nous utilisons ultérieurement. C'est ce qui constitue le chapitre II.

Nous enchaînons à ce moment sur une description de la version du logiciel de communication qui tourne sur un des VAX de l'Institut d'Informatique. Cette description

constitue le chapitre III du présent ouvrage.

A ce stade nous abordons ce qui constitue l'un des deux grands axes de notre mémoire, à savoir l'étude de la sécurité d'acheminement des messages sur EUNET. Cette étude constitue le chapitre IV.

Le chapitre V est, quant à lui, consacré au second des deux grands axes de cet ouvrage, soit l'analyse des problèmes liés à la confidentialité du réseau. Nous traitons dans ce chapitre des problèmes suivants : la confidentialité du contenu des messages, la confidentialité des parties communicantes, l'analyse du trafic, la modification de messages, et la duplication de messages.

Le sixième et dernier chapitre de ce mémoire constitue une conclusion, dans laquelle nous exposons les enseignements que nous en avons tiré, ainsi que les difficultés qu'aura posé son élaboration.

Chapitre II

Présentation du système d'exploitation UNIX

1. Introduction

Le système d'exploitation UNIX tend à devenir un système de plus en plus répandu. Sa grande force vient de sa portabilité (environ 90 % du système est écrit en langage de haut niveau : le langage C).

Nous présentons brièvement dans ce chapitre les concepts de UNIX qui sont utiles à la compréhension de cet ouvrage. Le lecteur qui désirerait approfondir sa connaissance de ce système d'exploitation peut se référer à [1] et [2].

2. Le système des fichiers

Il existe trois types de fichiers : les fichiers ordinaires, les répertoires et les fichiers spéciaux. Le système des fichiers est organisé de manière hiérarchique.

Ce système de fichiers est visualisé à la figure 2.1.

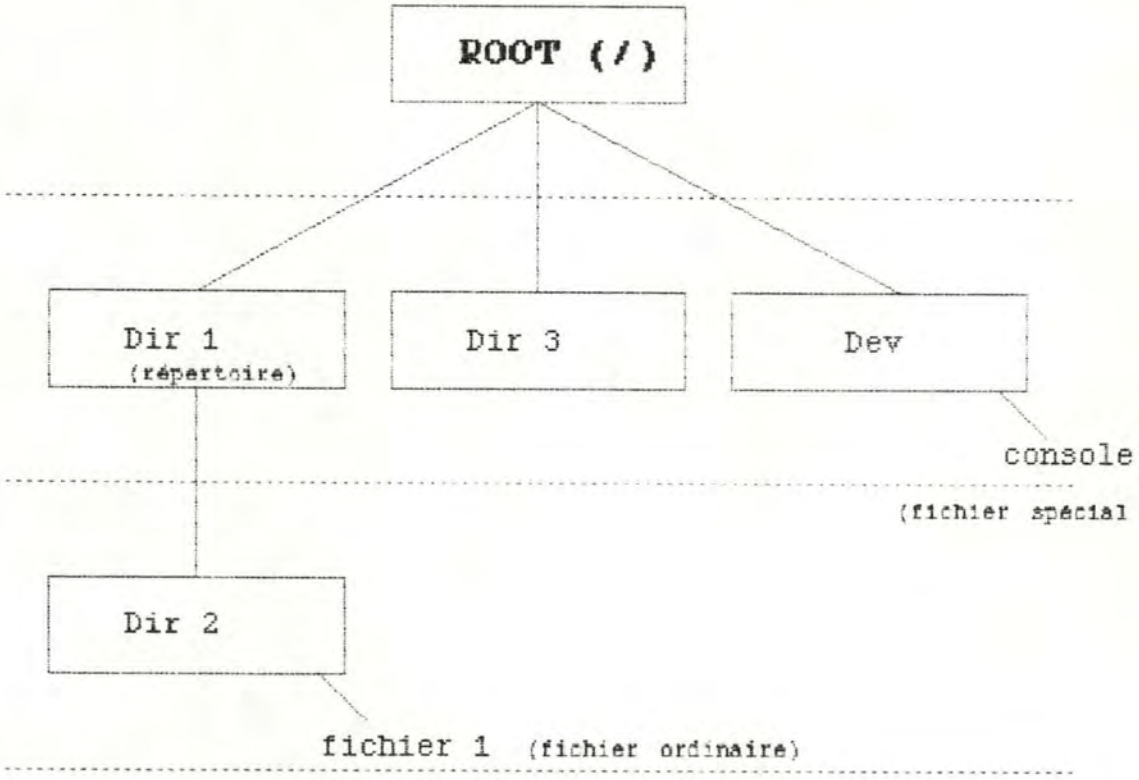

Figure 2.1 : le système de fichiers sur UNIX

La racine de la hiérarchie est un répertoire particulier qui s'appelle ROOT (noté /). Chaque répertoire contient la description soit d'un sous-répertoire, soit d'un fichier ordinaire, soit d'un fichier spécial.

Il existe deux méthodes pour pouvoir accéder à un fichier de la hiérarchie : spécifier un chemin d'accès de manière absolue en partant de la racine ROOT ou spécifier un chemin d'accès relativement à la position courante (un répertoire dans la hiérarchie).

Par exemple, le chemin d'accès absolu :

/dir1/dir2/fichier1

désigne le fichier "fichier1" se trouvant dans le sous-répertoire "dir2" du sous-répertoire "dir1", lui-même sous-répertoire du répertoire ROOT.

Si le répertoire courant est "Dir1", le chemin d'accès relatif :

dir2/fichier1

désigne le fichier "fichier1" se trouvant dans le sous-répertoire "dir2", lui-même sous-répertoire du répertoire "Dir1".

3. Le système de protection des fichiers [3]

Chaque fichier (que ce soit un fichier ordinaire, un répertoire ou encore un fichier spécial) possède trois groupes de protections : chaque groupe contient un bit pour la lecture, un bit pour l'écriture et un bit pour l'exécution. Le premier groupe est attaché au propriétaire du fichier, le deuxième groupe est attaché au groupe d'utilisateurs auquel appartient le propriétaire et le troisième groupe est attaché au reste du monde.

A chaque fichier correspond une séquence particulière de ces neuf bits, qui détermine les permissions d'accès que les utilisateurs ont sur lui. Chaque groupe de permissions est généralement exprimé en octal.

Par exemple, les protections 111101001 sont exprimées par le chiffre octal 07 pour le premier groupe, 05 pour le deuxième groupe et 01 pour le dernier groupe, c'est-à-dire une protection 0751 sur le fichier.

4. Le fichier PASSWD [4]

Le fichier "/etc/passwd" contient la description de tous les utilisateurs du système. La description des champs de ce fichier est donnée à la figure 2.2.

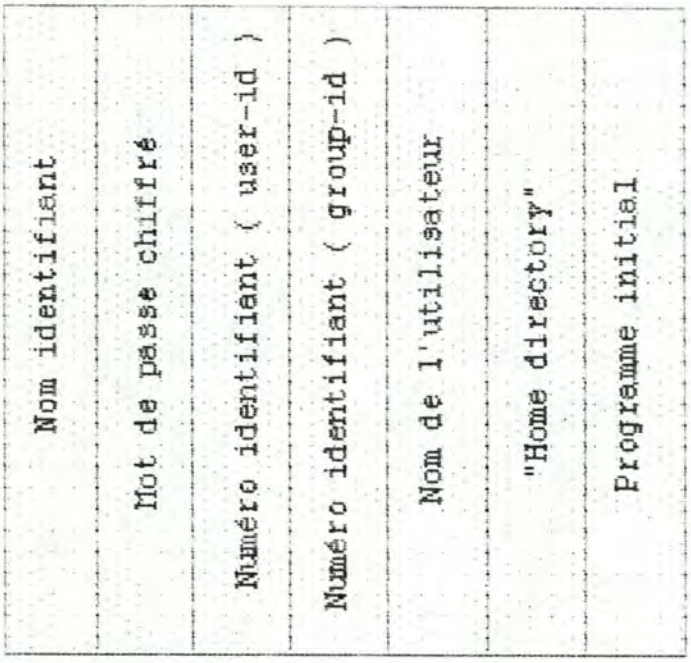

Figure 2.2 : description du fichier "/etc/passwd"

5. La phase de "login"

La phase de "login" est la phase de début de session qui correspond à la reconnaissance de l'utilisateur.

L'utilisateur introduit son nom identifiant (celui qui est repris dans la première colonne du fichier "/etc/passwd") ainsi que son mot de passe. Le mot de passe est alors chiffré et mis en correspondance avec celui repris par la deuxième colonne du fichier "/etc/passwd". S'ils sont identiques, l'utilisateur est considéré comme étant celui qu'il prétend être, et le programme désigné par la dernière colonne du fichier "/etc/passwd " est exécuté. Dans la plupart des cas, ce programme n'est autre que l'interpréteur de commandes, le programme "/bin/csh", encore appelé "shell".

6. Le mécanisme du SUID [5]

Le SUID est un mécanisme particulier par lequel un processus lancé par un utilisateur peut prendre temporairement les droits du propriétaire du programme, c'est-à-dire que le "user-id" effectif du processus devient le "user-id" du propriétaire du programme. Ce droit ne peut être donné que par le propriétaire du programme.

7. Le "super user"

Le "super user" est un utilisateur particulier dont le "user-id" est égal à zéro. Lorsque le "user-id" effectif vaut zéro, toutes les permissions sont accordées sur tous les fichiers.

Chapitre III

Description du réseau EUNET

1. Introduction

Ce chapitre présente le réseau EUNET et l'ensemble du logiciel de communication qui le supporte.

Après avoir fait la description générale du réseau, nous analysons et expliquons l'implémentation et le mode de fonctionnement de la version EUUG03. Cette version tourne actuellement sur un VAX de l'Institut d'Informatique des Facultés Universitaires Notre-Dame de la Paix à Namur.

2. Présentation générale du réseau

EUNET (European Unix NETwork) est un réseau européen de sites exploitant le système UNIX sous ses différentes versions.

La coordination est assurée par le EUUG (European Unix Users Group). En outre, chaque pays possède une association gérant les questions d'intérêt national.

2.1 Topologie

Le réseau est constitué d'un ensemble de sites dispersés sur l'ensemble du continent européen.

Il existe deux types de sites : des sites dits centraux ou "backbones" et des sites non centraux. Un site central est concerné par la question du recensement des sites et par le routage des informations au sein du réseau. Pour ce faire, chaque "backbone" possède les tables de routage pour l'entièreté du réseau. Il existe au moins un "backbone" par pays (le "backbone" belge est situé aux laboratoires de recherche de Philips à Bruxelles, et se nomme PRLB2).

Le réseau n'est pas géré de façon formelle. En effet, chaque couple de sites est entièrement libre d'ajouter une liaison UUCP. Cette décision ne nécessite que l'accord des deux sites et l'octroi de leurs droits d'accès respectifs. La seule contrainte de bon usage est d'avertir un "backbone" de l'établissement de cette nouvelle liaison afin qu'il puisse en tenir compte dans ses tables de routage. Ces modifications sont alors répercutées sur l'ensemble des "backbones".

Le réseau est schématisé à la figure 3.1.

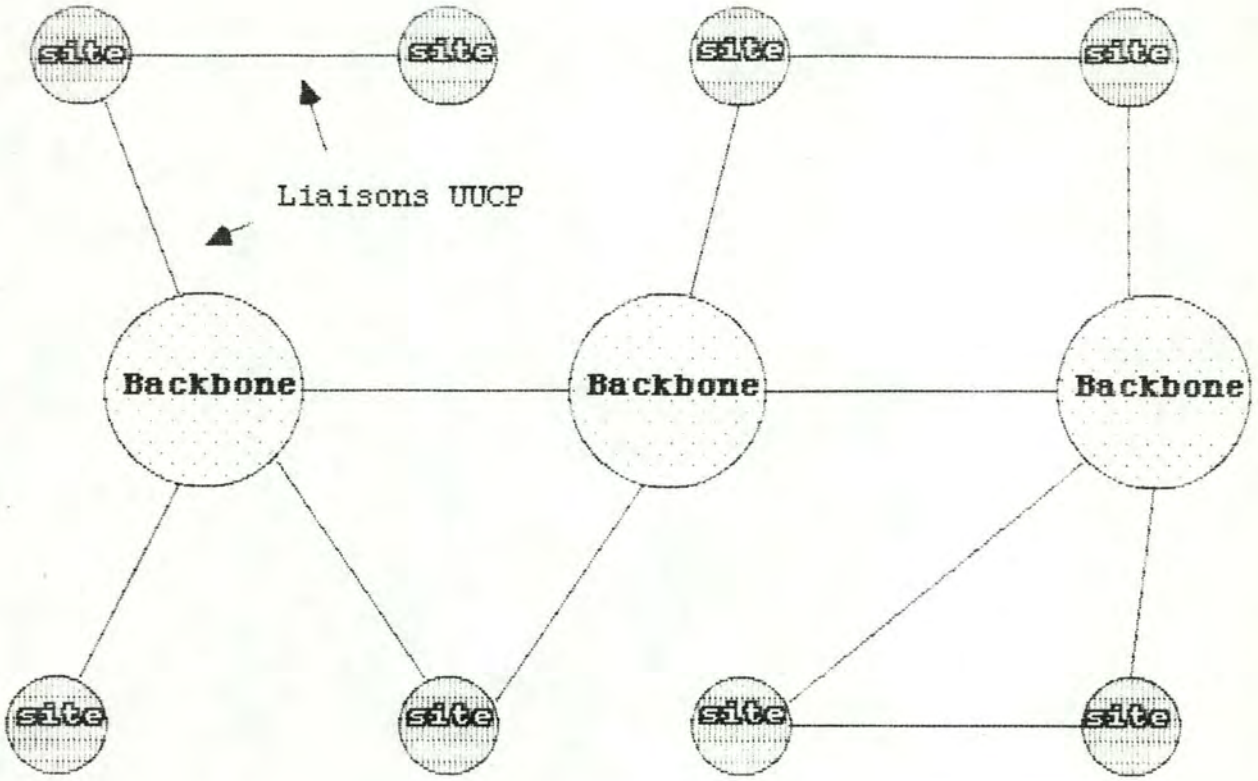

Figure 3.1 : topologie du réseau EUNET

2.2 Ouverture aux autres réseaux

EUNET est ouvert à d'autres réseaux via des passerelles. Ces réseaux sont :

- EARN (European Academic and Research NETwork), qui est un réseau constitué de machines IBM.
- ARPANET, qui est un réseau développé par le Département de la Défense des Etats-Unis.
- DECNET, CERNET, ...

De plus, le réseau EUNET peut communiquer avec le réseau USENET (Unix SErvices NETwork) qui est son homologue aux Etats-Unis.

Ces réseaux constituent un ensemble de 30 000 sites accessibles de par le monde.

2.3 Fonctionnalités

La fonction principale du réseau EUNET est de fournir un service de courrier électronique.

Une autre de ses fonctions est de fournir un service d'accès à des nouvelles. Ce service permet à un utilisateur du réseau d'obtenir des informations sur un sujet particulier.

3. Implémentation et mode de fonctionnement du réseau

3.1 Principe

Chaque site possède un ensemble de programmes qui permettent de composer des messages, de les envoyer à un destinataire situé sur le site local ou sur un autre site et de prendre connaissance de ceux qui ont été reçus. Cet ensemble peut être structuré en couches de programmes.

3.1.1 Structuration par couches [6]

Le but de chaque couche est de fournir des services aux couches de niveau supérieur, en leur épargnant des détails d'implémentation.

La couche de niveau "n" sur un site converse avec la couche "n" d'un autre site. Les règles et les conventions employées dans cette conversation sont communément appelées le protocole de couche "n".

En réalité, les données ne sont pas directement transférées de la couche "n" sur un site à la couche "n" sur un autre site (sauf pour la couche la plus basse). En fait, chaque couche transfère les données et les informations de contrôle à la couche immédiatement en dessous d'elle, tant que la couche la plus basse n'est pas atteinte. Cette couche s'occupe de la communication physique des informations.

Il existe une interface entre chaque paire de couches adjacentes. L'interface définit quels opérations et services la couche la plus basse offre à la plus haute.

3.2 Présentation par couches du logiciel de communication

Nous faisons pour commencer une description statique des couches du logiciel en commençant par la couche la plus basse. Ensuite, nous donnons la description dynamique de l'envoi et de la réception des messages sur le réseau.

3.2.1 Description statique des couches

Le logiciel de communication peut être décomposé en quatre couches : la couche physique, la couche gérant la transmission des informations sur le réseau (UUCP et UUCICO), la couche gérant la manipulation des adresses (SENDMAIL) et enfin la couche application (MAIL) (Fig 3.2).

Figure 3.2 : couches du logiciel de communication

A) Couche 1

Le moyen de communication utilisé par le réseau est quelconque. Les messages peuvent être acheminés par le réseau téléphonique commuté, une ligne louée, un réseau à commutation par paquets, ou tout autre moyen de transmission de données.

b) Le programme UUX

Comme pour le transfert de fichiers, l'exécution de commandes sur un site voisin s'effectuant en différé, il faut que la description de la requête soit enregistrée dans le "spool" d'entrée-sortie. C'est le programme UUX qui se charge de cette tâche; il crée les fichiers nécessaires dans le "spool" pour exécuter une commande sur un site voisin. Les arguments de cette commande peuvent provenir d'autres sites.

c) Le programme UUXQT

Le programme UUXQT se charge de l'exécution effective de toutes les commandes dont l'exécution a été demandée par les sites voisins. Pour que l'exécution puisse avoir lieu, il faut que tous les fichiers nécessaires soient disponibles et que ces commandes soient autorisées. La liste des commandes dont l'exécution peut être demandée à partir d'un site voisin se trouve dans le fichier XQTCMDS (voir description page A6 en annexe A).

d) Le programme UUCICO

Le programme UUCICO se charge du transfert effectif des fichiers qui ont été préparés par les programmes UUCP et UUX.

Pour ce faire, UUCICO effectue les opérations suivantes :

- parcourir le "spool" à la recherche d'un éventuel travail.
- Appeler le site voisin.
- Négocier un protocole de transmission avec le site appelé.
- Exécuter toutes les requêtes des deux sites.
- Enregistrer un certain nombre d'informations utiles.

UUCICO peut être activé selon deux modes : le mode MAITRE et le mode ESCLAVE. UUCICO lancé en mode MAITRE prend l'initiative du transfert.

Tous les fichiers que UUCICO ne peut correctement transmettre sont laissés dans le "spool".

A côté des programmes principaux, il existe un certain nombre de programmes auxiliaires.

2) Programmes auxiliaires

a) Programme de lancement de UUCICO en mode MAITRE

Le programme UUCALL permet de lancer facilement le programme UUCICO en mode MAITRE, avec en outre la possibilité de définir un niveau de "debugging" (1).

b) Programmes permettant d'obtenir des informations sur le réseau

- UUSTAT : ce programme permet d'obtenir des informations relatives aux demandes de transferts. Ces informations sont :

  - le numéro de la demande de transfert.
  - Le nom de l'utilisateur qui a formulé la demande.
  - Le nom du site voisin.
  - La date et l'heure d'émission de la demande.
  - La date et l'heure de l'obtention du statut de la demande.
  - Le statut de la demande sous forme d'un code ou sous forme de phrases explicatives.

---

(1) Le mécanisme de "debugging" permet d'obtenir des informations sur les opérations en cours. Il existe plusieurs niveaux de "debugging" auxquels sont chaque fois associés une certaine quantité d'informations.

- UUSNAP : ce programme permet d'obtenir un état courant des fichiers se trouvant dans le "spool" d'entrée-sortie. Il donne :

    - le nom des sites pour lesquels sont destinés un ou plusieurs fichiers.

    - Le nombre de fichiers de commandes, de données et de fichiers qui doivent être exécutés, relatifs à ce site.

- UUSUB : ce programme permet de définir un sous-ensemble du réseau et d'en obtenir des statistiques. Ces statistiques comprennent :

    - pour les connections :

        - le nom du site voisin.

        - Le nombre de fois que le site local a tenté d'appeler un site depuis la dernière suppression de fichiers pour ce site.

        - Le nombre de connections réussies.

        - La date et l'heure de la dernière connection réussie.

        - Le nombre de connections ayant échoué pour cause de périphérique occupé.

        - Le nombre de connections ayant échoué pour cause d'échec de "login" (voir le chapitre II, point 5, pour la description de ce mécanisme).

        - Le nombre de connections ayant échoué pour cause de non réponse (ligne occupée, site inactif, ...)

    - Pour le trafic :

        - le nombre de fichiers envoyés.

        - Le nombre de bytes envoyés durant la période qui était indiquée dans la dernière commande UUSUB dont l'option "uhr" était initialisée.

- Le nombre de fichiers reçus.

- Le nombre de bytes reçus durant la période qui était indiquée dans la dernière commande UUSUB dont l'option "uhr" était initialisée.

- UULOG : ce programme a pour fonction de faire un et un seul fichier des fichiers "LOGF.sys" (voir description page A7 en annexes), et permet aussi d'obtenir des informations sur le déroulement des communications d'un site ou d'un utilisateur donné.

c) Informations sur la configuration

- UUNAME : ce programme permet d'obtenir la liste des noms des sites voisins.

d) Gestion des formats de fichiers

- UUENCODE : ce programme a pour fonction de coder un fichier composé de caractères non ASCII en caractères ASCII. Pour ce faire, il prend trois par trois les caractères du fichier source et les code en quatre caractères ASCII.

- UUDECODE : ce programme reçoit comme entrée un fichier codé par le programme UUENCODE et le décode afin de restituer l'original.

e) Gestion du "spool"

- UUCLEAN : ce programme a pour fonction de supprimer du "spool" tout fichier plus vieux qu'un nombre donné d'heures.

C) Couche 3 [10] [11]

Cette couche a comme rôle principal la manipulation de l'adresse des messages. Elle est constituée du programme SENDMAIL et du programme RMAIL.

a) SENDMAIL

Le programme SENDMAIL a les fonctions suivantes :

- Traitement des arguments et analyse des adresses

Les adresses des destinataires sont collectées et les "aliases" (1) sont étendus. Si un destinataire est présent plusieurs fois, SENDMAIL n'en garde qu'un exemplaire.

- Collection du message

SENDMAIL collecte alors le message. Celui-ci est constitué de deux parties : l'en-tête et le contenu proprement dit. Il n'y a pas de contrainte de format sur le contenu du message excepté qu'il doit être constitué de lignes de texte (pas de données en binaire).

- Distribution du message

Pour chaque destinataire, SENDMAIL génère les requêtes UUCP et UUX nécessaires pour envoyer le message vers le site suivant du chemin d'accès.

---

(1) Noms de substitution grâce auxquels un nom peut remplacer un utilisateur ou un groupe d'utilisateurs.

b) RMAIL

Le programme RMAIL a la fonction suivante :

- il analyse les messages reçus par UUCP, rassemble les lignes "from" du message en une seule ligne de la forme "site1!site2! ... !site local" et transfère le message au programme SENDMAIL.

D) Couche 4 [12]

La couche 4 est constituée des programmes d'applications. Il s'agit du programme MAIL et du programme NEWS. Comme nous ne disposons pas de renseignements précis sur NEWS, nous ne l'aborderons pas dans le cadre de ce travail.

La fonction principale du programme MAIL est de fournir un environnement simple et agréable permettant d'envoyer et de recevoir du courrier. Ainsi, il permet de manipuler le courrier reçu (lecture, archivage) et de rédiger des messages grâce à un éditeur de texte. Il offre par ailleurs le mécanisme d'"aliasing".

3.2.2 Description dynamique des couches [9] [13]

Dans ce qui suit, nous expliquons le fonctionnement intra-couches et inter-couches lors de l'envoi et de la réception de messages sur le réseau.

L'envoi et la réception d'un message sont schématisés à la figure 3.3.

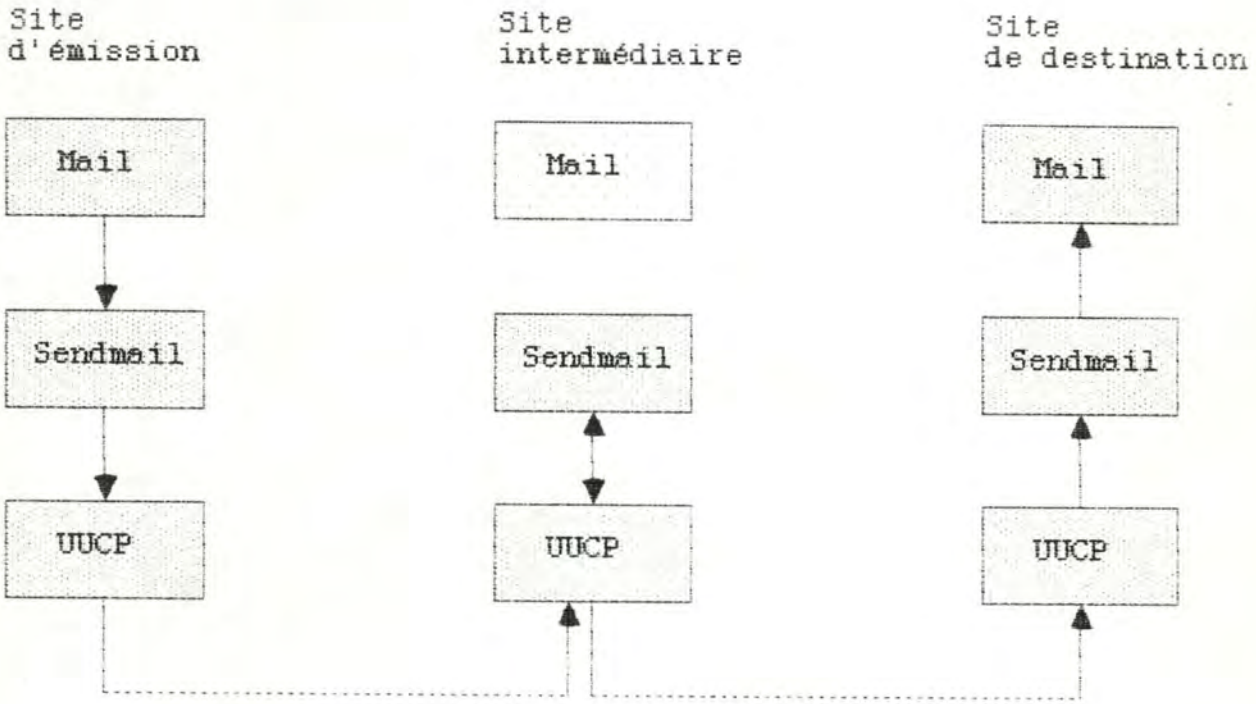

Figure 3.3 : cheminement d'un message à travers le réseau

Nous présentons maintenant le cheminement des messages en nous situant successivement sur le site de l'émetteur du message, sur un site intermédiaire quelconque et sur le site du destinataire.

A) Sur le site d'émission

Nous présentons le cheminement du message à travers les couches sur le site d'émission. Le point de départ du processus est le lancement du programme MAIL par un utilisateur.

1. Couche 4 : MAIL

Il rédige son message et désigne le ou les destinataires.

Quand il a terminé (c'est-à-dire qu'il a signifié au programme que son message peut être envoyé), MAIL invoque le programme SENDMAIL.

2. Couche 3 : SENDMAIL

SENDMAIL étend s'il y a lieu les noms de substitution ("aliases") et envoie le message à chaque destinataire. Deux cas peuvent se présenter :

- le destinataire est situé sur le site même.

- Le destinataire est situé sur un autre site.

Dans le premier cas, SENDMAIL se contente de déposer le message dans la boîte aux lettres du destinataire (c'est-à-dire dans le répertoire "/usr/spool/mail").

Dans le deuxième cas, il doit générer, pour chaque destinataire :

- une commande UUCP pour que le message soit transféré vers le premier site du chemin d'accès.

- Une commande UUX pour que soit exécuté le programme RMAIL sur ce premier site. RMAIL a comme paramètre le message à transférer.

3. Couche 2 : UUCP

Le programme UUCP prépare les fichiers nécessaires au transfert différé du message (voir description de UUCP au point 3.2.2 de ce chapitre). De façon similaire, le programme UUX prépare les fichiers pour l'exécution du programme RMAIL sur le site voisin. Ces fichiers sont placés dans le "spool" d'entrée-sortie.

A des instants déterminés, le programme UUCICO est activé. Cette activation peut être faite de plusieurs manières :

- par un démon du système d'exploitation (1).
- Par un des programmes UUCP, UUX, UUXQT ou UUCICO.
- Directement par l'utilisateur.
- Par un site voisin.

Quand il est activé dans le mode MAITRE, UUCICO inspecte le "spool" pour rechercher le travail à effectuer. Pour ce faire, il recherche les fichiers de commandes (préfixés par la lettre "C") et crée une liste de tous les sites à appeler. Lorsque l'option "sys" est spécifiée, seuls les fichiers de commandes concernant le site désigné par "sys" sont pris en considération.

Il essaie alors d'établir la communication avec un des sites de la manière suivante :

- il doit s'assurer qu'il n'y a pas déjà une communication entre les deux sites, donc qu'il n'y a pas déjà un UUCICO actif entre eux. Ceci est réalisé en testant la présence d'un fichier LCK. pour ce site (voir page A2 en annexe A). S'il n'y en a pas, UUCICO le crée et empêche ainsi toute nouvelle communication avec ce site.

- Les données nécessaires pour appeler le site sont trouvées dans les fichiers "L.sys", "L-devices" et "L-dialcodes". L'information contenue dans le fichier "L-devices" détermine le type de périphérique à utiliser (par exemple "DIR" pour ligne directe, ACU pour "Auto-call unit"). Si un de ces périphériques

---

(1) Un démon est un processus qui regarde à intervalles réguliers quels sont les programmes qui doivent être activés.

est disponible, UUCICO le réserve, sinon il se termine en signalant qu'aucun périphérique n'est accessible. Tous les fichiers sont laissés dans le "spool" en attendant la prochaine tentative de transfert.

- UUCICO atteint le site voisin grâce aux informations sur la vitesse de transmission et le numéro de téléphone dans le cas d'une "Auto-call unit". Il commence alors une séquence de "login" (voir chapitre II, point 5). Cette séquence est implémentée sous la forme "envoyer un message - attendre la réponse". S'il y a échec dans la phase de "login", UUCICO se termine.

  La syntaxe de cette séquence peut différer d'une version à l'autre.

  Il est important ici de se rappeler la structure du fichier "/etc/passwd" (voir chapitre II, point 4). Nous avons vu que l'on trouve dans ce fichier le nom du programme à exécuter juste après une phase de "login" réussie. Habituellement, c'est l'interpréteur de commandes (le "shell") qui est activé. Dans le cas d'une ligne du fichier concernant un site voisin, ce n'est pas l'interpréteur de commandes qui est activé, mais le programme UUCICO dans le mode ESCLAVE.

  Une fois la séquence de "login" achevée, le UUCICO esclave envoie un caractère de synchronisation et l'identification de son site. Le UUCICO maître envoie alors à son tour l'identification de son propre site et le niveau de "debugging" désiré s'il y a lieu.

  A ce moment, le UUCICO esclave va voir dans le fichier USERFILE si le nom de "login" du site appelant est déclaré. S'il ne l'est pas, il le signale au UUCICO maître et coupe la connection après avoir écrit un compte-rendu de la session dans le fichier "LOGF. sys".

- Il faut à présent déterminer un protocole de communication qui soit commun aux deux sites. Pour ce faire, le UUCICO esclave envoie un message "Pproto-list" (Protocole proto-list) où "proto-list" est une chaîne de caractères qui contient un caractère pour chaque protocole implémenté sur le site. Le UUCICO maître examine la liste et retient le premier caractère qui correspond à un protocole localement disponible. S'il n'en trouve pas, il envoie un message "UN" (Use No) pour le signaler, sinon il envoie un message "Ux" où "x" est le

caractère correspondant au protocole commun.

Pour être sûr que c'estt bien le même protocole, les deux UUCICO commencent à envoyer de l'information. Si la transmission réussit, il est retenu, et la transmission des données proprement dites peut commencer.

- Le comportement des deux UUCICO va être déterminé par une série de messages :

  - pour envoyer un fichier, le UUCICO maître envoie un message "Smsg" (Send msg), où "msg" détermine l'utilisateur, le répertoire cible et le mode de protection désiré (voir chapitre II, point 3). Si le UUCICO esclave refuse le fichier, il envoie un message "SN" (Send No) et le fichier est enlevé du "spool", sinon il envoie un message "SY" (Send Yes) et la transmission du fichier peut commencer.

  - Pour recevoir un fichier, de la même manière, le UUCICO maître envoie un message "Rmsg" (Receive msg) auquel, le UUCICO esclave répond par le message "TN" ou "TY" avant que la transmission ne puisse commencer.

  - Pour exécuter une commande sur le site voisin, de la même manière, le UUCICO maître envoie un message "Xmsg" (eXecute msg) auquel le UUCICO esclave répond par "XN" ou "XY" avant que la transmission du fichier contenant la commande ne puisse commencer.

  - Quand un fichier est reçu correctement et est placé dans le répertoire cible, le UUCICO du site receveur envoie un message "CY" (Copy Yes). Si le fichier ne peut être déplacé à sa destination finale, il envoie un message "CNx" (Copy No x) où "x" représente le numéro correspondant à la cause de l'échec.

  - Quand le UUCICO maître a effectué tout le travail pour le site avec lequel il est en liaison, il envoie un message "H" (Hang up). S'il y a du travail concernant le site maître dans le "spool" du site esclave, celui-ci envoie un message "HN". Le mode de chaque UUCICO s'inverse, le UUCICO maître devenant le UUCICO esclave et réciproquement. S'il n'y en a pas, le UUCICO esclave envoie un message "HY".

  - Quand le message "HY" est reçu, le UUCICO maître envoie également un message "HY" et les

échangent un message final "OO" et ferment la connection.

- Ils commencent alors une phase de terminaison :

  - le site esclave original commence à exécuter le programme UUXQT pour exécuter localement toutes les demandes qui viennent d'être reçues (fichiers préfixés pas la lettre "X" dont la partie "sys" du nom de fichier désigne le site local). Quand toutes les commandes ont été exécutées, il supprime les fichiers dans le "spool" et se termine.

  - Le site maître original continue à rechercher dans le "spool" les travaux pour d'autres sites. Quand tout est terminé, il exécute également un UUXQT, supprime les fichiers dans le "spool" et se termine.

B) Sur un site intermédiaire

Quand un site intermédiaire à reçu les fichiers correspondant à un message, le programme UUXQT exécute le programme RMAIL qui appelle lui-même le programme SENDMAIL. Nous nous retrouvons alors dans le cas du SENDMAIL sur le site d'émission.

C) Sur le site de destination

Comme dans le cas d'un site intermédiaire, SENDMAIL est appelé par RMAIL avec le fichier contenant le message comme paramètre. Il le dépose alors dans la boîte aux lettres du destinataire.

L'entièreté de ce mécanisme est schématisé par l'organigramme de la figure 3.4.

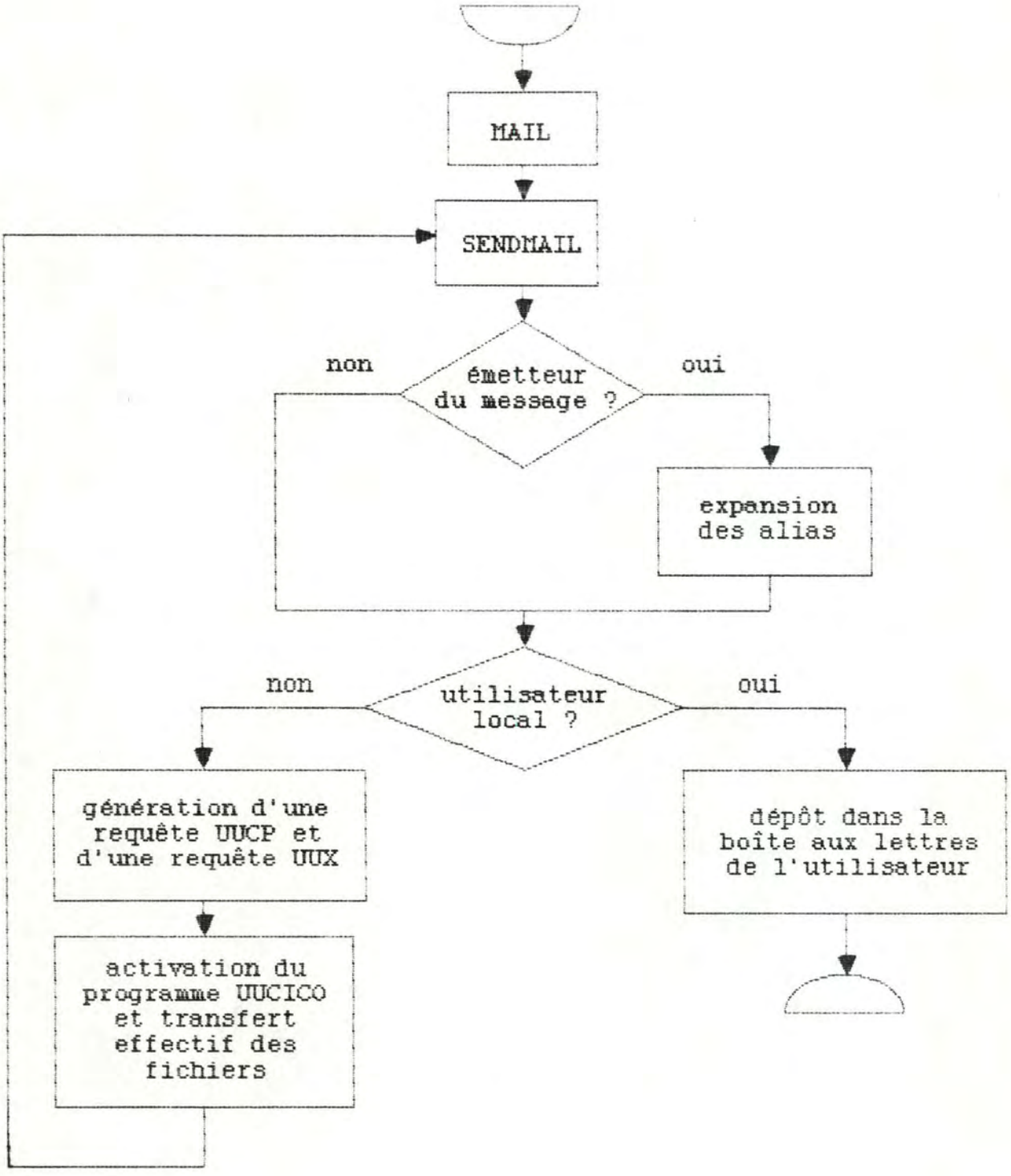

Figure 3.4 : transfert complet d'un message

Chapitre IV

Fiabilité d'acheminement sur le réseau EUNET

1. Introduction

Ce chapitre aborde la fiabilité de l'acheminement des messages sur le réseau EUNET, et plus particulièrement les réactions possibles en cas d'incident.

Pour des raisons quelconques telles que la déconnection du site où se trouve le destinataire ou de l'interruption prolongée de la liaison UUCP entre deux sites consécutifs se trouvant sur le chemin d'accès, il peut s'avérer impossible d'acheminer le message vers le site de destination.

Nous avons comme objectif de trouver des techniques qui permettent de prendre en charge les problèmes relatifs à l'acheminement des messages. Ces techniques sont basées d'une part sur la mise au courant de l'émetteur quand un état définitif du message est atteint et d'autre part sur la réaction dynamique aux incidents.

Pour bien cerner le problème, nous décrivons les mécanismes d'adressage et d'acheminement sur le réseau EUNET. Nous proposons alors des solutions aux lacunes, les modifications à apporter au système existant pour implémenter ces solutions, leurs avantages et leurs inconvénients, et leur applicabilité.

2. Situation sur EUNET

Dans un réseau de transmission de données, il faut disposer d'un mécanisme qui permette d'identifier un site auquel un message doit parvenir. Cela est effectué en lui attribuant une adresse, qui peut être soit globale à tout le réseau, soit locale à une partie de celui-ci. Habituellement, pour des raisons de facilité, l'adresse d'un site est vue comme un nom par l'utilisateur (ex : FUN-CS, qui identifie le site situé aux Facultés). Ce nom est mis en correspondance avec une adresse "réseau" de manière interne au système.

Comme il serait beaucoup trop coûteux d'avoir un maillage total de tous les sites entre eux, c'est-à-dire que chaque site aurait une liaison directe avec tous les autres sites du réseau, il faut déterminer un chemin d'accès pour faire progresser un message vers le site de destination. Ce chemin d'accès pourra être obtenu de différentes manières. Cela s'appelle le mécanisme d'adressage.

2.1 Mécanisme d'adressage

Chaque site possède un nom unique qui est connu par tout le réseau. Il existe deux possibilités pour faire parvenir un message vers le site de destination :

- déterminer explicitement un chemin d'accès, c'est-à-dire spécifier soi-même une suite de sites intermédiaires partant du site d'émission vers le site de destination. La syntaxe du chemin d'accès, pour faire parvenir un message du site A à un utilisateur situé sur le site B, pourrait être :

  site_1!site_2! ... B!utilisateur.

- Envoyer le message vers un "backbone" avec l'adresse du destinataire. Il faut spécifier explicitement le chemin d'accès jusqu'au "backbone" et celui-ci détermine un chemin d'accès jusqu'au site de destination grâce aux tables de routage qu'il possède. La syntaxe du chemin d'accès, pour faire parvenir un message du site A à un utilisateur situé sur le site B, pourrait être :

  site_1! ... !backbone!B!utilisateur

La détermination du chemin d'accès pour un message est effectuée une fois pour toutes, soit de manière explicite par l'émetteur, soit de manière automatique par le "backbone". Chaque site intermédiaire se contente de transmettre le message vers le site suivant du chemin d'accès déterminé initialement.

Il est possible depuis peu d'employer un format d'adressage conforme à la norme RFC 822 [14] déjà utilisée sur le réseau ARPANET. La syntaxe alors est la suivante :

< User-Id > @ < Domain-list >

avec - < User-Id > l'identifiant d'un utilisateur

- < Domain-List > ::= < Domain_n >. ... .< Domain_1 >

avec < Domain_* > le nom d'un site.

Par exemple, un utilisateur UT situé sur FUN-CS pourrait avoir comme adresse :

UT @ FUN-CS.UUCP

Il faut toutefois remarquer que ce format d'adresse n'est disponible qu'à partir d'un "backbone". L'adressage jusqu'au "backbone" doit toujours être dans le format standard décrit plus haut. En conséquence, la syntaxe d'un chemin d'accès pour atteindre l'utilisateur UT pourrait être :

site_1! ... !backbone!UT @ FUN-CS.UUCP

Une fois qu'un chemin d'accès a été déterminé, il faut faire parvenir le message au destinataire. Cela s'appelle l'acheminement du message.

2.2 Méthode d'acheminement [15]

L'acheminement comporte en fait deux aspects : la commutation des messages et l'adaptation des chemins d'accès.

2.2.1 Commutation

Pour chaque site intermédiaire du chemin d'accès, la commutation se limite à envoyer le message vers le site suivant déterminé précédemment, sans aucune décision de routage associée.

Le seul site qui peut prendre une décision de routage est un "backbone" dans le cas où la détermination du chemin d'accès est laissée à celui-ci.

2.2.2 Adaptation des chemins d'accès

Les tables de routage des "backbones" sont remises périodiquement à jour pour tenir compte de l'évolution du réseau. En effet, certaines liaisons UUCP peuvent être inutilisables pendant un certain laps de temps, d'autres peuvent avoir été rétablies. En outre, certains sites peuvent momentanément ou définitivement être déconnectés du réseau tandis que d'autres peuvent venir s'ajouter. L'information concernant l'état du réseau est distribuée sur tous les "backbones". Grâce à cela, ils peuvent adapter les chemins d'accès en tenant compte du dernier état connu du réseau.

2.2.3 Problèmes liés à la méthode d'acheminement

Les mises à jour des tables sont effectuées périodiquement et le mécanisme de communication fonctionne de manière différée. Il est dès lors tout à fait possible qu'entre le moment où un chemin d'accès à été déterminé et le moment où le message arrive sur un site intermédiaire quelconque de ce chemin d'accès, la configuration actuelle du réseau ait été modifiée.

Pour se fixer les idées, nous pouvons prendre comme exemple une suite de sites intermédiaires {S1, ... , Sk, Sk+1, ... , Sn} constituant un chemin d'accès particulier du site S1 vers le site Sn. Supposons que le message se trouve actuellement sur le site Sk (S1 <= Sk < Sn) et en attente

d'être transmis sur le site Sk+1. Cette situation peut être visualisée sur la figure 4.1.

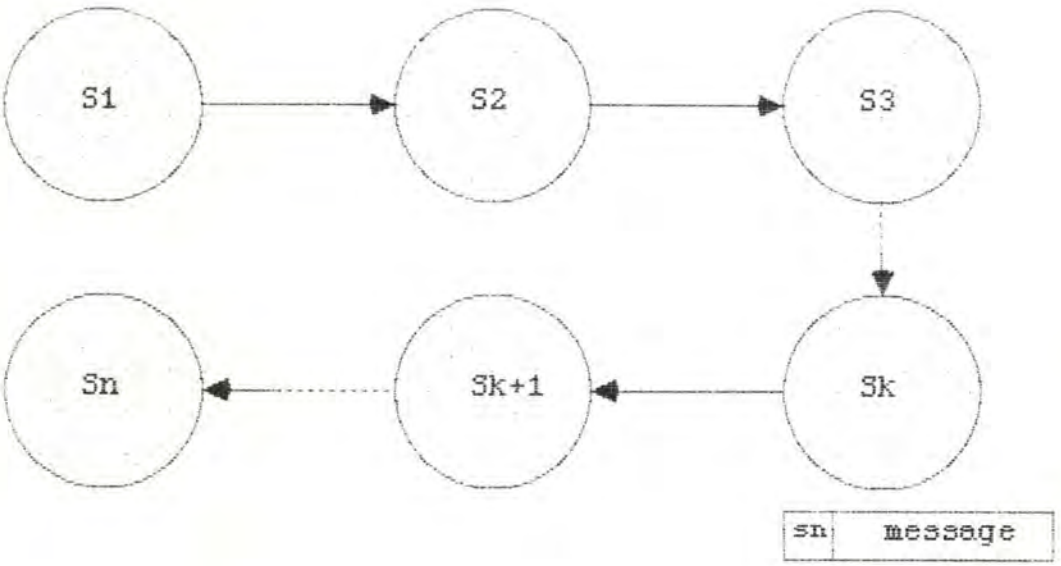

Figure 4.1 : acheminement du message

Trois scénarios peuvent alors être envisagés :

- scénario no 1 : la connection avec le site Sk+1 s'effectue normalement (la liaison UUCP entre les deux sites est opérationnelle et le site Sk+1 est actif). Le message peut donc être transmis vers le site Sk+1.

- scénario no 2 : la connection avec le site Sk+1 ne peut être établie et la cause de l'échec est momentanée (liaison UUCP momentanément en dérangement, site Sk+1 momentanément inactif). Ce cas n'est pas trop grave car dès que la cause de l'échec sera supprimée, le message pourra être correctement transmis vers le site Sk+1. La transmission sera juste quelque peu retardée.

- scénario no 3 : la connection avec le site Sk+1 ne peut être établie et la cause de l'échec est permanente (liaison UUCP définitivement coupée, site Sk+1 définitivement inactif). Ce cas est évidemment le plus grave, car à chaque tentative ultérieure de transfert du message, la même cause

d'échec sera rapportée, et celui-ci restera indéfiniment bloqué sur le site Sk.

Le scénario no 3 pose donc problème, car l'émetteur n'est pas averti que le message ne peut être transmis.

De plus, aucun accusé de réception n'est transmis lorsque le message arrive effectivement sur le site de destination. L'émetteur n'est donc jamais certain que le destinataire a bien reçu le message.

Dès lors, il faut mettre en oeuvre des techniques pour remédier à cette incertitude. Ces techniques sont analysées dans le point suivant.

3. Solutions

Certaines techniques peuvent être mises en oeuvre pour remédier à l'incertitude planant sur la réception du message par le destinataire. Nous distinguons dans ce qui suit deux types d'approches :

- la mise au courant de l'émetteur quand l'état définitif d'un message est atteint (soit le message est arrivé au site de destination, soit le message est bloqué définitivement sur un site intermédiaire du chemin d'accès).

- La réaction dynamique aux incidents qui peuvent survenir lors de l'acheminement du message (reroutage du message quand une ligne est coupée entre deux sites successifs du chemin d'accès, ...).

Nous pouvons nous baser sur le fait qu'un site intermédiaire garde un message tant que le site suivant du chemin d'accès ne l'a pas entièrement et correctement reçu (voir chapitre III). Un message ne peut donc pas disparaître entre deux sites successifs du chemin d'accès.

La première solution que nous apportons s'apparente aux deux approches et n'est donc pas classée dans un type d'approche particulier.

3.1 Etablissement d'une liaison UUCP

a) Principe

Prenons deux sites A et B faisant partie du réseau. A a un besoin d'échange avec B et ces deux sites voudraient avoir un acheminement fiable des messages échangés. Comme il n'existe pas de contrôle de bout en bout sur le réseau, ils peuvent décider d'établir une liaison UUCP entre eux. Ils bénéficient ainsi du contôle qui existe au niveau de la couche UUCP du logiciel de communication. En effet, l'émetteur situé sur le site A est certain que son message restera dans le "spool" d'entrée-sortie tant qu'il n'aura pas été entièrement et correctement reçu par le site B. Pour savoir si B a bien reçu le message, il suffit donc de vérifier que le message ne se trouve plus dans le "spool" du site A. Cette situation est schématisée à la figure 4.2.

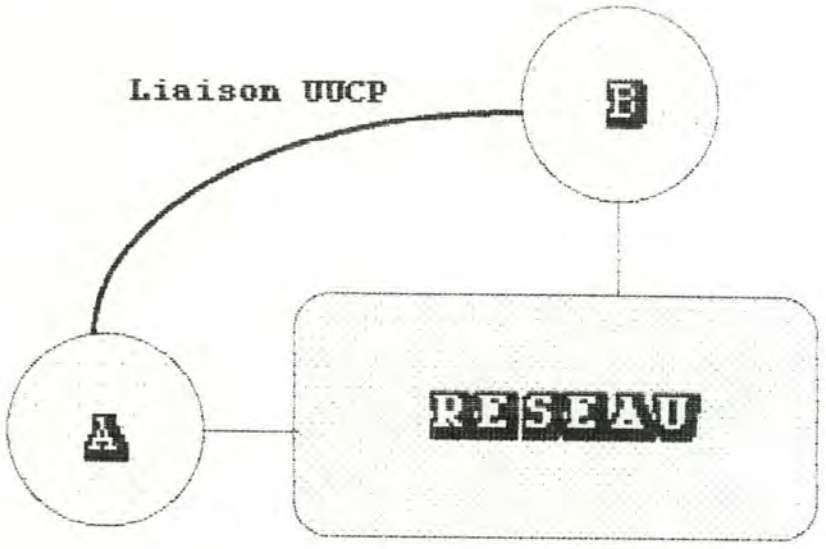

Figure 4.2 : établissement d'une liaison UUCP

b) Modifications sur le système existant

Cette liaison ne nécessite que l'adaptation de quelques fichiers du logiciel UUCP comme les fichiers "L.sys" et "USERFILE" (pour une description de ces fichiers, le lecteur peut se référer à l'annexe A), et la création d'une entrée avec le nom et le mot de passe du nouveau site voisin dans le fichier "/etc/passwd" (pour une description de ce fichier, voir le chapitre II, point 4).

En outre, chacun des deux sites impliqués doit nécessairement posséder le même moyen de communication (par exemple le réseau téléphonique).

Il existe cependant plusieurs problèmes :

1) le programme UUCLEAN (voir la description des programmes du logiciel UUCP en annexe A) met à jour régulièrement le "spool" d'entrée-sortie en supprimant les fichiers plus vieux qu'un nombre donné d'heures. S'il est activé à intervalles réguliers sur un site, il convient de positionner une option du programme qui permette d'avertir l'émetteur de la suppression de son message.

2) Dans l'implémentation à la Faculté d'Informatique, le "spool" a pour propriétaire l'"utilisateur" UUCP et le mode de protection 700, c'est-à-dire que lui seul a le droit d'aller y lire et écrire. L'émetteur ne peut donc pas vérifier lui-même la présence de son message. Une

solution serait de fournir un programme qui rechercherait dans le "spool" les messages de l'utilisateur non encore envoyés. Il aurait comme propriétaire UUCP et aurait le SUID positionné (voir le mécanisme du SUID au chapitre II, point 6) pour que le processus de l'utilisateur ait le droit d'aller lire les fichiers dans le "spool". Sa spécification pourrait être :

> "rechercher dans le "spool" les messages à envoyer pour l'utilisateur et afficher pour chacun de ceux-ci le nom du destinataire, le sujet ainsi que la date et l'heure de de sa rédaction "

Il pourrait en outre être invoqué avec différentes options :

- ssys : sélectionner uniquement les messages à destination du site "sys".
- hhour : sélectionner uniquement les messages qui sont dans le "spool" depuis plus de "hour" heures.

Ces options peuvent éventuellement être cumulatives.

c) Avantages

L'installation d'une liaison UUCP présente plusieurs avantages :

- elle peut être mise en place assez rapidement car elle ne demande qu'une adaptation de fichiers.
- La transmission est plus sûre car les messages ne doivent plus transiter par des sites intermédiaires.
- L'émetteur peut prendre immédiatement connaissance de l'état du message.
- Le délai de transmission d'un message est réduit au minimum.

d) Inconvénient

- L'inconvénient de cette solution provient de son coût. Pour bien comprendre la raison de l'augmentation du coût, il faut expliquer comment fonctionne la tarification de la transmission des fichiers au sein du réseau EUNET.

  En règle générale, chaque site ne paie que pour les fichiers qu'il envoie vers ses sites voisins. Prenons par exemple un message qui doit être envoyé vers le site C en passant par le site intermédiaire B. Le site A doit payer la transmission du fichier vers le site B tandis que B doit payer la transmission vers le site C. Cette règle s'applique également aux sites "backbone", qui supportent ainsi un coût élevé (beaucoup de fichiers transitent par eux afin d'être routé). Il existe au moins une exception à cette règle en ce qui concerne le "backbone" MCVAX, qui constitue le point de sortie du réseau vers les Etats-Unis. Le coût de transmission de tous les fichiers qui passent par ce "backbone" est refacturé aux autres "backbones".

  L'usage veut que, lorsque le nombre d'échanges entre deux sites devient important, ceux-ci établissent une liaison UUCP entre eux afin de ne pas faire supporter un accroissement de frais trop important aux autres sites intermédiaires impliqués dans l'acheminement.

  Maintenant que le principe de la tarification a été exposé, il est clair que l'augmentation du coût engendrée par l'établissement d'une liaison UUCP peut être importante, car il n'est plus réparti sur l'ensemble des sites intermédiaires. De plus, chacun des deux sites peut recevoir un surplus de messages à envoyer, car les "backbones" tiendront compte de cette nouvelle liaison dans leurs tables de routage.

  Un autre facteur à considérer est le coût d'envoi par message qui peut devenir prohibitif si les deux sites sont fort éloignés l'un de l'autre (par exemple l'Institut d'Informatique de Namur et un site situé dans le Michigan aux Etats-Unis). Cela est dû au fait qu'il faudra soit payer la location d'une ligne permanente, soit passer par un réseau existant (réseau téléphonique commuté, réseau X25, ...) et que plus la distance est grande, plus le tarif est élevé. Pour avoir des renseignements plus précis sur les tarifs en vigueur sur le réseau DCS (le réseau à commutation par paquets belge), le lecteur peut consulter l'annexe B.

e) Applicabilité

Sur base des avantages et des inconvénients que nous venons de citer, nous pouvons dire que l'établissement d'une liaison UUCP entre deux sites est envisageable lorsque :

- soit les sites ont un grand besoin de fiabilité et de rapidité dans l'acheminement des messages.

- Soit les sites ont un nombre d'échanges important dans le cas d'une ligne permanente et ne sont pas trop éloignés l'un de l'autre dans le cas d'un autre moyen de communication.

3.2 Mise au courant de l'émetteur

Nous analysons plusieurs techniques qui permettent de mettre au courant l'émetteur lorsqu'un état définitif du message est atteint :

- envoi d'un accusé de réception.

- Envoi d'un accusé de non réception.

- Tenue à jour d'un numéro de séquence concernant l'envoi et la réception des messages entre deux sites donnés.

3.2.1 Envoi d'un accusé de réception

Une première solution est d'avertir l'émetteur d'un message par un accusé de réception lorsque celui-ci a effectivement été reçu par le destinataire.

L'envoi de cet accusé de réception peut être pris en charge soit par le destinataire, soit par le système de communication du site de destination. Nous envisageons maintenant ces deux possibilités.

A) Envoi de l'accusé par l'utilisateur

a) Principe

Une solution est de laisser à l'utilisateur le soin de gérer lui-même l'envoi et la réception des accusés de réception. L'émetteur d'un message demande explicitement au destinataire de renvoyer par retour du courrier la confirmation de la réception. L'accusé de réception doit contenir le nom du destinataire, le sujet et la date d'émission du message, afin que l'émetteur puisse identifier le message qui correspond à l'accusé de réception.

b) Modifications sur le système existant

Appliqué tel quel, ce mécanisme ne demande aucune modification sur le système. Cependant, la tâche de l'utilisateur peut être facilitée par le système si celui-ci permet le positionnement d'un indicateur de rappel ou bien l'insertion d'une demande standard d'accusé de réception, à sa demande. Cette demande pourrait être exprimée par exemple en invoquant le programme MAIL avec une option particulière.

De la même manière, au niveau du destinataire, l'envoi d'un accusé de réception pourrait en partie être pris en charge par le système. Après avoir vu que l'émetteur du message attend un accusé de réception, l'utilisateur pourrait invoquer une commande qui aurait pour effet d'envoyer un accusé standard à l'émetteur. Ce mécanisme est identique à la commande REPLY (voir le manuel d'utilisation de MAIL [12].

c) Avantage

- La mise en oeuvre d'un tel mécanisme est très simple et rapide à implémenter.

d) Inconvénients

La gestion des accusés de réception est laissée sous l'entière responsabilité des utilisateurs finaux. Le système n'est pas sans failles pour plusieurs raisons :

- le destinataire peut omettre d'envoyer l'accusé de réception.
- Le problème de l'accusé de réception d'un accusé de réception se pose. En effet, le destinataire du message ne peut être sûr de la réception de son accusé de réception par l'émetteur qu'en demandant à celui-ci de lui renvoyer la confirmation ...
- Le message peut rester longtemps dans la boîte aux lettres du destinataire avant que celui-ci n'en prenne effectivement connaissance. L'émetteur pourrait supposer que le message n'a pas été reçu par le destinataire s'il tarde trop à le lire et à envoyer l'accusé de réception.

e) Applicabilité

Cette solution peut être envisagée quand la certitude de la réception par le destinataire n'est pas d'une importance capitale. Les utilisateurs peuvent alors se contenter de cette technique rudimentaire.

B) Envoi de l'accusé par le système

a) Principe

L'accusé de réception est géré de manière interne par le système de communication du site de destination.

Cette gestion peut être vue de deux côtés :

- gestion du côté de l'émetteur : l'émetteur peut vouloir ou non recevoir un accusé de réception à son message. Une méthode possible est d'appeler le programme MAIL avec une option particulière pour positionner un indicateur de demande d'envoi d'un accusé de réception.

- Gestion du côté du destinataire : si l'indicateur a été positionné, deux approches sont possibles quand le message se trouve sur le site de destination :

  - soit l'accusé est envoyé directement quand le message arrive sur le site.

  - Soit l'accusé est envoyé lorsque le destinataire prend effectivement connaissance du message.

b) Modifications sur le système existant

Plusieurs modifications sont à apporter au système existant :

- Sur le site d'émission : lorsque la demande a été exprimée par l'utilisateur, il faut positionner un indicateur. La première question à se poser est la localisation de cet indicateur dans le message. Deux solutions sont possibles, dans l'en-tête ou directement dans le message lui-même. Comme l'en-tête a un format standard, il est préférable de réserver une zone dans le message. Pour des raisons de facilité, la zone la plus adéquate se trouve au début du message. Sa longueur doit être décidée d'un commun accord entre les sites concernés. Le programme SENDMAIL ou le programme MAIL (voir chapitre III) peuvent être chargés du positionnement de l'indicateur, en l'accolant au début du message original.

- Sur le site de destination : la meilleure solution est d'envoyer l'accusé de réception dès que le message arrive sur le site. Ainsi, le temps d'attente pour envoyer l'accusé est réduit au minimum. C'est le programme SENDMAIL qui s'occupe d'analyser les messages reçus. C'est donc lui qui doit envoyer l'accusé de réception lorsque d'une part le message est destiné au site et, d'autre part, l'indicateur est positionné. Il supprime ensuite la zone réservée à l'indicateur et dépose le message dans la boîte aux lettres du destinataire. L'indicateur de l'accusé de réception ne doit pas être positionné pour éviter le problème des accusés d'accusés de réception.

c) Avantage

- La solution est relativement simple à mettre en oeuvre et l'émetteur sera certain que le site de destination enverra un accusé de réception si le message est bien arrivé.

d) Inconvénient

- Cette solution a les mêmes inconvénients que la précédente, à savoir le fait que l'accusé de réception peut également ne pas arriver à destination.

e) Applicabilité

Ce type de solution peut être envisagé, de même que la solution précédente, quand la réception par le destinataire n'est pas d'une importance capitale. Cette solution est toutefois plus fiable car l'envoi de l'accusé de réception n'est plus exposé à l'oubli du destinataire.

3.2.2 Envoi d'un accusé de non réception

a) Principe

Contrairement à l'envoi d'un accusé de réception, l'accusé de non réception est toujours pris en charge par le système d'un site intermédiaire du chemin d'accès ou par le site d'émission.

Un site envoie un accusé de non réception quand il est "sûr" que le message ne pourra jamais être transféré vers le site suivant du chemin d'accès.

Il faut évidemment définir ce qui est entendu par "sûr". A intervalles déterminés par le système du site, le programme UUCICO (voir le chapitre III) essaie de transférer les messages en attente dans le "spool" d'entrée-sortie vers le prochain site du chemin d'accès.

Deux cas peuvent alors se présenter : le message est transféré ou le message n'est pas transféré. Si le message n'est pas transféré, il reste dans le "spool" en attendant la prochaine tentative de transfert.

Si la cause de l'échec est permanente, le message ne parviendra jamais au site suivant et si la cause est temporaire, il sera transmis avec du retard.

Il faut fixer à l'avance soit un laps de temps maximum de séjour dans le "spool", soit un nombre maximum d'essais de transmission, après quoi le message est supposé ne plus jamais pouvoir être transféré vers le site suivant du chemin d'accès. Un accusé de non réception est alors envoyé vers l'émetteur et le message est supprimé du site.

Il faut se rappeler que sur certains sites, tous les fichiers plus vieux qu'un nombre donné d'heures sont supprimés du "spool" d'entrée-sortie par le programme UUCLEAN (voir la description de UUCLEAN en annexe A). Sur ces sites, il faudrait donc au moins envoyer un accusé de non réception à l'émetteur du message lorsque celui-ci est supprimé du "spool" (à ce moment, le site est sûr que le message ne parviendra jamais au destinataire).

Cette méthode pourrait également être optionelle et choisie lors de l'invocation du programme MAIL.

b) Modifications sur le système existant

La gestion de l'indicateur sur le site d'émission, qui détermine si l'envoi d'un accusé de non réception est demandé ou non, est similaire en tout point à celle vue au point précédent.

Voyons maintenant les modifications à apporter sur les sites intermédiaires et le site d'émission.

Deux cas peuvent se présenter :

- le programme UUCLEAN est régulièrement activé et l'accusé de non réception est envoyé lors de la suppression du message dans le "spool" d'entrée-sortie. Cet envoi est déjà automatiquement pris en charge par UUCLEAN si l'option "-m" a été activée (voir la description de UUCLEAN en annexe A).

- Le programme UUCLEAN n'est jamais activé. Il faut alors fixer les conditions pour qu'un message soit considéré comme impossible à transmettre. Par exemple, un programme pourrait être régulièrement activé par le processus CRON (pour avoir une description de CRON, le lecteur peut se référer à [1]) du système d'exploitation. Il aurait pour but d'envoyer un accusé de non réception à tous les émetteurs dont les messages se trouvent dans le "spool" d'entrée-sortie du site depuis plus d'un certain temps et dont l'indicateur est positionné. Une fois les accusés envoyés, il supprimerait ces fichiers du "spool" pour éviter toute confusion ultérieure.

c) Avantages

- L'émetteur d'un message est prévenu rapidement s'il est impossible d'acheminer celui-ci vers son destinataire.

- Cette méthode peut être couplée avec une des méthodes précédentes relatives à l'envoi d'un accusé de réception, et ce afin de réduire au minimum le temps d'incertitude de l'émetteur.

d) Inconvénients

- Les modifications affectent tous les sites intermédiaires entre le site d'émission et le site de destination.

- Il se peut que l'accusé de non réception ne puisse parvenir à l'émetteur du message (de la même manière que l'accusé de réception). Prenons par exemple un chemin d'accès A - B - C - D, le site A voulant envoyer un message au site D (fig. 4.3).

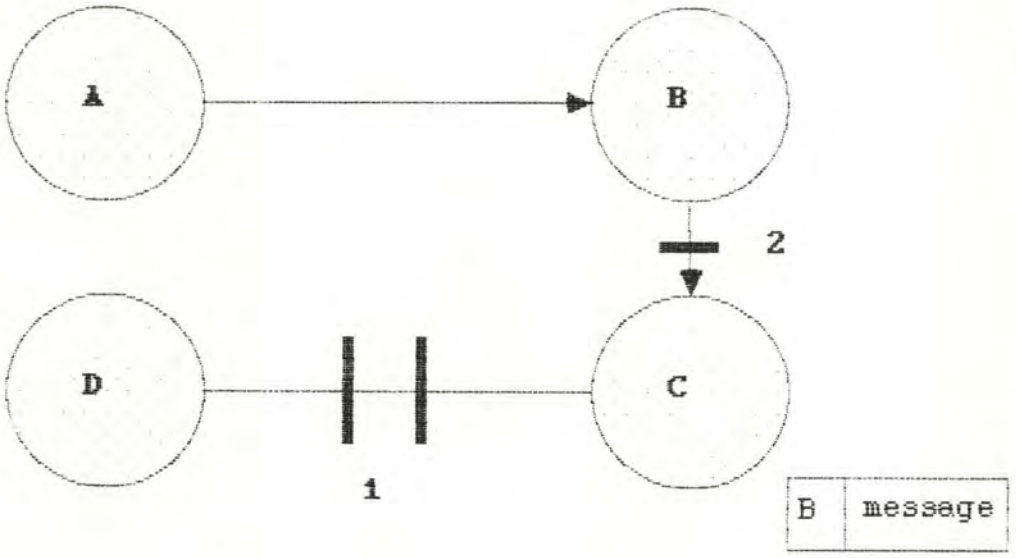

Figure 4.3 : envoi d'un accusé de non réception

Supposons que la ligne soit coupée en 1. Le site C ne peut envoyer le message vers le site D. Après un certain laps de temps, il envoie un accusé de non réception vers le site A. Si la liaison a entretemps été coupée en 2, le site A risque de ne jamais recevoir l'accusé de non réception.

e) Applicabilité

Cette méthode peut être envisagée lorsque l'émetteur doit être rapidement prévenu de l'impossibilité de délivrer son message au destinataire.

Comme le programme UUCLEAN peut être disponible sur tous les sites, ce mécanisme est celui qui est le plus rapide à mettre en oeuvre.

3.2.3 Tenue d'un numéro de séquence

a) Principe

Par paire de sites, un numéro de séquence pour l'envoi et un numéro de séquence pour la réception de messages entre ces deux sites sont tenus (ce mécanisme est déjà implémenté entre deux sites voisins par le logiciel UUCP).

Pour se fixer les idées, prenons deux sites A et B qui ont décidé de tenir un numéro de séquence (fig. 4.4).

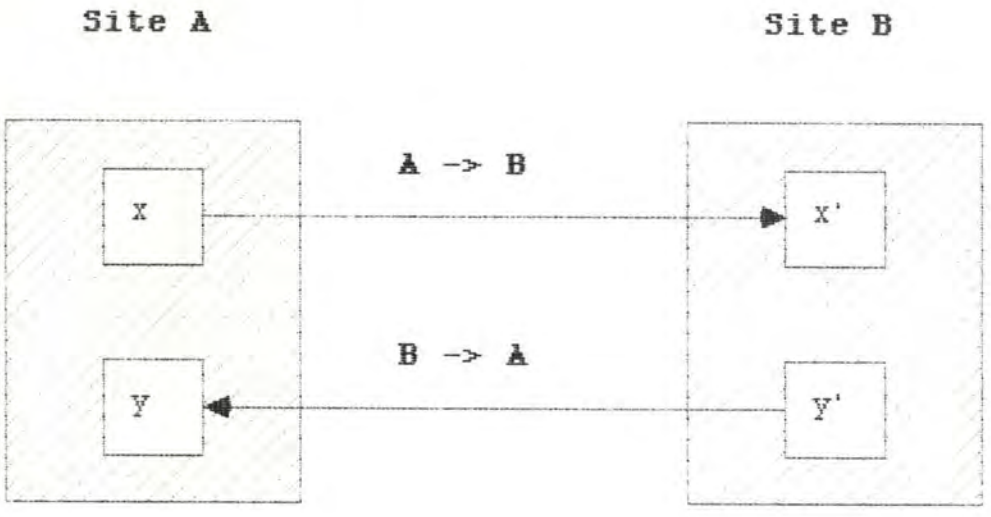

Figure 4.4 : tenue d'un numéro de séquence

Un utilisateur situé sur le site A envoie un message via le réseau à un utilisateur situé sur le site B. Le site A associe au message un numéro de séquence X et l'envoie à B. Quand B reçoit le message, il compare le numéro de séquence X du message avec le numéro X' attendu. S'ils ne sont pas égaux, il peut déterminer le ou les numéros des messages qui ne lui sont pas parvenus et en avertir le site A par retour du courrier.

Il faut alors que le site A puisse retrouver les émetteurs à partir des numéros de séquence renvoyé par B. Pour ce faire, A enregistre dans une table le numéro de séquence, l'émetteur et quelques informations utiles comme le destinataire et le sujet (pour que l'émetteur puisse identifier le message). Il indique à l'émetteur que le message n'est pas parvenu à son destinataire.

Le délai d'acheminement sur EUNET joue ici un rôle prépondérant. Comme il peut être variable, il est possible qu'un message "m1" à destination du site B envoyé en t0 arrive beaucoup plus tard qu'un message "m2" pour la même destination envoyé en t1 (fig. 4.5)

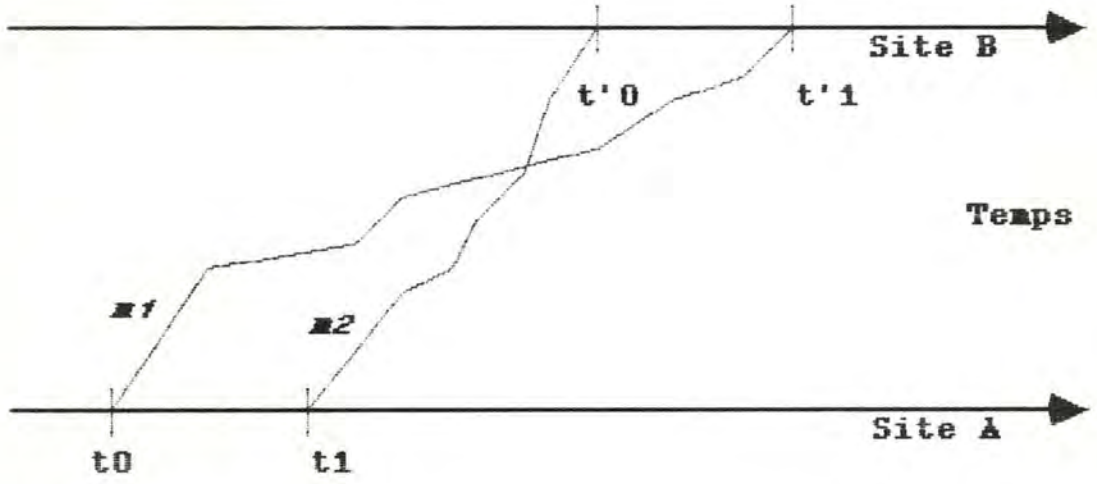

Figure 4.5 : délai variable d'acheminement

Le site B peut croire que le message "m1" s'est perdu alors que ce n'est pas le cas et l'émetteur du message "m1" sera averti à tort de sa disparition.

Pour cette raison, cette solution ne convient pas du tout pour un réseau comme EUNET.

Après avoir examiné des méthodes permettant de mettre au courant l'émetteur d'un message lorsque celui-ci a atteint un état définitif, nous examinons maintenant quelles sont les réactions dynamiques possibles en vue de faire parvenir un message malgré qu'un incident soit survenu.

3.3 Réaction dynamique aux événements

Quand le transfert d'un message est impossible entre deux sites successifs du chemin d'accès, il existe une solution alternative au fait d'avertir l'émetteur : tenter de faire parvenir le message en remédiant à l'incident. Cette méthode rend transparents pour l'utilisateur les échecs dûs au choix d'un chemin d'accès particulier, alors que le message est susceptible de suivre un autre chemin d'accès qui ne pose pas de problèmes.

3.3.1 Réexpédition du message

a) Principe

Cette technique est similaire à la technique de l'accusé de non réception vue précédemment, excepté qu'au lieu d'envoyer un message à l'utilisateur, le message est envoyé par un site intermédiaire au site d'émission qui réexpédie alors le message initial.

Pour cela, il faut soit que le site émetteur garde une copie des messages qui ont été émis, soit que le site intermédiaire renvoie le message initial avec l'avis.

Dans le cas où une copie des messages est gardée sur un site émetteur, un identifiant doit lui être associé pour pouvoir le retrouver. Cet identifiant peut être par exemple un compteur général pour le site, qui est incrémenté à chaque émission. Chaque message et son identifiant sont alors stockés par le logiciel de communication.

Le site d'émission peut recevoir un message notifiant qu'un site intermédiaire ne peut pas envoyer le message vers le site suivant du chemin d'accès. Ce message doit contenir l'identifiant du message.

Le site d'émission réexpédie alors le message vers le destinataire. Le choix du chemin d'accès pose problème. En effet, comme le message n'est pas parvenu à sa destination, il est raisonnable de supposer que le chemin d'accès, qui avait été déterminé lors de la première émission du message, n'est pas opérationnel.

Il faut donc laisser le soin à un "backbone" de déterminer un nouveau chemin d'accès, car celui-ci a dû être averti de toute modification de l'état du réseau (avec un certain délai comme nous l'avons déjà expliqué au point 2.3.2 de ce chapitre).

Pour des raisons de confidentialité, l'émetteur d'un message peut ne pas souhaiter que son message transite par certains sites. Pour cette raison, il a la possibilité de déterminer lui-même le chemin d'accès, indépendamment du "backbone". Il est donc inacceptable que, dans pareil cas, un "backbone" détermine le nouveau chemin d'accès.

Une solution est de ne pas réexpédier le message si l'émetteur du message en a exprimé le désir. Cela peut être réalisé en invoquant le programme MAIL avec une option particulière.

Cette solution n'est pas appropriée au réseau EUNET car les messages non transmis restent dans le "spool". Quand un site intermédiaire ne parvient pas à expédier le message vers le site suivant, il est tout à fait inutile de le réexpédier à partir du site d'émission. Il suffit de redéterminer un autre chemin d'accès à partir du dernier site atteint. C'est cette technique que nous allons analyser maintenant.

3.3.2 Reroutage du message

a) Principe

Lorsqu'un site intermédiaire remarque après un certain laps de temps qu'il n'a pu transmettre le message, il renvoie le message vers un "backbone" avec l'adresse du destinataire. Le "backbone" peut alors redéterminer un autre chemin d'accès. Ce mécanisme est illustré à la figure 4.6.

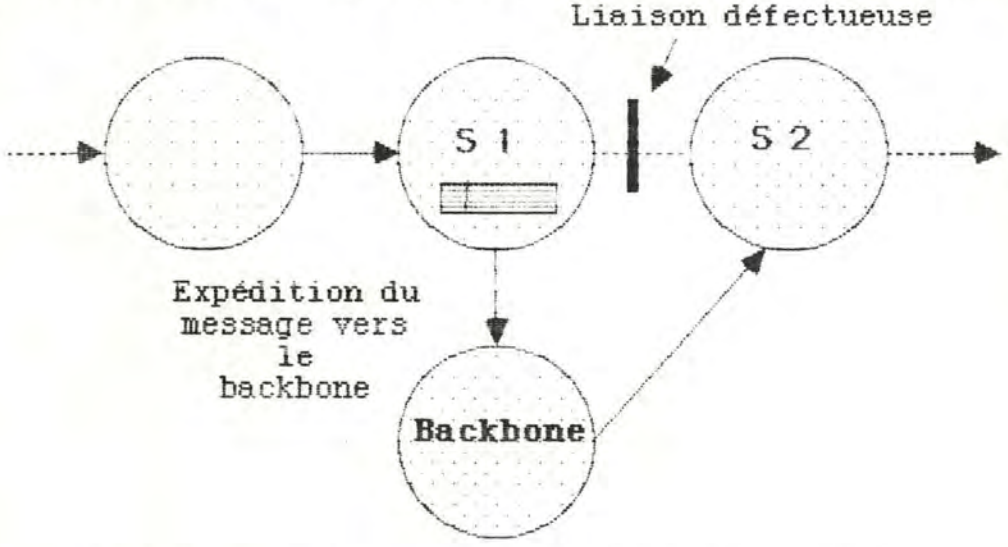

Figure 4.6 : redétermination d'un chemin d'accès

Supposons une liaison impraticable entre deux sites successifs S1 et S2 du chemin d'accès. Lorsque S1 se rend compte qu'il est impossible de transmettre le message vers le site S2, il l'expédie vers un "backbone" qui redétermine un nouveau chemin d'accès en fonction de l'indisponibilité de la liaison S1-S2.

Pour des raisons de confidentialité, il se peut que l'émetteur d'un message tienne lui-même à déterminer un chemin d'accès. Il est possible par ce moyen d'éviter que le message transite par certains sites intermédiaires qu'il juge inacceptables. Il ne faut donc pas que le système, en cas d'échec dans l'acheminement du message sur le chemin d'accès initialement déterminé, décide d'un nouveau chemin d'accès. Si l'émetteur ne veut pas de reroutage, il faut qu'il puisse le signaler au système. Cela peut être fait en invoquant le programme MAIL avec une option particulière.

b) Modifications sur le système existant

De la même manière que pour l'accusé de non réception, deux cas peuvent se présenter : le programme UUCLEAN est régulièrement activé ou il ne l'est pas. Le seul changement par rapport aux modifications introduites pour l'accusé de réception est que le site n'envoie plus un message vers l'émetteur du message mais qu'il le réexpédie vers un "backbone". Ce "backbone" peut être déterminé une fois pour toutes (celui avec lequel il existe une liaison UUCP ou à défaut le plus proche).

c) Avantages

Cette technique présente plusieurs avantages :

- l'émetteur est à peu près certain que le message arrivera à son destinataire (seul le cas d'une liaison défectueuse, qui est un point de passage obligé pour atteindre le site de destination peut rendre tout reroutage inopérant).
- La gestion est totalement transparente pour l'utilisateur.
- La détermination d'un nouveau chemin d'accès est effectuée directement, ce qui assure une perte de temps minimum.

d) Inconvénients

Cette technique présente toutefois quelques inconvénients :

- elle ne fonctionne pas lorsque la liaison défectueuse est un point de passage obligé pour acheminer le message vers le site de destination.

- Elle ne fonctionne pas non plus lorsqu'il n'est pas possible d'expédier le message vers un "backbone".

- Les modifications doivent être effectuées sur tous les sites entre le site de l'émetteur et le site du destinataire du message.

e) Applicabilité

Cette technique est appropriée dans tous les cas car elle assure qu'un message arrivera à sa destination, sauf dans les circonstances que nous venons de décrire, et de plus, dans un temps raisonnable.

Elle ne sera toutefois envisageable que sur un sous-réseau, où tous les sites effectuerons les mêmes modifications sur leur système de communication.

3.4 Conclusions

Par des méthodes relativement simples, il est possible de mettre en oeuvre des mécanismes qui accroissent la fiabilité et donc la qualité du service offert par le réseau.

Suivant le degré de fiabilité désiré, plusieurs des mécanismes vus précédemment peuvent être implémentés simultanément, pour finalement réduire au maximum les risques de disparition d'un message.

Le problème majeur vient du fait que le logiciel de communication qui supporte le réseau fonctionne de manière différée, ce qui introduit des délais d'acheminement difficilement maîtrisables.

Chapitre V

Confidentialité des messages sur le réseau EUNET

1. Introduction

Le domaine d'investigation couvert par ce chapitre est celui de la confidentialité des informations véhiculées par le réseau EUNET. Un réseau garantit la confidentialité des informations qui l'empruntent s'il est conçu pour parer à deux types d'attaques : les attaques passives et les attaques actives.

Une attaque passive est une attaque où l'"ennemi" (1) ne fait qu'"écouter" ce qui circule sur la ligne. Voydock et Kent distinguent dans [16] :

- la prise de connaissance du contenu des messages.
- La prise de connaissance de l'identité des correspondants.
- L'analyse du trafic.

Ces différentes attaques peuvent être rendues vaines si les mesures adéquates sont prises.

Quant aux attaques actives, Voydock et Kent distinguent :

- la modification de messages.
- La suppression de messages.
- Le retardement de messages.
- Le réordonnancement de messages.
- La duplication de messages.
- L'insertion de messages.
- Le blocage de la ligne.
- La tentative d'établissement d'une connection sous une fausse identité.

Les attaques actives, quel que soit le moyen utilisé, ne peuvent en fait qu'être détectées.

---

(1) personne non autorisée.

C'est une étude par attaque qui est proposée dans la suite de ce chapitre. Pour chacune des attaques que nous avons analysées, le lecteur trouvera :

- un exposé de l'attaque en elle-même. Nous situons ici l'attaque par rapport à son sujet : EUNET, et en donnons une représentation schématique.

- Une étude critique des mesures que l'on pourrait prendre dans l'état actuel des choses.

- Une présentation des techniques qui remédient aux éventuelles déficiences actuelles. Nous présentons à ce stade les méthodes que nous avons relevées dans la littérature, et opérons au terme de cette présentation un choix nuancé de l'une d'entre elles selon ses avantages et inconvénients.

- Des considérations d'implémentation de la solution qui aura été préconisée au terme de l'étape précédente.

Si la solution que nous préconisons venait effectivement à être implémentée, il nous faudrait encore considérer deux choses.

D'une part, quelle que soit la façon dont la technique choisie est implémentée, elle reste impuissante face aux "super users" des sites de l'émetteur et du destinataire de messages. En effet, ceux-ci ont accès à tous les fichiers, et peuvent demander l'obtention d'une image de toute partie de la mémoire centrale.

D'autre part, l'implémentation devrait être confiée à des personnes dignes de confiance, pour que celles-ci ne puissent inclure dans le texte des programmes, ce qui, dans la littérature, (cfr [17]) est qualifié de "cheval de Troie" (1).

---

(1) instructions insérées par le programmeur d'une application pour contourner les mesures de confidentialité assurées normalement par cette application.

2. Prise de connaissance du contenu des messages

2.1 Exposé de l'attaque

En général, l'émetteur d'un message ne désire pas que celui-ci soit lu par une personne autre que le destinataire du message ou lui-même. Or, le risque qu'un message soit lu par une tierce personne est omniprésent pour au moins deux raisons (voir figure 5.1).

Premièrement, le réseau est à commutation de messages. Tout message véhiculé par celui-ci subit par conséquent un stockage intermédiaire en chacun des noeuds du chemin d'accès. Un message peut de ce fait être lu par quiconque ayant accès aux fichiers du logiciel de communication, et notamment par le "super user" de chacun de ces noeuds.

Deuxièmement, il n'est pas exclu qu'un "ennemi" ne vienne s'interposer directement sur la ligne entre deux sites consécutifs du chemin d'accès entre l'émetteur et le destinataire.

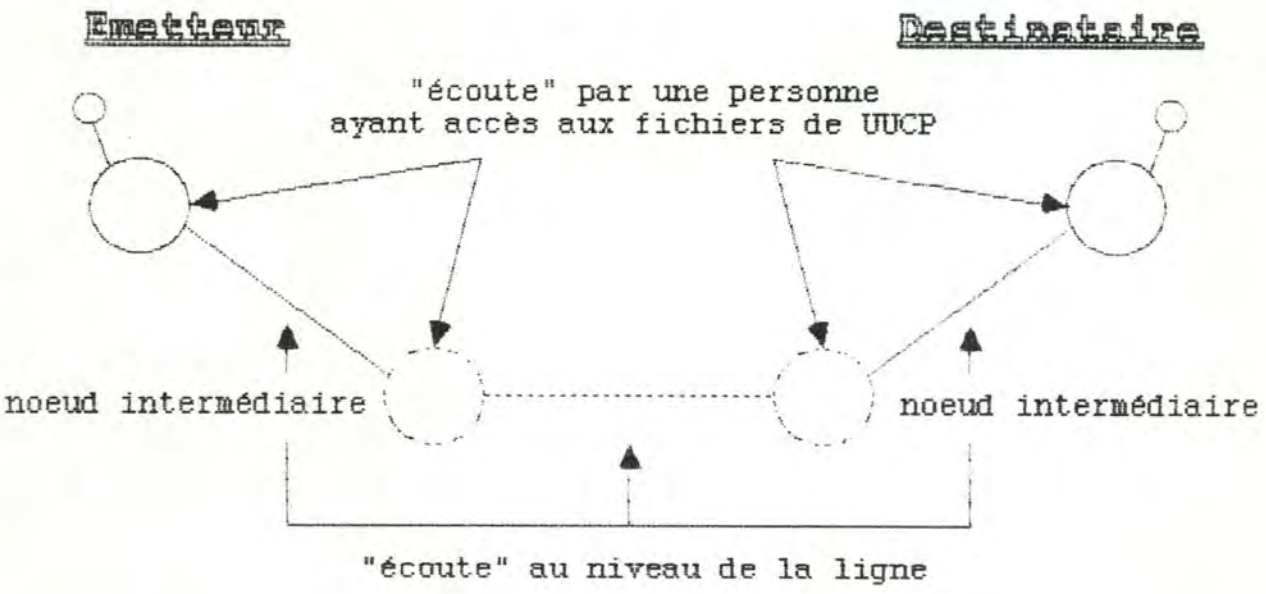

Figure 5.1 : prise de connaissance du contenu des messages

2.2 Situation actuelle

Le réseau, dans sa présente version, n'offre aucun moyen garantissant qu'un message confidentiel ne puisse être lu par un tiers.

Il est néanmoins possible dans l'état actuel des choses de diminuer les risques en remédiant au fait que les messages échangés entre le site local et un site distant puissent être lus par les personnes ayant accès aux fichiers du logiciel de communication, des noeuds intermédiaires du chemin d'accès. Il suffit pour ce faire d'établir une liaison UUCP entre le site local et le ou les sites distants avec lesquels la confidentialité des messages est souhaitée (cfr figure 5.2).

2.2.1 Etablissement d'une liaison UUCP

a) Principe

Les problèmes techniques liés à l'installation d'une liaison UUCP ont déjà été présentés dans le chapitre IV au point 3.1. Nous ne considérons donc ici que les avantages et inconvénients de cette technique qui sont directement liés au problème de la confidentialité.

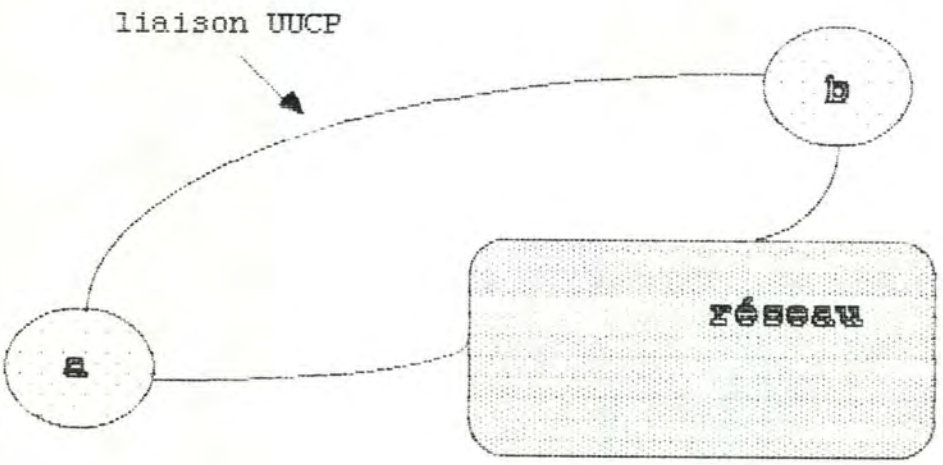

Figure 5.2 : installation d'une liaison UUCP

b) Avantage

- Le seul avantage de cette liaison directe est qu'il ne faille plus craindre la lecture des messages par les personnes ayant accès aux fichiers du logiciel de communication, des sites intermédiaires du chemin d'accès de l'émetteur au destinataire.

c) Inconvénients

- Cette façon de procéder n'empêche pas la lecture des messages par les utilisateurs ayant accès aux fichiers du logiciel de communication, des sites de l'émetteur et du destinataire du message. Le mode de protection de ces fichiers doit donc être modifié en conséquence.

- Cette technique, à elle seule, ne résout en rien le cas de l'interposition directe d'un "ennemi" sur la ligne. Elle peut cependant être appliquée en synergie avec le chiffrement au niveau de la ligne, que nous décrirons au point 2.3.1 du présent chapitre. Elle constitue alors une protection totale.

d) Applicabilité

Nous ne considérons ici que les conditions d'applicabilité de l'installation d'une liaison UUCP, lorsqu'elle est utilisée conjointement au chiffrement au niveau de la ligne.

Cette technique offrant une protection totale, est envisageable si les destinataires de messages confidentiels ne sont pas répartis sur un nombre trop élevé de sites et que ces sites ne sont pas trop éloignés.

Le réseau étant évolutif, il est cependant extrêmement rare que ces conditions soient satisfaites, c'est pourquoi nous en envisageons d'autres au point suivant.

2.3 Propositions de solutions

Une des façons de garantir l'acheminement confidentiel d'un message est de recourir au chiffrement.

2.3.1 Le chiffrement

Le chiffrement est une technique consistant à coder l'information à transmettre au moyen d'un algorithme et d'une valeur de clé, de telle façon qu'une personne n'ayant pas connaissance de la clé de déchiffrement ne peut en prendre connaissance.

Il existe deux classes d'algorithmes de chiffrement : les algorithmes à clés privées, et les algorithmes à clé publique. Pour un algorithme de chiffrement à clés privées, la confidentialité de l'information repose sur le caractère secret de ces clés (la clé de chiffrement est généralement la même que la clé de déchiffrement). Pour un algorithme à clé publique, la confidentialité de l'information ne repose que sur le secret de la clé de déchiffrement. La clé de chiffrement n'est ici un secret pour personne et peut être publiée (pour plus de détails en ce qui concerne les algorithmes de chiffrement, le lecteur peut consulter [18]).

Le chiffrement n'est certes pas la seule façon de procéder, puisqu'il est théoriquement possible de protéger physiquement la ligne, et de prendre les mesures nécessaires pour que seuls les correspondants (exception faite des "super users") aient accès aux fichiers. C'est cependant la seule qui soit d'un coût abordable.

La notion de chiffrement ayant été présentée, se pose encore le problème de sa localisation entre les émetteurs et destinataires de messages.

A) Localisation de l'opération de chiffrement

L'opération de chiffrement peut se situer directement au niveau de la ligne entre le site local et les sites voisins, ou se situer au niveau de l'émetteur et du destinataire des messages. Dans le premier cas, on parle de chiffrement au niveau de la ligne (voir figure 5.3), dans le second il s'agit du chiffrement de bout en bout (voir figure 5.4).

a) Chiffrement au niveau de la ligne

Le chiffrement au niveau de la ligne est généralement effectué dans un boîtier "hardware" contenant les clés de chiffrement-déchiffrement et une version câblée de l'algorithme de chiffrement. Il a l'avantage de rendre incompréhensible tout ce qui circule sur la ligne pour toute personne n'ayant pas connaissance de la clé de déchiffrement. Si quelqu'un "écoute" les informations circulant sur la ligne il ne peut donc ni lire le contenu des messages, ni connaître leur destinataire, ni même dire s'il y a des informations (voir [18]).

Ce mécanisme a cependant le défaut de laisser l'information à transmettre en clair en chacun des noeuds du chemin d'accès.

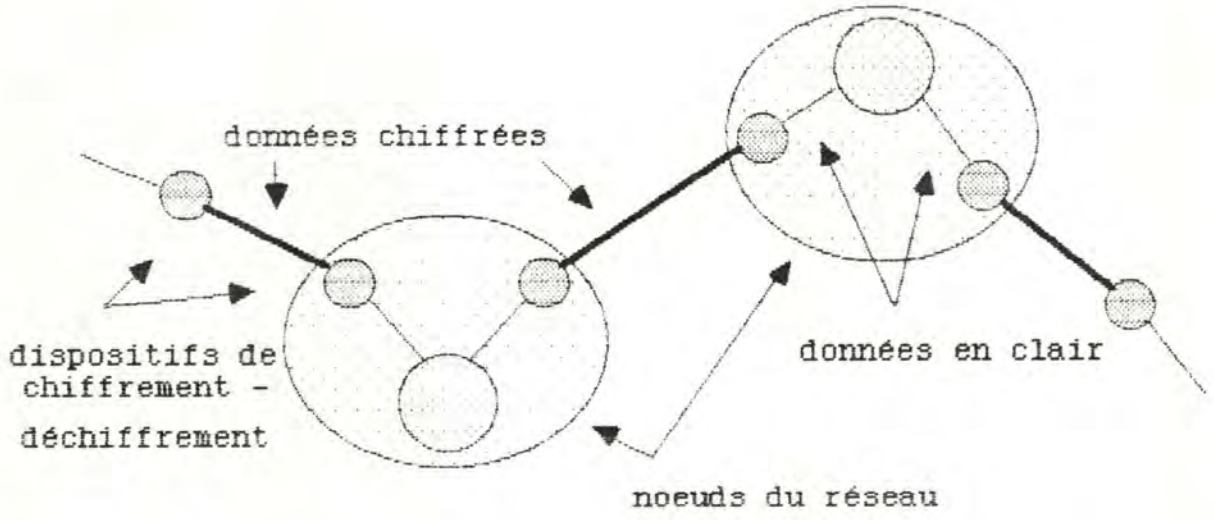

Figure 5.3 : chiffrement au niveau de la ligne

b) Chiffrement de bout en bout

Le chiffrement de bout en bout a l'avantage de ne laisser l'information en clair qu'aux deux extrémités de la ligne, c'est-à-dire avant le chiffrement et après le déchiffrement.

Il n'est cependant pas exempt de défauts, car l'utilisation de celui-ci implique que l'adresse du destinataire circule en clair sur la ligne, car elle doit être lisible par tout site du chemin d'accès. La conséquence de ceci est que n'importe qui peut prendre connaissance du destinataire du message.

Figure 5.4 : chiffrement de bout en bout

Pour que seuls l'émetteur et le destinataire de messages soient capables de les lire, il faudra donc utiliser le chiffrement de bout en bout. Le chiffrement au niveau de la ligne ne doit pas pour autant être écarté, car utilisé en synergie avec le chiffrement de bout en bout, il ne peut que contribuer à une confidentialité accrue des messages transmis.

Ayant situé l'opération de chiffrement par rapport au réseau, il nous faut encore étudier le problème de la gestion des clés.

B) Problème de la gestion des clés

Lorsque deux entités d'un réseau veulent entrer en communication de façon confidentielle, il faut qu'elles aient toutes deux connaissance d'une valeur de clé. Si l'algorithme de chiffrement est à clé privée, le destinataire du message doit connaître la clé de chiffrement de l'émetteur, tandis que si l'algorithme de chiffrement est à clé publique, l'émetteur du message doit connaître la clé de chiffrement du destinataire. La gestion de clés concerne tous les problèmes liés à la distribution de ces clés par le biais du réseau.

Ce problème prend une tournure différente selon que l'algorithme de chiffrement est à clé publique ou à clé privée. Nous l'envisagerons dans un premier temps en supposant que l'algorithme est à clé privée.

a) Gestion de la distribution de clés privées

Le problème de la distribution de clés privées est le suivant : si une valeur de clé doit transiter par le réseau, le seul moyen de la protéger d'un "ennemi" est de la chiffrer. Or pour qu'elle puisse être déchiffrée, il faut déjà qu'une valeur de clé ait été envoyée. Le problème semble donc être cyclique et n'avoir pas de solution.

Ce problème peut cependant être résolu si un petit nombre de valeurs de clés est distribué au préalable selon un canal sûr (de bouche à oreille par exemple) et si un site du réseau est désigné pour assurer le rôle de centre de distribution de clés (CDC).

La situation telle que décrite par Needham et Schroeder dans [19], est alors la suivante (cfr figure 5.5) :

Soit "n" le nombre d'entités susceptibles de communiquer dans le réseau; il faut que "n" clés privées soient distribuées pour les communications entre chacune d'entre elles et le CDC.

Supposons maintenant que l'entité "A" veuille communiquer avec l'entité "B". L'entité "A" ayant convenu d'une clé "Ka" pour ses communications avec le CDC et "B" d'une clé "Kb", ces clés n'étant connues que d'elles mêmes et du CDC.

Pour ce faire, "A" envoie au CDC une demande de connection à l'entité "B", et inclut dans son message un identificateur de message. Ce message n'est pas chiffré.

Le CDC après réception et analyse de la requête répond à l'entité "A" en lui envoyant la clé qu'elle devra utiliser pendant la connection à l'entité "B", soit "Kc", l'identificateur que "A" avait envoyé, une copie de la demande qu'avait formulée "A" et une information que "A" devra envoyer à "B" pour établir la connection et permettant à "A" de prouver son identité à "B". Ce message est chiffré en utilisant la clé de l'entité "A", soit "Ka". "A" est donc la seule entité du réseau qui puisse comprendre ce message et "A" sait que le message est authentique parce que chiffré avec la clé que seule elle et le CDC connaissent. De plus, "A" sait vérifier que le message reçu n'est pas la réplique d'un message antérieur grâce à l'identificateur qu'il contient.

Une fois que "A" a reçu ce message, elle envoie à "B" l'information qu'elle détient du CDC et qui était destinée à "B". Cette information comprend une identification de l'entité "A" ainsi que la clé "Kc". Elle est chiffrée avec la clé "Kb". A ce stade, "B" connaît la nouvelle clé, sait que c'est l'entité "A" qui veut entrer en communication, et

que le message reçu a été rédigé au CDC. Néanmoins, "B" ne peut savoir si le message qu'elle vient de recevoir est nouveau, ou si c'est une copie d'un message antérieur.

Pour s'assurer que le message qu'elle vient de recevoir n'est pas une copie d'un message antérieur, "B" envoie à "A" un identificateur chiffré en utilisant la clé de connection qu'elle vient de recevoir, soit "Kc".

"A" recevant cet identificateur effectue sur celui-ci une opération connue de "B" et renvoie le résultat à "B" en utilisant également la clé "Kc". Après réception de ce message, "B" est sûre que les messages reçus ne sont pas des répliques de messages antérieurs, et le transfert d'informations entre "A" et "B" peut commencer.

La situation telle que décrite ici est caractérisée par un CDC unique et donc une gestion de la distribution des clés centralisée. Ceci n'est pas la seule façon de résoudre le problème de la distribution des clés, il est également possible de la gérer de façon décentralisée, hiérarchisée etc... mais les principes sont les mêmes. Pour plus de détails, le lecteur peut consulter par exemple [20].

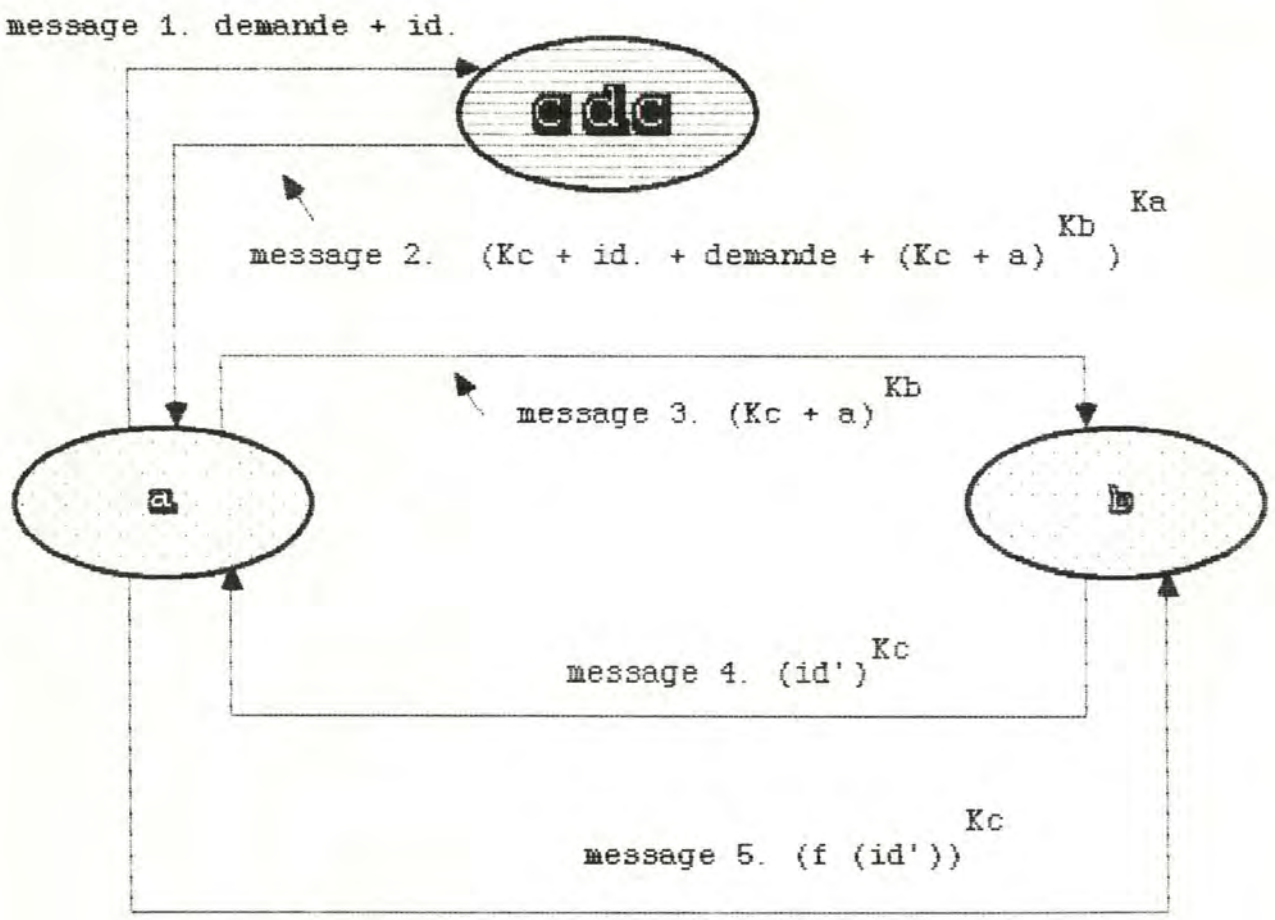

Figure 5.5 : gestion de clés privées centralisée

Ayant exposé le mécanisme de la distribution des clés quand celles-ci sont privées, montrons maintenant comment il se présente dans le cas de clés publiques.

b) Gestion de la distribution de clés publiques

Rappelons brièvement que pour un algorithme de chiffrement à clé publique, la clé privée de déchiffrement "K'" diffère de la clé publique de chiffrement "K", et ne peut être dérivée ni de la clé de chiffrement, ni d'un couple (texte en clair, texte chiffré).

Un utilisateur "A" après avoir généré une paire de clés (K,K'), peut donc publier sa clé "K". Un utilisateur "B", désireux d'envoyer un message à "A" utilisera la clé publique de "A" pour chiffrer son message, soit "K". Pour répondre à "B", "A" utilisera la clé publique de "B".

A priori, le mécanisme des clés publiques semble pouvoir faciliter l'établissement d'un canal de communication confidentiel. En effet, les clés de chiffrement étant publiques, leur transport par le biais du réseau semble pouvoir s'effectuer sans précaution aucune. Il paraît donc aisé de constituer une espèce de botin de clés publiques qu'il suffit de consulter pour obtenir la clé de chiffrement nécessaire pour converser avec une entité du réseau donnée. Il n'y a à priori aucun problème de distribution de clés et nul besoin d'une autorité centrale telle que le CDC dans le cas précédent.

Le raisonnement est cependant quelque peu hâtif. En effet, l'obtention d'un canal de communication assurant la confidentialité des messages véhiculés dépend du bon choix de la clé. Si ce choix est erroné, la protection apportée par le chiffrement à clé publique est nulle. L'exemple ci-dessous illustre cette affirmation.

Exemple :

Imaginons un site "A" désireux de communiquer avec un site "B". Pour ce faire, "A" envoie une demande de consultation du botin de clés publiques que l'on supposera situé sur le site "C". Mais au lieu que "C" reçoive la demande d'obtention de la clé publique de "B" c'est un "ennemi" "X" qui l'intercepte et - se faisant passer pour "C" - envoie à "A" une fausse clé (qui est en fait sa clé publique). Le site "A" ne peut s'en rendre compte et utilise donc la clé publique de "X" pour chiffrer le message destiné à "B". Lorsque "A" envoie son message à "B", "X" peut le déchiffrer sans problèmes, alors que c'est ce qui

devait justement être évité. Cette situation est décrite à la figure 5.6.

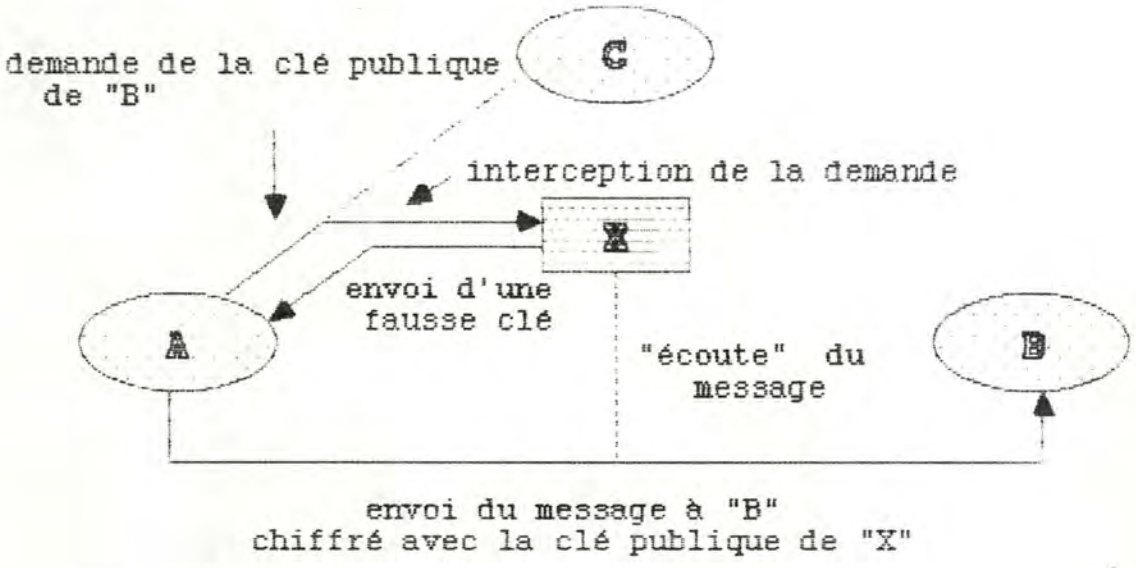

Figure 5.6 : distribution simpliste de clés publiques

Une autre raison qui contredit l'apparente simplicité de l'utilisation des clés publiques est que les clés utilisées doivent pouvoir être modifiées régulièrement pour des raisons de sécurité. La maintenance du "botin de clés publiques" pose donc le problème de l'authentification de toute demande de mise à jour. Il faut par conséquent qu'une autorité centrale assure la gestion des modifications et des consultations de ce "botin de clés publiques".

Si l'on tient compte des problèmes d'authentification soulevés ci-dessus et illustrés par l'exemple, la distribution des clés publiques peut être décrite comme suit (voir figure 5.7) :

Soit une entité "A" du réseau qui veut entrer en communication avec une entité "B". Pour ce faire il faut que "A" ait connaissance de la clé publique de "B" et que "B" ait connaissance de la clé publique de "A".

L'entité "A" commence par envoyer à une autorité centrale, que nous appellerons AC dans la suite, une demande d'obtention de la clé publique de l'entité "B", ainsi qu'une estampille. Ce message n'est pas chiffré.

L'AC répond à l'entité "A" en lui envoyant la clé publique de "B" ainsi qu'une copie de la requête et de l'estampille. Ce message est chiffré en utilisant la clé privée de l'AC. L'entité "A" peut déchiffrer ce message en

utilisant la clé publique de l'AC et est donc également sûre de la source du message qu'elle vient de recevoir. L'estampille garantit à "A" que ce qu'elle vient de recevoir n'est pas la copie d'un message antérieur et la copie de sa requête lui permet de vérifier qu'elle n'avait pas été modifiée. L'entité "A" peut à ce stade envoyer des messages à "B" car elle connaît la clé publique de "B".

Pour s'identifier au site "B" ainsi que pour éviter la répétition d'une séquence de messages antérieure, "A" envoie son nom ainsi qu'un identificateur, chiffrés en utilisant la clé publique de "B".

Lorsque "B" reçoit ce message, "B" exécute les deux premières étapes décrites précédemment pour obtenir de l'AC la clé publique de "A".

Une fois la clé publique de "A" obtenue, "B" envoie à "A" l'identificateur qu'elle vient de recevoir, ainsi qu'un autre identificateur, chiffrés tous deux en utilisant la clé publique de "A". "A" sait déchiffrer ce message et est sûre qu'elle communique vraiment avec "B" et que les messages reçus ne sont donc pas des copies de messages antérieurs.

"A" n'a plus qu'à envoyer le nouvel identificateur reçu de "B" pour que "B" soit sûre qu'elle communique vraiment avec "A", et non avec un "ennemi" lui envoyant des copies de messages antérieurs.

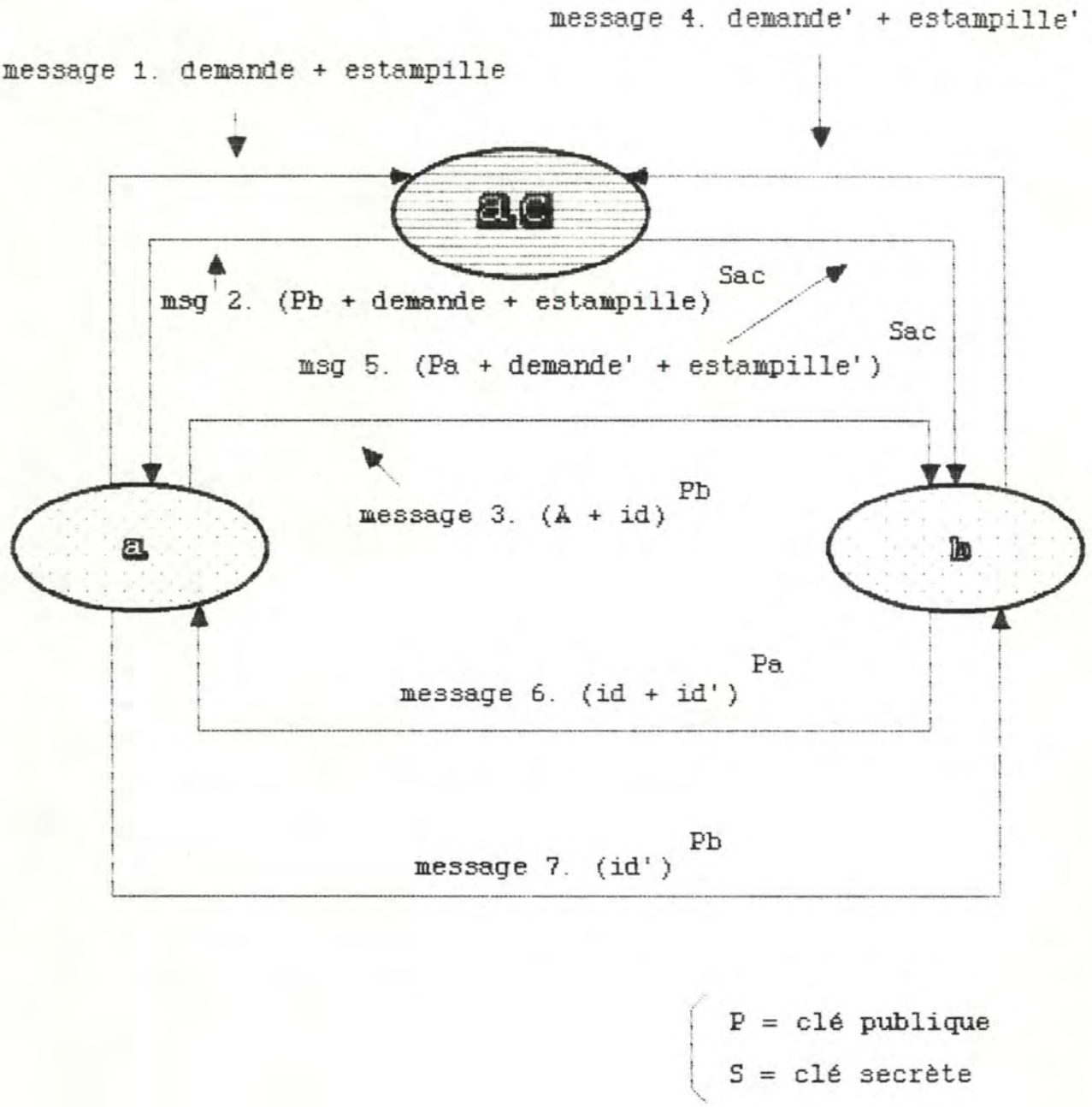

Figure 5.7 : gestion de la distribution de clés publiques

Arrivés au terme de la présentation des techniques de distribution des clés par le réseau, il nous faut voir si le réseau EUNET est à même de les assumer, dans lequel cas il nous faudra faire le choix de l'une d'entre elles. S'il ne l'est pas, c'est aux utilisateurs de celui-ci qu'incombera cette tâche.

c) Choix d'une technique de distribution de clés pour EUNET

Nous ne pouvons décider arbitrairement de confier la gestion de la distribution des clés au réseau EUNET sans considérer les avantages et inconvénients impliqués par un tel choix.

c.1) Avantage

- Le seul avantage offert aux utilisateurs s'ils confient la distribution de clés au réseau, est qu'ils n'ont nul besoin de se préoccuper des clés de chiffrement qu'ils utilisent pour leurs envois de messages. Cet avantage, bien qu'il soit unique, est cependant de taille.

c.2) Inconvénients

- Confier la distribution des clés au réseau implique que tout couple d'entités désirant communiquer de façon confidentielle doit être connecté à une tierce entité; le CDC dans le cas de clés privées ou l'AC dans le cas de clés publiques.

- Le temps de transmission des messages dont il faut assurer la confidentialité est élevé. En effet, le nombre de messages nécessaires pour obtenir les clés de chiffrement est de cinq dans le cas des clés privées et de sept dans le cas des clés publiques; et il ne faut pas perdre de vue que le réseau fonctionne en différé.

- Le nombre de modifications à apporter au réseau est très élevé. Il faut que soient désignés un(e) ou plusieurs CDC (AC), et qu'en tout noeud impliqué par cette distribution de clés soient installés les protocoles nécessaires pour la gérer.

c.3) Applicabilité

Le nombre de modifications à apporter aux programmes implémentant la version actuelle du réseau est tellement élevé qu'il n'est permis de les envisager que pour un sous-ensemble de celui-ci. Il n'est en effet pas pensable

d'imposer à tous les noeuds du réseau de modifier leurs programmes. D'autre part, le problème de la désignation des sites qui devraient jouer le rôle de CDC ou d'AC est quasi insoluble si le réseau est considéré dans sa totalité.

Quant au problème du temps de transmission des messages dont il faut assurer la confidentialité, il est relativement aisé de le pallier en se situant encore une fois dans l'optique d'un sous-ensemble du réseau. En effet, il suffit que les sites faisant partie de ce sous-ensemble s'entendent pour augmenter la fréquence de lancement du programme UUCICO (voir chapitre III point 3.2.1). Cette façon de procéder a comme conséquence de rapprocher le lancement effectif des requêtes de transferts et donc de faire du fonctionnement en "différé" du réseau un fonctionnement un peu plus "temps réel".

Sans pour autant exclure la possibilité que soient mises en oeuvre les modifications nécessaires pour faire du réseau EUNET (ou d'un sous-ensemble de celui-ci) un réseau gérant lui-même la distribution des clés, cela nous paraît cependant peu probable. Le réseau s'étant étendu de façon tout à fait informelle et anarchique, force nous est de penser que toute modification portant sur des sites autres que les sites de l'émetteur et du destinataire de messages est peu probable.

Ce raisonnement nous amène à proposer une solution au problème de la distribution des clés, où les modifications à apporter au réseau ne concernent que les sites de l'émetteur et du destinataire des messages. Les clés sont ici distribuées par les utilisateurs eux-mêmes.

c.4) Gestion de la distribution de clés par les utilisateurs

Le problème de la distribution des clés de chiffrement par les utilisateurs se trouve simplifié dans une large mesure si les clés qu'ils ont à communiquer sont publiques. En effet, si tel est le cas, ils n'ont pas à se préoccuper de leur non divulgation mais seulement de leur authenticité.

Le problème du transport authentique de la clé de chiffrement est quant à lui aisé. Le bouche à oreille, une lettre signée, ou le téléphone sont autant de moyens assurant ce transport dans les dites conditions.

D'aucuns reprocheront cependant à cette façon de procéder qu'elle implique la perte d'un des avantages du courrier électronique sur le téléphone qu'est l'asynchronisme. Mais comme les messages envoyés via un système de courrier électronique sont généralement courts (voir [20]), les valeurs des clés de chiffrement ne doivent que rarement être modifiées. Rares sont donc les

moments où les correspondants doivent êtres synchrones.

La solution que nous préconisons donc au terme de cette étude (voir figure 5.8) a le mérite de ne nécessiter qu'un minimum de modifications aux programmes existants. D'autre part, ces modifications sont localisées et ne portent que sur la couche "application" du protocole de gestion du réseau, à savoir le programme MAIL. La mise en oeuvre des modifications proposées ne devrait donc pas nécessiter un temps considérable, ce que laissent d'ailleurs présager les considérations d'implémentation du point suivant.

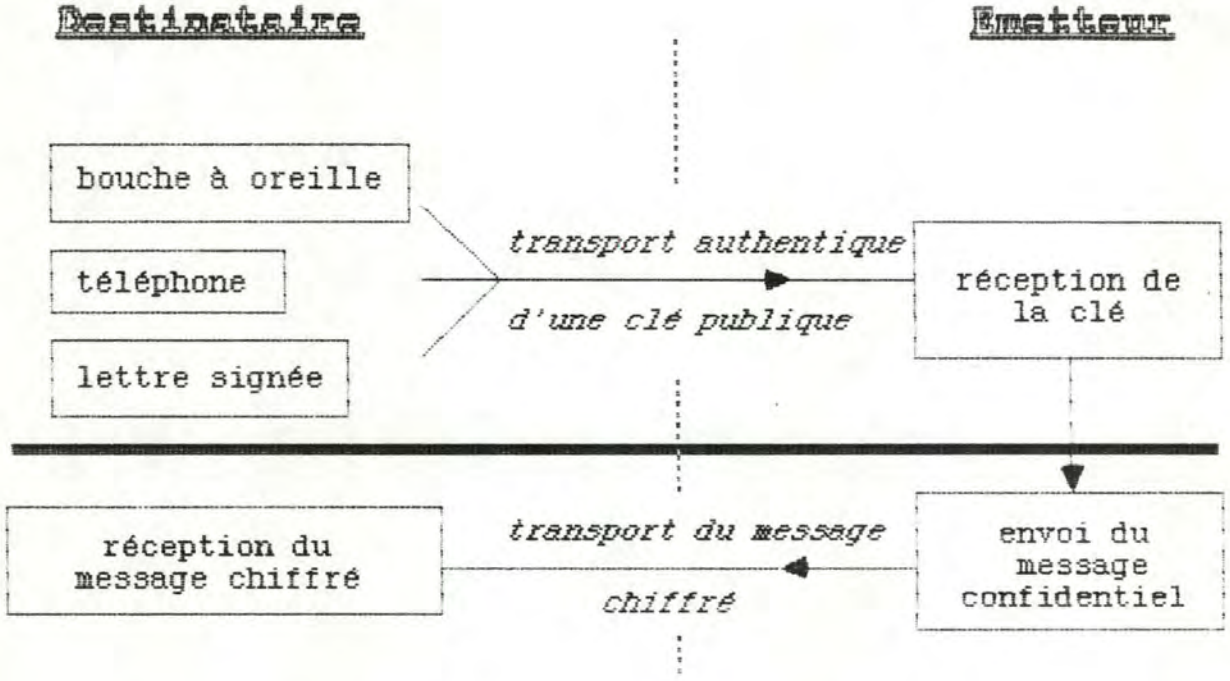

Figure 5.8 : distribution des clés par les utilisateurs

2.4 Considérations d'implémentation de la solution préconisée

Les considérations d'implémentation de la solution préconisée sont faites ici en trois points.

Dans un premier point nous étudions comment il faudrait implémenter le chiffrement. Un second point présente comment implémenter le déchiffrement. Le troisième et dernier point de notre exposé consiste quant à lui en une description des fonctions qu'il faudrait implémenter en matière de gestion de clés.

2.4.1 Considérations d'implémentation du chiffrement

Comme nous l'avons déjà souligné, les modifications que nous proposons n'affectent que la couche "application" du logiciel de communication, à savoir le programme MAIL.

Ces modifications peuvent être classées en deux catégories : les modifications de l'interface utilisateur, et les autres modifications. Voyons d'abord les modifications à apporter à l'interface utilisateur.

a) Modifications de l'interface utilisateur

Les modifications de l'interface utilisateur doivent être telles que la fonction de chiffrement soit invocable de manière analogue à toute autre commande (voir [12]). Ceci signifie, que d'une part le format de la commande d'invocation et que d'autre part l'endroit d'où elle peut être formulée soient similaires à toute autre commande. Ces considérations nous amènent à dire que la demande de chiffrement doit pouvoir être formulée selon les modes suivants :

- au niveau de l'interpréteur de commandes, en tant qu'option du programme MAIL.

  Par exemple : mail jules -c

  où l'option "-c" indique au programme MAIL que le message doit être chiffré après sa composition.

- Au niveau de l'interpréteur de commandes du programme MAIL.

Par exemple : mcipher jules

où la commande "mcipher" remplace l'habituelle commande MAIL, et signifie que l'édition du message doit être suivie du chiffrement de celui-ci.

- Au niveau de la composition même du message en tant que "tilde escape" (voir manuel d'utilisation du programme MAIL).

  Par exemple : ~k

  le caractère "c", quoique plus mnémotechnique pour évoquer le chiffrement existe déjà comme "tilde escape", c'est pourquoi nous avons opté pour un autre, en l'occurence "k".

- Au niveau des options par défaut du programme MAIL, en donnant la valeur "vrai" à l'option binaire que nous pourrions appeler "crypt".

  Par exemple : set crypt

  l'option "crypt" indiquant que tout message édité doit être suivi de son chiffrement.

Ayant passé en revue les modifications à apporter au niveau de l'interface utilisateur, décrivons maintenant les autres modifications.

b) Autres modifications

Au niveau du programme MAIL, l'effet de l'une des quatre façons d'invoquer la fonction de chiffrement est de positionner un booléen à la valeur "vrai". Ce booléen doit être testé lorsque la fin de l'édition d'un message est détectée. Le comportement du programme MAIL une fois la fin de l'édition d'un message détectée est conditionné par la valeur de ce booléen. Ce comportement est schématisé à l'aide de l'organigramme de la figure 5.9.

Nous commentons ci-dessous ce qui à la lecture de cet organigramme peut paraître confus.

- Tout message est constitué d'un en-tête et d'un corps. L'en-tête du message est composé d'un certain nombre de champs dont le lecteur peut trouver la description au point sept du manuel d'utilisation du programme MAIL. Le corps du message est quant à lui constitué par le contenu du message lui-même.

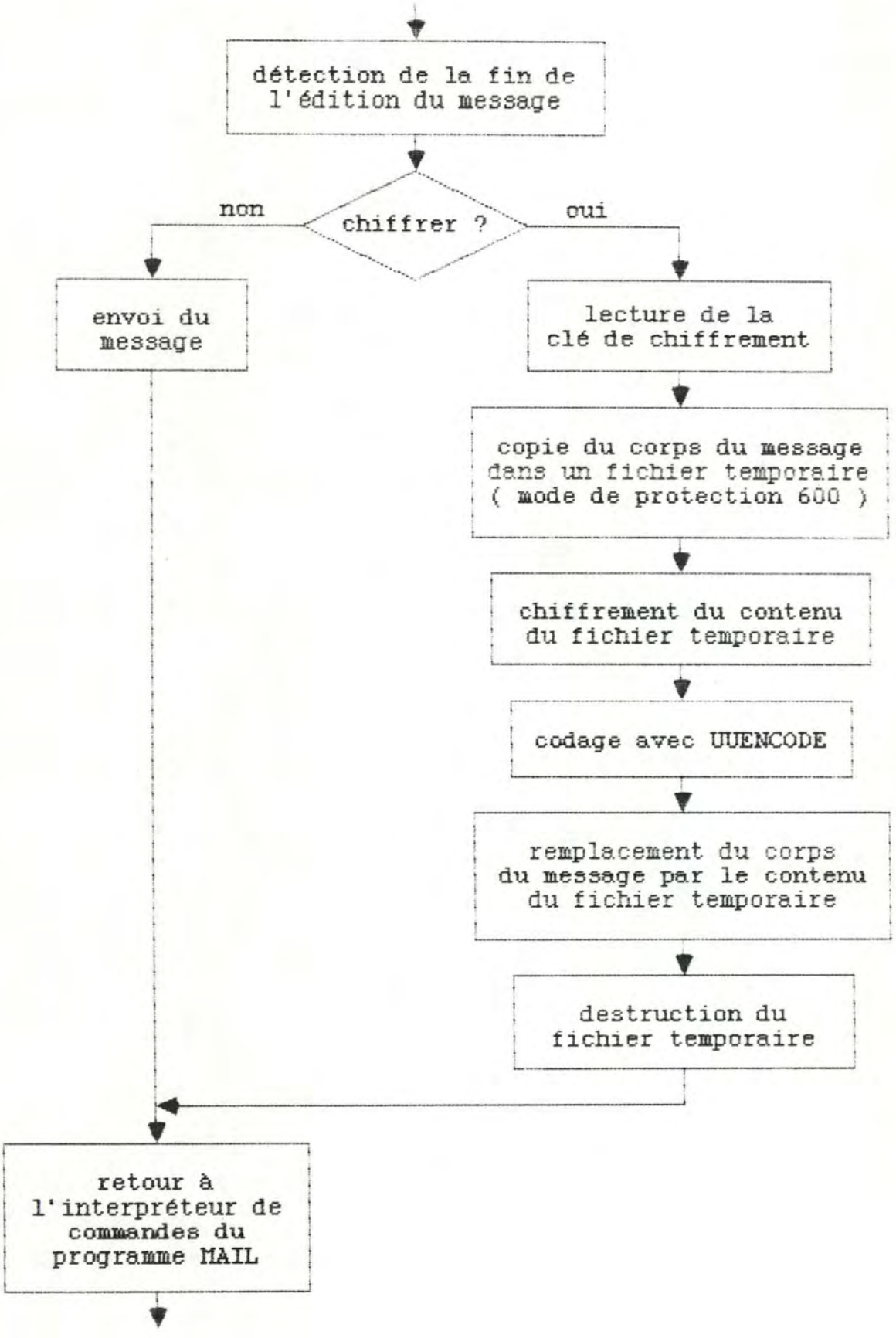

Figure 5.9 : gestion des demandes de chiffrement

L'opération de chiffrement n'affecte ici que le corps du message. Il faut en effet que l'adresse du destinataire ne soit pas chiffrée pour que le site local ainsi que les noeuds intermédiaires du chemin d'accès puissent déterminer le site auquel le message doit être envoyé. Or l'adresse du destinataire est contenue dans l'en-tête du message.

- Le mode de protection du fichier temporaire (de valeur 600) est défini de telle sorte que seul l'émetteur du message puisse y lire ou y écrire, exception faite du "super user" (voir chapitre II point 7).

- L'algorithme de chiffrement est un algorithme à clé publique, et doit être le même du côté de l'émetteur que du destinataire de messages.

- L'invocation du programme UUENCODE peut paraître surprenante à priori, mais s'explique tout simplement par le fait que le logiciel de communication ne permet que le transport de fichiers contenant des caractères ASCII. Or, suite au chiffrement, ceci risque de ne plus être le cas (voir description du programme UUENCODE au point 3.2.1 du chapitre III).

- Qu'il y ait chiffrement ou non, l'algorithme se poursuit par un retour à l'interpréteur de commandes du programme MAIL.

Ayant vu comment implémenter le chiffrement, plaçons nous maintenant dans l'optique du destinataire de messages chiffrés et considérons les modifications à apporter au programme MAIL pour qu'il incorpore le déchiffrement.

2.4.2. Considérations d'implémentation du déchiffrement

Comme pour les considérations d'implémentation du chiffrement, nous scindons la présentation des modifications ayant trait au déchiffrement en deux parties. La première est consacrée aux modifications de l'interface utilisateur, et la seconde décrit quant à elle les autres modifications.

a) Modifications de l'interface utilisateur

Les modifications à apporter à l'interface utilisateur pour qu'il incorpore les commandes nécessaires à l'invocation du déchiffrement doivent encore une fois être en accord avec ce qui existe (voir [12]). Il s'ensuit que la demande de déchiffrement doit pouvoir être formulée selon les modes suivants :

- au niveau de l'interpréteur de commandes du programme MAIL.

  Par exemple : decipher n

  où "n" est le numéro d'un message existant que l'utilisateur désire déchiffrer.

- Lors de la composition d'un autre message sous la forme d'un "tilde escape" (voir manuel d'utilisation du programme MAIL).

  Par exemple : ~u n

  où "n" est le numéro d'un message existant que l'utilisateur désire déchiffrer.

La présentation des modifications de l'interface utilisateur étant faite, passons aux autres modifications.

b) Autres modifications

Les modifications du programme MAIL ne relevant pas de l'interface utilisateur, sont schématisées au moyen de l'organigramme de la figure 5.10. Les quelques remarques ci-dessous servent à éclairer le lecteur quant à ce qui pourrait lui paraître confus ou incomplet à la lecture de cet organigramme.

- Le déchiffrement a lieu en deux temps. Dans un premier temps, le fichier temporaire est déchiffré au moyen du programme UUDECODE, et une fois cette opération effectuée, il est déchiffré selon la clé privée. La première étape du déchiffrement se justifie par le fait que le chiffrement du message s'était achevé par le codage à l'aide du programme UUENCODE. Ce programme a en fait la fonction inverse de UUENCODE (voir description du programme UUDECODE au point 3.2.1 du chapitre III).

- L'en-tête du message n'ayant pas été chiffré lors de l'envoi du message, il n'est également pas déchiffré.

- La saisie de la clé est effectuée en prenant certaines mesures de précaution. Celles-ci sont nécessitées par le fait que la clé de déchiffrement est privée. La saisie de la clé est donc effectuée sans écho sur le moniteur pour qu'un oeil indiscret ne puisse en prendre connaissance. D'autre part, comme il est préférable que cette clé ne subsiste en mémoire que le temps nécessaire au déchiffrement, sa valeur est détruite dès le déchiffrement effectué.

- La fonction de déchiffrement telle que schématisée ne détruit pas le message original après déchiffrement. Il faut en effet prévoir le cas où l'utilisateur se tromperait en introduisant sa clé de déchiffrement. Après déchiffrement, l'algorithme place le message déchiffré avec l'en-tête du message initial à la fin de la boîte aux lettres courante. Si l'utilisateur s'est trompé de clé, il peut supprimer le message qu'il vient de générer (au moyen des commandes du programme MAIL), et recommencer l'opération. L'utilisateur est donc libre de décider s'il désire garder la version chiffrée du message, la version déchiffrée, ou les deux. La correspondance entre les deux se faisant à l'aide de l'en-tête.

- En fin d'exécution de la requête, l'utilisateur se retrouve dans le mode qu'il avait quitté pour la formuler, c'est-à-dire soit le mode interpréteur de commandes du programme MAIL, soit le mode édition de messages.

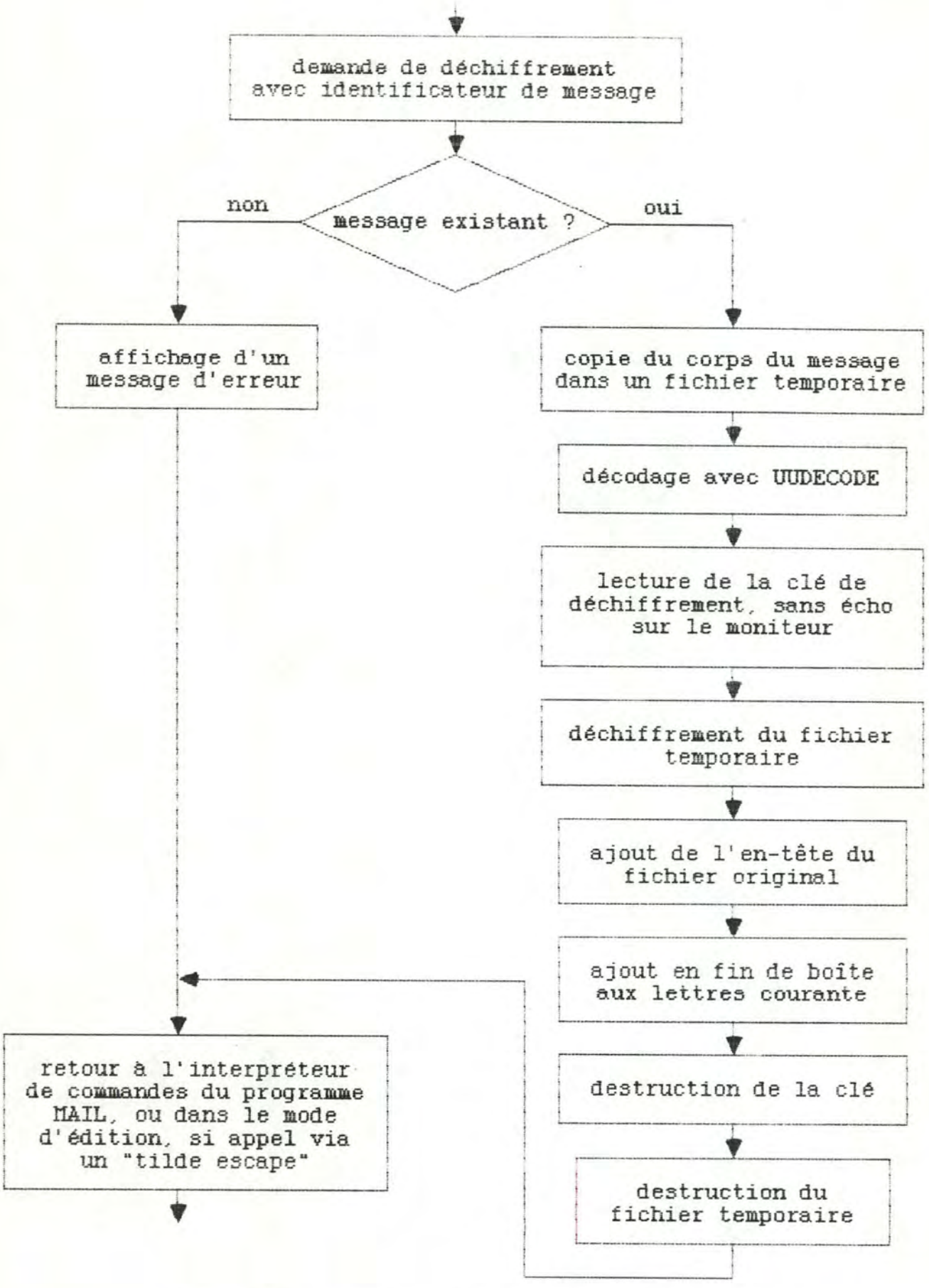

Figure 5.10 : gestion des demandes de déchiffrement

La présentation des considérations d'implémentation du chiffrement et du déchiffrement étant faite, il reste à analyser ce qu'il faudrait implémenter en matière de gestion de clés.

2.4.3 Considérations d'implémentation d'une gestion de clés

Telle que présentée ci-dessus, l'implémentation du chiffrement et du déchiffrement est déjà fonctionnelle. Leur utilisation deviendra cependant rapidement fastidieuse si l'utilisateur ne dispose pas de moyens pour gérer le stockage et la mise à jour des clés de chiffrement de ses correspondants, ainsi que le moyen de générer son couple de clés : (clé publique, clé privée).

Il faut donc que l'utilisateur puisse invoquer un gestionnaire de clés. Nous présentons ici les fonctions que celui-ci devrait offrir aux utilisateurs pour remplir correctement sa fonction. Voyons d'abord les modifications qu'il faudrait apporter à l'interface utilisateur.

a) Modifications de l'interface utilisateur

Il va de soi que l'invocation de ce gestionnaire de clés doit encore une fois pouvoir se faire comme pour toute autre commande du programme MAIL, c'est-à-dire en accord avec l'interface utilisateur existant (voir [12]). Il doit donc être appelable selon les modes suivants :

- au niveau de l'interpréteur de commandes du programme MAIL.

  Par exemple : keygest

- Lors de la composition d'un message sous la forme d'un "tilde escape" (voir manuel d'utilisation du programme MAIL).

  Par exemple : ~g

Voyons maintenant les modifications ne relevant pas de l'interface utilisateur.

b) Autres modifications

Les autres modifications réalisent l'implémentation des fonctions que devrait permettre ce gestionnaire de clés. Ces fonctions sont les suivantes :

- ajout d'un couple (destinataire, clé publique).
- Modification de la clé d'un couple (destinataire, clé publique) identifié au moyen du nom de destinataire.
- Suppression d'un couple (destinataire, clé publique) identifié au moyen du nom de destinataire.
- Génération du couple de clés de l'utilisateur, après introduction par celui-ci d'un paramètre secret servant à le générer.

En fin d'exécution du gestionnaire de clés, l'utilisateur se retrouve dans le mode qu'il avait quitté lors de l'invocation de celui-ci, c'est-à-dire soit le mode interpréteur de commandes du programme MAIL, soit le mode édition de messages.

3. Prise de connaissance de l'identité des correspondants

3.1 Exposé de l'attaque

Une personne qui parviendrait à lire l'en-tête d'un message, qui contient entre autres l'identité de l'émetteur et du destinataire, pourrait obtenir des informations utiles, pouvant porter préjudice aux correspondants. Par exemple, dans le cas d'une entreprise, une firme concurrente pourrait être fort intéressée par les échanges de cette entreprise avec l'extérieur, ou encore, dans le cas d'une université, un professeur pourrait ne pas tenir à ce qu'un autre professeur sache qu'il communique avec une certaine personne.

Il faut donc, quand cela s'avère utile, avoir à sa disposition des mécanismes qui permettent de masquer l'identité des correspondants.

L'attaque peut avoir lieu sur un site ou directement sur la ligne de transmission. Ces mécanismes doivent donc tenir compte de ces deux localisations possibles d'une attaque.

Cette attaque est illustrée à la figure 5.11

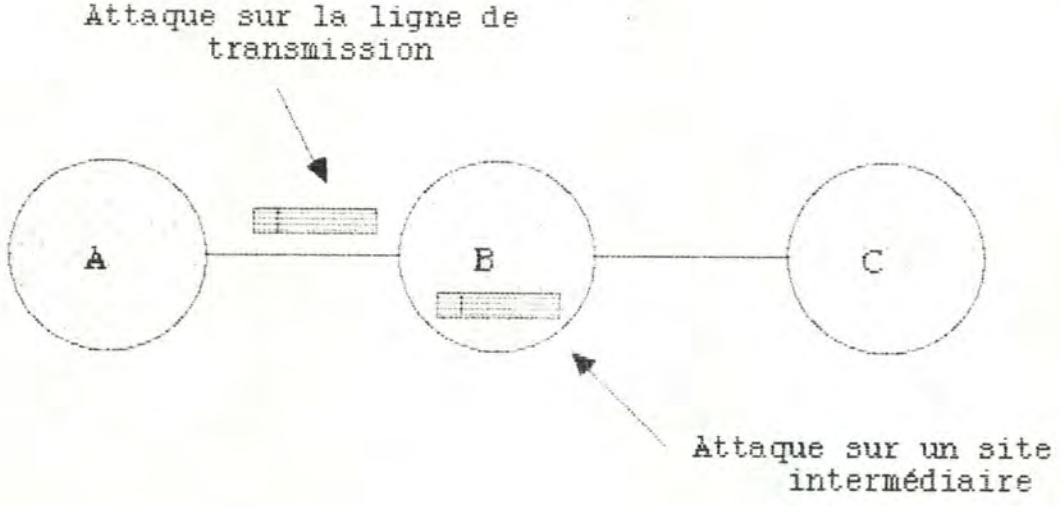

Figure 5.11 : attaque sur l'identité des correspondants

3.2 Situation actuelle

De par l'implémentation du mécanisme d'adressage sur le réseau EUNET, le chemin d'accès doit être lisible sur tous les sites intermédiaires, car ceux-ci doivent connaître le site suivant vers lequel il faut faire parvenir un message. De plus, ils connaissent le site d'où celui-ci provient.

Actuellement, le nom des correspondants se trouve en clair tout au long de l'acheminement d'un message. Toute personne ayant accès aux fichiers du logiciel de communication UUCP sur un site quelconque, et notamment le "super user", peut dès lors prendre connaissance de l'identité des correspondants de tous les messages se trouvant dans le "spool" d'entrée-sortie de ce site. Cette prise de connaissance peut également avoir lieu au niveau des lignes de transmission.

En outre, certaines informations comme l'émetteur d'un message et le chemin d'accès complet vers le destinataire sont enregistrées dans le fichier SYSLOG.sys (pour une description de ces fichiers, voir en annexe A). A l'Institut d'Informatique de Namur, ce fichier est accessible par tous les utilisateurs. Un exemple d'une ligne d'un de ces fichiers est donné à la figure 5.12.

```
Bma prlb2 (8/20-08:13-359) XQT QUE'D ( rmail cervax!ut )
```

Figure 5.12 : ligne du fichier SYSLOG.prlb2

Sur les sites où cette situation est analogue, la première chose à faire est de ne permettre l'accès à ce fichier qu'à l'administrateur du réseau ou, mieux encore, de le supprimer purement et simplement si les informations qu'il contient ne sont pas importantes dans la gestion du réseau.

Une solution optimale consisterait à garantir que personne, excepté l'émetteur et le destinataire du message, ne puisse connaître plus qu'une seule paire de sites intermédiaires (le site d'où provient le message et le site où va le message). Ainsi, personne ne pourrait induire la provenance et la destination finale du message.

Nous supposons dans ce qui suit que les différentes techniques de chiffrement proposées dans le point précédent sont acquises (chiffrement à clés privées, chiffrement à clés publiques).

Comme pour d'autres problèmes liés à la confidentialité, il est toujours possible d'établir une liaison UUCP entre deux sites pour empêcher que l'identité des correspondants ne puisse être accessible sur un ensemble de sites intermédiaires.

3.2.1 Etablissement d'une liaison UUCP

a) Principe

Le principe de l'établissement d'une liaison UUCP a déjà été présenté au point 3.1 du chapitre IV. Nous ne considérons ici que les avantages et les inconvénients de cette liaison qui sont directement liés au problème de la divulgation de l'identité des correspondants.

b) Avantage

- Le seul avantage de l'établissement d'une liaison UUCP est qu'une attaque ne peut plus avoir lieu sur un site intermédiaire.

c) Inconvénients

- La prise de connaissance de l'identité des correspondants peut encore avoir lieu sur le site d'émission et de réception.

- Cette liaison, à elle seule, ne résout pas le cas d'une écoute directe sur la ligne de transmission. Pour avoir une protection à ce niveau, il faut l'utiliser conjointement au chiffrement au niveau de la ligne.

3.3 Propositions de solutions

Nous distinguons dans ce qui suit la protection contre la divulgation de l'identité de l'émetteur et de l'identité du destinataire.

3.3.1 Identité de l'émetteur

L'identité de l'émetteur (le site et le nom de l'émetteur) d'un message peut être facilement traitée. Il suffit de chiffrer sur le site d'émission la ligne FROM de l'en-tête du message. La méthode de chiffrement doit absolument être à clé publique car le destinataire ne connaît pas, par définition, l'identité de l'émetteur. La distribution de clés publiques par le biais du réseau a déjà été discutée au point 2.3.1 de ce chapitre.

Cette méthode ne peut être appliquée à la ligne TO de l'en-tête du message car chaque site intermédiaire doit connaître le site suivant du chemin d'accès (l'identité du destinataire se trouve à la fin de cette ligne TO de l'en-tête).

Dans la plupart des cas, cela suffit à empêcher un "ennemi" d'obtenir des informations pertinentes sur l'identité des correspondants, ne connaissant pas l'identité de l'émetteur. Lorsque cette protection ne suffit pas, il faut alors mettre en oeuvre des techniques qui permettent de masquer l'identité du destinataire. Ces techniques sont analysées dans le point suivant.

3.3.2 Identité du destinataire

Plusieurs techniques peuvent être envisagées :

- chiffrement du message au niveau de la ligne.
- Chiffrement de l'identité des correspondants, de façon sélective suivant les sites intermédiaires du chemin d'accès.

A) Chiffrement au niveau de la ligne

a) Principe

Une solution est de chiffrer le message au niveau de la ligne, entre chaque paire de sites constituant le chemin d'accès vers le site de destination.

b) Avantage

- Cette solution remédie aux attaques provenant d'une écoute directe sur la ligne de transmission.

c) Inconvénients

- L'identité du destinaire est en clair sur chaque site intermédiaire, et toute personne pouvant lire les fichiers de UUCP peut prendre connaissance de l'identité du destinataire.

- Les modifications doivent être apportées à l'ensemble des sites du chemin d'accès.

d) Applicabilité

Cette façon de procéder est applicable lorsque la confidentialité de l'identité du destinataire sur les différents sites du chemin d'accès n'est pas un facteur crucial, mais que l'émetteur veut avoir une certaine protection au niveau des lignes de transmission, protection qui soit relativement simple à mettre en oeuvre.

Cette situation est typiquement le cas d'un sous-réseau où tous les sites se font mutuellement confiance.

B) Chiffrement sélectif

a) Principe

Le principe du chiffrement sélectif est de laisser le minimum d'informations sur le chemin d'accès en clair sur un site intermédiaire du chemin d'accès, c'est-à-dire le nom du site vers lequel il faut envoyer le message.

Chaque nom de site intermédiaire composant le chemin d'accès est chiffré sur le site d'émission. La clé de chiffrement pour un nom de site du chemin d'accès est la clé du site intermédiaire le précédant sur ce chemin d'accès. Cette technique est inspirée de [21].

Par exemple, supposons un chemin d'accès :

A!S1!S2!S3!S4!B!UT,

avec le site S1 ayant comme clé publique K1,
S2 ..... K2,
S3 ..... K3,
S4 ..... K4,
B ..... KB.

Cette situation peut être schématisée à la figure 5.13 :

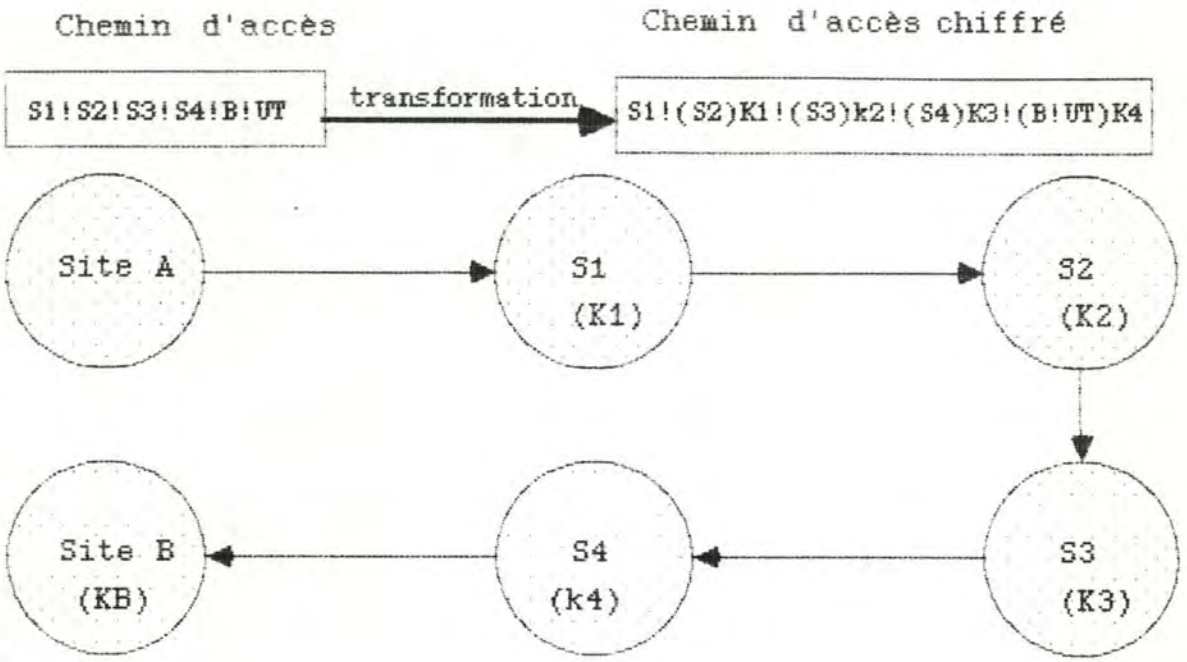

Figure 5.13 : chiffrement sélectif du chemin d'accès

La méthode de chiffrement doit être à clé publique car, par définition, un site intermédiaire ne connaît pas le site d'émission. Un système de chiffrement à clé privée ne peut donc être appliqué.

b) Avantage

- Cette technique est une solution à la fois pour le problème de l'écoute sur la ligne de transmission et pour la lecture sur un site intermédiaire.

c) Inconvénient

- L'inconvénient de ce mécanisme est de devoir chiffrer, sur le site d'émission, l'ensemble des noms de sites. Il faut donc obtenir autant de clés publiques (ce problème à déjà été discuté dans le point 2).

d) Applicabilité

Ce mécanisme convient particulièrement bien pour garantir la confidentialité de l'identité du destinataire. En effet, un site intermédiaire ne peut jamais déchiffrer que le nom du site suivant.

Malheureusement, cette méthode est inapplicable sur l'entièreté du réseau EUNET à cause de la distribution des clés publiques. Par contre, il est tout à fait possible de l'envisager sur un sous-réseau constitué de quelques sites, comme nous l'avons dit dans le point 2.3.1 de ce chapitre. Dans ce cas, la distribution des clés publiques par le biais du réseau peut être envisagée.

4. Analyse du trafic

4.1 Exposé de l'attaque

Une troisième attaque passive possible est l'analyse du volume des messages envoyés par le réseau de communication.

Le type d'informations obtenues par cette attaque peut avoir une valeur considérable. Prenons comme exemple les analyses des transmissions durant la seconde guerre mondiale, où une augmentation du volume des transmissions signifiait un regain d'activité de l'ennemi dans le secteur et pouvait constituer un renseignement très précieux.

Un utilisateur peut vouloir éviter que quelqu'un remarque que les échanges de messages avec certains utilisateurs varient. Il faut donc qu'il ait un mécanisme à sa disposition pour pallier ce problème.

L'analyse du trafic est en fait constituée en majeure partie par l'analyse du nombre de messages qui sont échangés au sein du réseau, mais aussi l'analyse de la longueur de ces messages.

Cette attaque est illustrée à la figure 5.14.

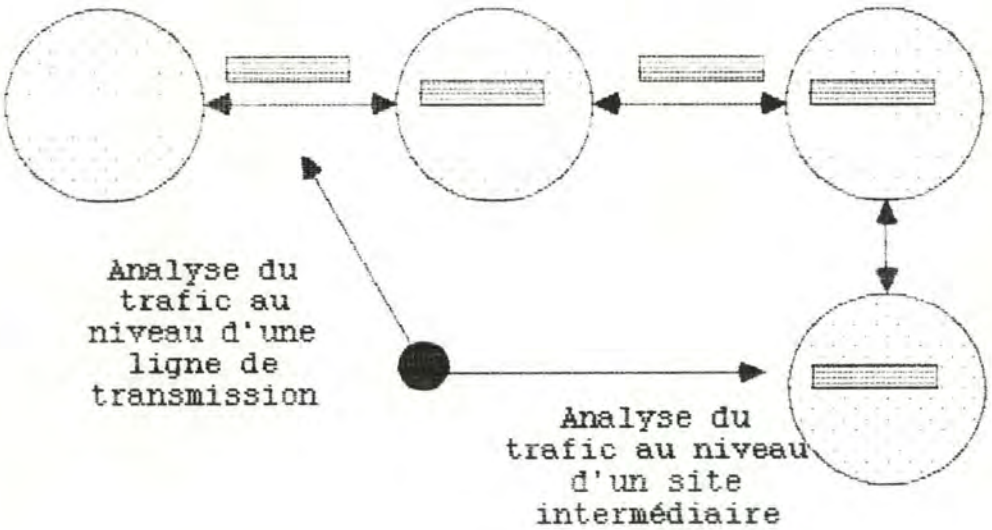

Figure 5.14 : analyse du trafic

4.2 Situation actuelle

Aucune mesure particulière n'est actuellement prise pour empêcher qu'une personne analyse le volume des messages échangés.

Une manière de limiter les endroits d'où peut provenir une attaque est l'établissement d'une liaison UUCP. Ainsi, l'analyse du trafic ne peut se produire que sur le site d'émission et le site de destination, et sur la ligne de transmission entre ces deux sites. Pour remédier à ce problème, les techniques que nous abordons dans le point suivant peuvent être envisagées comme complément à cette liaison.

Nous allons maintenant déterminer des techniques qui empêchent, d'une part l'analyse de la fréquence d'échanges de messages et d'autre part, l'analyse de leur longueur.

4.3 Propositions de solutions

4.3.1 Fréquence des messages

A) Envoi en permanence de messages sur le réseau

a) Principe

Une solution au problème de l'analyse de la fréquence des messages est de faire circuler en permanence sur le réseau le même nombre de messages. Certains de ceux-ci n'auront évidemment aucune signification tandis que d'autres seront tout à fait valides. Une personne voulant analyser le trafic sur le réseau ne pourrait de ce fait pas obtenir des renseignements significatifs.

Pour que les messages n'ayant pas de sens ne puissent pas être détectés par un tiers, il faut qu'ils soient chiffrés, en ne produisant pas chaque fois le même texte chiffré. il faut donc modifier le contenu et la longueur de chaque message vide de sens. Une solution à ce problème est de choisir la longueur du message de manière aléatoire et d'en construire le contenu en choisissant aléatoirement des caractères. Une autre manière pour masquer la longueur des messages est d'employer une longueur standard comme nous le verrons dans le point suivant.

Le mécanisme de chiffrement peut être indifféremment à clé publique ou à clé privée. Le problème de la distribution de ces clés a déjà été abordé au point 2.3.1 de ce chapitre.

Le site de destination doit pouvoir déterminer si un message est un valide ou vide de sens. Une solution est de convenir de la position dans le message d'un indicateur qui, une fois le message déchiffré, permettrait de savoir s'il est valide ou non.

Cette solution est schématisée à la figure 5.15.

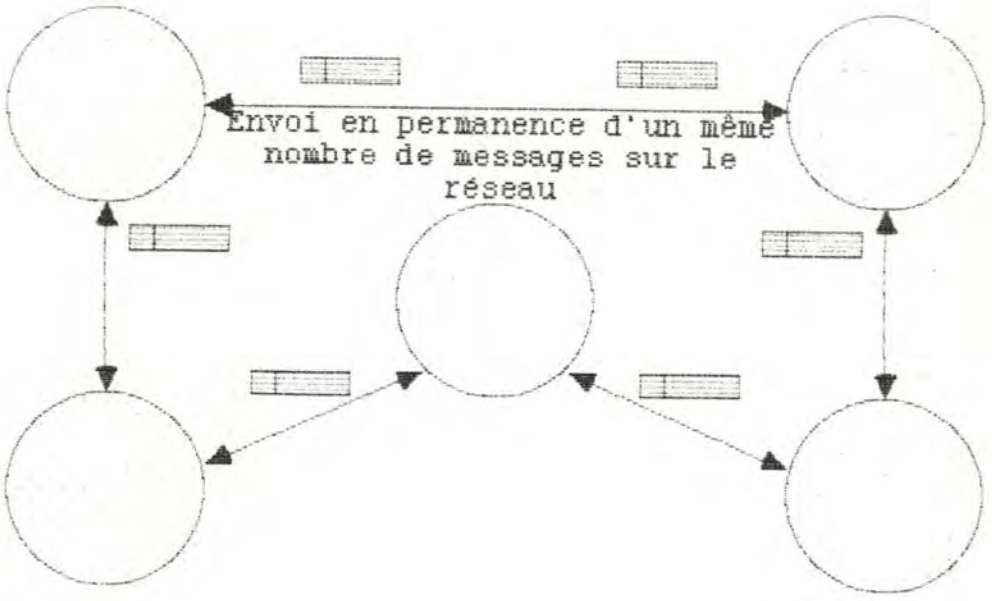

Figure 5.15 : envoi permanent de messages sur le réseau

b) Avantage

- L'avantage de cette technique est qu'elle masque totalement la fréquence réelle des messages.

c) Inconvénients

- Cette solution présente le gros désavantage d'accroître le volume des communications et par le fait même d'accroître le coût total de transmission des messages sur le réseau.
- Si le nombre de messages devant être effectivement transférés est supérieur au nombre préalablement fixé, certains de ceux-ci seront retardés dans leur acheminement vers le site voisin car ils devront attendre l'activation suivante de UUCICO.

d) Applicabilité

Cette solution est applicable lorsque d'une part le besoin de masquer la fréquence des messages est d'une

importance considérable, et d'autre part que les sites concernés par la mise en place d'un CDC ou d'un AC pour la distribution des clés ne sont pas nombreux. Cela peut être envisagé dans le cas d'un sous-réseau comprenant quelques sites, chacun d'eux relié à un CDC (AC) par une liaison UUCP.

4.3.2 Longueur des messages

A) Détermination d'une longueur standard

a) Principe

Une solution au problème de l'analyse de la longueur des messages est de leur fixer une longueur standard.

Deux cas peuvent se présenter lors de l'envoi d'un message :

- le message est plus court que la longueur standard.
- Le message est plus long que la longueur standard.

Si le message est plus court, il suffit de le compléter avec une séquence de caractères non significatifs, déterminée préalablement pour que la séquence puisse être enlevée à la réception du message (figure 5.16). Pour qu'un "ennemi" ne puisse pas détecter ces caractères non significatifs, il faut chiffrer de bout en bout le message. Pour que le chiffrement de cette séquence ne produise pas toujours le même résultat, il faut employer un algorithme de chiffrement qui tienne compte de ce qui a déjà été chiffré [22].

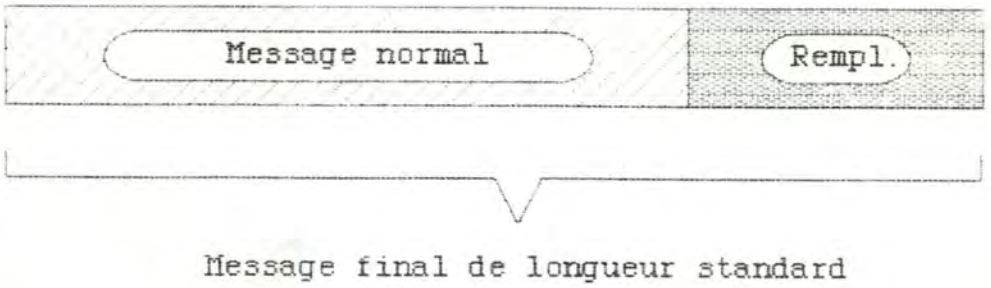

Figure 5.16 : remplissage du message

Si le message est plus long, il faut le couper en plusieurs parties de longueur standard, avec éventuellement la dernière partie complétée par une séquence de caractères non significatifs (figure 5.17), de la même manière que dans le cas où le message est plus court. Chaque partie du message doit pouvoir être remise dans le bon ordre, afin de reconstituer le message initial. Pour ce faire, il faut ajouter à chaque partie un indicateur signalant s'il y a encore une autre partie à recevoir ainsi qu'un numéro de séquence permettant de reclasser les différentes parties (il faut se rappeler ici que le temps d'acheminement vers un même site de destination peut être variable d'un message à l'autre, comme nous l'avons vu au chapitre IV).

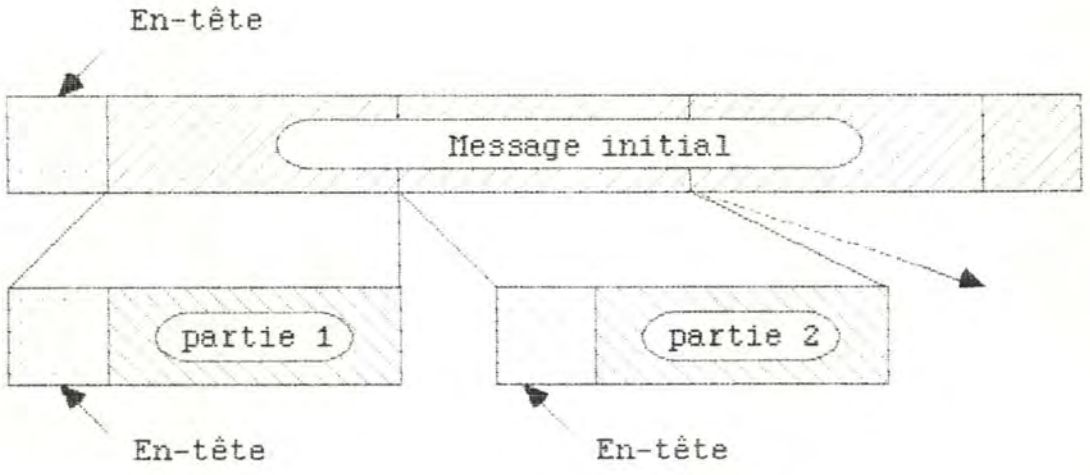

Figure 5.17 : éclatement du message

Cette longueur peut être fixée en examinant préalablement la taille moyenne des messages qui circulent sur le réseau. Il faut ensuite réaliser le meilleur compromis entre une longueur trop grande (qui augmenterait considérablement le nombre de caractères transmis) et une longueur trop petite (qui augmenterait le temps d'acheminement du message complet)

b) Avantage

- L'avantage de cette technique est qu'elle masque totalement la longueur réelle des messages.

c) Inconvénients

Cette technique présente également des inconvénients :

- dans le cas où le message est plus court que la longueur standard, il y a un accroissement du nombre de caractères échangés, et donc une augmentation du coût de la transmission.

- Dans le cas où le message est plus long, il doit être divisé en plusieurs parties, chacune étant envoyée séparément vers le site de destination. Avant que le message ne puisse être déposé dans la boîte aux lettres du destinataire, il faut que toutes les parties constituant le message initial aient été reçues. Or, le risque qu'une ou plusieurs parties du message restent bloquées pendant un certain temps sur un site intermédiaire augmente. Le temps d'acheminement total du message peut donc dans certains cas être fort augmenté. De plus, à cause de la redondance de l'en-tête dans chaque partie et du remplissage de la dernière de celles-ci si sa longueur est inférieure à la longueur standard, le nombre total de caractères envoyés, et donc le coût, sont plus élevés.

d) Applicabilité

Cette solution est uniquement envisageable lorsque l'analyse de la longueur des messages est un facteur d'insécurité élevé. Si ce n'était pas le cas, le coût d'exploitation d'une telle solution serait inacceptable.

Voyons maintenant comment implémenter pratiquement ces mécanismes sur le réseau EUNET.

C'est au niveau du site, et non au niveau de l'utilisateur, que ces mécanismes doivent être implémentés. En effet, ils font partie de la politique générale de gestion du réseau par le site.

4.4 Considérations d'implémentation de la solution préconisée

4.4.1 Fréquence des messages

Il faut qu'à chaque activation du programme UUCICO (voir chapitre III pour une description détaillée de ce programme), un même nombre de messages soient envoyés. Si le nombre de messages devant être effectivement envoyés n'est pas atteint, alors des messages vides de sens sont transmis. L'organigramme schématisant l'émission d'un message est donné à la figure 5.18 et l'organigramme schématisant sa réception est donné à la figure 5.19.

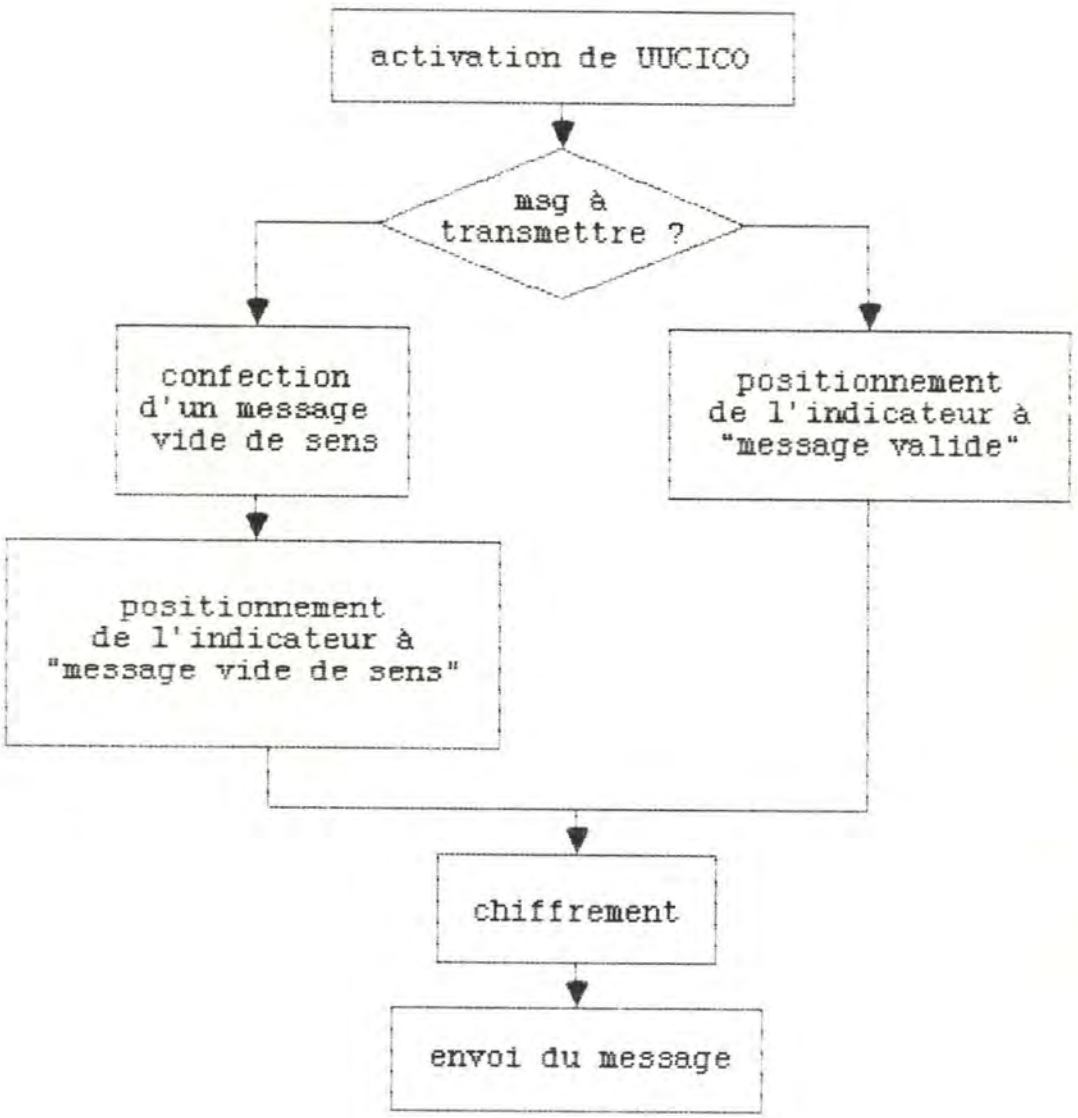

Figure 5.18 : transmission d'un message

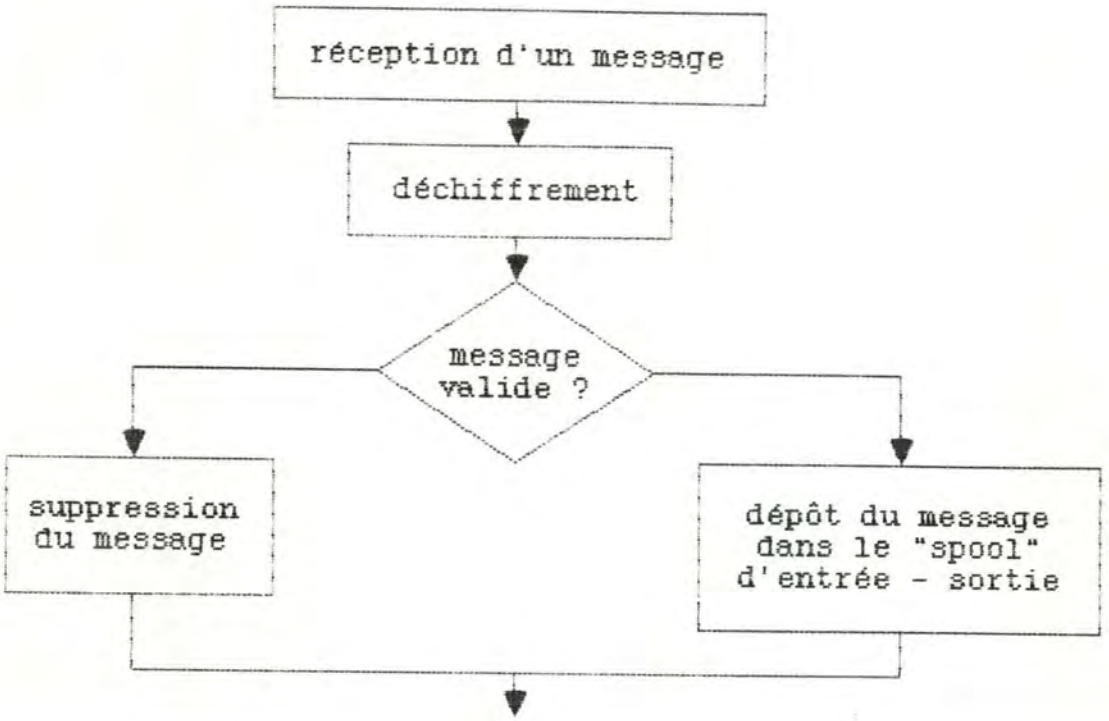

Figure 5.19 : réception d'un message

4.4.2 Longueur des messages

L'opération de remplissage ou d'éclatement des messages doit se dérouler après l'analyse de ceux-ci par le programme SENDMAIL (voir le chapitre III pour une description de ce programme) et avant le transfert effectif par le programme UUCP.

Toutes les parties du message doivent avoir le même en-tête, afin d'être acheminées correctement. L'indicateur doit être positionné et le numéro de séquence mis à jour.

Les opérations qui doivent être effectuées sur le site d'émission sont décrites par l'organigramme de la figure 5.20. Les opérations sur le site de destination sont, quant à elles, décrites par l'organigramme de la figure 5.21.

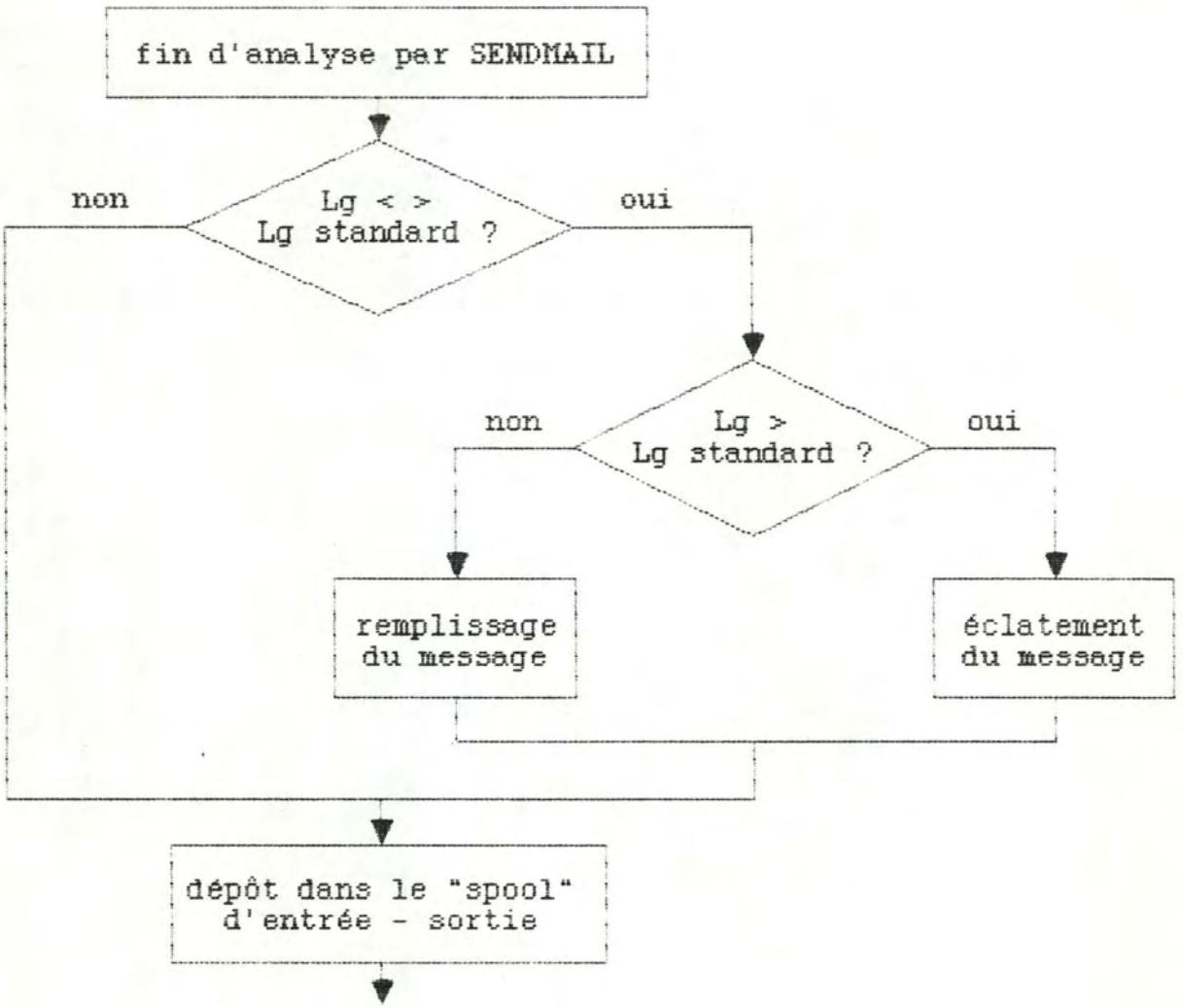

Figure 5.20 : opérations sur le site d'émission

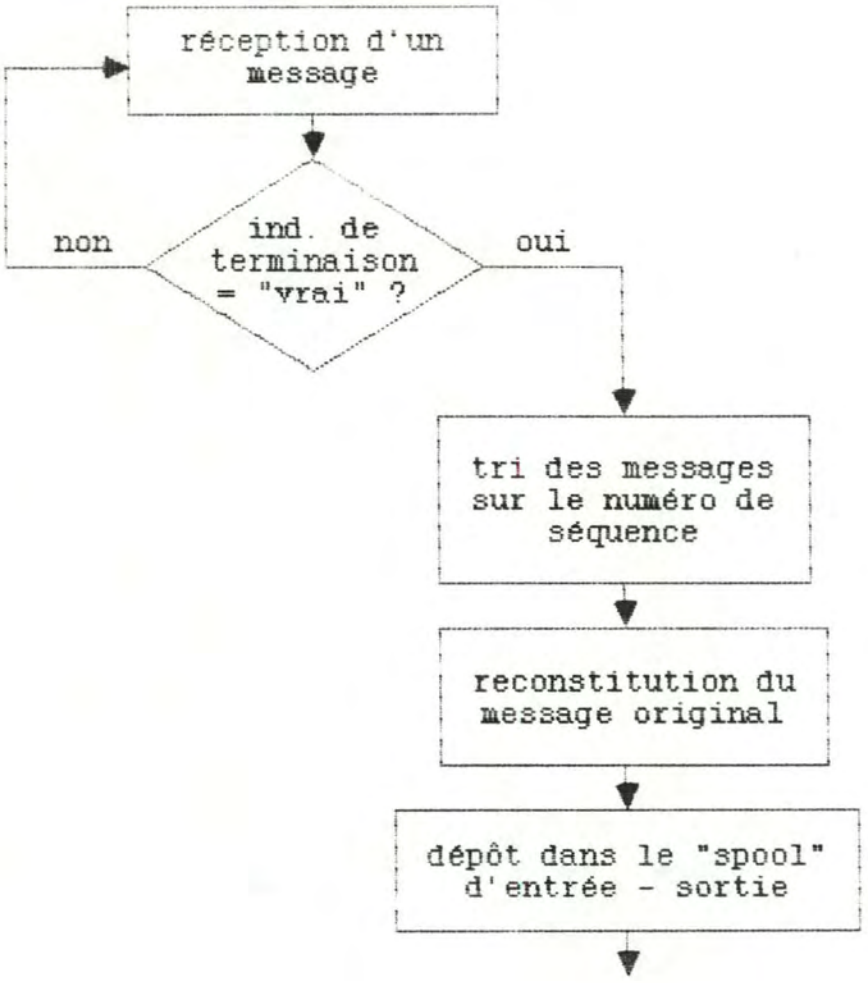

Figure 5.21 : opérations sur le site de destination

5. Modification de messages

5.1 Exposé de l'attaque

Il est des cas où le récepteur d'un message désire avoir la certitude que celui-ci lui soit bien destiné, et que ni l'adresse de l'émetteur, ni le contenu du dit message n'aient été modifiés depuis son envoi. Cette attaque est schématisée à la figure 5.22.

Cette certitude ne pourra être acquise qu'au prix de l'utilisation de techniques spécifiques d'authentification. Les raisons qui justifient la mise en oeuvre de ces techniques sont les mêmes que celles qui justifiaient les méthodes de protection des messages contre toute lecture par un tiers dans le point 2.1 de ce chapitre.

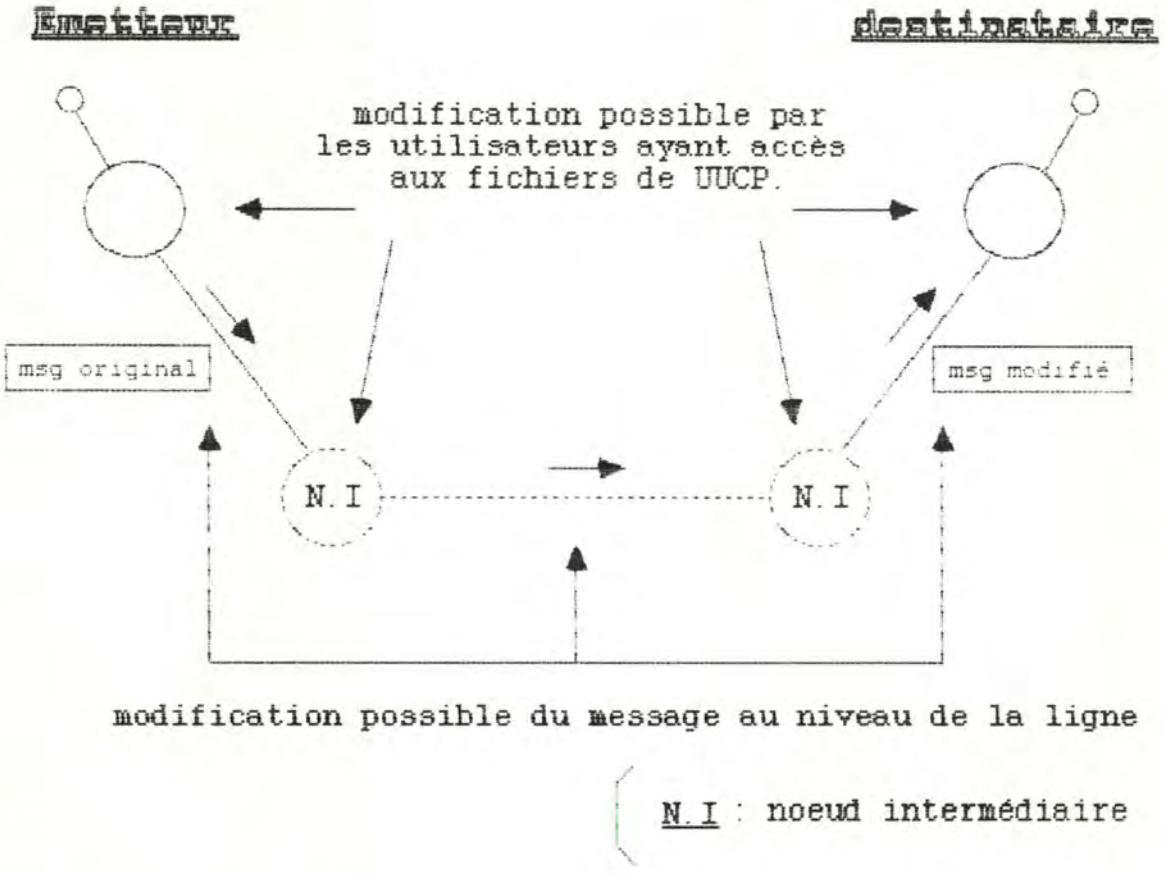

Figure 5.22 : modification de messages

5.2 Situation actuelle

L'utilisateur du réseau ne dispose pas, dans l'état présent, de moyens lui permettant de s'assurer que les messages qu'il reçoit sont authentiques.

Il est néanmoins possible dans l'état actuel des choses, de diminuer le risque d'encourir de telles attaques. Il suffit pour ce faire d'établir une liaison UUCP entre le site local et le ou les sites qui veulent se soustraire à cette attaque.

5.2.1 Etablissement d'une liaison UUCP

a) Principe

Le principe de l'établissement d'une liaison directe a déjà été présenté au point 3.1 du chapitre IV. Nous ne considérons ici que les avantages et inconvénients de l'établissement d'une liaison directe qui sont directement liés au problème de la modification de messages.

b) Avantage

- Le seul avantage que présente l'installation d'une liaison directe est qu'il ne faille plus craindre la modification de messages par les personnes ayant accès aux fichiers du logiciel de communication, des sites intermédiaires du chemin d'accès de l'émetteur au destinataire.

c) Inconvénients

- Cette façon de procéder n'empêche pas la modification des messages par les utilisateurs ayant accès aux fichiers du logiciel de communication, des sites de l'émetteur et du destinataire de messages.

- Cette technique, à elle seule, ne résout en rien le cas où l'ennemi entreprendrait son attaque au niveau de la ligne. Elle peut cependant être combinée avec la technique de chiffrement au niveau de la ligne. Utilisées conjointement, ces techniques n'empêchent

les modifications que dans la mesure où l'"ennemi" ne sait prendre connaissance des messages que dans leur forme chiffrée. Elles ne permettent cependant pas la détection de telles attaques.

d) Applicabilité

Si le caractère authentique des messages ne revêt pas une importance capitale, et que les destinataires de messages dont le caractère authentique est souhaité ne sont pas répartis sur un nombre trop élevé de sites, et que ces sites ne sont pas trop éloignés, cette façon de procéder peut suffire. Ces conditions sont cependant rarement réunies. Nous envisageons donc d'autres solutions au point suivant.

5.3 Propositions de solutions

Comme pour toute autre attaque active, les solutions qui sont proposées ici ne visent qu'à détecter si oui ou non il y a eu attaque.

Les moyens de détection offerts par la théorie (cfr [18]) peuvent être classés en deux catégories. Une première catégorie se base sur l'utilisation d'une clé privée connue de l'émetteur et du destinataire. La deuxième catégorie, quant à elle, ne nécessite pas l'utilisation de clés.

Analysons dans un premier temps les méthodes relevant de la première catégorie.

5.3.1 Méthodes d'authentification utilisant une clé

Les méthodes d'authentification utilisant une clé peuvent encore être partitionnées en deux classes. La première de ces deux classes comprend toutes les techniques utilisant le chiffrement comme moyen de détection de modifications. La deuxième classe, quant à elle, n'utilise pas le chiffrement.

Voyons d'abord comment fonctionnent les techniques de la première classe.

A) Techniques d'authentification utilisant le chiffrement

a) Principe

Les techniques basées sur le chiffrement fonctionnent comme suit :

le message à authentifier est divisé en blocs de taille équivalente (des blocs de 16 bits par exemple). Le dernier bloc est éventuellement complété par des zéros. Ces blocs sont ensuite additionnés modulo "n" ("n" dépend de l'implémentation) et la somme ainsi obtenue est accolée en fin de message avant que celui-ci ne soit chiffré. Cette somme porte le nom de "code de détection de manipulation" (CDM). Si maintenant le message est modifié pendant son acheminement vers le destinataire, la probabilité que la relation entre le message et le CDM soit modifiée dans le message déchiffré tend vers 1. Autrement dit, la quasi totalité des messages modifiés pendant leur transfert est

détectée. Cette technique est illustrée à la figure 5.23.

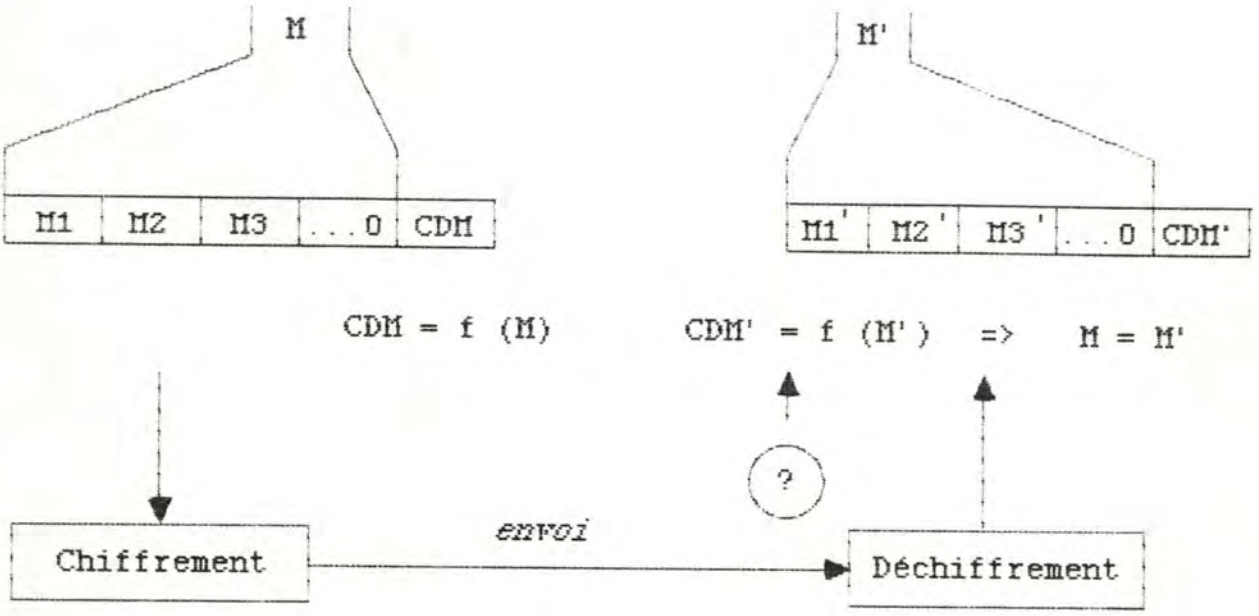

Figure 5.23 : authentification utilisant le chiffrement

Voyons ci-dessous, quels sont les avantages et inconvénients de cette technique.

b) Avantage

- L'avantage de cette façon de procéder est qu'elle combine en une seule opération le chiffrement et l'authentification des messages. Elle empêche donc toute modification ainsi que toute lecture par une personne non autorisée.

c) Inconvénients

- Cette façon de procéder implique que tout message que l'utilisateur désire authentifier soit également chiffré.

- Les clés de chiffrement et d'authentification sont ce faisant les mêmes, ce qu'un utilisateur particulièrement méfiant pourrait vouloir éviter.

d) Applicabilité

Cette technique est particulièrement intéressante dans le cas où tout message à authentifier est également confidentiel. Il faut cependant que le chiffrement utilisé pour assurer la dite confidentialité soit à clé privée.

En ce qui concerne EUNET, cette façon de procéder n'est pas des plus recommandables. La raison principale à cela est que la distribution des clés privées est difficilement envisageable pour ce réseau, comme nous l'avons déjà souligné dans le point 2.3.1 de ce chapitre.

Voyons maintenant quel est le principe des techniques non axées sur le chiffrement.

B) Techniques d'authentification n'utilisant pas le chiffrement

a) Principe

Les techniques d'authentification n'utilisant pas le chiffrement sont basées sur l'emploi de ce qui dans la littérature est appelé un authentificateur.

Un authentificateur est une fonction "A (K, M)", où "K" représente une clé privée connue seulement de l'émetteur et du destinataire du message, et "M" un message à authentifier, de longueur quelconque.

La fonction "A (K, M)" doit dépendre de chaque bit du message et est basée sur un algorithme qui peut être connu de tous. La valeur "A (K, M)" est ajoutée en fin de message, et est transmise avec lui. Le récepteur du message fait le même calcul en utilisant la même clé, et rejette pour non authenticité tout message pour lequel la valeur calculée n'est pas égale à la valeur reçue.

La qualité requise pour qu'un authentificateur soit efficace est que la modification d'un seul bit du message ou de la clé ait pour effet de modifier au moins la moitié des bits du résultat du calcul.

Une des manières d'implémenter cette méthode consiste à

réserver un champ "Q" de taille donnée en fin du message à authentifier. Ce champ "Q" est soit une constante donnée connue de tous, soit une valeur connue seulement de l'émetteur et du destinataire du message, dans lequel cas "Q" fait partie de la clé. Une fois la valeur de la fonction "A (K, M)" calculée, le champs "Q" est remplacé par le résultat. Cette technique est illustrée à la figure 5.24.

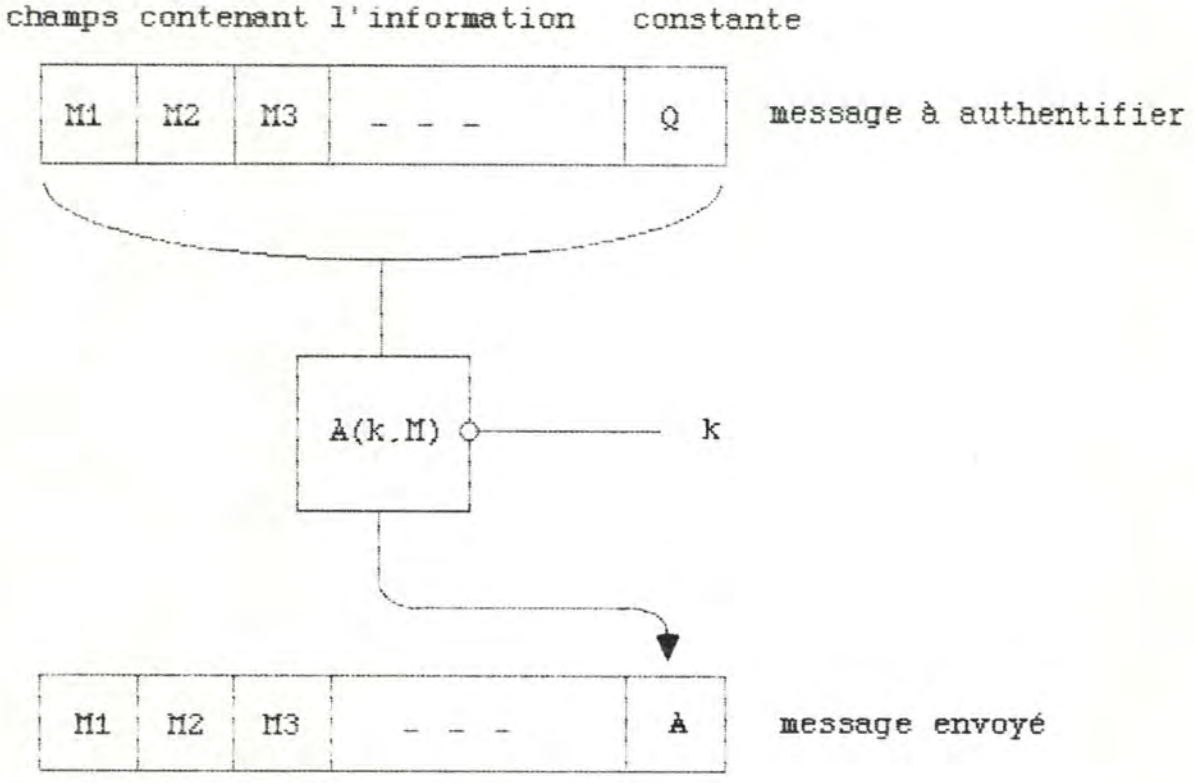

Figure 5.24 : authentification au moyen d'un authentificateur

Voyons quels sont les avantages et inconvénients de cette technique.

b) Avantage

- Cette technique préserve l'indépendance entre l'authentification et le chiffrement. Un utilisateur peut donc décider d'authentifier son message, de le chiffrer, ou de combiner les deux mécanismes comme bon lui semble. Les clés utilisées pour chacune de ces opérations peuvent également être différentes.

c) Inconvénients

Cette technique ne présente pas d'inconvénient notoire.

d) Applicabilité

Cette technique apparemment dépourvue d'inconvénients n'en est cependant pas exempte dès qu'il est question de l'adapter aux messages véhiculés par EUNET. En effet, cette technique exige l'utilisation d'une clé privée. Or la distribution de celles-ci par le biais de EUNET n'est pas chose triviale comme il a été mis en évidence dans le point 2.3.1 du présent chapitre.

La présentation des techniques utilisant une clé privée étant effectuée, passons à présent aux techniques n'en utilisant pas.

5.3.2 Méthodes d'authentification n'utilisant pas de clé

a) Principe

Les méthodes d'authentification n'utilisant pas de clé fonctionnent comme suit : au lieu d'utiliser un authentificateur "A (K, M)", ces méthodes utilisent une fonction à sens unique "A (M)", fonction seulement du message. Pour authentifier un message "M", la quantité "A (M)" est calculée et transmise au destinataire du message. Le destinataire du message refait ce calcul et rejette pour non authenticité, tout message pour lequel la quantité reçue diffère de la quantité calculée.

Pour que les modifications de messages soient détectables, il faut qu'il soit impossible à un "ennemi" de concocter un message ayant même valeur d'authentificateur que le message qu'il désire modifier. Autrement dit, il faut que la fonction d'authentification soit telle que, étant donnée une valeur "Ax" de celle-ci, il soit extrêmement difficile de trouver un message "M" tel que "A (M)" = Ax.

Il faut remarquer que cette technique d'authentification ne fonctionne que si le destinataire du message est sûr que la valeur "A (M)" reçue est authentique. Celle-ci doit donc avoir été transmise par un moyen permettant d'en vérifier l'authenticité, comme par exemple le téléphone.

Voyons quels sont les avantages et inconvénients de cette technique.

b) Avantages

- L'avantage principal de cette technique est évidemment qu'elle supprime tout problème de gestion de clés.

- L'authentification des messages est indépendante des mesures qui pourraient être prises pour leur assurer un caractère confidentiel.

- La longueur des messages ne se trouve pas augmentée comme elle l'était après application des autres techniques.

c) Inconvénient

- L'inconvénient majeur de cette technique est que l'authentificateur doit lui-même être authentifié. Le lecteur est en droit de se demander à quoi cette technique peut servir, car il paraît insensé de communiquer un authentificateur, alors que le message lui-même pourrait être transmis. Il est des cas où cette technique est néanmoins intéressante. Par exemple dans le cas de messages confidentiels, ou dans le cas de messages très longs, ou encore dans le cas de messages qui ne peuvent être acheminés que par un réseau de données, comme par exemple un programme objet.

d) Applicabilité

Cette technique d'authentification est certainement celle qui soit la plus adaptée au réseau EUNET, parce que d'une part, elle ne nécessite pas l'utilisation de clés, et d'autre part parce qu'elle ne nécessite qu'un minimum de modifications aux programmes existants. De plus, ces modifications sont localisées et ne concernent que la couche "application" du réseau, à savoir le programme MAIL.

C'est donc cette méthode que nous préconisons, et dont nous présentons les considérations d'implémentation.

5.4 Considérations d'implémentation de la solution préconisée

Il faut donner à l'émetteur de messages d'une part, et au destinataire de messages d'autre part les moyens de gérer l'authentification des messages qu'ils envoient ou reçoivent.

Situons nous d'abord du point de vue de l'émetteur de messages.

5.4.1 Point de vue de l'émetteur

L'émetteur de messages doit disposer de commandes lui permettant d'obtenir la valeur d'authentificateur associée aux messages qu'il envoie. Les modifications à apporter au programme MAIL pour qu'il incorpore cette fonction relèvent de deux catégories. Les modifications de l'interface utilisateur constituent la première, la seconde est, quant à elle, constituée des autres modifications. Voyons d'abord les modifications à apporter à l'interface utilisateur existant.

a) Modifications de l'interface utilisateur

Les modifications à apporter à l'interface utilisateur doivent être telles que la commande d'obtention de l'authentificateur d'un message soit analogue à toute autre commande (voir [12]). Cette demande doit donc pouvoir s'effectuer selon les modes suivants :

- au niveau de l'interpréteur de commandes, en tant qu'option du programme MAIL.

  Par exemple : mail jules -a

  où l'option "-a" indique au programme MAIL que la valeur de l'authentificateur du message destiné à l'utilisateur "jules" soit affichée sur le moniteur après achèvement de son édition.

- Au niveau de l'interpréteur de commandes du programme MAIL.

  Par exemple : mauthentic jules

  où la commande "mauthentic" remplace l'habituelle

commande MAIL, et signifie que l'édition du message doit se terminer par l'affichage sur le moniteur de sa valeur d'authentificateur.

- Au niveau de l'édition même du message, sous la forme d'un "tilde escape" (voir manuel d'utilisation du programme MAIL).

  Par exemple : ~a

- Au niveau des options par défaut du programme MAIL, en donnant la valeur "vrai" à l'option binaire que nous pourrions appeler "authenticate".

  Par exemple : set authenticate

  cette option indiquant que pour tout message édité, la valeur d'authentificateur doit être affichée sur le moniteur.

Les modifications à apporter à l'interface utilisateur ayant été présentées, passons aux autres modifications.

b) Autres modifications

Les modifications à apporter au programme MAIL, et ne relevant pas de l'interface utilisateur sont schématisées à l'aide de l'organigramme de la figure 5.25.

L'effet de l'une des quatre façons d'invoquer la fonction d'authentification est de positionner un booléen à "vrai". Ce booléen doit être testé à la fin de l'édition de tout message. Les actions entreprises selon sa valeur ne nécessitent pas de complément d'information, si ce n'est que, dans le cas où le message à envoyer doit également être chiffré, la valeur d'authentificateur n'est calculée qu'après chiffrement. En effet, dans ce cas, c'est le message chiffré qui doit être authentifié.

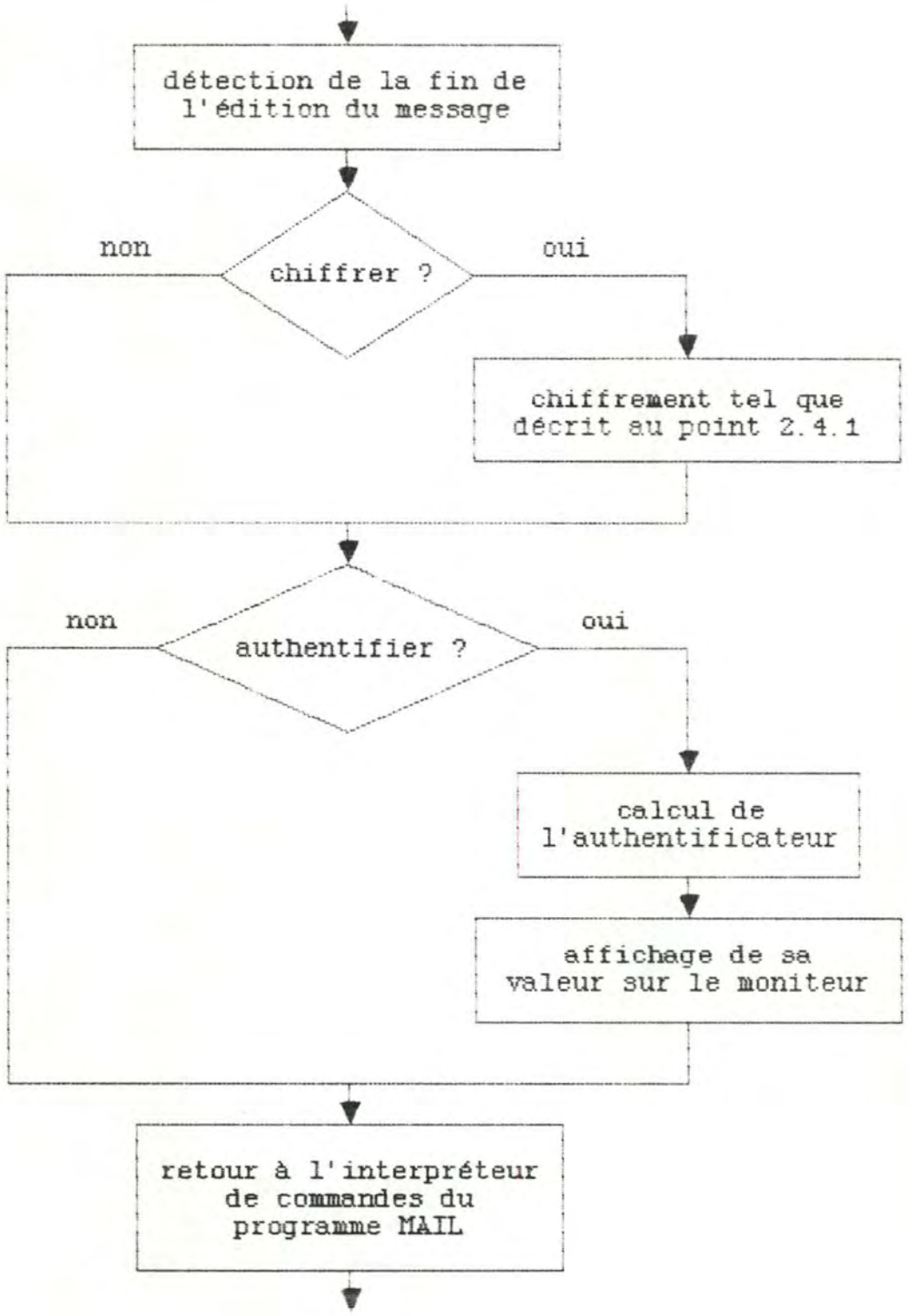

Figure 5.25 : gestion des demandes d'authentification du point de vue de l'émetteur

Après avoir présenté les modifications à apporter au programme MAIL en ce qui concerne l'émetteur de messages, passons aux modifications du programme MAIL concernant leur(s) destinataire(s).

5.4.2 Point de vue du destinataire

Le destinataire de messages doit disposer de commandes lui permettant d'obtenir la valeur d'authentificateur d'un message donné. Voyons d'abord les modifications à apporter à l'interface utilisateur pour qu'il intègre cette fonction.

a) Modifications de l'interface utilisateur

La valeur d'authentificateur d'un message doit pouvoir être obtenue selon les modes suivants :

- au niveau de l'interpréteur de commandes du programme MAIL.

  Par exemple : authentic n

  où "n" est le numéro d'un message existant dont l'utilisateur désire avoir la valeur d'authentificateur.

- Au niveau de la composition d'un message, sous la forme d'un "tilde escape" (voir manuel d'utilisation du programme MAIL).

  Par exemple : ~a n

  où "n" est le numéro d'un message existant.

Voyons maintenant les autres modifications.

b) Autres modifications

Les modifications ne relevant pas de l'interface utilisateur concernent le calcul et l'affichage sur le moniteur de la valeur d'authentificateur du message identifié par le numéro donné en paramètre. Leur implémentation ne nécessite pas davantage d'explications.

6. Duplication de messages

6.1 Exposé de l'attaque

Quelle que soit la technique utilisée pour s'assurer qu'un message n'ait pas été modifié entre son envoi et sa réception, celle-ci n'empêche nullement un tiers de prendre connaissance d'un message et d'en réinjecter ultérieurement une copie sur le réseau. Si la copie du message est envoyée avant que les correspondants aient changé de clé, elle ne sera rejetée que si les moyens de détection adéquats sont mis en oeuvre.

La duplication de messages est une attaque qui peut être menée à partir d'un noeud du chemin d'accès ou directement au niveau de la ligne entre émetteur et destinataire de messages, comme nous le montre la figure 5.26.

Nous ne considérons dans ce point que le problème de la duplication des messages des utilisateurs. La duplication de messages propres au logiciel de communication concerne plus particulièrement le problème de l'établissement d'une connection sous une fausse identité.

Remarque : cette attaque est davantage crainte dans le cas de transactions bancaires que dans le cas d'une messagerie électronique.

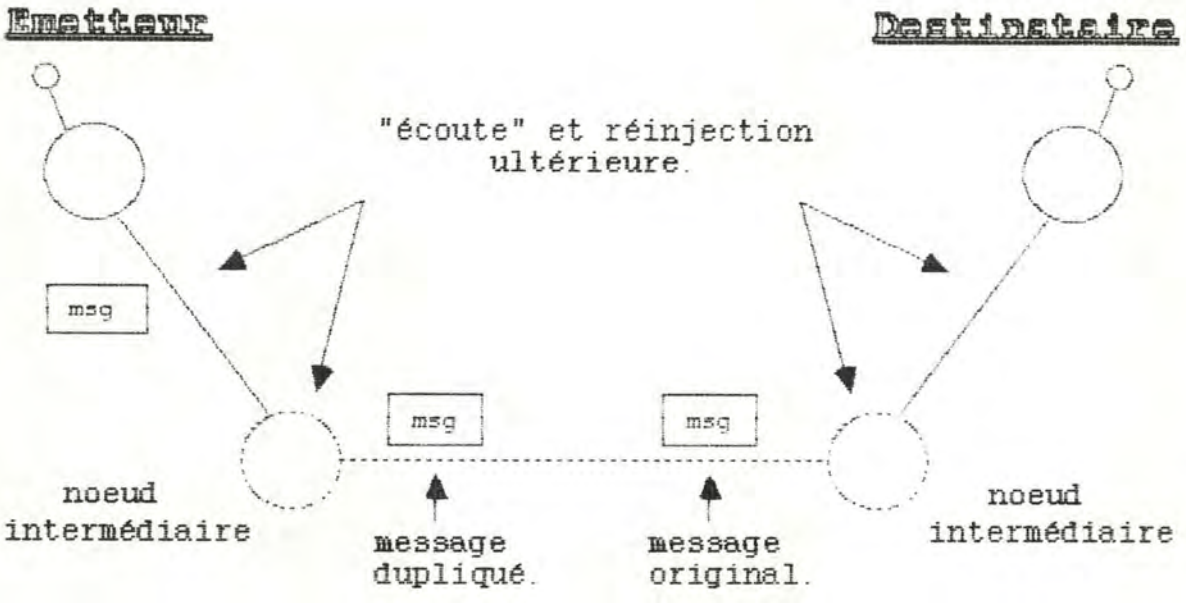

Figure 5.26 : duplication de messages

6.2 Situation actuelle

Le logiciel de communication tel qu'il existe actuellement est incapable de rejeter un message qui serait le duplicata d'un message antérieur.

L'utilisateur n'ayant pas les moyens de détecter les duplicata et craignant la réinjection de copies de ses messages ne peut donc qu'essayer de réduire, dans la mesure du possible, les endroits d'où de telles attaques peuvent être menées.

Les moyens dont l'utilisateur dispose pour agir de la sorte sont en fait extrêmement limités et se résument à l'établissement d'une liaison UUCP entre son site et le site de son destinataire.

6.2.1 Etablissement d'une liaison UUCP

a) Principe

Le principe de l'établissement d'une liaison UUCP a déjà été présenté au point 3.1 du chapitre IV. Nous ne considérons ici que les avantages, inconvénients et conditions d'applicabilité de l'installation d'une liaison directe qui sont directement liés au problème de la duplication de messages.

b) Avantage

- Le seul avantage de l'installation d'une liaison UUCP est qu'il ne faille plus craindre la réinjection de duplicata, par les personnes ayant accès aux fichiers du logiciel de communication, des sites intermédiaires du chemin d'accès de l'émetteur au destinataire.

c) Inconvénients

- Cette façon de procéder n'empêche pas la réinjection d'une copie d'un message par les utilisateurs ayant accès aux fichiers du logiciel de communication, des sites de l'émetteur et du destinataire de messages.

- Cette technique ne résout en rien le cas où l'"ennemi" entreprendrait son attaque au niveau de la ligne.

d) Applicabilité

Considérant le fait que les messages dont il est question ici sont des messages destinés à des opérateurs humains, installer une liaison UUCP pour diminuer le risque de recevoir une copie d'un message précédent, semble être inutile. En effet, si la réinjection d'un duplicata suit de près l'émission de l'original, il y a fort à croire que le destinataire se rende compte de la chose et qu'il rejette de ce fait le message de source douteuse. Par contre, si la réintroduction d'un message est opérée longtemps après l'émission de l'original, un utilisateur doutant de l'authenticité du message reçu peut en demander la confirmation par un moyen tel que par exemple le téléphone, ce qui n'est que peu contraignant, parce que rare.

Dans tous les cas, installer une liaison UUCP est ici dépourvu d'intérêt.

6.3 Propositions de solutions

La solution au problème de la duplication de messages est très simple en principe, puisqu'il suffit de rendre tout message différent des précédents et de rejeter tout message déjà reçu. Cette façon de procéder est néanmoins très laborieuse, et ne peut constituer une solution générale au problème. Voyons donc dans ce qui suit les techniques qui nous sont proposées dans la littérature (cfr [18]).

6.3.1 Utilisation d'un numéro de séquence

a) Principe

Une des façons de procéder est d'adjoindre à tout message un numéro de séquence incrémenté d'une unité pour chaque message envoyé. Chaque entité du réseau garde, pour toute autre entité, le numéro du dernier message reçu, ainsi qu'une trace du prochain numéro à envoyer (cfr figure 5.27).

L'authentification porte ici sur le message ainsi que sur le numéro de séquence, de sorte que ni le message, ni le numéro de séquence qui y est joint ne peuvent être modifiés sans que ces modifications ne soient détectées. Etant donné que toute entité du réseau sait quel est le prochain numéro de séquence qu'elle doit recevoir, les duplicata sont détectables et peuvent donc être rejetés.

b) Avantage

- Cette méthode permet non seulement la détection des duplicata, mais permet également la détection du réordonnancement des messages par un "ennemi", ainsi que la détection de messages perdus ou insérés.

c) Inconvénients

- Cette technique implique qu'il faille tenir à jour, pour tout site, le numéro du prochain message à envoyer, et le numéro du prochain message à recevoir. Il est cependant possible d'alléger quelque peu la quantité de numéros à stocker si au lieu de vérifier que le numéro reçu est égal au précédent incrémenté

de un, on se contente de tester qu'il est supérieur au dernier numéro reçu. Cette règle moins stricte permet de n'utiliser qu'un seul numéro pour les messages à envoyer mais ne supprime pas pour autant la nécessité de mémoriser, pour toute entité communicant avec l'entité locale le numéro du dernier message reçu.

- Cette technique exige des messages qu'ils soient reçus dans l'ordre dans lequel ils sont envoyés.

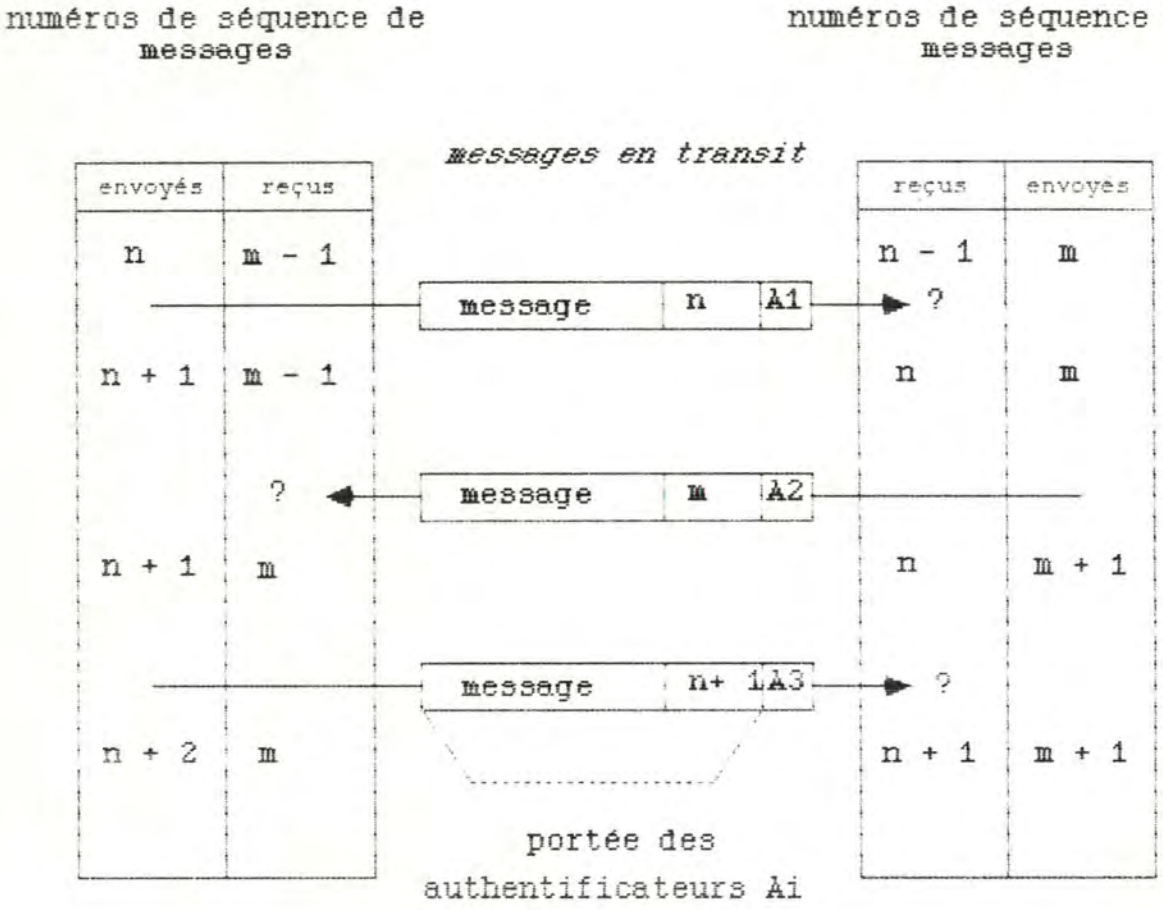

Figure 5.27 : adjonction d'un numéro de séquence

D) Applicabilité

Cette technique n'est applicable pour EUNET que si les mesures adéquates sont prises pour assurer que les messages véhiculés entre deux sites arrivent dans l'ordre dans lequel ils ont été émis.

Pour satisfaire cette condition, il est nécessaire d'installer une liaison directe entre les sites concernés et le site local. Or, comme nous l'avons déjà souligné, installer une liaison directe pour les seuls besoins de la détection de duplicata est une solution à proscrire.

La tenue de numéros de séquence permet, comme nous l'avons déjà dit, de détecter d'autres attaques que la seule détection de duplicata. A ce sujet, il faut encore ajouter que le logiciel UUCP offre la possibilité de gérer des numéros de séquence entre deux sites. Confier la gestion des numéros de séquence à UUCP est cependant dépourvu d'intérêt parce que aucune mesure accompagnative n'est prise pour assurer l'authenticité des dits numéros.

En définitive, nous pouvons affirmer que la tenue de numéros de séquence pour assurer la détection des duplicata, n'est pas recommandable dans ce contexte.

A côté de cette première technique, la littérature en propose une autre qui, quant à elle est basée sur l'utilisation d'estampilles.

6.3.2 Utilisation d'estampilles

a) Principe

La gestion de numéros de séquence peut devenir fastidieuse si le nombre d'entités avec lesquelles l'entité locale communique est élevé. Il est possible de contourner ce problème si au lieu d'ajouter à tout message un numéro de séquence, on lui adjoint son heure d'émission. L'authentification du message doit encore une fois porter sur l'entièreté du message, en ce compris son heure d'émission (voir figure 5.28).

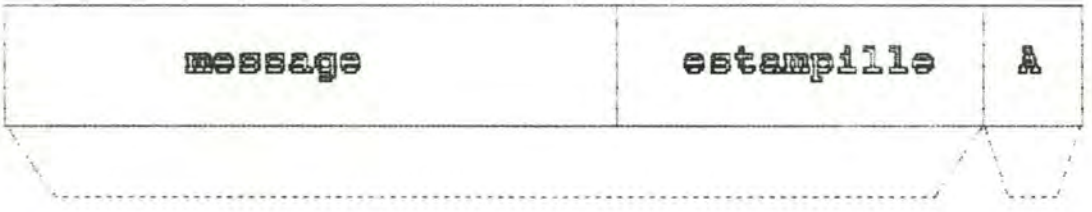

Figure 5.28 : message avec estampille

Si la transmission des messages par le réseau est relativement rapide, le fait de recevoir un message en un temps n'excédant pas une durée donnée peut suffire pour détecter les duplicata.

Si le réseau est à commutation de messages, les messages sont retardés en chaque noeud. Ceci n'empêche pas la méthode de fonctionner, à condition que les messages soient reçus dans l'ordre dans lequel ils ont été envoyés, car dans ce cas l'heure de l'envoi d'un message est plus ou moins équivalente à un numéro de séquence.

b) Avantages

- Cette méthode supprime la nécessité de gérer des numéros de séquence avec tout site susceptible d'entrer en communication avec le site local.

- Cette technique permet, comme la précédente, la détection d'autres attaques, comme par exemple le réordonnancement de messages.

c) Inconvénient

- Cette technique exige dans le cas d'un réseau à commutation de messages, que les messages soient reçus dans l'ordre dans lequel ils ont été émis.

d) Applicabilité

Comme pour la technique précédente, il convient de prendre les mesures nécessaires pour assurer que les messages véhiculés entre deux sites arrivent dans l'ordre dans lequel ils ont été émis. Les mêmes considérations d'applicabilité sont donc de mise ici.

Avant d'envisager l'une ou l'autre des méthodes pour EUNET, il importe d'avoir déjà fait le choix d'une technique d'authentification.

Si nous optons pour la technique préconisée dans le point 5 du présent chapitre, nous faisons d'une pierre deux coups. En effet, la méthode de détection des modifications

que nous préconisons implique d'office une communication par un moyen autre que le réseau; en l'occurence un moyen permettant l'envoi d'une valeur d'authentificateur de façon authentique. Tout message n'ayant pas donné lieu à cette communication n'est donc pas authentique et peut être rejeté.

Si nous optons pour une autre méthode d'authentification, vérifier qu'un message n'est pas un duplicata en communiquant dans les conditions d'authenticité décrites ci-dessus, reste - économiquement parlant - la solution la plus indiquée. Ceci est la conséquence du fait que les messages véhiculés par le réseau sont destinés à des opérateurs humains (nous avons déjà discuté de la question au point 6.2.1 de ce chapitre).

Chapitre VI

Conclusion générale

Arrivés au terme de ce mémoire, il nous reste à mettre en exergue ce qui en constitue l'apport original, et à faire l'inventaire des difficultés que nous avons rencontrées lors de son élaboration.

Comme le dit Denning D. dans [17], les mécanismes assurant la confidentialité d'un produit informatique ayant des exigences en ce domaine, se doivent d'être simples et en nombre restreint. Leur localisation dans les couches les plus basses du système est donc celle qui soit la plus indiquée car, ce faisant, les couches supérieures reposent sur des bases solides. Ceci signifie que la confidentialité doit être, dès le départ, l'un des maîtres mots du développement du produit.

Ceci ne fut pas le cas du réseau EUNET, qui s'est développé, rappelons-le, à partir de ce qui n'était au départ qu'un logiciel de copie de fichiers entre sites. Les conséquences ne tardent d'ailleurs pas à se faire sentir dès qu'il est question d'intégrer au logiciel existant ce qui manque pour lui conférer le degré de confidentialité désiré. Le lecteur aura sans doute constaté qu'il n'existe jamais de solution qui soit à la fois réellement facile à l'usage et simple à implémenter. Que ce soit pour le transfert d'une clé ou pour autre chose, la communication par un moyen autre que le réseau lui-même est souvent de mise. Ceci situe quelque peu les problèmes intrinsèques au fait que des questions aussi fondamentales que la confidentialité n'ont pas eu la place qu'elles auraient dû avoir lors de l'élaboration du logiciel.

En ce qui concerne le problème de la fiabilité d'acheminement, la situation est quelque peu différente, car le réseau comporte déjà certains aménagements à cet égard.

Pour ce qui est des difficultés que nous avons rencontrées, nous pouvons dire que celles-ci sont d'ordres divers. La principale fut certainement l'obtention de la documentation nécessaire à la compréhension du fonctionnement du réseau. Celle-ci est en effet relativement pauvre, quand elle n'est pas constituée exclusivement des programmes sources.

Une autre des difficultés auxquelles nous nous sommes heurtés fut que nous n'ayons jamais pu faire une réelle évaluation de la faisabilité des solutions que nous proposions. La meilleure évaluation eut été, sans aucun doute, l'implémentation de l'une d'entre elles. Nous n'avons cependant pas disposé du temps qu'il aurait fallu pour ce faire. Ceci reste certainement l'une des principales voies qui restent ouvertes au terme de ce mémoire.

Bibliographie

[1] BOURNE, S. R. (1983), *The Unix System*, Bell Laboratories.

[2] ROCHKIND, Marc J. (1985), *Advanced Unix Programming*, Prentice - Hall software series.

[3] GEURTS P. (1986), EUNET : *Problèmes de Sécurité sur FUN-CS*, Rapport à usage interne.

[4] MORRIS R. et THOMPSON Ken (1978), *Password Security : A Case History*, Bell Laboratories.

[5] RITCHIE Dennis M. (1978), *On the Security of Unix*, Bell laboratories.

[6] TANNENBAUM A. S. (1981), *Computer Networks*, Prentice - Hall

[7] NOWITZ D. A. (1978), *Uucp Implementation Description*, Unix Programmer Manual, Version 7, Vol. 2., Bell laboratories.

[8] *Uucp Installation and Administration*, Ultrix Programmer Manual.

[9] HOEBANX Pascal (1985), *An Analysis of the Uucp Package*, Technical Note Nr. 49, Louvain-la-Neuve.

[10] ALLMAN E. (1983), *SENDMAIL - An Internetwork Mail Router*, Britton-Lee Inc.

[11] ALLMAN E. (1983), *SENDMAIL, Installation and Operation Guide*, Britton-Lee Inc.

[12] SHOENS K. (1983), *MAIL Reference Manual.*

[13] NOWITZ D. A., et LESK M. E. (1978), *A Dial-Up Network of Unix Systems*, Unix Programmer Manual Version 7, Vol. 2., Bell Laboratories.

[14] MACCHI C. et GUILBERT J.F. (1983), *Transport et Traitement de l'Information.*

[15] CROCKER D. H. (1982), *Standard for the Format of Arpa Internet Messages.*

[16] VOYDOCK V. L. et KENT S. T. (1984), "Security Mecanisms in a Transport Layer Protocol", *Computer Network*, vol. 8, pp. 433-449.

[17] DENNING D. E. R. (1982), *Cryptography and Data Security*, Addison-Wesley Publishing Company

[18] DAVIES D. W. et PRICE W. L. (1984), *Security for computer Networks*, John Wiley & Sons.

[19] NEEDHAM R. M. et SCHROEDER M. D. (1978), "Using Encryption for Authentication in Large Networks of Computers, *Communications of the ACM*, Vol. 21, Nr. 12, pp. 993-999.

[20] POPEK G. J. et KLINE C. S. (1979), "Encryption and Secure Computer Networks", *Computing Surveys*, Vol. 11, Nr. 4, pp. 331-356.

[21] CHAUM D. L., (1981), "Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms", *Communications of the ACM*, Vol. 24, Nr. 2, pp. 84-88.

[22] ERDEM H., (1987), *Security in Computer Networks*, A Dissertation submitted for the Obtention of the Degree of Doctor in Applied Sciences, V.U.B.

Annexes

Annexe A

1. Fichiers

1.1 Couche 2

1.1.1 Fichiers de gestion des communications

A) L.sys

Le fichier "L.sys" donne, pour chaque site voisin, la procédure que doit suivre le site local pour s'y connecter. Chaque entrée dans ce fichier a le format suivant :

"system-name" "time" "device" "speed" "phone" "login",
où :

- "system-name" est le nom du site voisin.
- "time" est une information spécifiant les jours et les heures dans la semaine où le site voisin peut être appelé.
- "device" est le périphérique à utiliser pour appeler ce site.
- "speed" est la vitesse de transfert à utiliser, celle-ci s'exprime en "bauds".
- "phone" est le numéro de téléphone du site à appeler. Il est constitué d'une abréviation alphabétique optionnelle et d'une partie numérique. L'abréviation doit apparaître dans le fichier "L-dialcodes" décrit plus loin.
- "login" est l'information servant lors de la phase du "login". Elle contient principalement le nom de "login" et le mot de passe du site local sur le site voisin. Comme plusieurs sites peuvent se connecter sous un même nom, la phase de "login" ne permet pas d'identifier un site de façon univoque. C'est le nom de site qui permet de l'identifier. Celui-ci doit donc être unique.

  Elle a le format suivant :

  expect send [expect send] ..., où :

  - "expect" est une chaîne de caractères que le site local s'attend à recevoir du site se connectant, et

- "send" est la chaîne de caractères que le site local envoie une fois la chaîne "expect" reçue.

Exemple d'entrée dans ce fichier :

```
Prlb2 Any ACU 300 mh7654 login uucp ssword: word
```

B) L-dialcodes

Le fichier "L-dialcodes" contient des entrées du format suivant :

"abbreviation" "dial-seq" où :

- "abbreviation" est une abréviation d'un numéro téléphonique.
- "dial-seq" est le numéro correspondant à l'abréviation.

C) L-devices

Le fichier "L-devices" contient une entrée par périphérique utilisé pour la connection au réseau. Chaque entrée dans ce fichier a le format suivant :

"type" "line" "call-unit" "speed", où :

- "type" est le type de périphérique tel que : ACU pour "Auto-Call Unit", Pad pour une sortie asynchrone ou DIR pour une ligne permanente.
- "line" est le nom du périphérique à utiliser pour accéder à la ligne.
- "call-unit" est l'unité d'appel automatique associée (une valeur '0' de ce champ indique que la connection est permanente).
- "speed" est la vitesse en bauds de la ligne.

D) LCK..str

Chacun des fichiers "LCK..str" indique que le site ou le périphérique "str" est occupé. L'existence de tels fichiers empêche l'établissement de plusieurs communications entre deux mêmes sites et toute tentative d'utilisation d'un périphérique déjà occupé.

E) /.ADM/SLAVE.sys

Chacun des fichiers "/.ADM/SLAVE.sys" est un fichier de "debugging" des opérations effectuées dans le mode ESCLAVE pour le site "sys".

F) /.ADM/MASTER.sys

Chacun des fichiers "/.ADM/MASTER.sys" est un fichier de "debugging" des opérations effectuées dans le mode MAITRE avec le site "sys".

G) TM.pid.ddd

Les fichiers "TM.pid.ddd" sont des fichiers temporaires créés dans le "spool" lors d'un transfert d'un site voisin sur le site local.

- "pid" est un identificateur de processus.
- "ddd" est un numéro à trois chiffres initialisé à zéro lors de l'établissement de toute nouvelle connection et incrémenté d'une unité à chaque fichier copié.

Une fois le fichier complet reçu, il est copié vers sa destination. Si une erreur survient lors de cette copie, le fichier n'est pas détruit.

H) SUBDIRS

Le fichier "SUBDIRS" contient des entrées du format suivant :

"prefix" "sub-directory", où :

- "prefix" est le préfixe d'un fichier.
- "sub-directory" est le nom d'un sous-répertoire de "/usr/spool/uucp".

I) sys/C./filename

Un sous-répertoire "sys/C." est créé pour chaque site voisin. Chaque fichier de ces sous-répertoires est un fichier de commandes relatif au site "sys". Chaque entrée d'un de ces fichiers est associée à un fichier de données.

Chaque entrée d'un fichier de commandes a deux formats possibles :

- pour les fichiers de données à envoyer, le format est :

  "type" "s_file" "d_file" "log" "opts" "spl_f" "prot" "usr" où :

  - "type" est le type de commande ("S" pour Send).
  - s_file" est le nom complet du fichier source.
  - "d_file" est le nom complet du fichier de destination.
  - "log" est le nom de "login" de l'utilisateur qui a demandé le transfert.
  - "opts" est la liste des options.
  - "spl_f" est le nom du fichier de données associé dans le "spool".
  - "prot" est le mode de protection du fichier de données sur le site voisin.
  - "usr" est le nom de l'utilisateur du site voisin qui doit être averti du transfert.

- Pour les fichiers de données à recevoir, le format est :

  "type" "s_file" "d_file" "log" "opts" où :

  - "type" est le type du fichier ("R" pour Receive).
  - "s_file" est le nom complet du fichier à envoyer.
  - "d_file" est le nom complet du fichier de destination.
  - "log" est le nom de "login" de l'utilisateur qui a demandé le transfert.

- "opts" est la liste des options.

J) /sys/D./filename

Un sous-répertoire "sys/D." est créé pour chaque site voisin. Chaque fichier "filename" de ces sous-répertoires est un fichier de données.

K) /sys/X./filename

Un sous-répertoire "sys/X." est créé pour chaque site voisin. Chaque fichier "filename" de ces sous-répertoires est un fichier qui doit être exécuté par, ou pour le site "sys".

Chacun de ces fichiers possède les entrées suivantes :

- "U" "usr" "system" où :
  - "U" indique qu'il s'agit de la ligne "utilisateur".
  - "usr" est le nom de l'utilisateur.
  - "system" est le nom de l'émetteur de la commande.
- "F" "file_name" "real_name" où :
  - "F" indique qu'il s'agit de la ligne "fichiers". Il peut y en avoir plusieurs.
  - "file_name" est le nom unique du fichier utilisé lors de la transmission.
  - "real_name" est le nom de fichier d'où l'on a tronqué le chemin d'accès.
- "I" "file_name" où :
  - "I" indique qu'il s'agit de la ligne "entrée standard"
  - "file_name" est l'entrée standard.
- "O" "file_name" "system_name" où :
  - "O" indique qu'il s'agit de la ligne "sortie

standard".

- "file_name" est la sortie standard.

- "system_name" est le nom du système destinataire des résultats.

- "C" "command_line" où :

- "C" indique qu'il s'agit de la ligne contenant la commande à exécuter.

- "command_line" est la commande qui doit être exécutée.

1.1.2 Fichiers de gestion de la sécurité

A) USERFILE

Le fichier USERFILE contient des informations sur le préfixe des chemins d'accès autorisés pour les transferts de fichiers entre le site local et les sites voisins.

Il contient des entrées du format suivant :
"login" "sys" "[c]" "path-name" "[path-name]" ... où :

- "login" est le nom de "login" pour un site voisin.

- "sys" est le nom du site voisin.

- "c" est un indicateur optionnel indiquant s'il faut rappeler ou non le site voisin.

- "path-name" est un préfixe du chemin d'accès obligé pour le site voisin.

Ce fichier est utilisé comme suit :

- pour l'envoi d'un fichier du site local sur un site voisin, les préfixes des chemins d'accès obligés pour le fichier à transférer sont ceux se trouvant dans la première ligne du fichier USERFILE pour laquelle le champ "login" est égal au nom de "login" de l'utilisateur demandeur du transfert. Si une telle ligne n'y figure pas, la première ligne trouvée ayant un champ "login" vide est utilisée.

- Pour la réception d'un fichier d'un site voisin sur le site local, les préfixes des fichiers autorisés pour le fichier de destination sont ceux se trouvant dans la première ligne du fichier USERFILE pour

laquelle le champ "sys" est égal au nom de système d'où émane la demande. Si une telle ligne n'y figure pas, la première ligne trouvée ayant un champ "sys" vide est utilisée.

Exemple d'entrée dans ce fichier :

```
u,m /usr/xyz /usr/spool
u,  /usr/spool
```

Cette entrée permet à tout site voisin de se connecter sous le nom "u", mais si le nom du site se connectant n'est pas "m", il peut seulement demander le transfert de fichiers dont le nom commence par "/usr/spool".

B) XQTCMDS

Le fichier "XQTCMDS" dresse de façon exhaustive la liste des commandes dont un utilisateur distant peut demander l'exécution. Ce fichier peut contenir au maximum 24 commandes.

C) SEQF

Le fichier "SEQF" contient une entrée par site voisin avec lequel les administrateurs du site local ont décidé de converser en prenant en considération un numéro de séquence. Les entrées initiales sont constituées du nom de site. Lors de la première communication avec un de ces sites, le numéro de séquence ainsi que l'heure sont ajoutés. Ces deux items sont mis à jour lors de toute communication ultérieure.

1.1.3 Fichiers de gestion de données concernant le réseau

A) L_stat

Le fichier "L_stat" contient des informations sur les connections avec les sites voisins.

B) R stat

Le fichier "R_stat" contient des informations concernant le trafic entre le site local et les sites voisins.

C) L sub"

Le fichier "L_sub" contient des informations concernant les connections avec les sites voisins d'un sous-ensemble du réseau. Ce sous-ensemble peut être défini par le programme UUSUB décrit plus loin.

D) R sub

Le fichier "R_sub" contient des informations concernant le trafic entre le site local et les sites voisins d'un sous-ensemble du réseau. Ce sous-ensemble peut être défini par le programme UUSUB décrit plus loin.

E) /. ADM/LOGF. sys

Chacun des fichiers "/. ADM/LOGF. sys" contient le déroulement des communications avec le site "sys".

F) SYSLOG

Le fichier "SYSLOG" contient le détail des transmissions ayant eu lieu. Il spécifie pour chacune d'entre elles l'utilisateur, le site, la date et l'heure, le nombre de bytes transférés et le temps utilisé.

G) /. ADM/STST. sys

Chacun des fichiers "/. ADM/STST. sys" contient pour le site voisin "sys", des informations concernant les échecs éventuels qui ont été constatés lors d'un "login", d'un appel ou de la vérification d'un numéro de séquence. Il contient également un indicateur d'état actif quand deux sites sont en train de converser. Pour les échecs

ordinaires (appel, "login"), ce fichier empêche tout nouvel essai pendant environ une heure. Pour des échecs provenant d'un numéro de séquence, ce fichier doit être supprimé avant toute nouvelle tentative de transfert.

2. Programmes

2.1 Couche 2

2.1.1 UUCP

Le programme UUCP crée les fichiers nécessaires dans le "spool" d'entrée-sortie pour copier un fichier d'un site sur un autre. Les sites impliqués doivent obligatoirement se trouver dans le fichier "L.sys".

La syntaxe de la commande est la suivante :

UUCP [options] source destination

Les différentes options sont :

- C : cette option force la copie dans le "spool" du fichier à transférer.

- c : cette option permet d'empêcher la copie du fichier dans le "spool", celui-ci étant directement pris là où il se trouve lors du transfert effectif.

- d : cette option permet de créer les répertoires nécessaires pour copier le fichier.

- f : cette option permet de ne pas créer de répertoires intermédiaires pour la copie du fichier.

- esys : cette option permet de spécifier le site sur lequel la commande UUCP doit être exécutée.

- gletter : cette option permet de modifier l'ordre d'exécution de la commande en spécifiant une priorité différente.

- m : cette option permet d'indiquer qu'il faut prévenir le demandeur après accomplissement du travail demandé.

- nname : cette option permet de signaler au destinataire qu'un fichier lui a été envoyé.

- r : cette option permet de mettre la requête en file d'attente, sans que soit lancé le transfert effectif.

- sdir : cette option permet d'imposer un autre répertoire que le "spool" d'entrée-sortie.

- xd : cette option permet de définir un niveau de "debugging".

2.1.2 UUX

Le programme UUX crée les fichiers nécessaires dans le "spool" d'entrée-sortie pour exécuter une commande sur un site voisin, avec des arguments pouvant provenir d'autres sites. Les sites impliqués doivent obligatoirement se trouver dans le fichier "L.sys".

La syntaxe de la commande est la suivante :

UUX [options] command-string

Les différentes options sont :

- p : cette option permet de spécifier comme entrée standard de la commande UUX, l'entrée standard de "command-string".

- - : cette option a le même effet que p.

- r : cette option permet de mettre la requête en file d'attente, sans que le transfert effectif soit lancé.

- xd : cette option permet de définir un niveau de "debugging".

- gletter : cette option permet de modifier l'ordre d'exécution de la commande en spécifiant une priorité différente.

- n : cette option permet de refuser l'envoi d'une confirmation au demandeur lorsque sa requête est accomplie.

- z : cette option permet de refuser l'envoi d'une confirmation au demandeur lorsque sa requête est accomplie - avec succès.

- c : cette option permet d'empêcher la copie des fichiers dans le "spool" d'entrée-sortie, ceux-ci étant directement pris là où ils se trouvent lors du transfert effectif.

2.1.3 UUCICO

Le programme UUCICO a les fonctions suivantes :

- parcourir le "spool" d'entrée-sortie à la recherche d'un éventuel travail.

- Appeler le site voisin.

- Négocier un protocole de transmission avec le site appelé.

- Exécuter toutes les requêtes de deux sites.

- Enregister un certain nombre d'informations utiles.

UUCICO a deux modes possibles : le mode MAITRE et le mode ESCLAVE. Il peut être invoqué avec différentes options :

- sans paramètres, il est lancé en mode ESCLAVE.

- rx : cette option permet de spécifier explicitement le mode dans lequel UUCICO doit être lancé (MAITRE ou ESCLAVE).

- xd : cette option permet de définir un niveau de "debugging".

- t : cette option force l'appel immédiat dans le mode MAITRE.

- ddir : cette option permet d'imposer un autre répertoire que le "spool" d'entrée-sortie.

2.1.4 UUXQT

Le programme UUXQT exécute toutes les requêtes devant être exécutées localement. Ces commandes doivent se trouver dans la liste des commandes autorisées (c'est-à-dire dans le fichier "XQTCMDS").

La syntaxe de la commande est la suivante :

UUXQT [options]

UUXQT peut être invoqué avec une option :

- xd : cette option permet de définir un niveau de "debugging".

2.1.5 UULOG

Le programme UULOG a pour fonction de faire un et un seul fichier des fichiers "LOGF.sys" et permet aussi d'obtenir des informations sur le déroulement des communications d'un site ou d'un utilisateur donné.

La syntaxe de la commande est la suivante :

UULOG [options]

UULOG peut être invoqué avec différentes options :

- ssys : cette option permet d'imprimer les informations concernant le déroulement des communications avec le site "sys".

- uuser : cette option permet d'imprimer les informations concernant le déroulement des communications de l'utilisateur "user".

- xd : cette option permet de définir un niveau de "debugging".

2.1.6 UUCLEAN

Le programme UUCLEAN a pour fonction de supprimer du "spool" d'entrée-sortie tout fichier plus vieux qu'un nombre donné d'heures.

La syntaxe de la commande est la suivante :

UUCLEAN [options]

UUCLEAN peut être invoqué avec différentes options :

- ddir : cette option permet d'imposer un autre répertoire que le "spool" d'entrée-sortie.

- m : cette option permet d'envoyer un message au propriétaire du fichier supprimé.

- ppre : cette option permet de spécifier le préfixe des fichiers qui doivent être supprimés.

- ntime : cette option permet d'indiquer que ce sont les fichiers plus vieux qu'un nombre "time" d'heures qui doivent être supprimés (si l'option "- p" est employée, seuls les fichiers dont le préfixe correspond sont considérés).

- xd : cette option permet de définir un niveau de "debugging".

2.1.7 UUENCODE

Le programme UUENCODE a pour fonction de coder un fichier composé de caractères non ascii en caractères ASCII.

La syntaxe de la commande est la suivante :

UUENCODE file

2.1.8 UUDECODE

Le programme UUDECODE a pour fonction de décoder un fichier qui a été codé par UUENCODE, afin de retrouver le fichier original.

La syntaxe de la commande est la suivante :

UUDECODE file

2.1.9 UUCALL

Le programme UUCALL permet de lancer facilement le programme UUCICO en mode MAITRE, avec possibilité de "debugging".

La syntaxe de la commande est la suivante :

UUCALL [options] system

UUCALL peut être invoqué avec différentes options :

- f : cette option permet de suspendre temporairement les appels vers le site "system".

- pX : cette option permet d'indiquer qu'il faut seulement transférer les fichiers dont la priorité est inférieure ou égale à "X".

- i : cette option permet d'indiquer qu'il faut appeler le site "system" uniquement s'il existe du travail pour lui.

- r[N] : cette option permet d'indiquer qu'il faut rappeler le site "system" "N" fois si le fichier "STST. system" existe.

- tN : cette option permet d'abandonner l'appel après "N" secondes.

2. 1. 10 UUSTAT

Le programme UUSTAT imprime des informations concernant les demandes de transfert. Ces informations sont :

- le numéro de la demande.
- Le nom de l'utilisateur qui a émis la demande.
- Le nom du site voisin concerné.
- la date et l'heure d'émission de la demande.
- La date et l'heure de l'obtention du statut de la demande.
- Le statut de la demande sous forme d'un code ou sous forme de phrases explicatives.

La syntaxe de la commande est la suivante :

UUSTAT [options]

UUSTAT peut être invoqué avec différentes options :

- sans options, il donne des informations sur toutes les demandes émises par l'initiateur de la commande.

- xd : cette option permet de définir un niveau de "debugging".

- chour : cette option permet de supprimer les informations concernant toutes les demandes plus vieilles que "hour" heures. Cette option peut seulement être initialisée par l'"utilisateur" UUCP ou le "super user".

- jall : cette option permet d'obtenir des informations sur toutes les demandes.

- mmch : cette option permet d'obtenir des informations sur l'accessibilité du site "mch" (ces informations indiquent le nom du système, la date et l'heure de la dernière obtention d'un statut et le statut du site

lui-même sous forme de phrases explicatives). Si "mch" est égal à "all", des informations sur tous les sites connus du système sont obtenues.

- kjobn : cette option permet de supprimer la demande dont le numéro est "n". Cette demande doit appartenir à la personne qui exécute la commande à moins qu'elle ne soit "super user".

- uuser : cette option permet d'obtenir des informations sur toutes les demandes émises par l'utilisateur "user".

- chour : cette option permet d'obtenir des informations sur toutes les demandes plus vieilles que "hour" heures.

- yhour : cette option permet d'obtenir des informations sur toutes les demandes plus jeunes que "hour" heures.

- ssys : cette option permet d'obtenir des informations sur toutes les demandes concernant le site "sys".

- v : cette option permet d'obtenir les informations sous forme de phrases explicatives plutôt que sous forme de code.

2.1.11 UUSNAP

Le programme UUSNAP permet d'obtenir un état courant des fichiers se trouvant dans le "spool" d'entrée-sortie. Il donne les informations suivantes :

- le nom des sites pour lesquels sont destinés un ou plusieurs fichiers.

- Le nombre de fichiers de commandes, de données et de fichiers qui doivent être exécutés, relatifs à ce site.

La syntaxe de la commande est la suivante :

UUSNAP

2.1.12 UUNAME

Le programme UUNAME permet d'obtenir la liste des noms des sites voisins.

La syntaxe de la commande est la suivante :

UUNAME [option]

UUNAME peut être invoqué avec une option :

- l : cette option permet d'obtenir le nom du site local.
- sans option, il donne le nom de tous les sites voisins

2.1.13 UUSUB

Le programme UUSUB permet de définir un sous-ensemble du réseau et d'en obtenir des statistiques. Ces statistiques comprennent :

- pour les connections :
  - le nom du site voisin.
  - Le nombre de fois que le site local a essayé d'appeler un site depuis la dernière suppression des informations.
  - Le nombre de connections réussies.
  - La date et l'heure de la dernière connection réussie.
  - Le nombre de connections non réussies pour cause de périphériques occupés.
  - Le nombre de connections non réussies pour cause d'échec de "login".
  - Le nombre de connections non réussies pour cause de non-réponse (ligne occupée, site hors-circuit, ...).
- pour le trafic :
  - le nombre de fichiers envoyés.
  - Le nombre de bytes envoyés durant la période qui était indiquée dans la dernière commande UUSUB

dont l'option "uhr" était initialisée.

- Le nombre de fichiers reçus.

- Le nombre de bytes reçus durant la période qui était indiquée dans la dernière commande UUSUB dont l'option "uhr" était initialisée.

La syntaxe de la commande est la suivante :

UUSUB [options]

UUSUB peut être invoqué avec différentes options :

- xd : cette option permet de définir un niveau de "debugging".

- asys : cette option permet d'ajouter le site "sys" au sous-ensemble.

- dsys : cette option permet de supprimer le site "sys" du sous-ensemble.

- csys : cette option permet de tester la connection vers le site "sys". Si "sys" est égal à "all", alors toutes les connections sont testées.

- uhr : cette option permet de rassembler les statistiques concernant le trafic durant les "hr" dernières heures.

- r : cette option permet d'obtenir les statistiques sur le trafic.

- l : cette option permet d'obtenir les statistiques sur les connections.

- f : cette option permet de supprimer les statistiques concernant les connections.

Annexe B

1. Tarifs en vigueur au départ du réseau DCS

1.1 Relations intérieures

- Un appel coûte automatiquement 0.15 FB qu'il aboutisse ou non.

- La taxe à la durée (durée d'ouverture de la connexion) est de 0.10 FB par unité de trente secondes.

- La taxe au volume est de 0.20 FB par décasegment entre 8H. et 18h.30, et de 0.10 FB entre 18h.30 et 8h. sans utilisation du PAD de la RTT. Avec utilisation du PAD de la RTT, ces tarifs passent à 0.60 FB (tarif de jour) et 0.30 FB (tarif de nuit) par décasegment.

Pays	Tarifs (T.V.A. 19 % non comprise)		
	Durée FB/minute indivisible	Volume FB/10 segments indivisibles	
		pas d'utilisation de l'A.D.P.	avec utilisation de l'A.D.P. (1)
AUSTRALIE	4,00	2,20	2,50
BAHRAIN	uniquement pour le trafic entrant		
BRESIL	6,00	3,00	3,30
CANADA	4,00	2,20	2,50
COREE	6,00	3,20	3,30
COTE D'IVOIRE	4,00	2,20	2,50
ETATS-UNIS D'AMERIQUE	4,00	2,20	2,50
FR-ANTILLES	1,20	0,90	1,20
FR-GUYANE	1,20	0,90	1,20
FR-REUNION	1,20	0,90	1,20
GABON	4,00	2,20	2,50
HONG KONG	6,00	3,20	3,50
INDONESIE	6,00	3,20	3,50
ISRAEL	4,00	2,20	2,50
JAPON	5,00	3,00	3,30
NOUVELLE ZELANDE	4,00	2,20	2,50
SINGAPOUR	6,00	3,00	3,30
SUD-AFRICAINE (Rép.)	6,00	3,20	3,50
TAIWAN	6,00	3,20	3,50

- La redevance bimestrielle d'abonnement pour le raccordement direct ou le code d'identification (NUI) est également due : voir "Tarifs du réseau DCS en service intérieur".
- Pour chaque appel, que celui-ci ait abouti ou non, une taxe d'établissement d'appel de 0,15 FB est due.
- D'autres pays seront accessibles d'ici peu.

(1) Accès via le réseau téléphonique : la taxe pour une communication téléphonique zonale
via le réseau télex : 1 U.T. par minute.
En cas d'utilisation de terminaux à couplage acoustique la taxe de la communication téléphonique zonale ou interzonale est due.

1.2 Relations internationales

Pays	Tarifs (T.V.A. 19 % non comprise)		
	Durée FB/minute indivisible	Volume FB/10 segments indivisibles	
		pas d'utilisation de l'A.D.P.	avec utilisation de l'A.D.P. (1)
ALLEMAGNE (Rép. Féd.)	1,20	0,90	1,20
AUTRICHE	1,60	1,00	1,30
DANEMARK	1,20	0,90	1,20
ESPAGNE	1,60	1,00	1,30
FINLANDE	1,20	0,90	1,20
FRANCE	1,20	0,90	1,20
GRECE	1,20	0,90	1,20
IRLANDE	1,20	0,90	1,20
ITALIE	1,20	0,90	1,20
LUXEMBOURG	0,40	0,44	0,87
NORVEGE	1,20	0,90	1,20
PAYS-BAS	1,20	0,90	1,20
PORTUGAL	1,60	1,00	1,30
ROYAUME-UNI	1,20	0,90	1,20
SUEDE	1,20	0,90	1,20
SUISSE	1,20	0,90	1,20